Human
Systems
Information
Analysis
Center

SOAR
STATE OF THE ART REPORT

April 2002

Cognitive Systems Engineering in Military Aviation Environments:

Avoiding Cogminutia Fragmentosa!

A report produced under the auspices of The Technical Cooperation Programme
Technical Panel HUM TP–7 Human Factors in Aircraft Environments

Editors:
Michael D. McNeese
Michael A. Vidulich

Distribution A:
Approved for public release; Distribution is unlimited.

Cognitive Systems Engineering in Military Aviation Environments:

Avoiding Cogminutia Fragmentosa!

Cognitive Systems Engineering in Military Aviation Environments:

Avoiding Cogminutia Fragmentosa!

EDITORS:
MICHAEL D. MCNEESE
MICHAEL A. VIDULICH

Human Systems Information Analysis Center
Wright-Patterson Air Force Base, Ohio, 2002

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	April 2002	State-of-the-Art Report

4. TITLE AND SUBTITLE
Cognitive Systems Engineering in Military Aviation Environments: Avoiding Cogminutia Fragmentosa

5. FUNDING NUMBERS
SPO700-98-D-4001

6. AUTHOR(S)
Michael D. McNeese & Michael A. Vidulich (Editors)

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
HSIAC Program Office
AFRL/HEC/HSIAC Bldg 196
2261 Monahan Way
Wright-Patterson AFB OH 45433-7022

8. PERFORMING ORGANIZATION REPORT NUMBER
HSIAC SOAR 02-01

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Defense Technical Information Center
ATTN: DOD IAC Program Office (DTIC-AI)
8725 John J. Kingman Road, Suite 0944
Ft. Belvoir, VA 22060-6218

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.
Available solely through HSIAC for $45.00 (US).

12b. DISTRIBUTION CODE
A

13. ABSTRACT (Maximum 200 Words)

Complex environments place workers in an information-rich world with little time to make sense out of events surrounding them, assess their plans, make appropriate decisions, or perform multiple activities. This lack of "cognitive engineering" may produce cogminutia fragmentosa, where one's cognitive world breaks down into small, isolated strands of thought as unanticipated events transpire. When this persists, there is no longer an *interface* between the worker's cognitive world and the work for which he is responsible. Cognitive Systems Engineering(CSE) has been a response to work environments that produce cogminutia fragmentosa. As cognitive systems are proposed for complex environments, such as in military aviation domains, CSE is frequently utilized to understand and analyze various components of operator expertise and the interaction of expertise with specifications of the work domain.

This report on CSE resulted from an international symposium, Cognitive Systems Engineering in Military Aviation, co-sponsored and organized under the auspices of The Technical Cooperation Program (TTCP). The symposium was held in conjunction with the 9th meeting of TTCP Group HUM (Human Resources and Performance), Technical Panel 7 (TP7) Human Factors in Aircraft Environments, hosted by the U.S. Air Force Research Laboratory (USAFRL) and held in Dayton, Ohio, 22-26 May 2000.

14. SUBJECT TERMS
Cognitive Systems Engineering, Cognitive Task Analysis, Systems Design, Military Aviation, Human Factors, User-Centered Design, Systems Analysis, User Requirements, Decision Making

15. NUMBER OF PAGES
402

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UNLIMITED

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

Contributors

David Beevis
140 Newton Drive
Willowdale, Ontario M2M 2N3
Canada

Michael C. Bonner
Land Operations Division
Defence Science and Technology Organisation
PO Box 1500
Edinburgh, SA 5111
Australia

Klaus Christoffersen
Institute for Ergonomics
The Ohio State University
290 Baker Systems
1971 Neil Avenue
Columbus, OH 43210
United States

Blair Dickson, PhD
Qinetiq, Centre for Human Sciences
Farnborough, Hampshire GU14 0LX
United Kingdom

James R. Easter
Aegis Research Corporation
501 Grant Street; Suite 475
Pittsburgh, PA 15219
United States

Jack L. Edwards
AIM
206 Keewatin Avenue
Toronto, Ontario M4P 1Z8
Canada

Robert Eggleston, PhD
U.S. Air Force Research Laboratory
AFRL/HECA
2255 H Street
Wright-Patterson AFB, OH 45433-7022
United States

William C. Elm
Aegis Research Corporation
501 Grant Street; Suite 475
Pittsburgh, PA 15219
United States

James W. Gualtieri, PhD
Aegis Research Corporation
501 Grant Street; Suite 475
Pittsburgh, PA 15219
United States

Keith C. Hendy
Human Engineering Sector
Defence and Civil Institute of Environmental Medicine
PO Box 2000
1133 Sheppard Avenue West
Toronto, Ontario M3M 3B9
Canada

Howard Howells
Qinetiq, Centre for Systems Simulation
Farnborough, Hampshire GU14 0LX
United Kingdom

Eva Hudlicka, PhD
Psychometrix Inc.
1805 Azalea Drive
Blacksburg, VA 24060
United States

Frederick Lichacz
Defence and Civil Institute of Environmental Medicine
Toronto, Ontario
Canada

Gavan Lintern, PhD
Aptima, Inc
12 Gill St, Suite 1400
Woburn, MA 01801
United States

Michael D. McNeese, PhD
School of Information Sciences & Technology
The Pennsylvania State University
2M Thomas Building
University Park, PA 16801
United States

Christopher A. Miller, PhD
SIST
1272 Raymond Ave.
St. Paul, MN 55108
United States

Nicholas Milton, PhD
Epistemics Ltd
Strelley Hall
Nottingham NG8 6PE
United Kingdom

Neelam Naikar, PhD
Air Operations Division
Aero & Maritime Research Laboratory
PO Box 4331
Melbourne, Victoria 3001
Australia

Kit Pleydell-Pearce, PhD
University of Southampton
Highfield, Southampton SO17 1BJ
United Kingdom

Scott S. Potter, PhD
Aegis Research Corporation
501 Grant Street; Suite 475
Pittsburgh, PA 15219
United States

John Reising, PhD
U.S. Air Force Research Laboratory
AFRL/HEC
2255 H Street
Wright-Patterson AFB, OH 45433-7022
United States

Emilie Roth, PhD
Roth Cognitive Engineering
89 Rawson Road
Brookline, MA 02445-4509
United States

Penelope Sanderson, PhD
University of Queensland
ARC Key Centre for Human Factors and Applied Cognitive Psychology
McElwain Building
St. Lucia, Queensland QLD 4072
Australia

Nigel Shadbolt, PhD
University of Southampton
Electronics and Computer Science
Highfield, Southampton SO17 1BJ
United Kingdom

Robert M.Taylor
DSTL, Human Sciences
Farnborough, Hampshire GU14 0LX
United Kingdom

Jeni Tennison, PhD
Jeni Tennison Consulting Ltd.
4 Dudley Court
Beeston, Nottingham NG9 3HZ
United Kingdom

Michael A. Vidulich, PhD
U.S. Air Force Research Laboratory
AFRL/HECP
2255 H Street
Wright-Patterson AFB, OH 45433
United States

Sharon Whitecross, PhD
University of Bristol
Department of Experimental Psychology
Social Sciences Complex
8 Woodland Road
Clifton, Bristol BS8 1TN
United Kingdom

David D.Woods, PhD
Institute for Ergonomics
The Ohio State University
290 Baker Systems
1971 Neil Avenue
Columbus, OH 43210
United States

Several of the illustrations were provided by commercial publishers and those companies retain the right to the individual figures and associated material.

This material may be reproduced by or for the U.S. Government pursuant to the copyright license under the clause at 52.227–7013 (May 1987).

Additional copies of this state-of-the-art report (SOAR) are available for US $45^{00} (plus shipping and handling) from—

Human Systems IAC Program Office
AFRL/HEC/HSIAC
2261 Monahan Way, Building 196, Room 8
Wright-Patterson AFB, OH 45433–7022

Tel: (937) 255–4842
Fax: (937) 255–4823
E-mail: sales@wpafb.af.mil
URL: http://iac.dtic.mil/hsiac

Preface

Cognitive Systems Engineering (CSE) is making an impact on a number of different domains in which people utilize their various abilities, skills, and knowledge to overcome and change challenging situations. Of contemporary importance is the domain of military aviation. As new challenges are created within this broad community, the need for CSE will become even greater to make a difference in how complex systems come to be used by individual users or crews. The new millennium already is offering a variety of advanced information technologies for military aviation. Coupled with decreasing resources and necessitated reductions in crew size, the role of CSE looms as an extremely relevant field of study—for both theoretical development as well as practical application.

With these ideas in mind, it is our privilege and pleasure to welcome you to an innovative new State-of-the-Art Report (SOAR) that introduces readers to cognitive systems engineering as it relates and applies to military aviation domains. It is our hope to present a broad—yet poignant—integration of perspectives, issues, methods, and applications that afford a first-look understanding of CSE for use within aviation fields of practice. The book will consist of nine distinct chapters that approach CSE in a special way. The chapters are taken from internationally respected authors and provide the reader with a thorough understanding of the foundation of CSE as well as how it relates to different facets of military aviation. As a reader, we believe you will discover an active and illustrative review of the state-of-the-art developments that scientists, engineers, managers, developers, and students must be aware of for furthering their knowledge and understanding. Having introduced our motivations for organizing the book, let's briefly survey the direction the book intends to take.

Complex environments of the 21st century place workers in an information-rich world with little time to make sense out of events surrounding them, assess their plans, make appropriate decisions, or perform multiple activities. In many cases, computational support and advanced interfaces for work activities have not been engineered with cognition or context in mind. Unfortunately, this lack of "cognitive engineering" may produce what we refer to as "cogminutia fragmentosa," where the worker's cognitive world breaks down into small, isolated strands of thought as unanticipated events transpire (mental stovepipes). There can be a loss of meaning or control as the worker becomes separated from the demands of his or her work, and may remain lost in terms of comprehending the emerging elements of a situation. When cogminutia fragmentosa persists, there is no longer an interface between the worker's cognitive world and the work for which he or she is responsible. In other words, the worker cannot properly adapt to the situation encountered (i.e., a maladaptive state exists). If this state continues, errors, failure, and even catastrophic disasters are highly proba-

ble. This state may also contribute to affective and emotional responses (e.g., fear, anxiety, rage), which further complicate agent-environment transactions. However, all is not lost. We are now at a point in history where it is not uncommon to observe human factors practitioners referring to "cognitive systems engineering" as their method or tool of choice to respond to work environments that produce cogminutia fragmentosa. Indeed, as first-of-a-kind cognitive systems are proposed for complex environments, such as in military aviation domains, CSE is frequently utilized to understand and analyze various components of operator or team expertise (e.g., cognitive skills, engagement rules, specific knowledge); and the interaction of expertise with specifications of the work domain. As CSE is applied to real-world settings, agent-environment transactions can be quantitatively or qualitatively modeled (represented) and then used as a basis to predicate elements of a design (e.g., a human-computer interface, a decision support system). Typically, CSE practitioners engage workers through a variety of CSE methods that capture multiple facets of how work is transacted from agents to environment.

This book highlights the perspectives and foundations of an international community of practitioners who have both developed and applied CSE. One can see that the field emerges from several corridors that, in turn, produce alternative methodologies/approaches to address military aviation domains. Differing philosophies and techniques spawn incisive pathways of integration in the development of design artifacts. Because the aviation domain is fraught with multifarious levels of complexity and is demonstrative of cogminutia fragmentosa, we believe it supplies an excellent foundation for reviewing, assessing, communicating, and evaluating some of the principles (and nuances) inherent within various programs of CSE. The SOAR will emulate this objective by presenting the following sections for readers (along with their respective first authors):

- Foundations and Perspectives (Reising, Eggleston, McNeese, Woods)
- Methodological Pursuits (Potter, Neelam, Hendy)
- Innovations, Integration, and Application (Taylor, Hudlicka)

As editors of the book we challenge the reader to contrast/compare philosophies of use, theories of origin, goals, benefits, methods, tools, experiences, constraints and problems of applications, lessons learned, and examples as a means to generate new levels of understanding—as they relate to the specific constraints encountered in military aviation.

Michael D. McNeese
University Park, Pennsylvania

Michael A. Vidulich
Wright-Patterson Air Force Base, Ohio

January 2002

About Human Systems IAC

The Human Systems Information Analysis Center (HSIAC) is the gateway to worldwide sources of up-to-date human-factors information and technologies for designers, engineers, researchers, and human-factors specialists. HSIAC provides a variety of products and services to government, industry, and academia promoting the use of ergonomics in the design of human-operated and manned systems.

HSIAC's primary objective is to acquire, analyze, and disseminate timely information about ergonomics. On a cost-recovery basis, HSIAC will perform the following functions—

- Distribute human-factors and ergonomics technologies and publications.
- Conduct customized bibliographic searches and reviews.
- Prepare state-of-the-art reports and critical reviews.
- Conduct specialized analysis evaluations.
- Organize and/or conduct workshops and conferences.

HSIAC is a Department of Defense Information Analysis Center sponsored by the Defense Technical Information Center, Fort Belvoir, Virginia. It is technically managed by the U.S. Air Force Research Laboratory Human Effectiveness Directorate, Wright-Patterson Air Force Base, Ohio, and operated by Booz Allen Hamilton, McLean, Virginia.

HSIAC
ARMY NAVY AIR FORCE NASA FAA NATO

Foreword

Cognitive Systems Engineering (CSE) theories, methods and their application have received increased attention by human factors and ergonomics professionals who design complex human systems. This is particularly true for the stressful, information-overloaded, time-constrained, lethal work environment within the military. Military aviation is only one of the important domains and is the primary focus of this report. The reasons for this increased emphasis are many and compelling. Most operator-interfaces with complex weapon systems are not designed with the cognitive work requirements of the operators as a formal consideration. Indeed, even the Command, Control, Communications, Computers, (C4), Intelligence, Surveillance and Reconnaisance (ISR) [C4ISR] systems whose sole purpose is to support human situation awareness and decision making are rarely designed from a top-down, human-centered viewpoint. Improperly conceived and interfaced automation can lead to design-induced human error, particularly catastrophic in the aviation domain. This situation will only become exasperated as more uninhabited systems, such as unmanned combat air vehicles, are fielded. This comprehensive report examines in detail the various CSE foundations and theories, practical methods, and finally examples of applications to the design of complex systems. Chapters authored by leading experts in this increasingly important field provide a provocative analysis of progress, successes, and remaining challenges. Differences of opinion are intentionally presented to stimulate a thorough assessment of the state-of-the-art. More development is needed to formalize the methods that can be consistently applied in to bridge the remaining gap between CSE and complex system design. This single report informatively lays out these issues and serves as a guidepost for the way ahead. It is highly recommended reading for CSE researchers striving to mature theories and methods, and designers whose goal is to provide future warfighters with highly effective work-centered systems.

Maris Vikmanis
Chief, Crew System Interface Division
Human Effectiveness Directorate
U.S. Air Force Research Laboratory
Wright-Patterson Air Force Base, Ohio

Acknowledgments

The SOAR is predicated on and developed, in part, from an international symposium, Cognitive Systems Engineering in Military Aviation, cosponsored and organized under the auspices of the five-nation (United States, United Kingdom, Canada, Australia, New Zealand) forum for international research collaboration, namely The Technical Cooperation Program (TTCP). The symposium was held in conjunction with the 9th Annual Meeting of TTCP Group HUM (Human Resources and Performance), Technical Panel 7 (TP7) Human Factors in Aircraft Environments, hosted by the U.S. Air Force Research Laboratory (AFRL) and held in Dayton, Ohio, 22–26 May 2000. The purpose of TTCP HUM TP7 is to facilitate collaborative research and information exchange on human factors issues relevant to the extension of operational performance of advanced military aircraft. This symposium provided a timely opportunity to bring together key researchers and human factors specialists to discuss recent developments in cognitive systems engineering and to consider the implications for human factors issues in aircraft environments.

The holding of the International Symposium and the year 2000 meeting of the TP–7 Human Factors in Aircraft Systems panel were financially supported by the Crew System Interface Division of the U.S. Air Force Research Laboratory (AFRL/HEC, Mr. Maris Vikmanis, Division Chief).

In addition to acknowledging the above support, the editors would like to thankfully give credit to an outstanding effort on behalf of Mr. Jeffrey A. Landis, Editor and Publications Manager, Ms. R. Anita Cochran, Associate Editor, Mrs. RoseAnn Venis, Associate Editor, and Ms. K. Ahnie Senft, Graphic Artist, all with the Human Systems Information Analysis Center, Wright-Patterson Air Force Base, Ohio.

Furthermore, we give special acknowledgements to our authors and their incisive efforts in writing these chapters. Without their contributions we could have not endeavored such an effort. Also, we would like to thank reviewers of the book, including Dr. Joe McDaniel, U.S. Air Force Research Laboratory, who serves as the U.S. Government Technical Manager for the Human Systems Information Analysis Center.

M. D. M.
M. A. V.

Table of Contents

List of Figures xxvii

List of Tables xxxi

1 Cognitive Engineering and Its Relationship to Future Aviation Systems 1

Abstract 1
1.1 Introduction 2
1.2 Problems with Current Systems 2
1.3 PHILOSOPHY OF AUTOMATION 3
1.3.1 Early Automation Philosophy 3
1.3.2 Today's Automation Philosophy 4
1.3.3 Future Automation Philosophy 5
1.4 Operator and Electronic Associate Are A Team 5
1.4.1 Team Trust 6
1.5 Building the Electronic Associate 7
1.6 Unmanned Combat Air Vehicle (UCAV) and the Electronic Associate 8
1.6.1 Operator's Station 9
References 11

2 Cognitive Systems Engineering at 20-Something: Where Do We Stand? 15

Abstract 15
2.1 Introduction 15
2.2 Why Cognitive Systems Engineering? 18
2.3 Cognitive Systems Engineering Genotypes 20
2.3.1 CMU Genotype 21
2.3.2 OSU Genotype 26
2.3.3 RISO Genotype 31
2.3.4 UCSD Genotype 40
2.3.5 Conceptual Distinctions 43
2.4 CSE Engineering Practices 46
2.4.1 Analysis 46
2.4.2 Design 54

2.4.3 Evaluation 59
2.5 CSE Deployment 60
2.5.1 CTA and Other Approaches to Work Analysis 61
2.5.2 Judgment in CSE 65
2.6 Cognitive Systems Engineering: Where Do We Stand? 71
References 72

3 Discovering How Cognitive Systems Should Be Engineered For Aviation Domains: A Developmental Look at Work, Research, and Practice 79

Abstract 79
3.1 Introduction 80
3.2 Query I: What Is It? 82
3.2.1 Orientation 82
3.2.2 General Definitions 83
3.2.3 What Does Cognitive Systems Engineering Consist Of? 84
3.2.4 Viewpoints and Converging Themes 84
3.3 Query II: What Are the Formative Conditions? 86
3.3.1 Human Factors Engineering 86
3.3.2 Knowledge Engineering 87
3.4 Query III: What Are the Objects of Interest of CSE? 88
3.4.1 Understanding Collaboration In Context 89
3.5 Query IV: What are Representative Approaches/ Premier Examples of CSE? 90
3.5.1 Theory-Driven Examples 92
3.5.2 Method-Driven Examples 96
3.6 Query V: What Has Transpired? 97
3.6.1 A General View 97
3.6.2 The Neonatal Stage 99
3.6.3 The Toddler Stage 100
3.6.4 The Formative Years 100
3.6.5 The Preteens 101
3.6.6 The Restless Teenager 103
3.7 Query VI: What Has Evolved/What Has Been Learned? . .108
3.7.1 Summary 110
3.8 Query VII: Conclusions—What's Next? 111
3.8.1 The Challenges 112
References 113

4 Balancing Practice-Centered Research and Design 121

Abstract 121
4.1 Broken Pipes 122
4.2 Complementarity as a Strategy to Balance Research and Design 123
4.3 Moving Target 127
4.4 Synchronizing Parallel, Interlocking Cycles 128
4.5 Imbalance 130
4.6 Points of Balance 131
4.7 Different Roles for Human-Factors-Related Work 132
4.8 Complementarity and Synchronization 132
4.9 Acknowledgments and Recommended Reading 133
References 134

5 Using Intermediate Design Artifacts to Bridge the Gap Between Cognitive Analysis and Cognitive Engineering 137

Abstract 137
5.1 Introduction 138
5.2 Developing a Design Thread From Cognitive Analysis to Decision Aiding 140
5.2.1 Representing the Way the World Works—Building a Functional Abstraction Hierarchy . . .141
5.2.2 Modeling Cognitive Demands—Deriving DRs . .144
5.2.3 Capturing the Means for Effective Decision-Making—Identifying Information Requirements .146
5.2.4 Linking DRs to Aiding Concepts—Developing and Documenting a "Model of Support" 148
5.2.5 Developing Prototypes That Instantiate the Aiding Concept 149
5.3 An Illustrative Case: An Ecological Interface Supporting Command Decision Making 150
5.3.1 Functional Abstraction Hierarchy (FAH) 151
5.3.2 Decision Requirements 154
5.3.3 Supporting Information Requirements 154
5.3.4 Display Task Description 155
5.3.5 Rapid Prototype of "Choose Combat Power" Display 156
5.4 Discussion 162
5.5 Acknowledgment 164
References 164

6 Cognitive Work Analysis for Air Defense Applications in Australia 169

Abstract 169
6.1 Background to CWA 170
6.1.1 Introduction 170
6.1.2 Work Domain Analysis 171
6.1.3 Control Task Analysis 171
6.1.4 Strategies Analysis 173
6.1.5 Socio-Organizational Analysis 173
6.1.6 Worker Competencies Analysis 173
6.2 CWA and the System Life-Cycle 174
6.2.1 Requirements 174
6.2.2 Specification 175
6.2.3 Design 175
6.2.4 Simulation 176
6.2.5 Evaluation of Designs 176
6.2.6 Implementation 176
6.2.7 Test 177
6.2.8 Operator Selection 177
6.2.9 Operator Training 177
6.2.10 Routine, Nonroutine, and Maintenance Activity .178
6.2.11 Research (HF Studies) 178
6.2.12 Upgrades 178
6.2.13 System Retirement 179
6.2.14 Examples of CWA in Context 179
6.3 Evaluation of Designs 179
6.4 Analysis of Human-System Integration 184
6.4.1 AEW&C Activity Context 184
6.4.2 AEW&C Control Tasks 185
6.4.3 AEW&C Human-System Integration 186
6.5 Training-System Design 188
6.5.1 Training Objectives and Design Objectives 191
6.5.2 Measures of Effectiveness and Data Collection . .191
6.5.3 Basic Training Functions and Scenario Generation 193
6.5.4 Physical Functions 193
6.5.5 Physical Context and Physical Attributes 194
6.6 Information Work Spaces 195
6.7 Conclusions 197
References 198

7 Analyzing the Cognitive System From a Perceptual Control Theory Point Of View 201

Abstract 201
7.1 Introduction 202
7.2 Human Systems Analysis 204
7.2.1 Structured Analysis Techniques 205
7.2.2 Domains of Analysis 206
7.2.3 Reliability of Human Systems Analyses 208
7.3 Perceptual Control Theory 209
7.3.1 Information Processing Models and PCT 213
7.4 PCT Systems Analysis 216
7.4.1 Decomposing the Goal Structure 217
7.4.2 Analyzing the Cognitive and Perceptual Components 219
7.5 Analyses Emerging From the PCT Approach 223
7.5.1 Stability Analysis 224
7.5.2 Support to Higher-Level Goals 226
7.6 Application of PCT Systems Analysis 228
7.6.1 Land Forces Command System 228
7.6.2 Intelligent Adaptive Interfaces 230
7.6.3 Two Views 231
7.6.4 The Notion of Plans 232
7.6.5 The Nature of Hierarchies 233
7.6.6 Hierarchies and Plans 234
7.6.7 Hierarchies Revisited 236
7.6.8 Final Thoughts on Shared Concepts for Intelligent Adaptive Interfaces 237
7.7 Discussion 238
7.8 Summary 245
References 246

8 Cognitive Cockpit Engineering: Coupling Functional State Assessment, Task Knowledge Management, and Decision Support for Context-Sensitive Aiding 253

Abstract 254
8.1 Introduction 254
8.1.1 Cognitive Design Requirements 254
8.1.2 Background—The Development of Intelligent Pilot-Aiding Systems 257
8.1.3 Cognitive Systems Engineering Challenges—Supporting Adaptiveness 259

8.2 The Cognitive Cockpit 264
8.2.1 Assisting Cognitive Work 264
8.2.3 Functional Integration and Cognitive Control . . .267
8.2.4 Functional Architecture—
Agents, Communication, and Tasks 273
8.3 Monitoring the Pilot—Cognition Monitor 275
8.3.1 COGMON Functions—Pilot State Assessment . .275
8.3.2 COGMON Measures 276
8.3.3 COGMON Implementation 279
8.4 Monitoring the Environment—
Situation Assessor Support System 280
8.4.1 The Development of Knowledge-Based Systems .281
8.4.2 SASS Functions—Situation Assessment and
Tactical-Decision Making 282
8.4.3 SASS Methodology 283
8.4.4 SASS Implementation 286
8.5. Monitoring the Mission Plan and Configuring the
Cockpit—Tasking Interface Manager 287
8.5.1 TIM Functions—Task, Timeline, Interface, and
Automation Management 287
8.5.2 TIM Implementation 290
8.6 Prototyping, Simulation, and Testing Intelligent Aiding . . .293
8.6.1 COGSIM Functions—Specification, Analysis,
Development, and Test 294
8.6.2 COGSIM Methodology 295
8.6.3 COGSIM Implementation 296
8.6.4 COGSIM Demonstration and Test 300
8.7 Conclusions 304
References 306

9 Beyond Cognitive Systems Engineering: Assessing User Affect and Belief States to Implement Adaptive Pilot-Vehicle Interaction 315

Abstract 315
9.1 Introduction 316
9.2 Affective and Belief States: Effects on Performance and
Assessment 320
9.2.1 Effects of Affective States on Performance 320
9.2.2 Assessment of Affective States 321
9.2.3 Effect of Belief States on Performance 325
9.3 Related Work in Affect and Belief Assessment and
Adaptation 327
9.4 ABAIS Adaptive Methodology and System Architecture . .330

9.4.1 ABAIS Adaptive Methodology 330
9.4.2 ABAIS System Architecture 330
9.5 ABAIS Demonstration Task: Fighter Pilot "Sweep" Task .333
9.5.1 Task Context 333
9.5.2 Human Context 334
9.5.3 Summary of Demonstration Task Events 334
9.6 Cognitive, Affective, Personality Task Analysis (CAPTA) .334
9.6.1 Description of the CAPTA Process 335
9.6.2 Use of CAPTA to Support the ABAIS Architecture Development 337
9.7 User State Assessment and Behavior Prediction 342
9.7.1 User Affect and Belief State Assessment 342
9.7.2 Affect and Belief State Impact Prediction 348
9.8 Adaptation 349
9.8.1 Strategy Selection 349
9.8.2 GUI Adaptation 349
9.9 Demonstration of ABAIS System Performance 352
9.10 Summary, Conclusions, and Future Work 354
9.10.1 Summary 354
9.10.2 Conclusions 355
9.10.3 Future Work 356
9.11 Acknowledgements 359
References 359

Acronyms **367**

Author Index **371**

Subject Index **385**

List of Figures

Figure 1.1: Impact of automation on increasing mental distance between operator input and system output. 3
Figure 1.2: Examples of rules to specify the relationship between the pilot and pilot's associate (PA). 7
Figure 1.3: Adaptive UCAV system diagram. 10
Figure 2.1: Three aspects of the state of development of cognitive systems engineering. 17
Figure 2.2: Naming convention to identify different strains or genotypes ofcognitive systems engineering. 21
Figure 2.3: The cognitive triad model of work-in-context. 27
Figure 2.4: Multiple sources of cognitive demand. 29
Figure 2.5: The abstraction hierarchy. 33
Figure 2.6: A notional abstraction decomposition space for an enroute air traffic control work domain. 34
Figure 2.7: The decision ladder modeling tool used to represent work activity in information processing terms. 37
Figure 2.8: The seven-stage model of activity. 41
Figure 2.9: Human work conceptualized from three different perspectives: An agent view, a task view, or a domain view. 45
Figure 2.10: The CACSE conceptual representation of cognitive engineering as a human-centered development process that may be integrated with a software engineering development process. 57
Figure 2.11: An example of a hierarchal task analysis representation for a volunteer scheduling system. 62
Figure 2.12: A partial GOMS of the Atropos system using the GLEAN 3 GOMS formalism. 63
Figure 2.13: An abstraction hierarchy model of the student volunteer scheduling work domain. 65
Figure 2.14: A structural means-end relation embedded in an AH representation. 68
Figure 3.1: Capturing cognitive systems engineering. 86
Figure 3.2: A general framework of cognitive systems engineering. .98
Figure 4.1: A practice-centered approach to research. 123
Figure 4.2: Discovering patterns in cognition at work. 124
Figure 4.3: Leveraging research to generate useful design concepts. 125
Figure 4.4: The envisioned world problem: A moving target for design. 127

Figure 4.5: Mis-synchronized cycles. 129
Figure 4.6: The engine of innovation: Balanced cycles. 130
Figure 4.7: The engine of frustration: A recipe for design by trial and error. 131
Figure 5.1: A sequence of analysis and design steps creates a continuous design thread that starts with a representation of domain concepts and relationships through development of decision support requirements to creation of visualization and aiding concepts and rapid prototypes with which to explore the design concepts. 141
Figure 5.2: A sample goal-means decomposition from Woods and Hollnagel (1987) for one port (primary-system thermodynamics) of a nuclear power plant. 144
Figure 5.3: Functional abstraction hierarchy of military command and control with the "Apply Military (Combat) Power" portion highlighted. 151
Figure 5.4: Geographic map providing the scenario context for the "choose combat power" visualization. 157
Figure 5.5: JOE visualization showing combat power of Red and Blue forces at the time that the 2nd and 4th Blue battalions and 5th Red battalion reach the "choke point" south of the Burke bridge. 158
Figure 5.6: Updated JOE display reflecting the changes in relative combat power based on additional units selected. 160
Figure 6.1: Five general areas of constraints (boxes) work together to shape possible and effective action trajectories (arrows in center area). 171
Figure 6.2: Five phases of CWA with iconic representations of their most familiar analytic products. 172
Figure 6.3: Actual and potential uses of the five phases of CWA over the life-cycle of a complex sociotechnical system. 175
Figure 6.4: (a) A global view of the AEW&C abstraction hierarchy. (b) A sample of functions from each layer of the AEW&C abstraction hierarchy. 181
Figure 6.5: Structure of the group responsible for evaluating AEW&C designs. 182
Figure 6.6: The activity context for AEW&C. 185
Figure 6.7: Some of the control tasks for the surveillance activity of AEW&C with connections to aspects of the underlying work domain analysis. 186
Figure 6.8: Decision ladders linked to show human-system integration. 187

Figure 6.9: Connection between the functional structure of a work domain, training needs, and the functional requirements of training systems. 189
Figure 6.10: The abstraction hierarchy offers a means for tracing the impact of various design decisions on training. . . .190
Figure 6.11: (a) A global view of the F/A–18 abstraction hierarchy. (b) A sample of functions from each layer of the F/A–18 abstraction hierarchy. 192
Figure 6.12: A set of tools used for knowledge acquisition and knowledge representation in each phase of a cognitive work analysis for identifying the design-relevant properties of an information-action workspace. 195
Figure 6.13: A typology of display formats for different levels of abstraction. 196
Figure 7.1: Relationship between the systems engineering (in dark gray) and human engineering processes (in light gray). 206
Figure 7.2: The Perceptual Control Theory model. 210
Figure 7.3: Multiple agents interacting through their influence on shared environmental variables. 212
Figure 7.4: A framework for cognitive work analysis. 239
Figure 8.1: Cognitive engineering methods. 256
Figure 8.2: Relationship between system adaptiveness, human workload, and predictability. 262
Figure 8.3: Initial conceptual prototype for the DERA cognitive cockpit. 265
Figure 8.4: Offensive air functions. 267
Figure 8.5: Task net of decision making. 269
Figure 8.6: Strategy for aiding skill-, rule-, and knowledge-based behavior. 270
Figure 8.7: Decision ladders for control task analysis. 271
Figure 8.8: COGPIT agents architecture. 273
Figure 8.9: COGPIT agents, processes, and tasks. 274
Figure 8.10: COGMON structure. 276
Figure 8.11: The CommonKADS models. 284
Figure 8.12: SASS replanning task process. 286
Figure 8.13: Flow of information across functions 289
Figure 8.14: General architecture for tasking interfaces. 291
Figure 8.15: Prototype tasking interface GUI. 293
Figure 8.16: Structure of the COGPIT analytical and test work. . . .294
Figure 8.17: Generic format of initial decision-aiding taxonomy. . . .296
Figure 9.1: ABAIS system architecture. 319

Figure 9.2: High-level diagram of key demonstration task components. 319
Figure 9.3: Summary of GUI/DSS adaptation strategies. 333
Figure 9.4: Example of an affect state transition diagram resulting from cognitive-dynamic behavior analysis. . . .342
Figure 9.5: Sources of information for deriving pilot's affective state. 343
Figure 9.6: Frames 1 and 2 of the demonstration scenario: No adaptation occurs. 353
Figure 9.7: Frame 5 of the demonstration scenario: Adaptation occurs to enhance visibility and status of ambiguous radar returns. 354
Figure 9.8: Frame 9 of the demonstration scenario: Adaptation occurs to redirect attention and enhance visibility of incoming data to prevent fratricide. 355

List of Tables

Table 3.1: Continuing Objectives 82
Table 5.1: Template of Typied Decision Types and Associated Knowledge Acquisition Questions. 146
Table 5.2: DRs Associated With the "Choose Combat Power" Visualization Concept 154
Table 5.3: Supporting Information Requirements Associated With the "Choose Combat Power" Decision Requirement 155
Table 5.4: Display Task Description for the "Choose Combat Power" Display 156
Table 5.5: Mapping of Graphical Elements to DRs for the "Choose Combat Power" Display 159
Table 7.1: Hierarchical Goal Analysis Four Levels Deep 218
Table 7.2: A Template for Analyzing the Cognitive and Perceptual Aspects of a PCT Loop 220
Table 7.3: Pick Lists for Cognitive/Perceptual Categories Describing Goal-Directed Human Activites 222
Table 7.4: Template for Analyzing Non-Intelligent (Machine) System Components 223
Table 7.5: Illustration of a Completed Template for an Activity Involving Both Internal and External Variables 224
Table 7.6: Illustration of a Completed Template for an Activity Involving Internal Only 225
Table 7.7: Stability Analysis 226
Table 7.8: Analysis of the Upward Flow of Information in the System (Support to Higher-Level Goals) 227
Table 7.9: Classification of Goal Relationships 237
Table 7.10: Some Loose Equivalencies Between Procedures Used in MFTA, CWA, and PCT Systems Analysis 242
Table 8.1: Prioritized Areas of SASS Support and Assessed Case of Supply 285
Table 8.2: Summarized COGPIT Functional Decomposition 298
Table 8.3: Baseline Cockpit Interface Technologies 299
Table 8.4: PACT System for Pilot Authorization of Control of Tasks 301
Table 8.5: Basic Experimental Test Design 302
Table 8.6: Summary of COGPIT Engineering Methods, Tools, and Techniques 305

Table 9.1: Effect of Emotion and Personality Traits on Cognition: Examples of Empirical Findings 321
Table 9.2: Summary of Possible Affective and Belief State Influences on Pilot Behavior 335
Table 9.3: Relationship of CAPTA to ABAIS Architecture Components 337
Table 9.4: Examples of Pilot Cues, Situations, and Decisions 339
Table 9.5: Example Rules Predicting Impact of Anxiety on Performance 339
Table 9.6: Examples of Static Factors Used During User State Assessment 345
Table 9.7: Examples of Rules for Compensatory Strategies Selection 350
Table 9.8: Pilot Information Preference Profile: Categories of Information and Related GUI Modification Options . .353

1 Cognitive Engineering and Its Relationship to Future Aviation Systems

JOHN M. REISING
U.S. Air Force Research Laboratory

ABSTRACT

The discussion of Cognitive Systems Engineering (CSE) presupposes two things. First, that military aviation systems are susceptible to cogminutia fragmentosa. One of the hallmarks of cogminutia fragmentosa is that the user of a system cannot effectively adapt to encountered situations. This inability is due to a mismatch between the human's understanding of the current situation's properties, potentiality, and constraints and the actual state of the real world. One of the best examples of the problem is the difficulty that has been experienced in effectively using automation to aid pilots. The second assumption is that the application of cognitive systems engineering to the design of military aviation systems is a likely palliative for the problem. For example, to improve the situation with automation, it must become more sophisticated in its interactions with the human team members. This more sophisticated and cooperative future automation has been referred to as an Electronic Associate (EA). Designing the EA demands a much better understanding of the human crew so that the EA can work well with the crew. CSE may provide the insights to accomplish this goal. It is vital that useful tools for avoiding cogminutia fragmentosia be developed and validated because pilots of future military aircraft will be confronted with even more capable and complex systems to control. Also, some future systems, such as the Unmanned Combat Aerial Vehicle (UCAV), will be moving the operator completely out of the aircraft. Using CSE should be tested as a means for designing such challenging and important future systems.

1.1 INTRODUCTION

To understand the nature of the following paper, it is first necessary to appreciate the context in which it was written. I was asked to give an opening address to welcome the delegates to the CSE Workshop. Following my address, I was asked to write this paper. The paper reflects my address, and does not discuss the latest developments in CSE. However, it does show some potential payoffs that cognitive systems engineering can bring to the aviation arena.

As I read the title of the workshop, "Cognitive Systems Engineering and Military Aviation Environments: Avoiding Cogminutia Fragmentosa!," I became curious as to the meaning of "cogminutia fragmentosa." I began reading the abstract carefully and came across a major implication—if the user suffers from this phenomenon, "the user cannot properly *adapt* to the situation encountered." Since one of the goals of future systems is to have both the operator and the machine adapt, serious problems can result. Fortunately, the abstract reports that CSE is the "method or tool of choice to respond to work environments that produce cogminutia fragmentosa." Following this logic, CSE will have a major impact on future systems. In the discussion that follows, I hope to show that this premise is indeed true.

1.2 PROBLEMS WITH CURRENT SYSTEMS

Aviation Week and Space Technology (1995) published a special issue regarding automation concerns in commercial aircraft. One of the concerns was automation surprises, that is, the operators did not know why the automation was performing various tasks. They could not penetrate the decision logic. The reason for this problem is shown in Figure 1.1, and it centers around the mental distance between the operator input and the system output.

In older aircraft, there was a direct connection between the pilot's movement of the control yoke, through the connecting cables, to the control surfaces. Later as autopilots came in to the aircraft, the pilots indirectly controlled the surfaces through these devices. Further, with the introduction of flight management systems, more technology was placed between the pilot and control surfaces. The pilot's mental model of how the system worked became severely strained and resulted in a lack of understanding of the relationship between the automation and the control output. Some of these cockpit problems were undoubtedly due to the automation philosophy which drove the task sharing between the crew and the automation.

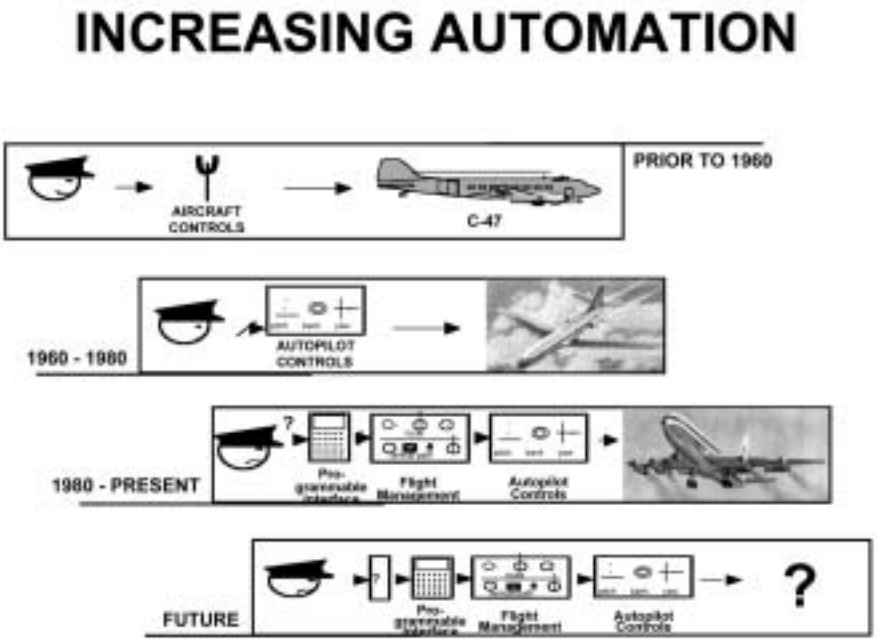


Figure 1.1: Impact of automation on increasing mental distance between operator input and system output.

1.3 PHILOSOPHY OF AUTOMATION

1.3.1 Early Automation Philosophy

"…It appears that the best arrangement is one in which inanimate components work as a team with human operators to provide safe and accurate control and guidance" (Draper, Whitaker, & Young, 1964, p. 5). The key concept in this philosophy is that the operator and the machine form a team. The active participation of the operator, a key component of the teaming arrangement, was reinforced in 1966, in a paper by Knemeyer and Yingling of the Air Force Flight Dynamics Laboratory. They expressed the thought that we cannot expect pilots, in an emergency, to cope with a problem which they have not been following. By monitoring only and not performing, they will fail to notice important information and, consequently, will not react properly (Knemeyer & Yingling, 1966). They went on to assert that automation is desirable, and it should exist not as a conflict but as an aid, an adjunct to the pilot.

For several years (early to mid-1960s) the Air Force research and development community pursued a technically sound and logical solution to the

pilot/autopilot integration problem. The concept was first evaluated by the Germans during World War II and was introduced in this country with names such as "pilot control force steering" or "force wheel steering." The system linked the pilot to the control surfaces through the autopilot by placing electronic force sensors in the control column and rudder pedals of the aircraft. The control pressures applied by the pilot were converted to electronic signals which were sent to the autopilot computer where they were summed with the commands being provided by the basic flight director system to, in turn and accordingly, move the control surfaces. This system provided the means for the pilot, in an emergency or otherwise, to assume control of the aircraft smoothly and in a conventional fashion without having to uncouple or disengage the autopilot. To a certain extent this concept has been overcome by the introduction of computer controlled "fly-by-wire" flight control systems in which the pilot provides inputs to the flight control computer which sums them with inputs from the aircraft's attitude sensors and manipulates the control surfaces to provide the required flight vector. However, this does not provide the pilot with the degree of autopilot control visualized in the original force wheel steering concept.

1.3.2 Today's Automation Philosophy

Despite the work in the early 1960s, the team arrangement design philosophy has rarely, if ever, been carried out in the implementation of automation in today's aircraft systems. To create teamwork, the designer must examine the roles of the human and machine in the system (Gagne, 1962). One of the key components in this process is function allocation between the human and the machine. Ideally this division of responsibilities between the two "team members" occurs by taking into account the strengths and weaknesses, workload limitations, etc., of each and then assigning their roles accordingly. However, in actual system design that is not how the process usually occurs; function allocation is largely a myth and is rarely applied in system design and development (Fuld, 1993). What basically happens is that everything possible is automated, and the human operator gets left with doing what the machine cannot do or what the machine fails to do because of a malfunction.

"Somewhat paradoxically, machines that can do more, and do it faster, provide the basis for systems that are increasingly demanding of the human operator, particularly in terms of cognitive requirements" (Howell, 1993, p. 235). The demand comes about because the operator is "not in the loop," but rather is a bystander—so long as the system functions normally. When emergencies occur, the operator is expected to take control of the system, diagnose the problem, and bring the system back to the nominal state. However, as was discussed in Section 1.1, a design driver should be to make the operators an

integral part of an automated system so they can adapt properly and avoid cogminutia fragmentosa to bring the system back to the nominal state. In other words, what is needed is adaptive allocation. "In this form, a change in allocation is triggered by some change in, for example, the performance level of the human operator. Particular concern is given to prevention of task overload (or, conversely under load) imposed on the human" (Hancock, 1996).

1.3.3 Future Automation Philosophy

It is interesting to note that nearly thirty years after the teamwork automation philosophy was espoused, it has once again come to the forefront. The term currently used is *human-centered automation* (Billings, 1991), which starts with the operator as the heart of the system and then incorporates the automation. From the operators' point of view automation is designed as it should be—to augment or assist operators in areas where they show limitations (Wickens, 1993). Although this automation philosophy is consistent with earlier years, the implementation can be much more difficult because many more avionics systems are contained aboard modern combat aircraft. On the other hand, because of present day capabilities in computers and software, the resulting product can be much closer to a true team. Operator-machine relationships are created which emulate those occurring between two human crewmembers—mutual support and assistance. This support can come about through a sophisticated decision aid which I will now call an Electronic Associate (EA).

1.4 OPERATOR AND ELECTRONIC ASSOCIATE ARE A TEAM

To function effectively, the operator and the EA must work together as a close-knit team. The ideal relationship between operator and machine can be likened to that of the good manager and his staff. The manager must be sufficiently aware of the work of the staff to be able to predict problems, but not so involved that their work is hindered. He must be involved enough to be able to offer assistance when called upon, and yet the manager must not micromanage and risk becoming overloaded and prevented from making the strategic decisions. The good manager will know which staff members can be relied on to act without supervision, as the pilot will form opinions about which of the aircraft systems require frequent attention. As in the conventional management situation, the aircraft's EA system must maintain the knife-edge balance of providing sufficient data exchange without swamping the pilot system manager; allowing sufficient machine autonomy without alienating the manager; and providing sufficient control without overloading the manager.

It must be remembered that the EA will be performing many of the functions now allocated to the second human crewmember in today's military aircraft such as a Royal Air Force's GR–1 Tornado or a U.S. Air Force's F–15E Strike Eagle. The effectiveness of these aircraft is significantly affected by the degree to which the two human crewmembers become an integrated team. Southeast Asia experience with F–4 Phantoms demonstrated this to be the case. With changing crewmembers, a conscious effort to build up the team relationships was critical. The problem arose because crew rest, illness, sortie generation requirements and individual rotation schedules precluded the same team from flying together on each mission. The result was that often a pilot and weapons system officer, who had not worked together before, were paired for a particular mission. Time and again it was proven that learning the individual aspects of each team member became an important factor in a successful mission.

1.4.1 Team Trust

One essential feature of a successful team is trust in the other partner. This in turn implies that the partner behaves in a rational and reliable manner; a partner cannot initiate actions which, even though they are logical to one, appear to be illogical to the other. To avoid arbitrary actions, there must be some overall governing rules which provide the logical structure under which both members operate. As examples of explicitly stated governing rules, consider the three laws of robotics (Asimov, 1950).

- A robot may not injure a human being, or through inaction, allow a human being to come to harm
- A robot must obey the orders given to it by human beings except where such orders would conflict with the first law
- A robot must protect its own existence as long as such protection does not conflict with the first or second law.

These rules provide the guidance required to allow the robot to perform its job in a reasonable and consistent manner. If the word "pilot" is substituted for the word "human" in the above example, a possible basis for governing the behavior of the EA exists. The three laws stated above are only examples of governing rules, and they would require major changes to be applicable in a military setting. For instance, without modification the ideal robot would not allow the pilot to take off, knowing that he was deliberately going in harm's way. The point is, however, that rules of this type provide the basis for consistent behavior for the EA and thereby provide a basis of trust for the pilot. It is through this trust that an effective team can be built. However, trust is not acquired instantaneously; it must be built up gradually. Trust can be envisioned

to develop in three stages. At first trust is based on the predictability of individual behaviors. In the second stage trust is based on dependability. "…Dependability may be thought of as a summary statistic of an accumulation of behavioral evidence, which expressed the extent to which a person can be relied upon" (Muir, 1987, p. 532). In the third stage of trust, faith is the major component because one team member is willing to bet that the other member will be dependable in the future.

Once the trust is built between the crew member and the EA, the continued overall efficiency of the system depends on such factors as machine accuracy, compliance with the suggestions of the EA, and degree of faith in the continued accuracy of the decision aid (Raeth & Reising, 1999; Riley, 1989).

1.5 BUILDING THE ELECTRONIC ASSOCIATE

One of the primary goals of the U.S. Air Force's Pilot's Associate Program was to create a software helper to better enable the operator to cope with the increasingly complex automation. The level of autonomy given to the software helper was that of an associate. "The 'associate relationship' is characterized by a mixed-initiative approach to collaborative problem solving between one or more human actors and a subordinate but semiautonomous computer system with sufficient depth and range of intelligence and capabilities to accomplish a full task domain" (Miller & Riley, 1995).

Analogous to Asimov's three laws of robotics, Principles of Interaction (Lizza, Rouse, Small, & Zenyuh, 1992) were created in the Pilot's Associate

The Effort to Require of the Pilot to Control the PA Must Be Less Than the Effort Saved by the PA.

Plans May Be:

- Approved or Rejected Explicitly
- Approved or Rejected Pre-Mission
- Approved or Rejected Implicitly by Pilot Action
- Ignored, with Predictable Results
- The PA Must Operate In a Predictable Manner.
- All PA Actions Must Be Tailorable by the Pilot.
- The PA Is Required to Monitor the Pilot, Not the Other Way Around.
- The PA Must Follow the Pilots Lead.

Figure 1.2: Examples of rules to specify the relationship between the pilot and pilot's associate (PA).

Program, which would specify some of the operating rules between the pilot and the EA. A partial list can be seen in Figure 1.2.

One of the major ways to assist the operator is seen in the statement, "The PA is required to monitor the pilot, not the other way around." The same philosophy was carried on in the Rotorcraft Pilot's Associate RPA program conducted by the U.S. Army. The philosophy was expanded in that not only would the EA monitor the operator's performance, but would also understand the commander's *intent* and plan accordingly. One way of understanding the commander's intent is to look at the strategies employed in performing particular tasks. Cognitive Task Analysis (CTA) is well-suited for this arena. "…CTA attempts to discover the strategies that people actually use to perform a task, along with the cues and relationships and patterns required by the strategies" (Klein, 1995, p. 530).

The key result of the monitoring and planning is the ability of the EA to off-load tasks from the air crew in times of high workload. For the EA to achieve this, a very detailed cognitive task analysis is required to determine which tasks the crew is required to perform in the course of the mission. Once the task model is programmed, the EA can then monitor the task model as a basis for nominal performance. If the crew falls behind in performing these tasks because of the uncertainties involved in actual combat situations, the EA can then aid the crew by assuming the tasks they shed. The RPA was successfully flight tested, and various functions of the associate were exercised. Since the completion of the RPA program, the U.S. Air Force and the Defense Advanced Research Projects Agency, along with Boeing, have decided to incorporate the RPA's associate's software into the Unmanned Combat Aerial Vehicles (UCAV) program (Dornheim, 1999).

1.6 UNMANNED COMBAT AIR VEHICLE (UCAV) AND THE ELECTRONIC ASSOCIATE

The U.S. Air Force, as well as a number of other countries, has an unmanned reconnaissance vehicle in operational status. The Predator is in the Air Force inventory, and it is planned that another vehicle, Global Hawk, will also soon enter operational status. These types of vehicles are used for a variety reconnaissance and intelligence gathering activities. Their performance in Bosnia has proven to be very worthwhile, and new platforms are under development (*Aviation Week & Space Technology*, 1996; Goodman, 1996). Further in the future, a combat version of an uninhabited vehicle is a strong possibility. In fact, the creation of a combat vehicle was one of the major recommendations of the *New World Vistas (NWV): Air and Space Power for the 21st Century* report. "An effective UCAV will be enabled in the next century as the result of the simultaneous optimization of information flow, aircraft performance and

mission effectiveness" (McCall & Corder, 1995, p. 34). Lest anyone think the inhabited aircraft is a thing of the past, the report goes on to say, "The UCAV will not completely replace the inhabited aircraft for decades, if ever, but the presence, or absence, of a pilot is now a design trade that can be made in a logical way" (McCall & Corder, 1995, p. 34).

One reason these vehicles are being considered is performance. Without having to take into account the physical limitations of the pilot, UCAVs could be designed to pull 10 to 20 times the force of gravity, something not possible in inhabited aircraft (Fulghum, 1996). This increased performance, in turn, enhances survivability for the vehicle. The operator's survivability will also be increased, since he or she will be remote from the UCAV.

1.6.1 Operator's Station

There are at least three different stations being contemplated, depending on the operator's distance from the UCAV—the back seat of a two-seat fighter/attack aircraft (as exemplified by current aircraft such as the F–15E Strike Eagle or the Tornado GR1), one of the crew of an airborne command and control aircraft, or at a ground location. The first would be closest to harm's way since the fighter/attack aircraft could be in the battle area, but somewhat removed from the ground target. The second would be further from the target area and would be in an aircraft flying over friendly territory. The third could be in enemy territory, but just as likely would be in friendly territory at some considerable distance from the target area. In each of the three operator stations, the operators would have to be given information sufficient to provide them with the situation awareness needed to successfully complete the mission.

The ground-based operator's station located far from the battle area may pose the most difficult challenge in providing adequate information to the operators. Various sensor assets would have to be combined with databases to give them both a general overview of battle area as well as a real time "view" of the target area. Head-mounted displays and projection systems are two possible candidates for providing a synthetic 3-D world to the operators. The mode of operator control could be through voice, hand motion wearing a 3-D cursor controller such as a data glove, or some bezel mounted keys on real, panel-mounted displays. Regardless of how the operators interface with the station, physical requirements of the ground-based operator station would be less than its airborne counterpart which has to undergo the rigors of flight.

The overwhelming number of tasks for the operator of the ground-based UCAV control station will be cognitive rather than physical. Being remote from the vehicle, the operator will not experience any of the physical stresses imposed by the flight regime; however, to stay aware of the vehicle status and its progress in the mission, the operator may have a high cognitive load.

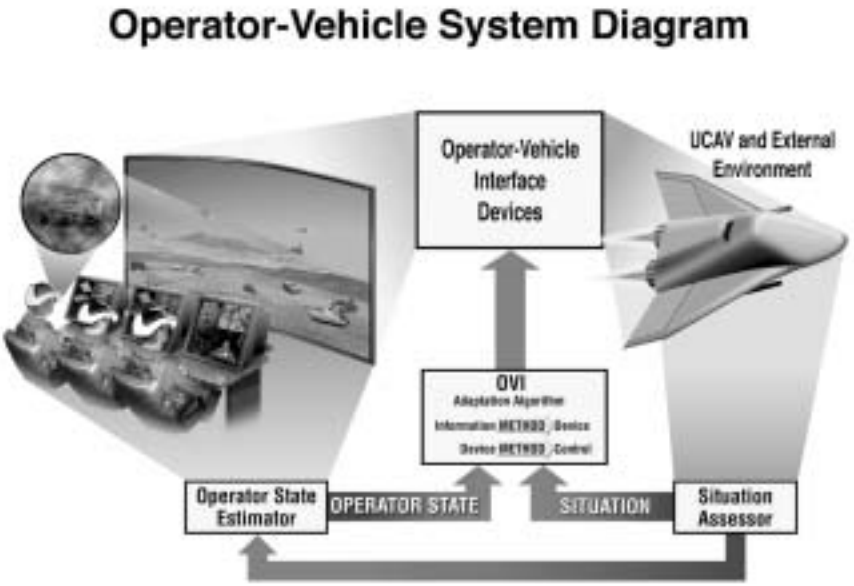


Figure 1.3: Adaptive UCAV system diagram.

Because of the vehicle's level of autonomy, it can perform many routine functions on its own. For example, the vehicle could reconfigure control surfaces if it sustained battle damage. It may also be able to autonomously adapt to certain mission events such as missile threats, thereby placing additional cognitive load on the operator to maintain adequate situation awareness if unexpected events occur. Since the operator's main function may be to intervene in these unanticipated events, cognitive work analysis is crucial if the success of the system is to be achieved (Vicente, 1995).

Would the operator be a pilot? This issue will always be debated, but if the UCAV is designed to be semiautonomous, i.e., preprogrammed to fly a route and avoid threats, but capable of accepting commands remotely, there will probably be little, if any, "hand flying" required. The operator of this air vehicle may not be a pilot and may control more than one vehicle at a time. The system diagram illustrating the entire control loop is shown in Figure 1.3.

Maybe a weapon systems officer would be a better operator? Maybe a targeting or intelligence specialist? However, no matter who the operators are they may have to adapt to changes in the mission capability of the vehicle by issuing additional control commands. These commands are not necessarily "stick and rudder" flight inputs, but rather can be added at a much higher level. For example, a voice command could be issued, such as "UCAV 2, target reassignment. Attack second priority target." These commands are at a much higher

level than just simple course corrections and could place a higher cognitive load on the operator.

One of the crucial aspects of this control loop shown in Figure 1.3 is the ability of *both* the vehicle and the operator to adapt. Not only can the operator adapt to changes in aircraft state, but by monitoring the operator state through physiological and other means such as behavioral task models, the EA contained in the ground-based software can assume tasks if the operator is cognitively overloaded. Since both the operator and the vehicle can adapt, cognitive systems engineering is especially crucial in this operator-vehicle system if we are to avoid cogminutia fragmentosa.

REFERENCES

Asimov, I. (1950). *I, robot.* New York: Fawcett.

Aviation Week & Space Technology. (1995, January 30). Automated cockpits: Who's in charge? *Aviation Week & Space Technology*, pp. 1–24.

Aviation Week & Space Technology. (1996, April 1). *Aviation Week & Space Technology.*

Billings, C. (1991). *Human centered automation* (NASA Tech Memo 103885). Moffett Field, CA: Ames Research Center.

Draper, C. S., Whitaker, H. P., & Young, L. R. (1964, September). The roles of men and instruments in control and guidance systems for spacecraft. In P*roceedings of the XVth International Astronautical Congress*, Warsaw, Poland.

Dornheim, M. (1999, October 18). Apache Tests Power of New Cockpit Tool. *Aviation Week and Space Technology*, 46–49.

Fuld, R. B. (1993, January). The fiction of function allocation. *Ergonomics in Design, 1*, 20–24.

Fulghum, D. (1996, February 5). Pilots to leave cockpit in future Air Force. *Aviation Week & Space Technology*, 26–28.

Gagne, R. M. (1962). *Psychological principles in systems development*. New York: Holt, Rinehart, & Winston.

Goodman, G. W., Jr. (1996, July). New eyes in the sky. *Armed Forces Journal International.*

Hancock, P. (1996, October). The future of function allocation. *Ergonomics in Design, 4*, 24–29.

Howell, W. C. (1993). Engineering psychology in a changing world. In L. W. Porter & M. R. Rosenzweig (Eds.), *Annual Review of Psychology* (pp. 231–263). Palo Alto, CA: Annual Reviews.

Klein, G. (1995). The value added by cognitive task analysis. In *Proceedings of the 39th Annual Meeting of the Human Factors Society* (pp. 530–533). Santa Monica, CA: HFS.

Knemeyer, S., & Yingling, G. L. (1966, November). Results of the USAF pilot factors program as they apply to flight safety. In *Proceedings of the 19th Annual International Air Safety Seminar,* Madrid, Spain.

Lizza, C., Rouse, D. , Small, R., & Zenyuh J. (1992, July). Pilot's Associate: An evolving philosophy. In T. Emerson, M. Reinecke, J. Reising, & R. Taylor (Eds.), *The Human-Electronic Crew: Is the Team Maturing* (Tech. Rep. WL–TR–92–3078). Wright-Patterson AFB, OH: Wright Laboratory.

McCall, G. H., & Corder, J. A. (1995). *New World Vistas, Air and Space Power for the 21st Century. Summary Volume.*

Miller, C., & Riley, V. (1995, December). Achieving the associate relationship; lessons learned from 10 years of research and design.. In J. Reising & R. Taylor (Eds.), *The Human-Electronic Crew : Can We Trust The Team?* (Tech. Rep. WL–TR–9–3039). Wright-Patterson AFB, OH: Wright Laboratory

Muir, B. M. (1987). Trust between humans and machines, and the design of decision aids. *International Journal of Man-Machine Studies, 27*, 527–539.

Raeth, P., & Reising, J. M, (1999). Transitioning basic research to build a dynamic model of pilot trust and workload allocation. *Mathematical and Computer Modeling, 30*, 49–165,

Riley, V. (1989). A general model of mixed-initiative human-machine systems. In *Proceedings of the 31st Annual Meeting of the Human Factors Society* (pp. 124–128). Santa Monica, CA: HFS.

Vicente, K. (1995). Task analysis, cognitive task analysis, cognitive work analysis: What's the difference? In *Proceedings of the 39th Annual Meeting of the Human Factors Society* (pp. 534–537). Santa Monica, CA: HFS.

Wickens, C.D. (1993). *Engineering psychology and human performance*. New York: Harper Collins.

2 Cognitive Systems Engineering at 20-Something: Where Do We Stand?

ROBERT G. EGGLESTON
U.S. Air Force Research Laboratory

ABSTRACT

Cognitive Systems Engineering (CSE) is a multidisciplinary field that is both a scientific endeavor and an engineering practice that emphasizes user-centered analysis and design. This chapter presents an analysis of the state of CSE that concentrates on its technical development. The analysis considers the field from three different perspectives: conceptual foundations, engineering tools and practices, and deployment and use in systems design. Four distinct cognitive engineering frameworks that span the field are identified and discussed as a way to make clear similarities and differences across the conceptual landscape that serves to define the field. The CSE "toolkit" is briefly discussed in terms of knowledge elicitation, knowledge capture, and design innovation support. Some areas of current confusion and ambiguity are also presented as a means to highlight possible impediments to continued growth and technical maturity of the field. The chapter concludes with comments about the apparent direction of CSE developments as they relate to large-scale system design.

2.1 INTRODUCTION

Cognitive Systems Engineering (CSE) is both a field of scientific study and an approach to human-centered engineering practice for the design of interactive systems. While it has been defined in many different ways by various authors, at its core CSE seeks to understand how to model work in ways *directly useful for design* of interactive systems. Further, it seeks to develop concepts, principles, tools, and methods that enable work-centered design engineering of interactive systems. As an engineering practice, CSE exploits its scientific understanding and principle-based engineering toolkit to produce design products

that feed into the full interactive systems development process. Products range from models of interaction, at several levels of detail, to quantitative estimates of performance for design configurations, to the innovation of support systems derived from the CSE body of knowledge.

This chapter is a commentary on the field of CSE[1]. It evolved from a discussion on the "maturity" of the CSE field that I led at the TP7 Workshop on CSE. The purpose of the commentary is to consider the state of development of CSE as a cross-disciplinary field of study and practice.

CSE has followed a logical progression in its development. In the early years, major attention was directed toward establishing the conceptual foundations of the field. Leaders of the field concentrated first on establishing the constructs used to characterize work and second, on the concepts for how to utilize these characterizations to model work in ways that would be directly useful for the practice of engineering design. This conceptual foundations period, which occurred largely in the 1980s and earlier, was followed by a time when the focus of attention shifted to extending models of work and creating new data collection and analysis tools that made more explicit the cognitive aspects of work as practiced. Utilization of these knowledge elicitation and cognitive task analysis processes represents a major part of CSE as an engineering practice. This engineering practice period has dominated the 1990s. As we embark on the 21st Century, CSE seems poised to move into a third period or stage. There are early signs that the new focus will concentrate on how to improve the integration of CSE practices and products into systems engineering as an integral part of systems analysis, management, and design engineering practices for large-scale, interactive systems development.

In an effort to understand the state of CSE today, we will explore these three aspects of CSE development. The first task is to lay out the conceptual landscape of the field. Toward this end, I have identified four unique genotypes of CSE that represent its foundation and that also can provide a conceptual understanding of the field for the reader who has little prior background in this area. While more genotypes could be discussed in detail, the selected set is sufficient to span the conceptual landscape, and therefore reveal important similarities and differences among the various approaches being adopted by practitioners in the field. Thus, a review of these genotypes can be used both to add structure to an otherwise confusing landscape and to denote progress and maturity of the field in different ways. The discussion starts with a historical overview to uncover the reasons behind each genotype and then concentrates on detailing the conceptual distinctions fostered by these frameworks.

To be useful as engineering tools, the various CSE frameworks must be populated with information germane to a specific work domain in the context of a specific systems development project. With such information, CSE is able to support decision-making throughout the analysis, design, and evaluation processes of systems development. The second part of the chapter considers the

1. Both the terms Cognitive Engineering (CE) and cognitive systems engineering have been used in the literature to identify this approach to human factors engineering and engineering psychology. Hence, these terms may be treated as synonyms. Sometimes cognitive systems engineering is identified with a particular approach to CE, and this can add confusion. In this chapter I will use the term cognitive systems engineering to identify the field.

development of CSE from this engineering practice perspective. It begins with a look at each CSE genotype in the context of performing a cognitive task analysis in support of systems engineering. The discussion continues by looking at the more direct design and evaluation support aspects of CSE. As part of this presentation, CSE development is considered from two perspectives. The first perspective considers progress in terms of the innovation of new concepts and tools offered by CSE to improve systems engineering for more cognitively intensive human-machine systems. The second perspective considers progress in terms of the deployment or use of CSE concepts and engineering tools by CSE developers and practitioners. If the concepts and engineering tools are well formed, then it follows logically that they should be used with a higher degree of reliability. If the constructs are less mature or poorly communicated, then we can expect inconsistencies in deployment due to lack of clarity of concepts, limitations in supporting tools, or limitations in educating users about important distinctions. These issues are briefly considered in the discussion mainly by way of a few examples of CSE concepts from the different genotypes.

In summary, the commentary answers the question, *Cognitive systems engineering at twenty-something—where do we stand?* in three interrelated ways. As shown in Figure 2.1, it reviews the progression of major conceptual distinctions that serve to define the conceptual forms and theoretical maturity of the field. Second, it identifies recent additions to the CSE engineering tool kit that are motivated by these conceptual distinctions to see the maturity of the engineering processes. Finally, it looks at how CSE has been deployed as a means to see its maturity in terms of extent of use and clarity of application of CSE methods in systems development.[2]

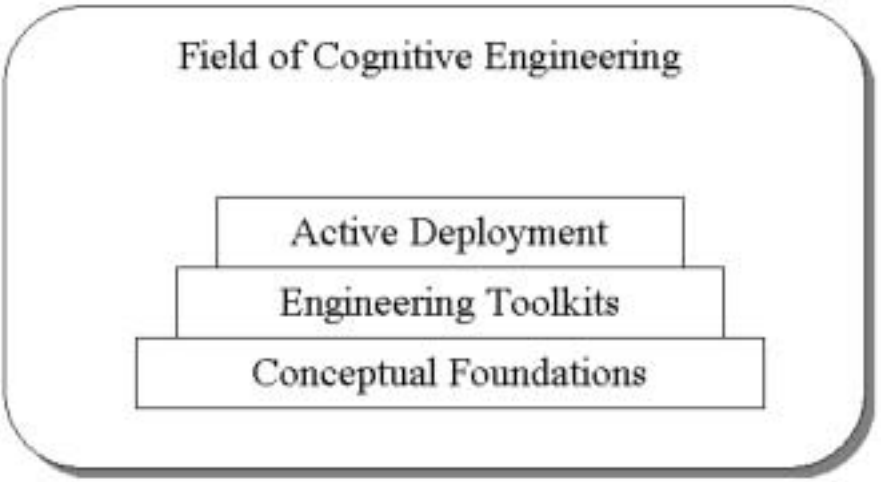


Figure 2.1: Three aspects of the state of development of cognitive systems engineering.

2. Although this commentary reviews several developments in CSE, the chapter is not intended to be a comprehensive review and survey of the field. Items have been selected to show the progression of development and the current state of CSE in terms of the three identified aspects.

2.2 WHY COGNITIVE SYSTEMS ENGINEERING?

CSE is what I do. This sentence is a paraphrase of how Donald Norman, who coined the term *cognitive systems engineering*, chose to introduce this new field of study and engineering practice (Norman, 1981, 1986, 1987). It reflects the fact that cognitive systems engineering was born as an intensely personal endeavor. Norman recognized that the movement of computer technology from the data processing center to the desktop was a harbinger of the change in work that could be expected for a large segment of the work force. In the future (i.e., today!), humans would spend much more time doing cognitive work (e.g., planning, analyzing, problem solving, decision making, negotiating, and coordinating), and computer-enhanced tools would accomplish much of the physical work (e.g., crunching numbers, assembling components, moving items, etc.) under supervisory control. Trained as both an electrical engineer and a cognitive psychologist, Norman had a strong interest in influencing the design of these new computer tools from the perspective of human performance. But the existing database regarding human performance in terms of cognitive factors, derived from engineering psychology, human factors engineering, computer science, and cognitive science, did not seem adequate. In his view, new tools were needed that concentrated at once on the fundamental understanding of cognitive work and on user-centered engineering of work support tools. This new field, CSE is thus a composite of science and engineering. It places an emphasis on a theoretical understanding of cognitive work, as well as on engineering design innovation and systemization to produce interactive systems that better support work in all of its dimensions. As Norman put it, it is the field of *applied cognitive science* (Norman, 1986).

Norman was not alone in his observation that new thinking and tools were needed to enable safe and productive work in the computer age. At about the same time David Woods and his colleagues, working in the power generation industry, called for a cognitively oriented engineering design practice to incorporate machine-based decision support in the control room (Hollnagel & Woods, 1983; Woods, 1986). Incidents like the Three Mile Island nuclear disaster heightened world awareness of the importance for good design in the operator control room of large, complex, and potentially dangerous production processes. The Woods group became aware firsthand of how the computer served to increase the cognitive distance between the operator and the details of process control. Further, this awareness was not limited to scientists and engineers in the United States. In fact, Jens Rasmussen, working quietly at the Danish Atomic Energy Commission Research Establishment (RISO), foresaw the need for a CSE long before the term was coined (Rasmussen, 1976). Rasmussen's work dates to the late 1960s, but in general was not published in widely available open source literature until the 1980s. He saw the problem in terms of a need to support adaptive problem-solving behavior as a means to

ensure safe operation of a power production facility. To him, currently accepted approaches to reliability engineering at the time did not adequately address the factors, cognitive and otherwise, that enable an operator to make adaptive and effective adjustments to meet evolving contingencies. Also in the 1980s, Stewart Card was motivated to attempt to convert advances made in the scientific studies of cognitive psychology and artificial intelligence into engineering models of human behavior that more adequately represented cognitive factors. Existing analysis tools used by human factors practitioners tended to gloss over these mental operations. As a result, cognitive characteristics of work tended to be under-represented in the analysis and design of complex person-machine systems. Card attempted to redress this problem by explicitly including cognitive variables in information-processing models of human performance.

It is evident from the foregoing that many individuals involved in the engineering research and development community were converging in the 1980s on the same conclusion: *There is a growing need to incorporate a deeper understanding of cognitive behavior into the design of modern work tools and systems.* I take this convergence as marking the formation of CSE as a new field even though, as noted, some lines of work extend back further in time. While not inclusive of all research that could rightly be included in this incipient field, these four lines of work (Norman, Woods, Rasmussen, and Card) certainly deserve to be considered as members of the central core of the CSE movement.

There are at least three important common characteristics across these four research and development programs. All of them were driven by the need to solve real-world work problems that required an effective interface between humans and machines or large machine-based processes. They all recognized the need to ensure that technology is designed to *aid the user in solving problems in work.* Finally, they all saw the need for CSE as both an applied science of joint cognitive systems and an engineering methodology or practice for the design of such systems. Thus, CSE would need to mature as a science and as an engineering practice to meet the expectations of its founders. Indeed, this has been the case.

CSE is now, by my reckoning, twenty something. Looking back, we see the 1980s as a period when considerable effort was devoted broadly to establishing conceptual foundations for the field. Researchers struggled with basic questions regarding the nature of work and how to make the user interface pleasant to use or "nonintrusive." In short, it was a time for constructing the CSE conceptual toolkit. Emphasis appeared to shift around the beginning of the decade of the nineties. With tools in hand, albeit sometimes in rough form, and with some design engineering experience, the deployment of CSE grew to a more sizable level of participation in large-scale projects and the development of support system prototypes to address hard and complex problem-solving situations in the context of ill-defined, high risk work. Thus, during this period we can see how the conceptual underpinnings of CSE are used to improve

design engineering tools and practices while simultaneously creating design innovations. In some instances, the coupling of engineering tools to theoretical development is tight and obvious; in others it is looser and less obvious.

It should come as no surprise that each of the four mentioned variants of CSE held different views on the conceptual and engineering tools needed in the field. The founding scientist-practitioners provided a hint of these differences by way of the variations in motivation for addressing cognitive work. These variations highlight the fact that CSE comes in different colors or flavors, based on different theoretical views. To contribute to the fundamental advancement of the field, one must understand these differences. Second, because the theoretical divide has proven to be quite large in CSE, if these differences are not understood the larger acquisition and development community will be less able to make informed judgments about how to capitalize on what CSE can offer to systems design and development. Third, there are differences and inconsistencies in how some common terms that span the entire field of CSE are used. These differences stem, in part, from the meanings ascribed to the same or similar terms by the different variants. Thus, practitioners and users alike can avoid confusion by having a better understanding of these nuances. In sum, variants within the CSE field define different frameworks that provide the conceptual landscape for the field. An inspection of these frameworks, therefore, provides a window into the maturity of the field: What types of frameworks have been established? What are the key differences between them? To what degree have they been formalized?

2.3 COGNITIVE SYSTEMS ENGINEERING GENOTYPES

The remainder of this section provides further elaboration of the various threads that serve to demarcate the CSE conceptual landscape. I call the frameworks that have evolved from the four identified starting points CSE genotypes, or simply genotypes for short.[3] Given the differences in theoretical stance and perspective among the four strains, this seems like an appropriate use of the term. Each genotype is labeled by using the name of the institution that reflects the affiliation of the principal champion at the time a strain was established, with one exception. The Woods group was originally formed at the Westinghouse Research and Development Center; however, over time this group generally has been recognized as associated with The Ohio State University; hence I will use OSU to label this brand of CSE. Following this scheme, the other three genotypes are CMU, RISO, and UCSD (see Figure 2.2). They will be discussed in alphabetical order.

3. Kim Vicente has also used the term genotype as a means of distinguishing between perpectives of CSE espoused by Jens Rasmussen and David Woods. Here I have followed his lead and adopted the term for a more comprehensive analysis of the field.

CSE Genotypes		
Name	Origin	Champion
CMU	Carnegie Mellon University	Card[4]
OSU	The Ohio State University	Woods[5]
RISO	RISO National Laboratory (Denmark)	Rasmussen
UCSD	University of California at San Diego	Norman

Figure 2.2: Naming convention to identify different strains or genotypes of cognitive systems engineering.

2.3.1 CMU Genotype

The expressed aim of this genotype is to establish a unified framework that is able to reflect scientifically founded knowledge about human performance in a manner that is directly useful for "engineering style" calculations used in the design and development process for interactive systems. To achieve this goal, Card and his colleagues proposed to develop "problem-oriented" models of human performance. These models would provide design engineers a vehicle by which they could easily represent specific work tasks in terms of human information processing. Ideally, a model would provide a quantitative analysis of projected human performance for any particular task. Thus, such a model could be used to make quantitative predictions with regard to design factors, or for the comparison of alternatives. In addition, human-centered models could also be used to help formulate alternative task structures in a manner that would make explicit the demands placed on human physical and cognitive processing resources. If task structure and system design constraints are already formulated, then the model could be used to access the physical and cognitive cost of achieving the task with those conditions.

To achieve this goal, Card, Moran, and Newell developed two cognitively oriented modeling frameworks; one directly focused on the human and the other focused on the task expressed in human problem solving terms. The Model Human Processor (MHP) uses knowledge, derived largely from cognitive psychology and cognitive science, to specify elements of an information-processing architecture in a computable form (Card et al., 1983, 1986). For example, the model specifies elemental components of and derives processing times for human perceptual, cognitive, and motor activities. The architecture considers the ability and constraints of the human to mentally chunk, forget, store, and retrieve information and to reason with information.

While it has become traditional to refer to architectures of this nature as cognitive models, in fact, they include perceptual and motor processes also. The emphasis placed on cognition is due to the fact that earlier information-processing models used in design engineering tended not to make cognitive fac-

4. Card was affiliated with CMU at the time he was working on the Human Model Processor and GOMS modeling methods. He susbsequently moved to Xerox PARC, but the CSE work has been sustained at CMU by others.

5. Woods and colleagues were affiliated with the Westinghouse Research Center when their work on CSE began. However, it has been sustained largely by Woods and colleagues at OSU since that time.

tors explicit, although there are some exceptions.[6] Models based on information theory or control theory, for example, consider the human as an (signal) information processor but the characterization resulting from these theoretical frameworks did not address details of cognitive work (D'Azzo & Houpis, 1966; Shannon & Weaver, 1949). Thus, CSE constitutes a "new look" at an old problem in human factors engineering: how to model the human in terms useful for design engineering. The new look from this genotype emphasizes the explicit modeling of mental operations, as opposed to using a single global construct like information. One could argue that this new look dates at least to Broadbent's (1958) seminal work on attention, but a more thorough cognitive development did not occur until Neisser (1967) and Simon and Newell's work on production systems (Newell & Simon, 1972). Card, Moran, and Newell (1983) represents the first time someone attempted to transition these new gains in understanding human cognition into engineering tools for the use of designing better interactive systems.

An important aspect of the MHP framework is that it provides a standard set of parameterized information processing primitives and a fixed process architecture that includes cycle times for three independent processes (perception, cognitive, and motor). As a result, it has the means to produce models of human behavior founded on established scientific understanding of human information processes. Card et al. (1983) carefully selected primitives that span information processing for an intact agent or user and established effective approximations and parameters for these components in a manner that was faithful to the scientific database, while simultaneously avoiding unsolved and controversial scientific issues in the study of human cognition. Because the model served to integrate, in process form, knowledge from three diverse areas of behavioral research, the MHP can be used to simulate complete human behavior. In effect, the model becomes a unified theory of behavior that emphasizes cognition. Thus, it provides the design engineer with a tool that can be used to analyze task work in terms of perceptual, motor, and cognitive factors. In this way cognitive factors can be included in the assessment of a range of system design issues.

The second modeling framework developed by this group goes by the acronym GOMS, which stands for Goals, Operators, Methods, and Selection rules. Whereas the MHP framework represents a synthesis of knowledge on human information processing, GOMS represents a synthesis of what is meant by the term work or task from a cognitive perspective. In general terms, work is conceptualized as being a goal-driven activity that is attained through rational action. Further, work is regarded as problem solving. These views are captured in the *Problem Space Principle* and the *Rationality Principle*. The Problem Space Principle is predicated on the notion that a person's activity can be characterized as applying a sequence of actions to transform an initial state into a final goal state. In other words, work is a problem to be solved, and problem

6. It is traditional to call models like the MHP cognitive models or cognitive architectures because of the emphasis placed on human thinking processes. They usually devote some attention to other aspects of information processing such as perception and motor processes as well, but the treatment may be less well developed. Thus, they may more correctly be called models of human behavior; however, we will follow established tradition and use the term cognitive modeling here.

solving is regarded as a search process that transforms an abstract, cognitive goal into a physical goal that can be obtained by a sequence of mental and physical actions. The second guiding concept, the Rationality Principle, asserts that a person will develop methods to become efficient at problem solving, given the structure of the task environment, human knowledge, and processing capabilities and limitations. The GOMS constructs implement these principles.

In GOMS, a goal is defined to be what a person is trying to accomplish at a functional level of description. Thus, it is a symbolic or abstract expression stated in natural language that specifies what to accomplish but not how to accomplish it. A goal statement, therefore, acts as a top-level task definition. (At this level of abstraction, a goal statement of work is equivalent to a functional statement of a system in user-centered terms.) The remainder of the GOMS model expresses how cognitive operators reduce this abstract understanding to actions taken in the world. The reduction process involves a cycle of cognitive activity that is conceptualized in terms of the goals, methods, operators, and selection rules. An operator transforms a goal into a subgoal that is "closer" to the desired end state. A sequence of subgoals is often needed to traverse the problem space. A method is a sequence of operators and subgoals to accomplish a higher-level goal. A method acts like a chunk of knowledge that can be activated as a single unit. Its formation and use is dependent upon prior learning. It may be thought of as a procedure. Selection rules represent knowledge about which method to apply when there is more than one method available to complete the task. They are the control mechanism for the modeling framework. The cycle of decomposition terminates when execution of a subgoal directly yields an elemental action that achieves the desired goal state.

To illustrate the GOMS framework, Card et al. used the problem domain of text editing. Consider the situation where one wishes to delete a phrase. DELETE–PHRASE, then, may be an expression of the task and, hence, the active goal of the agent. To achieve this goal, the agent invokes a series of operators, like: MOVE–MOUSE (to beginning of the phrase), CLICK–MOUSE–BUTTON, WIPE–LETTERS, RELEASE–MOUSE–BUTTON, HIT–DELETE–KEY.[7] With experience, one may chunk this sequence of operators into a method that can be called the *MARK–AND–DELETE* method. The text editor may support other methods, as well. For example, the same task could be accomplished by following a *DELETE–CHARACTER* method (i.e., deleting one character at a time). This method would have to be invoked n times until no more characters were present. Now, a writer (agent) may (implicitly) formulate a heuristic that says: use the *DELETE–CHARACTER* method if the number of letters is $\leq x$; otherwise use the *DELETE–PHRASE* method. This heuristic acts as a selection rule. A GOMS model for a task consists of a description of a sequence of operators, methods, and selection rules like this to achieve the initiating task goal.

7. For any specific model, the analyst describes operators at a level appropriate to its planned use.

Within this basic structure, a GOMS model incorporates information processing capabilities and limitations of the actor associated with activating goals, operators, methods, and selection rules. In its simplest form, an analysis of the cognitive demand of any identified GOMS model can be expressed in terms of cognitive processing time for each GOMS construct. In other words, each construct is treated as an activity that consumes human processing resources (cognitive, perceptual, or motor) that is expressible in units of time. Task time then depends on the set of mental activities included in the problem solving process. Prior to learning a method, the model predicts some task performance time x. After learning, one or more methods may be formed; the model predicts a new task time y.

The initial family of models developed by Card et al. was strictly hierarchal in form (i.e., a linear sequence of GOMS operators). Subsequently, the GOMS family has been expanded to include models that support parallel processes and a more detail explication of cognitive processing. These include CPM–GOMS and NGOMSL. CPM–GOMS represents parallel processes for Cognitive, Perceptual, and Motor (CPM) activity (John Kieras, 1996a, John and Kieras, 1996b). For a CPM–GOMS model, operators for each type of process are arrayed in time, based on meeting constraints of the task and the internal dependencies between operations within the agent. Further, all operators have a time parameter—how long it takes them to execute. The task is then analyzed in terms of interconnections across these three streams of mental or human information processing operations. An important aspect of the analysis is to inspect the set of operators to discover which operations of the different class types can occur in parallel and thus co-occur during a time interval. As a result of this arrangement, a specific type (cognitive, perceptual, or motor) of operator may be initiated and completed before another co-occurring operator of a different type completes its work. In effect, one operator may be completely or partially "covered" in time by another one. Therefore, total task time depends on an analysis of this complex web of mental activities. For example, an agent can be reading and processing text while simultaneously moving the mouse into position. The portion of mouse movement that co-occurs with text reading is subsumed by the longer reading time, and thus it is not reflected in the final time estimation for the task. In this way, a time- and constraint-sensitive critical path is formed in a CPM–GOMS model to predict task performance time. The acronym CPM carries the dual meaning of cognitive, perceptual, and motor, and critical path method in CPM–GOMS.

NGOMSL is a structured natural language notation for representing GOMS models that includes the constructs and form of a cognitive architecture called the Cognitive Complexity Theory (CCT) (Bovair, Kieras, & Polson, 1990; Kieras, 1988; Kieras & Polson, 1985;). The CCT includes mechanisms for internal cognitive operators for such things as adding and removing information from working memory or setting up subgoals. The notation provides a

structural way to organize external keystroke-level operators and the more detailed internal cognitive operators. The relation between the external task and internal human processing operators is direct and thus can be used to produce an integrated and computable production system (Newell & Simon, 1972). Hence, the NGOMSL system provides a means for generating GOMS models in computational form. The system can be used to represent a task at multiple levels of detail. Because of the requirements of the computational architecture, NGOMSL enforces a more rigorous development process for creating GOMS models than is true of the other members of the GOMS family.

What have we learned about the conceptual orientation of this genotype of CSE? Clearly the heart of this genotype is a modeling framework aimed at representing, in some form, the mental activities predicted to be involved in completing a task. In broad terms, the conceptual focus is on work as a cognitive activity. More specifically, the conceptual logic of the CMU genotype can be summarized in the following way:

- Agents engage in work
- Problem solving involves cognition (often a large component of work)
- Any work can be modeled as problem-solving activity
- Once a goal is established, problem solving can be achieved by a goal decomposition process (i.e., goal-bound task is the focus of analysis)
- Formal models of work can be formed within a GOMS framework
- These models can be analyzed in terms of cognitive activity, and
- Thus they can predict cognitive work based on a task design.

The formal constructs of the CMU genotype are contained in the GOMS formalisms. MHP concepts are loosely associated with the original GOMS formalism, but they are more tightly associated with the CPM–GOMS formalism. The analyst who builds a GOMS model is free to select the level of resolution at which an operator or method is expressed. In this way, a GOMS model can be expressed at a level of approximation that is deemed by the analyst to be useful to meet the needs of specific design engineering conditions. When appropriate, more detailed models can be generated.

It is useful to note that the modeling strategy treats elements of the *work domain* (features in the environment) as properties of *work activities*. For example, a MOVE–MOUSE operator incorporates a domain object in activity language. The mouse object is not considered independently from the mouse-moving activity. Thus, work domain items are included only as elements of activities in the formalisms used by this genotype. Other CSE genotypes follow strategies that place emphasis on preserving the distinction between work domain objects and work activity objects.

It is also important to note that this genotype implicitly regards work as a collection or set of tasks ($n \geq 1$). As stated earlier, a task is defined by the acti-

vation of a driving goal state. A task exists for the duration of time between the initial activation of a driving goal and its termination by achieving a final goal-satisfying state.

2.3.2 OSU Genotype

The impetus behind this genotype was the desire to explore the use of emerging computer technology to produce decision support systems to aid control room operators in the electrical power generation industry. Woods (1986) noted that expert systems, a popular application of artificial intelligence research in 1980s, considered decision support in terms of using computer (reasoning) automation to produce answers for user consumption. That is, a knowledge-based or cognitive machine was designed as a tool for use by a worker. The expert system takes input from the worker, uses its knowledge and produces an answer for the user, often after periodically asking the user to clarify or provide more information. The knowledge-based reasoning output generated by the expert system is then available for use by the human. Woods took the stance that cognitive tools would receive better reception and provide more effective aiding if they worked as partners *with* the human worker in the process of decision-making and problem solving. In other words the human and the machine both develop an understanding of the problem and ways to solve it as a dynamic process, as opposed to the human feeding the expert system and digesting its response. This stance represents a shift from viewing aiding as an outcome that is provided by the intelligent machine toward a view of *aiding as a dynamic work process* involving both sources of cognitive ability—humans and smart machines. Because the process is dynamic and subject to unexpected events, this shift implies that a different kind of relationship must be established between the human and the decision-supporting tool. Hollnagel and Woods (1983) used the term *joint cognitive system* to distinguish this view from the outcome-oriented expert system approach.

Woods noted that workers utilized properties of the workplace, including engineering tools, as cognitive aids. This implies that cognition extends to artifacts in the world, as well as the mental abilities of the worker(s). Therefore, in some sense, cognition is distributed: it is both in the head and in the tools. Thus, to model cognitive work, one must model aspects of the work world and the worker in cognitively relevant terms. Further, this also implies that cognition is situated. Change aspects of the work world and "cognition" (information and its meaning) may also change. These observations, and others, drove Woods to analyze work *in context*. In other words, the interaction between the worker, the work environment and tools in the environment represent the appropriate unit of analysis. This has implication for how one represents the fundamental nature of work, and thus, how it can be supported through design engineering.

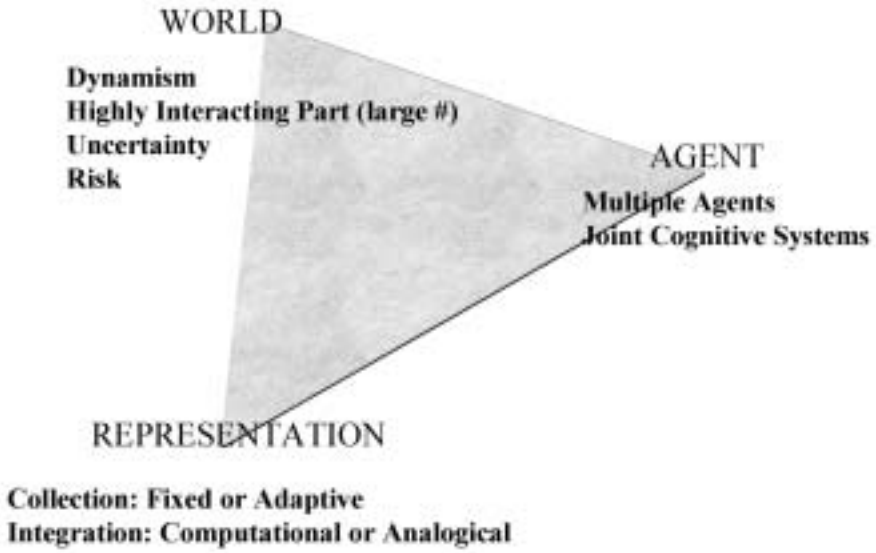


Figure 2.3: The cognitive triad model of work-in-context. (Adapted from Woods & Roth, 1988).

In an effort to represent work in the cognition-in-context framework, Woods developed the cognitive triad concept. This looks at work in terms of the *demand characteristics* of the work world, and considers work as problem solving. As shown in Figure 2.3, the triad is composed of agents, representations, and the work world. The triad represents the problem-solving space in terms of the interactions among agents, representations made visible in the human-tool interfaces, and the work world. The cognitive demand of the world is represented in terms of its dynamism (i.e., rates of change), the number of parts of a problem, and the extensiveness of interactions among parts, uncertainty, and risk in the situation. Risk expresses cognitive demand of the world in agent terms, while uncertainty may be due to both agent independent or agent relative factors. Taken together, these dimensions of the work world serve to characterize the complexity of the world that at once becomes part of the problem to be solved (i.e., the distributed work goal) while acting as impediments to goal satisfaction (i.e., constraints on the driving goal as a problem).

The representation construct addresses the fact that how a problem is expressed in a medium makes certain information or manipulation explicit and available to the agent at the expense of other information available in the world. Therefore, a representation invites a way to think and proceed, even though it does not preclude other ways to conceptualize and act. Thus, it makes some things apparent and cognitively easy and others opaque and cog-

nitively hard. There is ample empirical data to support these aspects of work (e.g., Norman, 1993; Wason & Johnson-Laird, 1972; Zhang, 1992).

The agent construct of the triad acknowledges internal cognitive factors (e.g., memory, reasoning skills, etc.) that impact how an agent interacts with tools and other properties of the work world. These are internal cognitive resources and constraints that must be considered in problem solving.

The cognitive triad represents an embodiment of the concepts, beliefs, and assertions of the OSU genotype. From this orientation, the field of study of cognitive systems engineering, in the broadest possible terms, is *humans at work with tools*. It asserts that worker behavior is influenced by the need to cope with the demands and constraints of the problem-solving world. Thus, work is taken to be problem solving *with tools*. Second, coping with the complexity of the work world itself is part of problem solving. Even though these aspects of the work world are incidental to the purpose-driven problem that sets the overarching goal that guides work in the first place, they are nevertheless part of the problem confronting the worker. This implies that the demand characteristics of the work place must be included in any model of work. The cognitive triad provides a theoretical framework that is able to capture these aspects of work in a general, context-independent way. Hence, it supports a scientific study of work conceived in this manner.

The representation and agent legs of the triad may also be viewed as expressing demand characteristics. By following the same logic, it is reasonable to assert that demand in both of these domains comes from *intended* and *unintended* sources, in the same manner as that found in the work world. Thus, like the work world, the representation(s) of the world available to the worker and the worker's internal demands both contribute to the solution of the work problem and constitute part of the work problem.

Figure 2.4 illustrates this point. The driving goal establishes the external demand from the world, as previously stated. The window to the world provided by tools, such as a computer or remote sensor, intends to make explicit information germane to solving the driving problem or goal. In this sense they are "demands" on an agent to use the available information for this purpose. Further, the agent is motivated and poised to deploy its resources, cognitive and otherwise, to use the available information in the service of solving the driving problem. Hence, these may also be regarded as positive "demands" on the worker. Now, as shown on the other side of the chart, these same sets of sources unintentionally introduce incidental factors that serve to generate negative "demands" that become part of the work problem. The world, for example, may include an array of items that, say, impede movement to a goal location. In a similar way, a representation of the world may be insensitive to certain changes and objects and thus add complexity for the human by requiring reliance on memory to keep them present because they are not immediately available for perception in the representation. The representation, in effect,

	Demand Characteristic	
	Incidental	Intended
World	Context factors	Problem goal
Representation	Hidden data	Direct information
Agent	Collateral states (e.g., emotional control, fatigue, etc.)	Goal-driven motivation; resource focusing

Figure 2.4: Multiple sources of cognitive demand. Some stem directly from the work problem and thus are "intended." Others result froom incidental factors. The locus of the source may be in the world, the representation used in the support artifact, or from the agent.

becomes part of the problem by what it makes opaque. In the same vein, the agent brings demands that unintentionally interfere with main-line problem solving. These include things like personal goals, adverse emotional states (and the ability to control them), and adverse physical states (e.g., fatigue) that all serve to "conflict" with the driving goal of the work problem. These unintended demands act to add complexity that the agent must work around by managing them. Hence, they are part of work even though they are not part of the initiating, driving problem. Thus, the world, representation, and agent are resources to solve a problem, and at the same time, they may, and generally do, add complexity to its solution. Both of these aspects are therefore included in the definition of work.

Pushing the demand language this far may seem a bit forced. Indeed, Woods does not make the incidental and intended demand characteristics distinction explicit, as done here. However, it serves as a useful way to make the point that the complexity of work comes from both factors directly associated with the driving goal of work, as well as from incidental factors that are distributed across the agent, representation, and other elements of the work world. It is for this reason that interactions among these constructs are treated as the unit of analysis by this genotype. In essence, there is a distributed network of goals involved in problem solving work. Thus, to model work, one must account for these nonproblem driven aspects of problem solving.

It is important to recognize that the OSU genotype views decision making and problem solving as related and open activities. That is, in a real-world context, the problem or decision requirement is subject to change while one is working on a solution to the driving problem. Other events occur that may change an aspect of the problem or the very nature of the problem itself. It is in this sense that problem solving is open: The "problem" is not a well-defined stationary task to be completed. It is in part because problem solving is open that we must understand cognition in context, or situated cognition, to design effective support systems to aid users in cognitive work. This view of problem-

solving work clearly contrasts with the view of work from the CMU genotype. A task-based model treats work as closed by limiting its definition to the driving goal. Thus work, as a technical construct, means different things to different cognitive engineers.

In addition to the cognitive triad, the OSU genotype has introduced a set of three interacting and partially overlapping constructs to represent factors that govern a human's intention to act in a work situation. These factors are (1) available knowledge, (2) attentional dynamics, and (3) strategic factors. These constructs were derived from an analysis of the origin of errors and the recovery from errors that have been observed over different work domains, ranging from commercial aviation to the medical operating room. Each factor consists of a set of cognitive variables that can be used as the basis for assessing properties of the work environment, including support machines and tools, and social systems that make work hard or error prone. For example, six cognitive variables have been identified under the knowledge factor: buggy knowledge, mental models, knowledge calibration, inert knowledge, simplifications and heuristics, and imprecise knowledge (Cook et al., 1992). Attentional dynamics deals with traps that invite inappropriate fixations and attentional shifts associated with establishing and maintaining situation awareness. Strategic variables include goal trade-offs and decision choices, including risk assessment. All three types of factors are implicated in the dynamics that shape work performance and that can be used to account for how errors arise in work. For example, Cook and colleagues illustrate how the design of a procedure for setting up an automatic infusion pump controller adds to work complexity and invites an error to emerge due to imprecise knowledge, memory lapse, and poor situation variables (Cook et al., 1992; Woods et al., 1994).

In sum, the theoretical emphasis of the OSU group has been focused on answering the question: What is the anatomy of the cognitive systems engineering problem? They have concluded that the scientific problem is the study of work as problem solving with tools. The engineering problem is how to design tools as (decision) support systems to aid the problem-solving work process. Their analysis indicates that work is open, cognition is distributed, and that dynamical and uncertain factors influence work. Rather than work being defined by the desire to achieve a single, task-defining goal, work is defined by a constellation of interacting goals incident to the work world. The cognitive triad is offered as a general framework to express the work problem to be solved. The knowledge-attentional-dynamics-strategies scheme serves as a general framework that identifies factors that influence intention formation during the course of work.

It should be obvious that these conceptual framing tools of the OSU genotype do not offer a well-formed specification for a process architecture of cognition. Rather, they serve as a context independent language that can be used to represent the work problem in an open manner. In effect, the OSU approach

regards the *actual* work problem to be emergent, and therefore, it is not appropriate to provide a specification for a process architecture to directly express the work problem. Accordingly, they offer frameworks that capture the problem indirectly. This is in marked contrast with the CMU genotype. GOMS is a specification for a process architecture for how to solve a work problem. The OSU genotype would claim that GOMS is a specification for how the *driving problem* can be solved, not a specification for the actual, emergent problem to be solved.

Finally, the OSU genotype may be regarded as an *ecological view of work.* Work is a dynamic process that involves the coevolution of factors from the world, tools, and agents. Again, it is useful to recognize the consequence of this technical distinction for how work is defined. A task approach begins with an analysis of work in the world, as would the ecological approach. However, the goal of the analysis for the task approach is to develop a model of a task that can be adequately expressed in terms of the driving goals or functions to be achieved. In this way the work formulated as a task can, in effect, be "lifted out" of the environment for use in a cognitive model as a basis for analysis. This, in turn, is used to support decision making about artifact design. The OSU genotype has a fundamental disagreement with this position. A single goal-driven task model of work does not preserve the dynamics of the work world. To preserve the open, dynamical attributes of work, it must be modeled in a way that maintains connection of the driving task with the constellation of goals embedded in the dynamics of the work world. A different type of "cognitive language" is needed to properly model work from this ecological perspective. As a result, it would produce a different kind of process architecture for a computational model of work.

2.3.3 RISO Genotype

The RISO genotype has much in common with the OSU strain. It too considers cognitive systems engineering from an ecological perspective. Work behavior is regarded as being shaped by factors in the environment, including the tools available to support work. Problem solving and decision-making are regarded as open processes, and cognition is distributed. Because work is open, in that it must contend with unplanned and unexpected conditions, a worker must be adaptable. This genotype focuses on adaptive behavior and how to support it. Its goal is twofold: to provide an engineering design framework that is able to effectively guide the development of an interactive system that is, first, robust to disturbances and second, error tolerant.

Support for adaptive work behavior is viewed as key to meeting these engineering goals. A system can be made to be robust to disturbances or unexpected conditions if it can behave adaptively in a way that preserves system stability while the driving problem is being prosecuted. Similarly a system can be

robust to error if it can make adaptive adjustments both to detect anomalies and then recover from them during the course of working toward the solution of the driving problem.

Like the CMU and OSU genotypes, the RISO genotype mainly considers work from a decision making/problem solving perspective. That is, the worker is seen as an adaptive decision maker or problem solver. Conceptually, work is considered to be both a component of a sociotechnical system and a characteristic of an agent. Work is both a part of the environment or work domain and the activity of an agent. In other words, the driving work goal is embraced by an agent and comes from the environment. The actual work reflects emergent behavior of the agent. The driving goal and other properties of the environment act as work goals or constraints on work that shape, but do not dictate, actual work. It follows from this view of work that any framework that can be used to model work must represent both the work domain and work activity, and the model must be sensitive to the dynamics in both aspects to understand what makes human work behavior robust and error tolerant.

Rasmussen has proposed a comprehensive framework that treats the work domain, work activity, and the work agent in separate subframeworks that are related to each other. The relation between frameworks acts like a process architecture for a nonlinear dynamic modeling system of work. As a result, the overall framework provides a detailed specification for modeling work, even though it only models work indirectly. This will be made clear in the following discussion.

The conceptual toolkit for the RISO genotype is organized around five interrelated frameworks developed by Rasmussen and his colleagues. For the purpose of discussion, I will refer to them in the following manner:

- Work domain
- Control task (CT)
- Control strategy (CS)
- Social/organizational (SO), and
- Worker competencies (WC).

2.3.3.1 Work Domain. The work domain framework provides a specification for how to model the field of practice in a way useful for design engineering. It proposes to model the work world in terms of two dimensions: a structural abstraction hierarchy and a whole-part or aggregation dimension. Together, these dimensions form a lattice that identifies the aspects of the work domain that are hypothesized to be important to guide the design of interactive systems.

The Abstraction Hierarchy (AH) identifies different types of functional properties of the work domain. A prototypical AH consists of five levels that follow a defined order. Somewhat different generic labels have been used to identify these levels over the years (Rasmussen, 1986, Rasmussen et al., 1994;

Functional Purpose	Domain Properties Represented
Purposes and Constraints	The purposes for which the system is being designed
Abstract Functions and Priority Measures	The intended casual structure of the work environment represented in terms of the flow of values and abstract physical properties
General Functions	Description of the basic process of the system in functional language
Physical Processes and Activities	Characteristics of the physical components associated with these processes and their connections
Physical Form and Configuration	Characteristics of appearances and spatial distribution of physical components

Figure 2.5: The abstraction hierarchy.
(Based on Rasmussen, 1986; Rasmussen, et al., 1994; and Vicente, 1999).

Vicente, 1999). The most recent generic labels are shown in Figure 2.5. The purposes and constraints level of the hierarchy is the place to represent what I have called the driving problem or goal of the domain[8]. It specifies the functional purpose(s) of the work domain. Goals are stated in terms of desired system outcomes. The priority measures level is the place to represent the values of the specific work domain in question. Values act as requirements derived from many different sources. They identify what the work environment regards as important; hence they are given priority. Typical sources of values include management/organizational; social, influence of government and regulatory bodies; and the constraints that derive from natural laws relative to product production (e.g., physical laws of nature). The strength of these value-based requirements generally will be different, and the perception of importance by the problem solver may vary throughout the course of completing an epoch of work. The general function level is used to represent the domain in terms of the general functions that must be achieved to yield the outcome product. It expresses a typical functional model for a system that identifies a set of necessary and sufficient factors to achieve the driving goal, without specifying the form of work production. The physical processes level is used to re-express the more abstract general functional model in terms of the physical variables available in the environment that can be used to implement the general functions. The last level, physical form, depicts the physical spatio-temporal layout of the workspace and its contained resources. It makes explicit the means by which the items at the physical function level of the hierarchy can be achieved. Thus, the physical form model may be regarded as providing objects expected to be

8. In the RISO genotype, the driving problem is considered to be a component of the work domain. At the highest level of abstraction, this problem is expected to map one-to-one and onto the driving problem adopted by a problem solver. In other words, the driving problem acts like an abstract task goal for an agent, as in the CMU genotype, but it remains as a property of the domain. Thus, there is a subtle distinction in the meaning of a driving problem when it is considered from the perspective of the RISO or CMU genotypes.

implicated in work (i.e., the work space as "figure") against the physical layout of the work site (i.e., the work space as "ground").

The second dimension of the work domain framework treats the domain as a system expressed at different levels of detail. It encodes a whole-part relation and is called the aggregation dimension, or as some prefer, the decomposition dimension (Eggleston, 1998; Vicente, 1999). When combined with the AH, the full lattice contains a family of languages, each of which concentrates on the domain as a system in terms of a different aspect, and each aspect is expressed at multiple levels of detail. In other words, the terms used in each row of the abstraction decomposition space serve to form a lexicon of a simple, single-focus "language." The constructs of the language compose a consistent set, with respect to the referent used to define each type of row. Because of this consistency, these sets are well formed and thus if used can help avoid logical paradox, such as the classic barber paradox. The total set of languages helps to provide a more complete picture of a work domain while preserving the logical distinctions among the languages.

A full Abstraction Decomposition Space (ADS) is generally quite large. For the purpose of illustration, only a highly compiled version of an enroute air traffic control domain is shown in Figure 2.6. More detail ADS descriptions would be included in a fully developed model (for more complete examples, see Bisantz & Vicente, 1994; Vicente, 1999).

Our interest for this discussion is on the structure of the framework and its use, and not on the details of the content for the framework for a specific work

	Abstraction-Decomposition Space		
	Decomposition-Dimension		
Abstraction Dimension	System	Subsystem	Component
Purpose	Aircraft Flow Control		
Priority Measures		Speed/Safety/ Economic Topology	
General Function		Aircraft Passage Flow Through Sector	In-bound; out-bound; and internal passage flows
Physical Function			Individual aircraft control states
Physical Form			Physical arrangement of transiting aircraft

Figure 2.6: A notional abstraction decomposition space for an enroute air traffic control work domain. The ADS is formed from a decomposition dimension and an abstraction dimension that follows the abstraction hierarchy.

domain; therefore, the discussion will be limited to these aspects. There are a few things to note. First, it is customary to leave some cells open. For example, there is generally little or no value to providing a highly compiled system level representation of the physical layout. Smaller scale objects are needed to show the important aspects of the spatial and temporal configuration of the workspace. The same holds true for the physical functions. It is important to individually represent the physical variables that can satisfy different functional uses. Thus, they are naturally expressed at a subsystem or smaller-size scale. Only cells useful for design decision-making are populated.

Second, there are many situations, especially early in the work analysis process, when the physical form level can be omitted from the representation without causing a serious loss of useful information. Third, the top four levels of the AH should be taken as the minimum number of levels needed to construct a "full" AH. In other words, the aspects of the work domain reflected in these levels are taken to be necessary to produce an adequate model of a field of practice. Some researchers appear to believe that the necessary and sufficient set of levels to represent a domain varies by domain. Others suggest that the five levels constitute both a necessary and sufficient set. How many levels are required and what criteria need to be used to select the appropriate number for a specific domain remains an open issue. No one has provided a logical or empirical basis that supports the five-level or open-size positions.

A fourth important point about the work domain framework is that it is an object-oriented representation. Even though it describes functional properties of the field of practice, the expression is in a structural form. It does not specify a process model. Rather, it specifies a structural model of the functional landscape relevant to interactive system design. As a result, every element is an expression of work in an event-independent form. It represents a complex structure or domain where activity takes place. It is both where work takes place and a source of resources and constraints that serves to provide the complex texture that emerges in work activity.

Finally, it is important to recognize that each abstraction level serves to completely represent the entire system or domain, in terms of the aspect being expressed. That is, each level or row of the ADS provides a separate and complete model of the work domain. Thus, the entire framework includes a minimum of five interrelated and complete descriptions of the work domain.

2.3.3.2 Control Task. The control task framework of the RISO genotype considers the use of the work domain in activity language. The functional purpose of the domain establishes the necessity for a control task to be accomplished. The control task framework is used to characterize the structure of the event-dependent process to achieve the domain goal. In other words, the control task language is used to describe what needs to be done to achieve the goals of the

work domain, within the prevailing resource and constraint characteristics of the domain. Prototypical processes are generated and expressed as categories of possible ways to accomplish work, as defined by the driving purpose.

Rasmussen has developed a decision ladder to act as the framing tool for control task description (Rasmussen, 1986; Vicente, 1999). As shown in Figure 2.7, the decision ladder characterizes a control task in terms of information processing or cognitive activities and knowledge states. It is important to note that this is not representing the information processing of any identified agent. It is a representation of goal-driven work in activity terms, regardless of who (human or machine) accomplishes it. In essence, the decision ladder takes linear information processing stage model and bends or folds it around the evaluate performance criteria activity, thereby creating an inverted "V" that resembles a ladder. The left side of the ladder considers work in terms of sensory and perceptual activities. The right side considers work in terms of problem-forming and execution activities. Again, it is important to keep in mind that even though the information processing language being used in the decision ladder is normally associated with human information processing, the framework focuses on the *nature of the control task(s)* of the work domain. It is not intended to be a model of how a human or any other specific agent accomplishes the task.

Like the work domain lattice, the decision ladder is a template that can be populated by an analysis of a specific control task. For a specific task, labels for the process activities and knowledge states are used that are relevant for the context. In addition, short cut paths or shunts that are supportable by domain resources, including resources for specified classes of agents, are also expressed in the decision ladder. Labeled arcs between data processing/knowledge state nodes are used to depict these aspects of supportable information processing behavior. For example, an agent may observe data from the environment and based on internal resources (e.g., memory) may leap to the conclusion that a specific task requires action and then proceed to formulate a procedure to accomplish the task. Once populated with context-specific information and known or asserted capabilities of different classes of information processing agents, an activity-based model of the nature of work can be mapped onto the decision ladder template.

As the last paragraph implies, the nature of a control task is not permanently fixed for a work domain. Rather, it is relative to the resources available in the work domain and thus, changes as these resources change. This, of course, includes the resources provided to the domain by any class of agent. It follows that the nature of a control task is more difficult to describe adequately as more resources in the work domain are able to dynamically change themselves. In principle, the decision ladder provides a modeling tool that can address these dynamical properties of work activity in a systematic but open manner.

According to the RISO genotype, an *actual control task form will be emergent.* However, it is expected to have the "stable" characteristics derived from

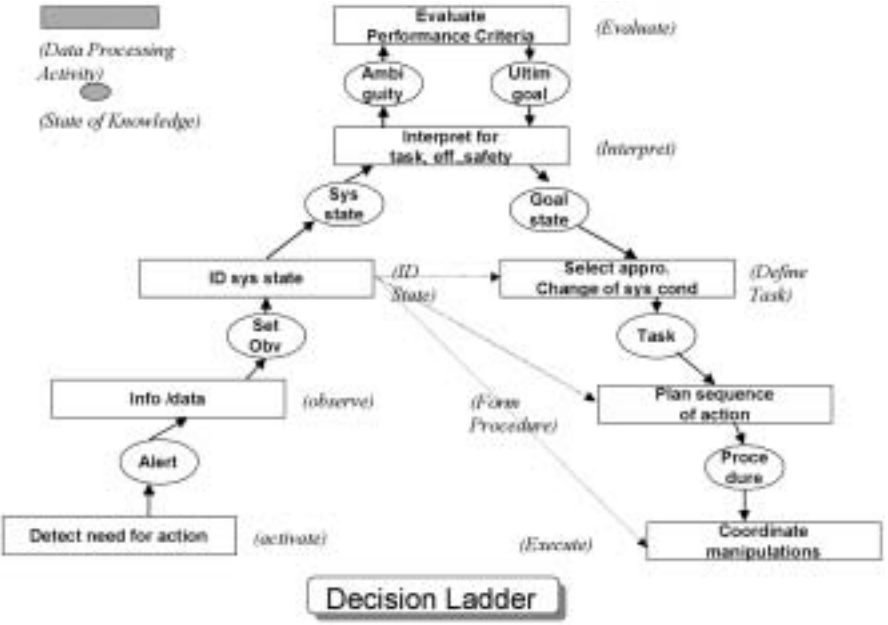


Figure 2.7: The decision ladder modeling tool used to represent work activity in information processing terms. Prototypical, expected, or observed information processing short cuts can be depicted by different style arrows. Three shortcuts are shown for illustration. (Adapted from Rasmussen, et al., 1994).

the driving goal of the domain, relative to possible agent abilities and constraints of the domain.

2.3.3.3 Control Strategies. The control strategies framework follows in the same spirit as that of the control task. It seeks to represent, in an open manner, the intrinsic nature of work strategies that can be used to accomplish a control task. The difference between the decision ladder and the control strategies frameworks is simply one of focus. While the control task model provides the basic shape or shapes of problem-solving work, a control strategy model provides different classes of methods by which these activities may be addressed by a class of agent. Each strategy class requires certain information-processing resources to accomplish the control task in a specified manner. Thus, in this way strategies serve to specify information requirements for agents that can accomplish the control task in different ways.

Rasmussen takes the position that a control strategies framework should be used to represent different reasonable ways an expert could work on the work problem. In other words, even though the control task and control strate-

gies frameworks are open and thus allow an infinite number of strategy types to be formed, for the purposes of guiding design engineering decision making, Rasmussen et al. (1994) recommend analysis be concentrated on the discovery of strategies employed by experts in the work domain.

A modeling tool used to express control strategies is the information flow map (Rasmussen et al., 1994). Vicente provides a nice illustration of how a form of a flow map can be used in this capacity (Vicente, 1999). Strategies may also be expressed as information processing modes and thus can also be represented by a decision ladder.

2.3.3.4 Social / Organizational. The social/organizational framework is like the others in that its purpose is to reveal the structure of the work domain in terms of its social and organizational properties. This includes how work is partitioned among agents and the type of leadership style that pervades relationships and interactions throughout the organization. These factors influence both the necessity for communication and the structural form communication may take, as well as constraints derived from different forms of communication. Both social and organizational factors add to the demand characteristics of the environment, and thus become part of the intended and unintended aspects of the domain that contribute to the nature of problem solving work (Rasmussen et al., 1994; Vicente, 1999).

2.3.3.5 Worker Competences. The last conceptual framework in the RISO genotype is known as worker competencies. As suggested, this framework is used to represent the capabilities and limitations of different classes of agents, human or machine. Again, the objective is to capture intrinsic agent constraints relative to the various shaping factors from the previously developed frameworks. In other words, the other frameworks provide insight into the form of constraints that have implications for what kinds of competencies of an execution agent should be needed to successfully adapt to contingencies and achieve the driving goal. Toward this end, Rasmussen (1983) devised the Skills-Rules-Knowledge (SRK) taxonomy as a framing device. This taxonomy consolidates agent information processing into three alternative cognitive processing modes. Each term in the SRK label reflects one of these control modes: skill-based, rule-based, and knowledge-based control. The knowledge-based control mode is used to represent all forms of deliberate reasoning. Using stored knowledge to form an analogy or logical syllogism is an example of deliberate reasoning. The rule-based control mode, as the name implies, represents rule following. It assumes the existence of rules that can be recovered from memory or activated by recognition (e.g., if the light is red, then stop the car.). The skill-based mode represents control following "natural laws"

(Rasmussen, 1986). An example is the ability to make adjustments to your car's position by noticing a loss of safe distance with the automobile in front of you. The natural law is carried by the perceived rate of closure expressed in retinal size information naturally conveyed by the structure of the leading automobile. In general, the three modes represent different ways the worker interprets information from the environment. It gives recognition to the fact that an object in the environment may have multiple meanings based on how it is interpreted. The skill-based, rule-based, and knowledge-based modes reflect differences in agent processing required to achieve an interpretation based on the communication form ascribed to an object. Knowledge-based control asserts controlled symbolic processing. Rule-based control asserts perceptual sign or cue processing that activates stored rules. Symbolic and sign-based interactions with the world reflect an arbitrary mapping of objects in the world to the symbol/sign interpretation languages (Rasmussen, 1983). Skilled-based control asserts the use of energy-based signals that lawfully map domain objects and events to the agent. Thus, in Rasmussen's terms, when an agent is able to operate based on the pickup of affordances in the domain, it is working under skill-based control (Gibson, 1979).

The key components of the RISO framework are the work domain, control task, and worker competencies subframeworks. All of the frameworks are related. The work competencies maps onto the control task, and the control task maps onto the work domain. In this way, event-independent and event-dependent representations are brought together to produce a kind of "process architecture" for this indirect model of work. An emergent process is dynamically formed between the work domain properties and the work activity properties relative to an active (worker) agent. Like the OSU approach, when populated for a specific domain, the set of frameworks that comprise the RISO genotype express the demand characteristics of the field of practice. This representation goes beyond an expression of the work field as a driving goal to be achieved, as in the CMU approach. As a result, a larger set of factors that contribute to problem solving complexity are included. Thus, the representation is able to account for both intentional and unintentional, or expected and incidental, demands that serve to make problem solving hard.

As indicated at the outset, The RISO genotype is concerned about adaptive problem solving behavior. Because the work domain framework represents multiple aspects of the field of practice, it is able to serve as a comprehensive domain map. As a result, if perturbations or unanticipated events occur, the degrees of freedom in the domain are available as potential (cognitive) resources that can be used by the agent in work activity to form an appropriate adaptive response. While normally not stated, the same argument applies to the other representations in the total framework. The control tasks and control strategies models also include degrees of freedom that can be exploited to meet unexpected demands. In this way the RISO analysis system is able to address adaptive behavior that is

"error tolerant" without specifying a specific adaptive process to be initiated based on specific work conditions. It models work as a self-organizing system using resources in and being constrained by the work environment.

The total framework may be regarded as a nontraditional specification language for a process architecture of work behavior. It is nontraditional because it tries to specify a self-organizing system within a constraint space. As a result, the specification lays out the interacting shaping factors and does not directly specify the emerging work process. Rather, it attempts to specify what are believed to be the *intrinsic characteristics of work* (Rasmussen, 1994; Vicente, 1999) that are largely, if not completely, responsible for work behavior. An analyst who fills in the framework for a selected domain completes the framing language. This is similar to how the GOMS framing language (or architecture) is used, as is also the case for the less developed cognitive triad language of the OSU genotype.

2.3.4 UCSD Genotype

The fourth genotype considers the field of CSE as a new joint science-engineering discipline to guide the design, construction, and use of systems. Its expressed goal is to produce, in a principled way, the ability to effectively relate relevant physical variables of the system to the psychological variables of the human user of the system. To achieve this goal Norman asserted that the field first needs theoretical tools to "understand what the user is doing" (Norman, 1986, p. 37). Thus, like the other genotypes, it recognized the need for conceptual tools that focused on understanding human work. Norman proposed a seven-stage model of activity as a framework to represent work activity. Like the CMU genotype, this framework is expressed as a specification for a goal-directed activity process. The seven-stage model is depicted in Figure 2.8.

This model includes both psychological variables and physical variables. The goal is to be able to link the physical variables to the cognitive variables. The seven variables depicted above the horizontal line express mental or cognitive activity. Physical activity, shown below the line, directly relates to physical states of the system and expresses an eighth variable class of the framework. As presented by Norman (1986), the *goal* construct represents the state the person wishes to achieve. An *intention* is treated as a mental activity that constitutes a decision to act to achieve the goal. The *action specification* construct is regarded as "a psychological process of determining the psychological representation of the actions that are to be executed by the user on the mechanisms of the system" (Norman, 1986, p. 37). The *execution* variable produces a mapping from intentions to desired system state that serves as a mental specification of the actions to be physically executed. *Physical activity* relates to the physical state of the system. *Perception* refers to the translation of the physical

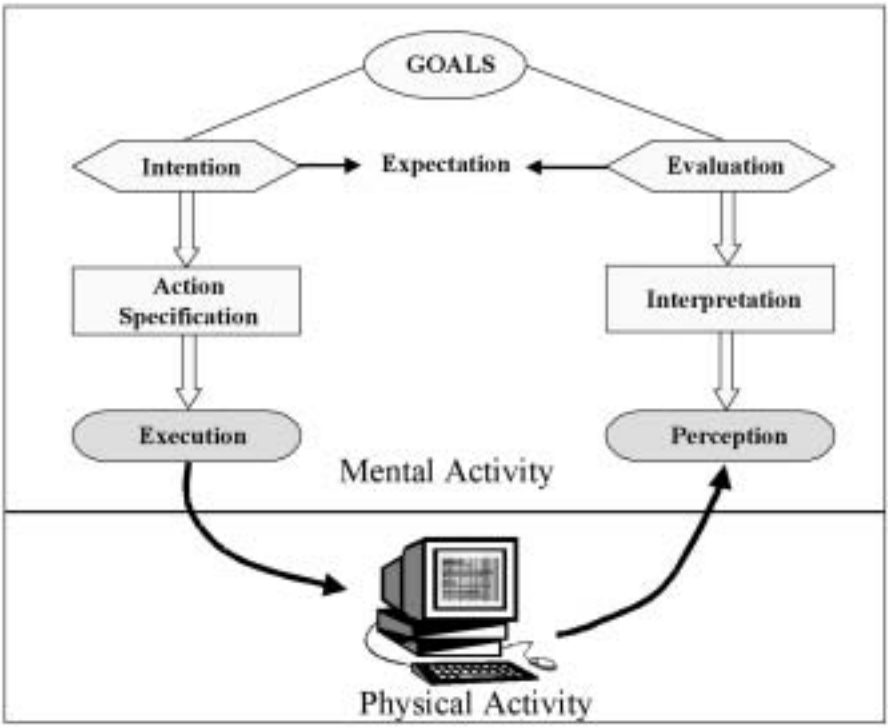


Figure 2.8: The seven-stage model of activity. (Adapted from Norman, 1986).

state of the system into the psychological state of perception that involves a layout of objects on a surrounding surface. It includes all perceptual operations. *Interpretation* is the process of converting the perceptual state into meaning with respect to the goal and intention(s). *Evaluation* is a process that compares interpreted meaning with active intention(s) and goals. The entire activity process iterates with new goals and intentions being formed as a natural outcome of the evaluation subprocess.

Like the GOMS modeling framework, the seven-stage model specification is expected to be used to produce an approximate model of problem-solving behavior. While it is presented in linear stage form, specific instances may include repeated and skipped steps. As a result, it is not as procedualized as the GOMS family of models.

In contrast to the other genotypes, the UCSD strain tends to focus on work in terms of direct interaction with a computer system. In other words, work is viewed in terms of human-computer interaction. The physical variables refer to the physical properties of the computer interface when work is accomplished directly with a computer. It is also clear that the specification for the seven-stage model is not as well developed as that for a GOMS model. The seven-stage

model provides cognitive constructs, but the actual process is sufficiently underspecified in that it does not constitute well-formed process architecture.

One use of the seven-stage model is to directly support interface design for interactive systems. The UCSD genotype has considered the cognitive nature of this design problem in terms of two gulfs, the gulf of execution and the gulf of evaluation, that separate the cognitive variables from the physical variables of the interactive system. The gulf of execution refers to the demand of the interactive system on the user to turn (cognitive) intentions into input to the computer. That is, the user, who thinks in terms of a "work language," must translate intentions into the "interface language" of the computer to use the computer in work. The gulf of evaluation is the demand on the user to interpret computer output relative to work need. The seven-stage model provides the designer with a device that can be used to make the cognitive nature of work explicit to better understand these demands.

To bridge these gulfs, the cognitive problem can be divided into two pieces. Each piece is associated with a specific pair of cognitive stages. These are known as the semantic distance and the articulatory distance problems. The semantic distance, on the input side, refers to the cognitive separation between the work-centered language of user intention and the semantics of the user interface language that must be used to enter the intention into the computer (e.g., menu labels, text input, etc.). On the output side it is the distance of the computer language to the cognitive stage of evaluation (e.g., actual content change versus desired change). Thus, semantic distance addresses the *content* problem of the user interface relative to the user's work focus, as reflected in the cognitive activities of intention and evaluation formation.

The articulatory distance addresses the *form* of expression of the information by the computer. A form mismatch on the input side increases the load on the action specification stage of cognition. On the output side, it increases the load on the interpretation stage. Thus, if the form of expression in the computer interface does not match the form of action specification or interpretation, then the user must overcome this distance by engaging in more cognitive work. It follows from this view that there are at least three cognitively inspired design goals:

- Minimize semantic distance from system operation to work
- Minimize articulatory distance from the system to work, and
- Maximize direct engagement with work.

The first two goals follow directly from the previous discussion. The third goal introduces another construct in the language of the UCSD genotype. Direct engagement is related to but different from the construct of distance. Engagement is said to be direct when the user's interaction with the computer results in a feeling of involvement with objects of the work domain (Hutchins,

Hollan, & Norman, 1986). In other words, as an engagement becomes more direct the computer effectively becomes more invisible. The user feels like he/she is doing goal-driven work rather than manipulating a tool to indirectly accomplish goal-driven work. Whereas semantic and articulatory distance are defined in relation to the operation of the computer as a tool in work, direct engagement is defined in relation to goal-driven work itself. Together, these constructs capture the demand characteristics of the problem-driving goal and those associated with the incidental demand characteristics (from the computer), in the same spirit as the OSU genotype. The major difference between these genotypes in this regard is essentially the scope of demand characteristics that can be explicitly considered. The UCSD genotype seems to limit incidental factors to the computer system, perhaps on the assumption that computers will usually mediate knowledge work.

The UCSD genotype is like the CMU strain in that it follows a cognitivist orientation toward work. The seven-stage model of activity is a process-based framework. It attempts to model work directly as an explicit process. However, in its current form it falls short of being a cognitive architecture because control mechanisms have not been adequately specified.

At the same time, the UCSD framework may also be regarded as considering the work domain in structural terms. The notions of semantic distance, articulatory distance, and direct engagement speak to the work domain as challenges to be addressed. Thus, like the RISO framework, separate models are used to capture aspects of work activity (seven-stage model) and the work domain. However, it is again clear that the conceptual formulation from this more ecological perspective is not developed to the same extent that it has been in the RISO genotype.

The UCSD genotype is closely associated with user-centered design of the human computer interface. This point is made clear by Norman's comments on the science of CSE. In his terms, CSE is about creating a science of user-centered design. This science produces principles that can be applied at the time of design and yields designs that are pretty good at meeting the three previously stated design goals during the first design iteration. Thus, CSE science includes a study of work, human activity, and how to convert this knowledge into useful design principles to impact design decision making. CSE is the science of work relative to design engineering.

2.3.5 Conceptual Distinctions

These four genotypes provide a wealth of conceptual distinctions that reflect the study of cognitive work in CSE for the express purpose of including better descriptions of cognitive factors in systems design. They make clear the difficulties involved in modeling work in a consistent, complete, and useful manner.

They point out that there are at least three alternative ways to consider cognitive work, all of which may be useful for engineering design. Cognitive work can be viewed as (1) a property of a worker or agent, (2) a property of a purpose-driven task, or (3) a property of a web of factors that defines the work domain, including the agent or agents. Although it is clear that there is no single way that can fully summarize the similarities and differences among the genotypes, for the purpose of considering the state of development of the field in terms of its theoretical basis, it is useful to review what each genotype has to say about these three views of work. Figure 2.9 shows this contrast.

The CMU genotype approximates work as a goal-driven task. As a result, all aspects of the work domain and agent cognitive processes considered are embedded in the task view (i.e., GOMS family of models). Conversely, the OSU genotype stresses interactions and a demand characteristic perspective of work. Thus, the agent and task are embedded in the work domain framework. The RISO genotype is also organized around the work domain, but it provides separate frameworks that may be regarded as relating more directly to the other views of work. The same may be said for the UCSD genotype, if Norman's cognitive control framework from cognitive psychology is included as a component. But in this case, the domain view is much more restricted, consisting of the human-computer interaction and a purpose-driven task.

The genotypes also differ in the degree to which they have refined their respective frameworks. The CMU framework provides a process architecture based on a transformation paradigm. Abstract, purpose-driven goals are transformed by a sequence of operators to subgoals and eventually into elemental actions. The RISO genotype is the most developed framework from the ecological perspective. Separate, interconnected sub-frameworks provide detailed coverage of all three views of work. They may be considered as containing an emergent process architecture from a self-organizing control perspective. The architecture is necessarily represented implicitly rather than explicitly. The explicit representation identifies factors that shape the emergent process, including intended and unintended driving forces embodied in the action agent.

The OSU and UCSD conceptual frameworks are not as thoroughly developed as the other two. The OSU orientation is consistent with a self-organizing notion of work and thus one might expect it to develop along the lines of RISO genotype. The UCSD genotype seems to provide a mixed case. It is formed from a traditional cognitivist perspective that emphasizes internal human information processing. The seven-stage model reflects this transformational process. However, it also gives recognition to experiential based behavior that involves attuning to the environment, which seems to suggest the importance of a self-organizing system being implicitly shaped by external factors. But it is not at all clear how these two different types of control processes interrelate to define the texture of work.

Conceptual Views of Work			
Genotype	Agent View	Task View	Domain View
CMU	MHP model	GOMS family of model	
OSU			Cognitive triad framework
RISO	SRK framework	Control task framework	Abstraction Decomposition Space framework
UCSD	Experiential and reflective control frameworks[9]	Seven-stage model	Gulfs of execution and evaluation; direct manipulation

Figure 2.9: Human work conceptualized from three different perspectives: An agent view, a task view, or a domain view. This chart shows the relation between these perspectives and the conceptual tools used by each of the four CSE genotypes.

Each genotype might be regarded as providing a different hypothesis about how to model work in a way useful for design engineering. The ecological stance of the OSU and RISO genotypes argues the position that work is emergent and should be modeled indirectly. The indirect approach is preferred because the model can be robust to change and supports dynamic error correction. These attributes are achieved in part by explicitly representing the coupling between *task activity* and the *work domain*. As disturbances arise, knowledge of the domain can be used to make adaptive adjustments in the work pattern. In other words, *work* emerges from the interactions of a work field, and a task relative to an agent or worker. Thus, the ecological position emphasizes the capture and representation of adaptive behavior in an open fashion. It is from this basis that the constructs to model work were selected.

The CMU and UCSD genotypes have elected to model work in a *direct, single goal-driven manner,* whereas the ecological approach, at least implicitly, considers multiple goals. An advantage of this approach is that work can be modeled as a closed process. As a result, it provides a means to more easily establish predictions of human performance based on the consequences of cognitive work. In general, one would expect to be able to make human work models faster from this perspective. It readily accommodates multiple levels of approximation that can be selected to meet the system development needs as they arise. However, this gain is offset by a reduction in the ability of the models to capture adaptive behavior in a robust manner, and it does not address error recovery unless it is explicitly built into a model. The indirect approach may also generate different levels of approximation, but because of the diversities of vari-

9. These control processes reflect Norman's view of data-driven and cognitive-driven behavior. They have been developed as part of Cognitive Psychology and represent an expression of the general notion that there are two modes of mental control, so-called automatic and control processes. It is not clear if this property of a cognitive model is formally included in the UCSD CSE genotype. It was not discussed in the overview presented here.

ables involved and the limitations of less-developed modeling tools for self-organizing, nonlinear dynamic systems, it is more difficult to cast these models in a computational form that can support performance prediction.

In summary, CSE has evolved a complex conceptual landscape. Work in this field is advancing the general scientific knowledge of human work. By nature, cognitive variables are ephemeral. They are of necessity abstractions. We cannot directly see or touch them. As a result they are open to a great deal of controversy, and they are difficult to define with precision. These four CSE genotypes provide a flavor of the range of variables that are believed to be useful for characterizing work to aid design engineering. They suggest that cognition resides both in the head and in the environment based on the characteristics of objects relative to an agent's work and knowledge state. While the conceptual distinctions allow us to discriminate among these different genotypes, it is fair to say more work is needed to improve the definition of terms and the creation of cognitively based work architectures to model human work.

2.4 CSE ENGINEERING PRACTICES

The groundwork for cognitively oriented engineering analysis was largely established during the 1980s. This provided basic conceptual tools to be used to support the human-centered design of interactive systems from the perspective of each of the genotypes. The concepts of work and approaches to modeling work address the subject matter of the field. The next issue is how to gather and analyze actual information about specific work to be used to populate the perspective frameworks to support design decision making from a human work perspective. Each genotype places its own emphasis on what to look for to characterize cognitive work. Each one has made its own contributions to the analysis, design, and evaluation aspects of systems engineering. The practice of CSE also has been maturing in each of these aspects of design and development. These developments are discussed in this section. For the purpose of organization, I will consider these developments separately in terms of support to the analysis, design, and evaluation aspects of systems development.

2.4.1 Analysis

CSE is most closely associated with human-centered analysis of an interactive system. Indeed, some seem to consider cognitive systems engineering as being the analysis activity that has come to be called Cognitive Task Analysis (CTA). There is more to CSE than CTA, but it should be obvious that CTA is heavily involved in the process of populating any of the previously discussed CSE frameworks to model work in a way useful for design engineering.

The origins of task analysis date back to the beginning of human factors engineering as an approach to user interface design. It includes a set of techniques to assess and describe human-machine interactions systematically relative to work, usually expressed as a task. Like all other aspects of systems engineering, a structured analytical process is useful to help ensure that all relevant issues have been identified and considered in the design, and to provide a rationale for identified requirements. Further, it is also the case that a systematic and structured process serves to lend credence to any judgments that must be made in the analysis. These same benefits apply equally as well to CTA.

CTA is mainly different from classical task analysis in two ways. First, as expected, it attempts to make explicit the cognitive aspects of task work. Second, for some forms of CSE, it includes a formal representation of the task or work domain, as well as work activities.

In broad terms, a CTA, as well as a behavioral task analysis, includes the use of techniques to extract and derive information about work that can be used to establish system requirements. Document analysis and various forms of knowledge elicitation techniques, including nonintrusive observation studies, are widely used as a means to gain deeper insights into work in general, and cognitive processes in work in particular. These methods are used by practitioners who approach the analytic work of systems engineering from one of the four CSE genotypes, as well as by those who may take a more eclectic CSE approach, and other human factors engineers, human-computer interface specialists, and systems analysts who do not identify with the CSE movement. However, even though the same basic *forms of data collection* are common, each practitioner will have preferences that influence how they probe, what they look for, and the judgments they make in the process of extracting information from the raw data. For example, a GOMS-oriented analyst will use document search techniques, interactions with subject-matter experts, and observational techniques to establish a functional, single goal-decomposition model of work, and use judgment to postulate a work process and level of expression granularity to include in a GOMS model of work. Obviously, the analyst will be primed to notice factors that suggest operators, methods, and selection rules. Other "incidental" factors may be missed, treated as noise, or considered to be second or third order issues, and thus they will not influence the GOMS representation. By contrast, a RISO-oriented analyst may be primed to consider some of these same observed "low priority" items as central shaping factors that need further investigation to understand them more fully. Thus, the CTA would probe in these areas. For example, the larger array of factors that serves to establish "priority values" in the work domain fall outside of the factors that directly define functions of a task. The GOMS-oriented analyst would tend to ignore or place less value on these factors, while the RISO-oriented analyst would try to capture the full set of domain values and learn about their interrelationships. Cognitive work, then, would be taken to be

involved as much with managing how these values are satisfied as it would be with operating on the more direct functional and goal-driven properties of work. This aspect of cognition would not be reflected in the GOMS model.

In summary, cognitive engineers and other human factors and human computer interaction specialists may use the same CTA techniques and methods. However, what they extract and how they use the available information is likely to vary widely depending on their orientation and skill.

Our main concern here is to inspect the progress that has been made in the development and use of CTA tools to improve the analysis of cognitive factors in work. Advances in CSE can be seen in three areas. First, more attention is being given to the development of complete systems of analysis, rather than the use of a disconnected set of techniques. Second, new methods have been developed that specifically probe for different aspects of work knowledge and that can be used to represent the qualitative data derived from elicitation and observational techniques. Third, computer support tools continue to be developed to support analysis generation, results sharing, and reusability of work models across projects and for the life cycle of a single project. Some examples of these advances are provided to give a flavor of the state of the art.

2.4.1.1 Systems of CTA. It should be obvious that the four CSE genotypes may be regarded as systems for CTA. Each provides a guiding framework for what knowledge needs to be gained from the analysis of specific work related to a specific development effort. Each framework provides a focus for what to notice, what to probe for, and what the analyst needs to understand. They provide the starting point for a systematic and thorough data collection plan. As indicated earlier, the CMU and RISO genotypes were the most developed frameworks in the 1980s. The OSU genotype has added a Functional Abstraction Hierarchy (FAH) to its framework as an adjunct to the cognitive triad. Other frameworks have also emerged from researchers whose origins are in the decision science community. These include the work on naturalistic decision making championed by Klein (1989; Klein et al., 1993; 1989), and cognitive analysis in support of the development of decision support systems originated by Zachary (1986). This latter work will be discussed in the next section. Here I will update the OSU genotype with a very brief discussion of the FAH.

Both the OSU and RISO genotypes approach cognitive task analysis from an ecological orientation. It is not surprising, then, to find some degree of convergence between these frameworks in terms of the tools used to represent the work domain or the field of practice. The RISO genotype has made a clear distinction between modeling the work domain and modeling work activities that occur in a domain (e.g., Rasmussen et al., 1994; Vicente, 1999). The Abstraction Decomposition Space (ADS) is one of the framework tools used by the RISO genotype to focus and organize a cognitive analysis with respect

to the work domain. More recently, the OSU genotype has developed an FAH as an engineering modeling tool for the same purpose. The abstraction dimension in this approach is based on Rasmussen's AH which is part of the ADS. What is unique about the FAH is that it also represents decision activities as part of the domain (Roth & Mumaw, 1995). Thus, it is used to represent knowledge gained from a cognitive task analysis in the form of a domain model that includes a description of the field of practice in relation to decision-making activity. Whereas the RISO genotype uses separate modeling tools to express the work domain and task activities in decision terms, the OSU genotype brings these two aspects of the CTA data together within a single FAH representation. The inclusion of decision processes in the FAH reflects the OSU design focus on the development of decision support systems.

2.4.1.2 Knowledge Elicitation and Representation. Many new knowledge-gathering techniques have been introduced to better reveal cognitive aspects of work. Klein and his colleagues have been particularly active in this area. They have developed the Critical Decision Method (CDM) and the Applied Cognitive Task Analysis (ACTA) set of methods. The CDM uses a retrospective strategy organized around critical incidents. A Subject-Matter Expert (SME) is asked to recall a specific hard problem or difficult incident, and then an interviewing process is used to issue cognitive probes to identify key decision points, shifts in the situation, and work strategies used. Importantly, eleven specific probe categories are used to acquire a detailed picture of the factors that influenced decision making (Hoffman, Crandall, & Shadbolt; 1998; Klein et al., 1989). The analyst uses probes to encourage the SME to talk about work in terms of analogies, the basis of or "whys" of choice, felt (perceived) time pressure, situation assessment, and hypothetical thinking.

ACTA is a collection of three techniques to capture the cognitive demanding aspects of a task (Militello et al., 1997). ACTA has the virtue that it is easy to learn and apply. The three techniques are called task diagram, knowledge audit, and simulation interview. The task diagram method asks an SME to break down a task into a set of three to six subtasks that are most "cognitively challenging." Using an interactive procedure between the analyst and SME, these subtasks are represented in a flow diagram. The knowledge audit is another probe technique. Importantly, it is organized around probe categories that are known to be important to detect differences between novice and expert performers. For example, a novice has more trouble communicating a big picture view of a situation than does an expert. The knowledge audit is one of the few techniques that use the construct of expertise as a basis for discovering cognitive details about problem solving and decision-making in a work context. The last ACTA method is the simulation interview. It provides a specific incident and asks the SME to engage in mental simulation of the unfolding sit-

uation. Different probes from the other techniques are employed to help focus and stimulate the process. All three techniques in ACTA are geared toward the discovery, from a cognitive perspective, of how SMEs actually work in their jobs. A detailed account of ACTA is provided in Militello et al., 1997.

In the process of using these and other CTA elicitation methods, the Klein group has also produced some new ways to represent the resulting data. For example, they have used an annotation technique to highlight elicited knowledge contained in interview transcripts resulting from the CDM (Hutton et al., 1997) Other representations include (1) a situation awareness chart to succinctly express how a situation is perceived to evolve; (2) a decision requirements table that highlights difficult judgments and indicates why it requires expertise, types of errors found to occur, and other data; (3) a critical cue inventory that lists the cues used to make a diagnosis; and (4) a cognitive demands table that is similar to the decision requirements chart (See Hutton et al., 1997 for examples).

All these techniques have emerged from a naturalistic decision making framework that has been advocated by Klein (Klein et al., 1989). This framework attempts to understand decision making in context, following the same argument of the OSU genotype. The Klein version of naturalistic decision-making could easily be considered another CSE genotype. It coevolved during the same general timeframe as the other genotypes. The framework is ecologically oriented and related to the OSU genotype in the emphasis it places on the demand characteristics of critical situations. Klein has developed a descriptive model, known as Recognition-Primed Decision Making that accounts for many decisions on the basis of recognizing situation characteristics that guide action taking without the apparent need to explicitly formulate options (Klein, 1989). Because naturalistic decision-making has traditionally been associated with the decision sciences, and because it is reasonably closely related to at least one other CSE genotype, it has not been developed as a separate genotype for the purpose of this commentary.

Another knowledge elicitation technique that focuses on decision-making is called the decision decomposition protocol. The full protocol includes a taxonomy of six decision support techniques and a set of 72 questions that focus on uncovering the goal structure of a decision maker. These questions are arranged into eight separate areas (Zachary, 1986). For example, there are questions on the source of goals (e.g., "What goals, if any, are imposed on the decision maker by higher authority?"), task dynamics (e.g., "Is the decision likely to occur more than once in the current context?"), choice criteria (e.g., "What is the decision maker's stance toward risk taking?"), and several other categories.

The decision decomposition protocol was the initial formulation of a comprehensive system for analyzing and modeling work from a decision-making perspective. The approach has evolved into a system known as COGNET. COGNET stands for COGnition as a NEtwork of Tasks. It is a conceptual

framework that aims to model real-time, multi-tasking, problem-solving work. The COGNET analysis and development process results in the formation of a computational model of problem solving. The architecture achieves a global problem-solving process by employing a blackboard structure that interacts with procedural knowledge that expresses tasks to be achieved in the prevailing situation. As a result, the analyst probes for information that can be used to formulate perceptual knowledge, declarative knowledge, procedural knowledge, and action knowledge. Data is derived from expert decision makers engaged in a realistic problem-solving context. The DDP is used first to establish characteristics of the work situation and explicitly seeks to discover the range of problems that make the work challenging. From this base, scenarios are formed and knowledge is elicited from experts performing the task by using think-aloud and question-answering protocols. Data from the elicitations are analyzed and formed into a representation known as the Decision Task Description language. This language has a well-formed syntax that is similar to that of GOMS. These descriptions are used to create the procedures that are expressed in the full computational model (Zachary et al., 1992; 1998). In sum, COGNET is a fully developed CSE system. It could also be treated as a separate CSE genotype, but again, like Klein's work, it is more closely associated with the decision sciences and for this reason was not identified as a genotype.

Three additional knowledge elicitation methods have emerged from work accomplished by military researchers. One method, called the Constraint Analysis and Synthesis Technique (CAST) considers work to emerge from a shaping process like the RISO genotype. The technique begins with a simulation interview that identifies a specific type of work, a current mission, and a detailed description of capabilities to be provided by a first-of-a-kind weapon system. A scenario is briefly sketched that includes a critical event. The SME fleshes out the scenario. The idea is that they will import their expertise into the situation. A series of probes are then used to uncover the factors that influenced why various attributes were included by the SME in the filled-in scenario. Finally, influence factors are evaluated by the SME in terms of influence perceived strength or fixedness of each factor. The method reveals the constraint profile, the origin of constraints, and their strength and thus identifies how cognitive work is shaped (Eggleston & McCracken, 1987). Strength/fixedness rating is a significant feature of CAST not found in other techniques.

The U.S. Air Force has also explored the use of a participatory design framework as a means that, in part, is used to improve the ability to uncover how experts conceptualize work (McNeese et al., 1995; Zaff, McNeese, & Snyder, 1993). One technique employs concept mapping that begins by providing a SME a core concept and using a nondirective and interactive approach to reveal the concepts that are important to the expert, and hence how they conceptualize work. A particularly challenging problem is how to synthesize a unified cognitive map based on inputs from a set of experts whose individual

maps may vary in terms used and other factors. McNeese and his colleagues have developed a tool to address this issue (Gomes, Lind, & Snyder, 1993).

Another technique that attempts to extract and understand expert cognitive skill is the precursor, action, result, and interpretation method. The technique is unique in that it uses a structured interview that has experts pose problems to other experts as a means to simulate representative work conditions. It attempts to uncover problem solving strategies and other forms of tacit knowledge that contribute to expertise. The verbal commentary resulting from this interchange is captured by an analyst. The analyst also asks probe questions to learn about precursors, interpretations, and other facets of the cognitive work (Hall, Gott, & Pokorny, 1995).

This limited sampling of cognitively based knowledge elicitation and representation techniques serves, in part, to illustrate the problems of how to acquire a useful characterization of work in cognitive terms. As Potter et al. (1998) pointed out, a cognitive task analysis attempts to understand how an expert conceptualizes work. It seeks to uncover:

- Technical terms and meanings used in a work context
- Sensitivities to what items are similar/dissimilar and why, and
- How items are organized (whole-part; cause-effect; spatial; etc.).

Potter et al. adhere to an ecological approach to CSE and subscribe to the OSU genotype. In general, the ecologically inclined practitioners of CSE tend to follow Potter et al. and regard CTA directly from the perspective of user work. Other practitioners that have been motivated by the cognitivist orientation of CSE tend to regard CTA as a means to elicit information about various forms of knowledge used by subject matter experts to accomplish work. In other words, work is seen as the deployment of knowledge in a skillful manner. CSE investigators and practitioners have classified work knowledge in many different ways. These include:

- Action knowledge
- Declarative knowledge
- Perceptual-based knowledge and control
- Procedural knowledge
- Rule-based knowledge
- Skill-based knowledge and control
- Strategic knowledge
- Metacognitive knowledge, and
- Attentional control skill knowledge.

It should be clear that there are several factors that make it hard to acquire information about work from a cognitive perspective. For example, it is gener-

ally regarded that experts have tacit knowledge that in part defines their expertise. The expert is able to demonstrate expertise but is not able to directly explicate all knowledge brought to bear on the issue. As a result, indirect probing and observational methods are needed to discover tacit knowledge. These methods are also needed to avoid the problem that arises when the SME attempts to "help" the analyst by providing what he or she thinks the analyst is looking for. This type of "experimenter demand" is well known in laboratory research performed in psychology. The cognitive analyst may ask, "Do you ever do X?" and the subject matter expert may think, "Well, I could think of it that way" and answer "yes" to the question. But in fact the SME does not consider the issue in that manner, usually for some very good reasons. As a result, the analyst is being inadvertently misinformed because the chosen way to make a probe stimulated an unfortunate demand for an answer. Some SMEs will answer, "Yes, but I do not think of it that way, and here is why." Thus, they provide an opening to overcome the demand of the original question. However, others will only provide the simple but misleading answer. Thus, the analyst must not only attempt to understand the SME's thinking process and skill, but he/she must also take steps to ensure the reliability and validity of the acquired data.

For these reasons the new techniques tend to emphasize observations and more open-ended probes. All the techniques mentioned above provide the cognitive engineer with more well defined and better engineered methods to elicit information about expertise and cognitive work processes. Some of the new cognitively oriented task analysis techniques appear to be more efficient than older methods for uncovering how experts work, especially in high stress, complex work situations. In spite of these gains, however, data collection and analysis remains a major time consuming process, and it has proven difficult to establish good ways to store and retrieve information for later analysis and use. Little attention seems to have been directed toward the issues of inter-rater judgment reliability and accuracy, and how to efficiently import the knowledge gained into models that can be used to further understanding and analysis. While there has been good progress, more is needed.

As stated at the outset, this commentary should not be regarded as a comprehensive review of CTA methods. Recent reviews of CTA methods have been produced by Potter et al. (1998) and Hutton et al. (1997), and the general topic is thoroughly discussed in a recent edited book by Schraagen, Chipman, and Shalin (2000). The Hutton et al. review provides brief descriptions of CTA methods and includes an evaluation of 14 selected methods. The evaluation is focused on the type of expertise that can be extracted from a method. Five evaluation criteria were used: resources required (to deploy the method), experience/skill required (to use the method), purpose of the analysis/method, task complexity, and expertise addressed. Potter et al. reviewed 20 cognitive systems engineering methods. These methods were grouped into six classes: (1) mapping semantic space, (2) functional domain modeling, (3) structured interviews/elici-

tation, (4) functional task modeling, (5) computational modeling methods and, (6) performance/observation methods, and participatory design methods. Each method is described and characterized in terms of four factors, tool support; product/output; relevance to systems design, and key issues. These two documents provide a good overview of the current CTA landscape. Other sources of information on CTA are Kirwan and Ainsworth (1992) and Cooke (1994).

2.4.2 Design

The systems development process is often characterized in terms of analysis, design, and evaluation activities. By nature, development for complex systems is iterative, and these three aspects of the process occur many times before the final product is produced. As a field, CSE provides support to all three aspects of systems engineering from a human work perspective.

Support to design from CSE comes in at least two forms. First, cognitive task analysis, used in conjunction with any of the genotypes or other perhaps less formal frameworks, naturally leads to the generation of functional requirements and specifications. Second, CSE also provides concepts, principles, and guidelines to support more detailed decision-making in the design engineering process. There have been new developments in both of these areas.

In general terms, each CSE genotype makes statements about what work information needs to be made available to a worker. The RISO system selects information items to express the so-called intrinsic characteristics of work for a given work domain. The OSU system focuses on information that defines decision making problems and what is needed (1) to aid the user to visualize or otherwise better understand the current situation, (2) to directly assist in the formulation of a course of action, which may include exploring different alternatives implicitly or explicitly, and (3) to better understand potential risks and unintended consequences of different decisions. In other words, the OSU system focuses on decision support. Both the CMU and UCSD systems consider the information needed to produce a procedure, perhaps with some flexibility, to achieve the goal state of a task. These types of information from all of the genotypes are a direct consequence of cognitive analysis. In this section, we are concerned with how is it conveyed to designers to impact the actual design of the interactive system artifact.

To address this issue, it is useful to first consider the different bases from which artifact design evolves. Design decisions may be characterized as coming from three different sources: analysis, management, and innovation. CTA serves as a work- and user-centered aspect of design by analysis. It reflects the gains derived from planning and using a systematic process. Certain items in any analysis stand out as key design requirements that must be reflected in the to-be-produced artifact. Others are more contentious and often reveal (some-

times difficult) trade-offs between alternative capabilities related to different technologies. Program management methods are used to resolve these issues and thus the to-be-produced artifact also reflects design by management. It is often said that design is a creative endeavor. The designer envisions or otherwise formulates a solution concept based on a general understanding of the design problem. The designer has an insight using tacit knowledge and skills. Design innovation of this nature occurs at the very beginning of a project and reoccurs many times by different designers throughout a project until the design is fully realized in the final medium.

Based on this view of design, contributions to the actual design artifact from CSE are conveyed to the final artifact in at least three ways. First, the CSE analysis methods are used in a systematic manner to provide an obvious case for certain user-centered factors that are persuasive to the project team and project manager on their own merits. Hence, they are included in the design. Second, the presentation of CSE factors may be communicated to senior project management in a manner that gains acceptance on the basis of compelling argument alone. In other words, it wins in a trade-off debate. Third, CSE information may be packaged and delivered to the designer working in the medium of the design artifact in a way that clearly relates CSE requirements to the items of the artifact that are available to be manipulated by the designer. Thus, the cost of inclusion by the hardware or software engineer is sufficiently low so that they are worked into the design without debate.

CTA is used more or less directly to impact artifact design from the design by analysis and design by management perspectives. Recent advances in CSE attempt to influence design by innovation in two ways. First, some effort has been devoted within the OSU genotype, under U.S. Air Force sponsorship, toward the development of a CSE-based design process that produces design artifacts that better link with software design tools and methods. Second, the RISO genotype has developed guidelines for an ecological approach to the design of the user interface for interactive systems and has recently formulated an initial design typology framework as another means of shaping user interface design.

Computer-Aided Cognitive Systems Engineering (CACSE) is the name of a tool that has been developed by the Logica Carnegie Group under Air Force sponsorship. The purpose of CACSE is to more formally integrate cognitive work analysis and cognitive systems engineering methods with the general systems development process (Logica Carnegie Group, 2000; Potter et al. 2000). It is generally recognized that every design organization subscribes to its own design methodology within a systems engineering framework. It may be modified in some ways to meet contractual requirements introduced by a customer. This implies that any design methodology that prescribes a rigid process using a fixed set of tools is unlikely to be adopted by any organization; and even if adopted, it is not likely to survive changes in the organization of procurements over time. Based in part on this belief and in part on the recognition that many

different tools can be used to effectively accomplish a CTA, the CACSE method allows for flexibility in method selections. In its current state of development, CACSE lays out a CTA framework that extends CSE products into actual artifact design and that provides a new descriptive formalism to represent decision requirements to the software engineer.

The general framework is shown in graphic form in Figure 2.10. It attempts to show the growth of understanding that occurs over time as more analyses of the current world are performed and more explorations of the so-called envisioned world (i.e., design concept and the form of the world in which it will operate) expressed by the designed artifact. Like the RISO genotype, CACSE divides knowledge acquisition methods focused on current work into those that attempt to (model) understand the work domain and those that provide understanding about the practitioners and activities in the domain. CTA analyses lead to a level of understanding that serves to produce hypotheses for ways to improve work performance. The envisioned world is explored through the development of prototypes. A major goal of the CACSE process is to produce some CSE artifacts that directly aid the developer in the prototyping process.

The FAH, briefly described earlier, is a major CTA tool in CACSE. A Display Task Description (DTC) template is a new invention of the framework that attempts to provide, in a succinct manner, a constellation of information about a decision requirement. The hope is that this information set will better equip the artifact designer to make good design choices based on a deeper understanding of the support requirement. The template identifies four interrelated categories of information: (1) the critical decisions, (2) aspects of the user who must make the decision, (3) supporting information requirements, and (4) the context in which the decision is required. With this set of information, CACSE provides guidance to the prototype design to facilitate proper interpretation of the decision requirement.

The CACSE framework has inspired the development of a software CACSE tool to serve as an analysis and design support system for system engineering from a CSE perspective. In its current form, CACSE is expressed as a software-based analysis environment that contains an FAH toolkit. It includes drawing, editing, and tracking tools for producing, modifying, and storing an FAH (Logica Carnegie Group, 2000).

A somewhat less ambitious effort has resulted in a similar software tool to aid the development and use of work domain analysis in the RISO genotype. This tool has been produced by Sanderson and her colleagues and is known as the Work Domain Analysis Workbench (WDAW) (Sanderson, Eggleston, Skilton, & Cameron, 1999; Skilton et al., 1998). Like the CACSE tool, it provides drawing, editing, and tracking tools but in this case they implement the features of an abstraction decomposition space representation of a field of practice. It provides several features to aid the analyst in constructing and editing a work domain model. The WDAW has been used on several projects by

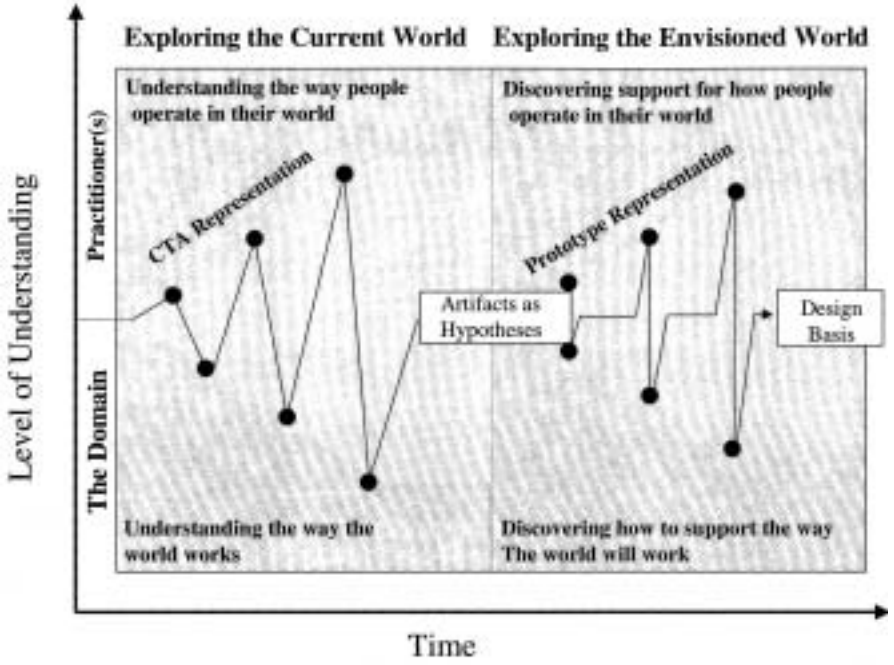


Figure 2.10: The CACSE conceptual representation of cognitive engineering as a human-centered development process that may be integrated with a software engineering development process. (Based on Potter et al., 2000)

Sanderson and her students, and it has recently been employed in a requirements definition study for a major U.S. Navy weapon system.

New computer tools have also been developed to improve support for GOMS modeling. For example, Baumeister et al. (2000) recently compared the use of four different computer aids for constructing GOMS models. These tools varied in ease of use and level of support provided to the analyst. In general, computer tools to support the cognitive analyst tend to reduce the time to produce analysis products, increase the opportunity for data reuse, provide some form of syntactical error checking, and help ensure that a derived work model is well formed with respect to the guiding constructs of a specific CSE genotype. As a result, they help to improve CTA consistency within a genotype.

An interface designer must convert requirements and specifications into a tangible interface artifact. In broad terms the designer must establish a concept for how to represent information, how to present this information to the user, and how to design the interaction with information in the interface. The RISO and UCSD genotypes offer some guidance about how to approach these basic aspects of interface design. The UCSD system embraces the notion of a direct

manipulation interface. By making explicit the characterization of two cognitive gulfs that separate the user from direct work and the need for an explicit first-person engagement with work, the UCSD system provides focusing guidance to the physical artifact designer. This guidance was discussed earlier during the description of the UCSD genotype. The RISO genotype has developed an interface design strategy known as Ecological Interface Design (EID) (Rasmussen, 1988; Vicente, 1999; Vicente & Rasmussen, 1992). EID argues for the merits of focusing on representing the work domain in the user interface, and for providing explicit representation of the domain at both the functional and physical levels of the AH. Further, it points out the need to support work at a skill-based, rule-based, and knowledge-based approach to performance. By providing appropriate abstract information in the presentation, the interface supports reasoning from first principles (knowledge-based), and reveals underlying dynamics that define the state of the system that may be masked when only physical features are presented in the interface expression. In this way, the interface representation and presentation help the user to visualize the work domain state and to recognize disturbances and how to address them. The representation of functional abstraction hierarchy information was shown to improve performance in a recent study by Pawlak and Vicente (1996). It has also been used to prototype the design of new displays for a nuclear power plant facility (Vicente & Tanabe, 1993), and a redesign of the fuel and engines displays for a C–130 Model E–H military aircraft (Dinadis & Vicente, 1999).

In its current form, EID does not provide specific guidance regarding the characteristics of the form of presentation that is best suited to assist visualization of information from each level of the AH. To improve on this state, Rasmussen (1999) has recently proposed a new taxonomy that can be used to identify the representational content for visualization at various levels of the AH. The taxonomy is currently in an early state of development. It attempts to suggest links between the range of representational content by abstraction level and paths to (visualization) presentation to support each type of content.

Given the number of different ways design decisions are made, it is difficult to assess CSE advancement in this area. Cognitive engineers have used their analyses to formulate support concepts (e.g., Potter et al., in press, Roth et al., 2000, Zachary et al., 1992, 1996). CACSE seems like a promising approach to improve the linkage between CSE and the bench-level artifact designer. But in its current state of development it appears to provide more for the CSE analyst than for the software engineer. For example, while CACSE is able to provide the software engineer with design requirements to support worker decision-making, it does not provide any assistance to the engineer with any specific insights about how the requirement relates to the form and content of an effective software object model for the system. More work is needed to narrow the gap and to increase the probability that improved work usage factors will be incorporated into the software design.

2.4.3 Evaluation

Evaluation occurs multiple times throughout the design development process. Requirements, specifications, technology options, design concepts, all are evaluated iteratively as knowledge is accumulated and the project situation changes. Little systematic work has been accomplished toward developing new CSE-oriented evaluation techniques. Rasmussen has sketched a strategy toward a systematic evaluation program that is consistent with the RISO philosophy of design, but it remains in a relatively immature form (Rasmussen et al., 1994). Kirlik and his students have been working on new ways to evaluate data from laboratory-based studies of complex work (Rothrock, 2001; Rothrock & Kirlik, in press).

Recent work by Benda and Sanderson (1998) provides an indication that a different type of evaluation, motivated by the RISO genotype, may gain in importance in the future. Rather than focus on making direct comparisons between alternative concepts, these authors consider the issue of how to evaluate the consequence of some type of change in the work situation. For example, what can be expected if a new piece of technology is introduced into a work domain? Can its impact on total work performance be predicted? Benda and Sanderson point out that many different types of change can map directly into specific modeling frameworks of the RISO system. For example, a technology change that is introduced can be mapped onto the work domain model. They illustrate an evaluation analysis of the impact of technology change by devising a formal notation system that characterizes change in terms of things like tightening and loosening constraints, contradicting current practice, adding affordances, etc. Benda and Sanderson used this notation to evaluate the predicted work domain change resulting from the introduction of a new electronic anaesthesia record-keeping system into a medical center. The basic approach was used to compare how the new technology would influence *coordination* in the domain. It was able to uncover coordination problems that were incidental to the goal-driven value of the specific technology and provided an explanation for why fewer records were signed (a legal requirement) when the new technology was in place than with the older manual recording method. Thus, this form of evaluation considers incidental as well as intentional effects of the introduction of new technology into a work domain.

This type of evaluation is aimed at the scale of the sociotechnical system. It appears to be a promising development that is able to provide systematic evaluation without reducing either the scope or complexity of the work domain in the evaluation process.

Of all the CSE approaches, GOMS modeling from the CMU genotype has received the most use for the purposes of evaluation. Because GOMS models predict outcome performance, they are frequently used to compare alternatives and used at the broad system level (Gray, John, & Atwood, 1993) as well as at

the detailed interface level to compare the details of design concepts. Recently, for example, Gray et al., (2000) has demonstrated that millisecond differences in cognitive processing can add up to produce a significant performance effect associated with interface details like button design. John and Kieras (1996) review several uses of GOMS in this comparative manner.

In summary, GOMS and other process models that capture cognitive factors seem to be useful for usability analysis, including error analysis (Freed et al., 1997; Freed & Shafto, 1997; Gray et al. 2000; John & Kieras, 1996). Both descriptive and computational models are used. The ecological approach to CSE is just beginning to pay more attention to this aspect of the system development process.

2.5 CSE DEPLOYMENT

There is little doubt that the use of CSE approaches has been increasing in the system design community. Examples cited in previous sections provide some evidence to support this claim. Several examples of use have been provided by John and Kieras (1996a), Klinger et al. (1993), Hutton et al. (1997), and Rasmussen et al. (1994) just to name a few sources. In broad terms, CSE has been applied to the information overload problem associated with issues of information superiority in the military (e.g., Flach & Kuperman, 1998; Klein, 1997; Woods, Patterson, & Roth, 1998); the analysis of first-of-a-kind systems (e.g., Flach et al., 1998; Rasmussen, 1998); the analysis and design of decision support systems (e.g., Potter et al., 2000; Roth et al., 2000); and a wide range of work aimed at design trade-off studies and usability analyses (e.g., Gray, John, & Attwood, 1993; John & Kieras, 1996b). This work has resulted in the creation of some novel designs that have been favorably received by both system developers, and it has shown the ability to make nonintuitive predictions of performance that have been validated by empirical investigation (Gray et al., 1993; Roth et al., 2000).

However, it is also important to recognize that there are some factors that may be impeding the rate of progress. These factors range from issues of clarity and ambiguity of use of CSE frameworks, methods, and techniques by CSE practitioners to issues of clarity of understanding of CSE by system designers and managers in the larger system development community. To complete this commentary on the state of the field, it is appropriate to devote some attention to these issues. In this section I will provide examples of these types of problems.

2.5.1 CTA and Other Approaches to Work Analysis

Work or task analysis techniques have been used in human factors and personnel subsystems for many years. Several of these techniques address task description in different forms. (cf. Kirwan & Ainsworth, 1992). Many human factors and Human-Computer Interaction (HCI) practitioners seem to be relatively insensitive to differences in understanding and modeling work tasks when using classical task analytic methods versus the newer, more cognitively oriented methods (Eggleston, 1998). For example, how is a hierarchal task analysis representation (an old technique) different from a cognitively driven, goal-based GOMS representation, or an AH representation of a work domain? Do they capture the same or different information? Do they lead to the same or different conclusions?

It is misguided to place too much emphasis on selective analysis tools. Each CSE genotype, for example, exploits the use of a wide range of techniques for data collection and to support analysis. What is extracted from the techniques differs, sometimes substantially, based on what CSE framework is used. Consistency of use of methods within a framework is probably more important than the specific method selected to meet a given data collection situation. Nevertheless, a mature field is able to make clear distinctions between factors of technical significance. A human factors engineer or human computer interaction practitioner should be able to appreciate the differences in framework and supporting techniques. To illustrate the point, it is instructive to consider the differences, say, in the use of a hierarchical task analysis with the GOMS framework, and the AH framework of the RISO genotype.

Hierarchical Task Analysis (HTA) is an analytic technique used to represent a task in a way that provides a clear and succinct expression of a task useful for design. It represents a task in terms of activities. It produces a hierarchy of a task in terms of operations and plans. Operators identify work requirements for a person in a system. Plans refer to the conditions or rules that specify when to apply an operator. This sounds strikingly similar to the general form of a GOMS model. Indeed, in the Baumeister et al. (2000) study that compared software support tools for GOMS modeling, the example case was originally analyzed by using an HTA. This provides a convenient example to illustrate the often subtle and overlooked differences among the different approaches to capture critical aspects of a task.

The work domain considered in the Baumeister et al. article is a student volunteer scheduling system, named Atropos, used at the annual meeting of CHI'98. Atropos is a software application provided to students to use for self-scheduling of volunteer activities to meet the needs established by the CHI'98 conference planning staff. First, the hierarchical task analysis focused on interaction with the Atropos system. The same is true for the GOMS model. In contrast, the RISO genotype considers work initially from the perspective of the

1.0 Manage SV Schedule
- 1.1 Login
  - 1.1.1 Type User ID
  - 1.1.2 Move Next Field
    - 1.1.2.1 Press <Tab>
    - 1.1.2.2 Home to Mouse
    - 1.1.2.3 Point to Field
    - 1.1.2.4 Click Field
    - 1.1.2.5 Home to Keyboard
  - 1.1.3 Type Password
  - 1.1.4 Press <Enter>
  - 1.1.5 Click<Quit>
- 1.2 Add Slots
- 1.3 View Slots
- 1.4 Delete Slots
- 1.5 Click <Quit>

Figure 2.11: An example of a hierarchal task analysis representation for a volunteer scheduling system. (Based on Baumeister, et al., 2000).

work domain, as opposed to work activity. The AH is a framework that can be used to represent the domain.

The basic structure of the HTA can be represented graphically (see Baumeister et al., 2000) or as an indented outline, where each entry is a goal/operator. For example, the top-level task goal (which is equivalent to a function or an operator) is Manage Student Volunteer Schedule. This is decomposed into five subgoals: Login, Add Slots, View Slots, Delete Slots, and Click <Quit>. The login (function) goal is further decomposed into five lower-order goals, four of which are defined as elemental task activities. The remaining subgoal or operator is decomposed into a set of five elemental tasks. These are presented in outline form in Figure 2.11).

Plan statements are also provided in an HTA. An example plan at the 1.0 level is:

Plan 0: Do 1.1
While not finished, if want to:
Add slot—do 1.2
View slots—do 1.3
Delete slot—do 1.4
Do 1.5

```
Method-for-goal:  Login Atropos
  Step 1.    Recall-LTM-item-whose Content is
    User-info and store-under <User-info>.
  Step 2.    Type-in User-ID of <User-info>.
  Step 3.    Accomplish-goal: Move-to Next-field.
  Step 4.    Type-in Password of <User-info>.
  Step 5.    Keystroke Enter-key.
  Step 6.    Decide:
    If System-response of <User-info> is
      Error-dialog,
   Then Accomplish-goal:
  Recover-from Login-Error-dialog.
  Step 7.    Return-with-goal-accomplished.

Selection-rules-for-goal:  Move-to Next-field
  If User-prefers-KB of  <User-info> is True,
      Then Accomplish-goal : Move-via KB.
  If User-prefers-KB of <User-info> is False,
     Then Accomplish-goal : Move-via Mouse.
Return-with-goal-accomplished.

Method-for-goal : Move-via KB
  Step 1.    Keystroke Tab-key.
  Step 2.    Return-with-goal-accomplished.

Method-for-goal : Move-via Mouse
  Step 1.    Point-to Password-field.
  Step 2.    Click Mouse-button.
  Step 3.    Return-with-goal-accomplished
```

Figure 2.12: A partial GOMS of the Atropos system using the GLEAN 3 GOMS formalism. (Adapted from Baumeister, et al., 2000)

While this HTA appears to be quite detailed, additional work is needed to understand the cognitive consequences of using the system. In other words, if one considers the hierarchal task analysis to be complete, important cognitive aspects of the task will not have been explicitly analyzed and considered in the design. A GOMS model may be used to extend the analysis to provide this additional information. In this way it identifies areas that remain under-specified by the hierarchal task analysis. Some additions from GOMS modeling include (1) forming sets of operators into methods, (2) decomposing operators

into mental operations, and (3) defining cognitive processing times for the elemental operators. An example of one GOMS model for the login operator is shown in Figure 2.12.

This example contains a fragment of a GOMS model that was developed using the GOMS toolkit called GLEAN3 (Kieras et al., 1995). GLEAN3 calculates a time for each execution statement in GOMSL. It adds 50 ms for the execution of every GMOSL statement. In addition it has a time value for each perceptual or motor operator action. For example, 200 ms is added for a mouse click. (These time values are based on studies of human performance.)

In this instance, the CTA accomplished from the GOMS perspective produces a computable model that can be used to predict performance times. It is also diagnostic of areas that may be inefficient from a cognitive load perspective. Thus, in this way the CTA adds additional insights that are not contained in the more traditional hierarchal task analysis method.

CTA often involves a great deal of judgment. Different analysts, for example, might approach the problem differently. An analyst with an ecological orientation, for example, would begin by modeling the student volunteer *work domain*, as opposed to the task activity, by using the Rasmussen AH or the Roth FAH. Figure 2.13 shows a partial, high-level AH for the student volunteer scheduling domain. On the surface this looks similar to the work activity model of the HTA. However, a closer look reveals several differences. First, the functional purpose of the domain acts as the driving goal of the domain and is functionally equivalent to the driving goal treated as the initiator for an activity. Thus, it is the same as the top-level goal used in the HTA and the GOMS model. But the abstract function of the AH codes priority measures that are not expressed in either the HTA or the GOMS representations. Four possible priority measures are identified in Figure 2.13 for the purpose of illustration. These are properties of the domain that influence activities, but are not direct goals of an activity. For this reason they are not captured by the HTA and GOMS approaches to analysis. The ecological view regards these aspects of the environment as important and in the RISO genotype explicitly directs the analyst to look for these features. It is expected that a violation of these values usually results in some undesirable cost that the agents will attempt to avoid, or some positive reward they will seek. These are the nondriving goals that serve to make work complex. Notice also that many of these priority measures go beyond the Atropos software system itself. That is, the work domain from the RISO perspective is larger than the software application, which is where it stops for the HTA and GOMS analysis.

The difference in the AH framework does not stop here. It is also apparent at the general function level. Only one of the items shown at this level deals with the Atropos application directly, make self-appointment option available. The other general functions are ignored by the HTA and GOMS models.

10. The means-end steps represented here are clearly incomplete, leaving out logical steps at the same level of granularity that should be included. However, the abbreviated presentation is sufficient to convey the general structure of a process means-ends, which is the focus of this discussion.

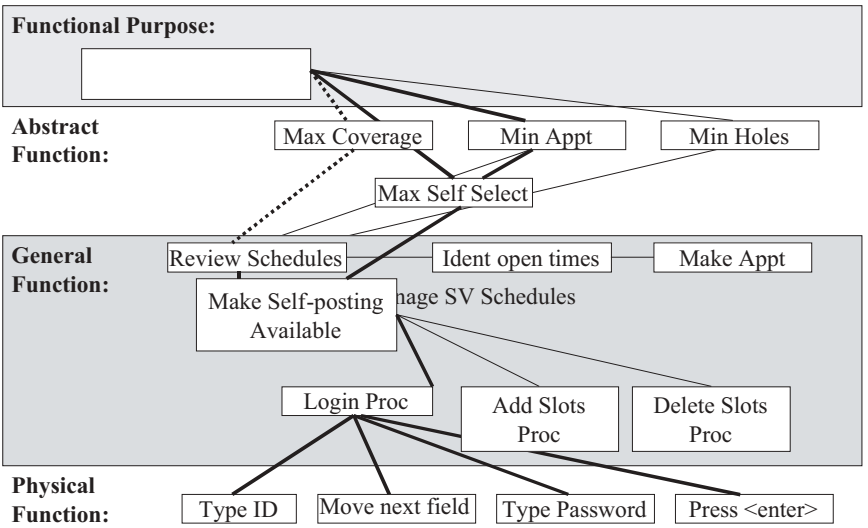


Figure 2.13: An abstraction hierarchy model of the student volunteer scheduling work domain.

For simplicity, only the make self-appointment function is further represented at the Physical Function level. At this point the physical functions are isomorphic with those in the HTA. But remember these only apply to one of the functions shown at the General Function level. Other physical functions would not show up in the HTA.

This brief illustration shows that the information encoded in the AH analysis is organized differently and includes some information covered in the HTA and GOMS, but also includes other information as well. Different conclusions (e.g., requirements) will be drawn from each of these analyses. All of them can provide valuable insights for use by the artifact designer.

2.5.2 Judgment in CSE

To populate any CSE framework requires substantial use of informed judgment by the cognitive engineer. For example, judgment is involved in developing an HTA, a GOMS model, or an AH. In fact, one uses judgment in how to set the system boundary to define the object of study and analysis in the first place. There is no escaping the need for judgment in the analysis of complex problems. However, an analytic process that includes formal analysis within a systematic process helps to reduce subjectivity and contributes to the quality of an analysis. This helps to minimize ad hoc contributions that are both difficult to defend and tend to cut across technical distinctions of different systems or frameworks, thereby adding confusion about the CSE field and its perceived

value. To advance the field, it is important to improve the degree of systemization, guidance, and training provided for each CSE framework.

Proponents of a specific CSE genotype become well versed in a specific brand of cognitive systems engineering. However, it is less clear that CSE developers, students, and practitioners of all forms of CSE fully appreciate and comprehend important distinctions among the different analysis and design approaches. Indeed, there is evidence that some important conceptual distinctions are not always well understood, and this may have a degrading effect on the knowledge gained from a CSE analysis. For example, while it is easy to understand that there is a difference between the work domain and the activities of workers in the domain, similar information about both aspects of a field of practice can be encoded as a property of the *domain*, a property of a *task*, or a property of the *worker*, as indicated earlier (see Figure 2.9). How information is encoded determines what will be made explicit and what will remain implicit in any given representation or model used by a particular genotype of CSE. These differences are important because the user of a representation will tend to be conditioned by the explicit representation and hence tend to miss potentially critical implicit information. Practitioners in a mature field would be sensitive to these differences and be able to evaluate their consequences in relation to different design decisions. Some examples of conceptual distinctions that are often confused by CSE developers and CSE practitioners alike are illustrated below.

2.5.2.1 Confusion Between Process and Structural Means-Ends Analysis. There is evidence that practitioners of CSE tend to be insensitive to technical distinctions promulgated by the different CSE genotypes. One example of this involves the concept of a means-ends analysis that is used both from the cognitivist and the ecological perspectives of CSE.

Vicente (1999) pointed out that the means-end relation coded in an AH is structural in nature and has a different meaning than the means-ends analysis that is consistent with a recursive goal decomposition process expressed in a GOMS model. To make the distinction clear, consider the simple problem of a person feeling dehydrated and setting a goal to solve this problem. In the tradition, a means-ends analysis is a process of recursively decomposing an abstract activity from an abstract goal state until an immediately actionable goal state is produced. The concept was introduced by Simon during the early development of the production system framework (Newell & Simon, 1972; Simon, 1996) and is consistent with the GOMS formalism. For this simple problem, the abstract goal (the end) may be represented as QUENCH_THIRST. A process means-end analysis looks for a way to accomplish this goal. The agent could, for instance, invoke the mental subgoal, GET_DRINK_OF_WATER, which, in turn activates the mental subgoal,

OPEN_REFRIGERATOR, etc., until the activated motor subgoal is a direct action (e.g., POUR_ AND_DRINK).[10] Thus, the means-end hierarchy encodes a process of activity that satisfies the initiating abstract goal.

The process means-end hierarchy may also be used to provide a rationale for an actor's behavior. Any sequence of three subgoals provides an answer to the general questions, why?, what?, and how?, beginning with the most abstract expression. Thus, for example we have:

- WHY: I'm thirsty.
- WHAT: Get glass of water.
- HOW: Get water from the refrigerator.

This same pattern of three questions is used by Rasmussen to highlight a structural means-end relation that maps onto the AH. It can easily be misinterpreted as reflecting a goal decomposition process in the form of a process means-end. It is not. The RISO genotype considers work as an emergent process that is *shaped* by properties of the work environment or domain. These are often called constraints. The emergent work perspective, therefore, is conceived to be a constraint satisfaction process. The constraint objects both serve to explain and *indirectly* cause the observed behavior. Because they operate in this indirect way, it is incorrect to view a structural means-end as goal decomposition, which is a direct transformational process. A structural means-end relation is intended to highlight an indirect shaping process.

To make the point concrete, consider a notional AH for a thirsty person in the context of a typical American home. The system purpose for the domain may be cast as the need to ACQUIRE NUTRITION, as show in the Figure 2.14. That is, the purpose of this domain is established by a physiological state of a human agent. The other levels of the AH are properties of the work domain that further act to shape behavior. For this simple example, the relevant physical form of the domain expresses characteristics of the home environment where the agent is located when a thirsty state occurs.

When an agent (person) is in a thirsty state, the system purpose is activated. This, in turn, serves to activate the other properties of the AH that provide the constraint net that guides the emergent behavior of getting a drink of water from the refrigerator. The process or work activity (of an agent) used to achieve the abstract purpose of the work domain maps results into the same observable outcome behavior as the process means-end, but in this case it is controlled by an indirect shaping process instead of a transformation process. Domain factors shape work activity:

- Acquire Nutrition < shapes need for quenching thirst activity
- Healthy Life Style < shapes choice of thirst quenching activity (e.g., biased toward selection of a low calorie liquid)

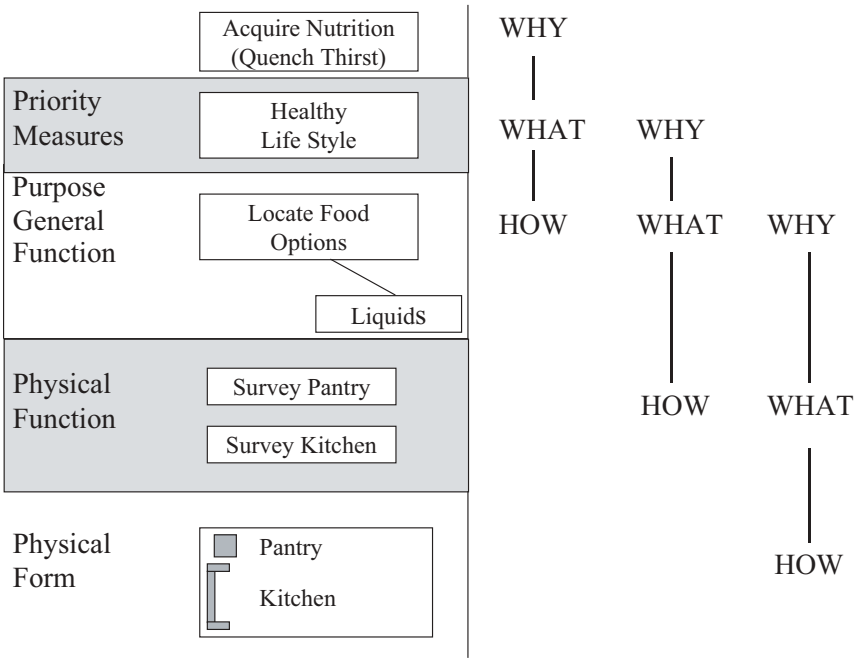


Figure 2.14: A structural means-end relation embedded in an AH representation.

- Locate Food < shapes recognition/reasoning activity (e.g., influences functional looking plan)
- Visually Survey Pantry/Kitchen < shapes action plan activity (e.g., influencing the physical looking plan), and
- Physical Pantry/Kitchen layout < shapes physical action activity (e.g., influences navigate/locomote to refrigerator while avoiding obstacles).

By inspection, it is clear that the AH dimension contains a WHY–WHAT–HOW means-end relation as a way to explain action. ACQUIRE NUTRITION: Why?—I'm thirsty. Because I value a HEALTHY LIFE STYLE, I select a low calorie option (answering the what question). I then recall there is cold water in the refrigerator (functionally answering the how question). In an indirect way, the structural objects of the domain "cause" the behavior of the agent to emerge. The more abstract "ends" are satisfied by the more concrete "means."

In contrast to a process means-end model of goal satisfaction, which establishes a specific trajectory of work, the structural means-end model only shapes outcome behavior and does not specify a given activity path. For example, the domain does not specify the need for water as a subgoal like it would

be in a goal transformation process. Rather, it specifies a constraint on the allowable types of drinks the agent may seek. The agent makes the selection (i.e., closes the degrees of freedom of the problem) as constrained by the shaping factors of the work domain. As a result, the structural means-end relation of the work domain influences the emergent behavioral trajectory while leaving open the actual selections made by the agent.

There are many instances where CSE researchers and practitioners have invoked the Rasmussen AH construct and proceeded to generate a process means-ends model instead of a structural means-end model. This indicates apparent confusion about a major aspect that distinguishes one CSE genotype from another. It is not clear why this happens. Does this mean that this distinction has not been clearly articulated in the literature? Does it mean that researchers and practitioners do not perceive it to be important? Or does it simply mean that there is a strong need for an analyst to create a closed or complete model of behavior instead of creating an indirect model that is incomplete with respect to specifying outcome behavior?

2.5.2.2 Internal Confusion Within the Ecological Perspective on CSE. Even within a given orientation to CSE, there is some evidence that technical distinctions are not well followed and may contribute to creating confusion for CSE students and practitioners. For example, Potter et al. (1998) provide an example of a functional abstraction hierarchy derived from a particular CTA. The illustration uses the functional categories from the Rasmussen AH. But the presentation of the content of the hierarchy creates an impression that is not consistent with the AH concept. Items depicted at the general function and physical function levels reflect whole-part decomposition and do not reflect the abstraction distinction implied by these levels. Further, the hierarchy is identified as an FAH, but it does not reflect any of the unique decision process properties of the FAH. These decision processes may actually be contained in the model and perhaps would be seen if the depiction were expanded, but based on the published data it looks like an example of an incorrectly formed AH, not a FAH. Thus, at best it is misleading and will tend to add confusion for students and users of the CSE product. At worst, it illustrates a muddle between a process means-end and a structural means-end relation within a single representation. What is presented can be interpreted as a mixture of the two types of relations. The authors may be using the AH or the FAH or they have developed an original variant of their own. This creates an opportunity for confusion and misunderstanding both among researchers within CSE who are trying to advance the field and among practitioners who are trying to use the CSE knowledge in systems design.

There is even uncertainty about constructs within a single CSE genotype. One example is suggested by the simple illustration used to show a structural

means-end relation. What are the requirements for defining a work domain in the RISO genotype? Some examples of work domains treat the agent or actor as being external to the domain. For instance, the human body has been taken as a work domain of patient resuscitation (Hajdukiewicz et al., 1999). Other work domains include the actor as a component of the domain. For instance, the medical staff involved in a patient operation is considered as an element in the surgical domain. Does this difference matter? Can a work domain be well formed if it does not include an actor? Under what condition might it be appropriate to define a work domain that does not include an agent? The point here is that this issue and others like it that speak to the precision of a genotype remain as open topics. Until they are addressed, we can expect there will be variations and perhaps inconsistencies in use. This represents another area where additional attention is needed to further advance the CSE field.

Some sources of confusion that reflect the state-of-the-art CSE field are based on an interaction between variations within the field and variations in the larger systems design community. The CSE field has many aims. Some focus on the problem of designing the user-system interface. Others focus on the design of decision support systems. Still others may regard the interactive system interface as a support system; hence it blends the HCI and the DSS aspects into a unified work interface. And some concentrate on the entire interactive system as the design object of CSE. These variations in aim mean some design objects can be in conflict. For example, a good CSE-based design for a stand-alone decision support system many be a poor design from the perspective of the user interface as an integrated, multifaceted support system. CSE recommendations therefore can potentially conflict with other CSE based recommendations. Such conflicts weaken both the creditability and perceived value of cognitive systems engineering by the larger design community.

How a design team is organized also influences how CSE can be applied in the design. For example, large-scale design projects tend to have separate teams that are responsible for the human computer interaction, decision support system, and human factors aspects of the system. As a result, this work organization constrains the possibility of some, perhaps highly desirable, CSE inspired design concepts. The interaction between a design team organization and the views of different CSE analysts may be in conflict. If technical decision makers for the project do not understand the reason for these differences, then they will not have a good basis for preferring one set of CSE requirements and recommendations versus another.

These same factors also contribute to the difficulty of improving the integration of CSE practices and products into the system development process. The situation is even more complicated. Currently there are several different approaches to system development. Some projects follow an incremental building strategy. Others follow a spiral development strategy. Both approaches are used when the required infrastructure for a product must be developed as part

of the product. Recently, more system developments have been for products that take advantage of an "open" infrastructure like the World-Wide Web. When creation of an infrastructure does not need to be part of the to-be-designed product, new variations in the development process are possible. What needs to be done to improve the integration of CSE into system engineering practices? All of the above factors make this a complicated question to answer.

2.6 COGNITIVE SYSTEMS ENGINEERING: WHERE DO WE STAND?

This commentary has touched upon many different factors that speak to the maturity of CSE as a science and as an engineering practice. As an applied science, CSE must deal with understanding work in the context of real-world conditions. There is no doubt that this is a difficult undertaking. The four CSE genotypes express different frameworks that in some sense serve as hypotheses of work. These frameworks are one sign of the progress of the field. The ecological perspective represents a bold move to model work in an indirect manner, as opposed to the more obvious direct manner typified by the cognitivistic perspective. This in itself may be regarded as a significant step forward. Based on the material covered in this chapter, it is clear that scientific progress is greater than this. In fact it is not confined to the nature of work. Both the science and the engineering practice of CSE have resulted in new ideas about how to capture and represent or model information extracted from subject-matter experts. This amounts to a scientific advancement in this applied area as well. The same can be said about the guidance CSE provides to the design of user support and the representation and presentation properties of the user interface of an interactive system.

These gains have made it possible to improve the engineering practices of user- or work-centered design. The inclusion of CSE analysis has produced new insights into human work that have suggested new forms of aiding or support and new forms of interface expression that, at least under some situations, have resulted in improved performance. There is little disagreement that CSE adds value to systems development.

But there are weaknesses in CSE in its present form. Terms used by different genotypes are not always well defined. Formal description languages used by these genotypes range from being fairly well formed to being weakly formed. There are wide variations in what CSE practitioners look for in a CTA, what they extract, and how they model work. The practice of CSE may be characterized as being in a guild stage. Any given practitioner tends to follow the belief and concepts of a selected recognized researcher in the field, and there is little fertilization or even understanding across different perspectives. The field depends more on good mentors than on a good system of education. And relatively little attention has been devoted to issues of reliability and validity of results.

The connection of CSE thinking and products to bench-level artifact designers is still an area that is underdeveloped. Object-oriented design has gained a major position, if not the dominant position, in software development. There have been many attempts to add structure to the software development enterprise. The Unified Modeling Language (UML) is the latest object-oriented framework being used in this capacity. It is a graphic language that can be used to visualize, specify, construct, and document the objects of software systems. While it may be treated as a development methodology, UML is better thought of as a language that can be used within any development system. Thus, while it adds structure it is not overly constraining. This is an important characteristic of UML, and one that should be emulated in a system engineering process that links CSE more effectively with software engineering. The CACSE approach appears to be headed down this path. More work along these lines is needed.

There is little doubt that CSE will continue to advance and increase the potential of positively influencing interactive system design. Based on the information presented here, some steps that can contribute to advancement of the field seem clear. There is a need to move beyond the guild stage. This will require improvements in the education of future CSE practitioners.

There is a need to improve the various frameworks used to both capture cognitively based knowledge from subject-matter experts and deploy it in models of work and human performance. This will require directing more attention to extending and sharpening the constructs used in these frameworks and fostering comparisons in a scientifically accepted manner. Scientific debate fuels progress.

There is a need to look for ways to increase reusability of CSE products and to extend CSE to cover the full life-cycle of a system. If the RISO genotype is right in the belief that intrinsic properties of the work domain can be discovered and represented, then it follows that this type of knowledge should be stable and hence reusable as new technology is available for possible insertion into the domain. Other areas of reusability are also possible. Standard units of cognitive actions, for example, may be established to facilitate activity modeling. Other possibilities for reusability also need to be explored.

CSE has enjoyed great interest and improvement over the past several years. More gains can be expected as we enter the 21st century.

REFERENCES

Baumeister, L.K., John, B.E., & Byrne, M.A. (2000, April). A comparison of tools for building GOMS models. In *Proceedings of the CHI'00, AMC,* 502–509. New York: Computer-Human Interaction.

Benda, P.J., & Sanderson, P.M. (1998). *Towards a dynamical model of adaptation to technological change.* Paper presented at OzCHI 1998, Adelaide, South Australia.

References

Bisantz, A. M., & Vicente, K.J. (1994). Making the abstraction hierarchy concrete. *International Journal of Human-Computer Studies*, 83–117.

Bovair, S., Kieras, D.E., & Polson, P.G. (1988). *The acquisition and performance of text-editing skill: A production system analysis* (Tech. Rep. No. 28). Ann Arbor, MI: University of Michigan.

Broadbent, D.E. (1958). *Perception and communication.* New York: Pergamon.

Card, S., Moran, T., & Newell, A. (1983). *The psychology of computer-human interaction*. Hillsdale, NJ: Erlbaum.

Card, S., Moran, T., & Newell, A. (1986). The model human processor: An engineering model of human performance. In K. Boff, L. Kaufman, & J. Thomas (Eds.), *Handbook of perception and human performance, Vol II. Cognitive Processes and Performance.* New York: Wiley.

Cook, R.I., Woods, D.D., & Howie, M.B. (1992). Unintentional delivery of vasoactive drugs with an electromechanical infusion device. *Journal of Cardiothoracic and Vascular Anesthesia, 6,* 1–7.

Cooke, N.J. (1994). Varieties of Knowledge Elicitation Techniques. *International Journal of Human-Computer Studies, 41*, 801–849.

D'Azzo J.J., & Houpis, C.H. (1966). *Feedback control system analysis and synthesis.* New York: McGraw-Hill.

Dinadis, N., & Vicente, K.J. (1999). Designing functional visualizations for aircraft system status displays. *International Journal of Aviation Psychology, 9*, 241–269.

Eggleston, R.G. (1998). Cognitive systems engineering: The latest fab or a true step forward as an approach to complex multi-person system analysis and design? *RTO Meeting Proceedings 4, Collaborative Crew Performance in Complex Operational Systems*, (Rep. RTO–MP–4) (pp 15–1–15–12). Neuilly-Sur-Seine, France: NATO.

Eggleston, R.G. & McCracken, J.R. (1987). *CAST: A constraint analysis and structuring knowledge elicitation tool*. (Internal Laboratory Report). Wright-Patterson AFB, OH: Aero Medical Research Laboratory

Flach, J.M., Eggleston, R.G., Kuperman, G.G., & Dominguez, C.O. (1998). *SEAD and the UCAV: A preliminary cognitive systems analysis,* (Tech. Rep. WP–TR–1998–0013). Wright-Patterson AFB, OH: U.S. Air Force Research Laboratory.

Flach, J., & Kuperman, G.G. (1998). *Victory by design: War, information, and cognitive systems engineering*, (Tech. Rep. AFRL–HE–WP–TR–1998–0074). Wright-Patterson AFB, OH: U.S. Air Force Research Laboratory.

Freed, M.A., & Shafto, M.G. (1997). Human-system modeling: Some principles and a pragmatic approach. In *Proceedings of the Fourth International Workshop on the Design, Specification, and Verification of Interactive System*, Granada, Spain.

Freed, M.A., Shafto, M.G., & Remington, R.W. (1997). *Employing simulation to evaluate design: The APEX approach.* NASA ASC website.

Gibson, J.J. (1979). *The ecological approach to visual perception*. Boston, MA: Houghton-Mifflin.

Gomes, M.E., Lind, S., & Snyder, D.E. (1993). A human factors evaluation using tools for automated knowledge engineering. In *Proceedings of the IEEE National Aerospace and Electronics Conference (NAECON)*, Dayton, OH. 661–664.

Gray, W.D., John, B.E., & Atwood, M.E. (1993). Project Ernestine: Validating a GOMS analysis for predicting and explaining real-world performance. *Human-Computer Interaction, 8* (3), 237–309.

Hajdukiewicz, J.R., Burns, C.M., Vicente, K.J., & Eggleston, R.G. (1999, September–October) Work domain analysis for intentional systems. In *Proceedings of the 43rd Annual Human Factors and Ergonomics Society meeting,* Houston, TX, 323–327. Santa Monica, CA: HFES.

Hall, E.M., Gott, S.P., & Pokorny, R.A. (1995). A procedural guide to cognitive task analysis: The PARI method; (Tech. Rep. AL/HR–TR–1995–0180). Wright-Patterson AFB, OH: Armstrong Laboratory.

Hoffman, R.R., Crandall, B.W., & Shadbolt, N.R. (1998). A case study in cognitive task analysis methodology: The Critical Decision Method for Elicitation of Expert Knowledge. *Human Factors, 40* (2), 254–276.

Hollnagel, E., & Woods, D.D. (1983). Cognitive systems engineering: New wine in new bottles, *International Journal of Man-Machine Studies, 18*, 583–600.

Hutchins, E.L., Hollan, J.D., & Norman, D.A. (1986). Direct manipulation interfaces. In D.A. Norman & S.W. Draper (Eds.), *User centered system design: New perspectives on human-computer interaction*. Hillsdale, NJ: Erlbaum.

Hutton, R.J.B., Pliske, R.M., Klein, G., Klinger, D., Militello, L., & Miller, T. (1997). *Cognitive engineering implications for information dominance* (Tech. Rep. AL/CF–TR–1997–0172). Wright-Patterson AFB, OH: Air Force Research Laboratory.

John, B.E., & Kieras, D.E. (1996a). The GOMS family of user interface analysis techniques: Comparison and contrast. *AMC Transactions on Computer-Human Interaction, 3* (4), 320–351.

John, B.E., & Kieras, D.E. (1996b). Using GOMS for user interface design and evaluation: Which technique? *AMC Transactions on Computer-Human Interaction, 3* (4), 287–319.

Kieras, D.E. (1988). Toward a practical GOMS model methodology for user interface design. In M. Helander (Ed.), *The handbook of human-computer interaction* (pp. 135–158). Amsterdam: North-Holland.

Kieras, D.E., & Polson, P.G. (1985). An approach to the formal analysis of user complexity. *International Journal of Man-Machine Studies, 22*, 365–394.

Kieras, D.E., Wood, S.D., Abotel, K., & Hornof, A. (1995) CLEAN: A computer-based tool for rapid GOMS model usability evaluation of user interface. *Proceedings of the ACM Symposium on User Interface Software and Technology*, 91–100.

Kirwan, B., & Ainsworth, L.K. (1992). *A guide to task analysis.* London: Taylor & Francis.

Klein, G.A. (1997). Implications of naturalistic decision making framework for information dominance (Tech. Rep. AL/CF–TR–1997–0155). Wright-Patterson AFB, OH: U.S. Air Force Research Laboratory.

Klein, G.A. (1989). Recognition-primed decisions. In W. Rouse (Ed.), *Advances in Man-machine systems research, 5*, (pp. 47–92). Greenwich, CT: JAI Press.

Klein, G.A., Calderwood, R., & MacGregor, D. (1989). Critical decision method for eliciting knowledge. *IEEE Transactions on Systems, Man, and Cybernetics, 19* (1), 462–472.

Klein, G.A., Orasanu, J., Calderwood, R., & Zsambok, C. (Eds.). (1993). *Decision making in action: Models and methods*. Norwood, NJ: Ablex.

Klinger, D.W, Androile, S. J., Militello, L.G., Adelman, L., & Klein, G. (1993). *Designing for performance: A cognitive systems engineering approach to modifying an AWACS human computer interface* (Tech. Rep. AL/CF–TR–1993–0093). Wright-Patterson AFB, OH: Armstrong Laboratory.

Logica Carnegie Group (2000, February). *CACSE tool user's manual*, Version 3.0. Pittsburgh, PA: Logica Carnegie Group.

McNeese, M.D., Zaff, B.S., Citeria, M, Brown, C.E., & Whitaker, R.D. (1995). AKADAM: Eliciting user knowledge to support participatory ergonomics. *International Journal of Industrial Ergonomics, 15* (5), 345–364.

Militello, L.G., Hutton, R.J., Pliske, R.M., Knight, B.J., Klein, G. (1997). *Applied cognitive task analysis (ACTA) methodology.* (Final Tech. Rep. Prepared for the Navy Personnel Research and Development Center (contract No N66001–94–C–7034). Fairborn, OH: Klein Associates.

Neisser, U. (1967). *Cognitive psychology.* New York: Appelton-Century-Crofts.

Newell, A., & Simon, H.A. (1972). *Human problem solving*. Englewood Cliffs, NJ: Prentice-Hall.

Norman, D.A. (1993). *Things that make us smart: Defending human attributes in the age of the machine*. Reading, MA: Addison-Wesley.

Norman, D.A. (1981) *Steps toward a cognitive engineering: System images, system friendliness, mental models* (UCSD Tech. Rep.). La Jolla: University of California Press.

Norman, D.A. (1984) Cognitive engineering principles in the design of human-computer interfaces. In G. Salvendy (Ed.), *Human-computer interaction* (pp. 11–16). Amsterdam: Elsevier.

Norman, D.A. (1986). Cognitive engineering. In D.A. Norman & S.W. Draper (Eds.), *User centered system design: New perspectives on human-computer interaction*.

Norman, D.A. (1987) Cognitive engineering—Cognitive science. In J. Carroll (Ed.), *Interfacing thoughts: Cognitive aspects of human-computer interaction*, (pp. 325–336). Cambridge, MA: MIT Press.

Pawlak. W. S., & Vicente, K. J. (1996). Inducing effective operator control through ecological interface design. *International Journal of Human-Computer Studies*, *44*, 653–688.

Potter, S.S., Roth, E.M., Woods, D.D. & Elm, W. (2000). Bootstrapping multiple converging cognitive task analysis techniques for system design. In J.M.C. Schraagen, S.F. Chipman, & V.L. Shalin, (Eds.) *Cognitive task analysis*. Mahwah, NJ: Erlbaum.

Potter, S.S., Roth, E.M., Woods, D.D., & Elm, W.C. (1998). *Toward the development of a computer-aided cognitive engineering tool to facilitate the development of advanced decision support systems for information warfare domains* (Tech. Rep. AFRL–HE–WP–TR–1998–0004). Wright-Patterson AFB, OH: Air Force Research Laboratory.

Rasmussen, J. (1976). Outlines of a Hybrid Model of the Process Plant Operator. In T.B. Sharidan & G. Johannsen (Eds.), *Monitoring behavior and supervisory control,*(pp 371–383). New York: Plenum.

Rasmussen, J. (1983). Skills, rules, knowledge: Signals, signs, and symbols and other distinctions in human performance models. *IEEE Transactions on Systems, Man, and Cybernetics. 13* (3), 257–267.

Rasmussen, J. (1986). *Information processing and human-machine interaction: An approach to cognitive engineerin*g. New York: Elsevier.

Rasmussen, J. (1988). A cognitive engineering approach to the modeling of decision making and its organization in process control, emergency management, CAD/CAM, office systems, and library systems. *Advances in Man-Machine Systems Research, 4,* 165–243.

Rasmussen, J. (1998). *Ecological interface design for complex systems: An example: SEAD–UAV systems* (Tech. Rep. AFRL–HE–WP–TR–1999–0011). Wright-Patterson AFB, OH: Air Force Research Laboratory.

Rasmussen, J. (1999). Ecological interface design for reliable human-machine systems. *The International Journal of Aviation Psychology, 9* (3), 203–223.

Rasmussen J., Pejtersen, A.M., & Goodstein, L.P. (1994). *Cognitive systems engineering*. New York: Wiley.

Roth, E.M., Gualtieri, J., Easter, & Potter, S.S. (2000). Bridging the gap between cognitive analysis and cognitive engineering. In *Proceedings of the Natural Decision Making Symposium*, Sweden.

Roth, E.M., & Mumaw, R. (1995). Using cognitive task analysis to define human interface requirements for first-of-a-kind systems. *Proceedings of the Human Factors and Ergonomics Society 39th Annual Meeting* (pp. 520–524). Santa Monica, CA: HFES.

Rothrock, L. (2001). Using time windows to evaluate operator performance. *International Journal of Cognitive Ergonomics, 5* (1), 1–21.

Rothrock, L., & Kirlik, A. (in press). Inferring rule-based strategies in dynamic judgment tasks. *IEEE Transactions on Systems, Man, and Cybernetics*.

Sanderson, P., Eggleston, R.G., Skilton, W., & Cameron, S. (1999, September-October). Work domain workbench: Supporting cognitive work analysis as a systematic practice. In *Proceedings of the 43rd Annual Human Factors and Ergonomics Society Meeting* (pp 333–337). Santa Monica, CA: HFES.

Schraagen, J.M.C., Chipman, S.F., & Shalin, V.L. (Eds.). (2000). *Cognitive task analysis.* Mahwah, NJ: Erlbaum.

Shannon, C.E., & Weaver, W. (1949). *The mathematical theory of communication.* Urbana, IL: University of Illinois Press.

Simon, H.A., (1996) *The sciences of the artificial* (3rd Ed.). Cambridge, MA: MIT Press.

Skilton, W., Cameron, S., & Sanderson, P.M. (1998). *Supporting cognitive work analysis with the work domain analysis workbench* (WDAW). Paper presented at OzCHI 1998, Adelaide, South Australia.

Vicente, K. (1999). *Cognitive work analysis: Towards safe productive and healthy computer based work.* Hillsdale, NJ: Erlbaum.

Vicente, K., & Rasmussen, J.(1992). Ecological interface design: Theoretical Foundations. *IEEE Transactions on Systems Man and Cybernetics, 22*, 589–606.

Vicente, K.J., & Tanabe, F. (1993). Event-independent assessment of operator information requirements: Providing support for unanticipated events. In *Proceedings of the American Nuclear Society Topical Meeting on Nuclear Plant Instrumentation, Control, and Man-Machine Interface Technologies* (pp. 389–393). LaGrange, IL: ANS.

Wason, P.C., & Johnson-Laird, P.N. (1972). *Psychology of reasoning*. Cambridge MA: Harvard University Press.

Woods, D.D. (1986). Paradigms for intelligent decision support. In E. Hollnagel, G. Mancini, & D.D. Woods (Eds.), *Intelligent decision support in process environments*. New York: Springer-Verlag.

Woods, D.D., & Hollnagel, E. (1983). *Cognitive systems engineering*

Woods, D.D., Johannesen, L.J., Cook, R.I., & Sarter, N.B. (1994). *Behind human error: Cognitive systems, computers, and hindsight*. Wright-Patterson AFB, OH: Crew System Ergonomics Information Analysis Center.

Woods, D.D., Patterson, E.S., & Roth, E.M. (1998). *Can we ever escape from data overload? A cognitive systems diagnosis* (Tech. Rep.AFRL–HE–WP–TR–1999–0195). Wright-Patterson AFB, OH: Air Force Research Laboratory.

Woods, D.D., & Roth, E.M. (1988). Cognitive engineering: Human problem solving with tools. *Human Factors*, *30*, 415–430.

Zachary, WW,. (1986). A cognitive based functional taxonomy of decision support techniques. *Human-Computer Interaction, 2*, 25–63.

Zachary, W.W., Ryder, J.M., Ross, L., & Weiland, M. (1992). Intelligent computer-human interaction in real-time, multi-tasking process control and monitoring systems. In M. Helander & M. Nagamachi (Eds.), *Human factors in design for manufacturability* New York: Taylor & Francis.

Zachary, W.W., Szczepkowski, J. Le Mentec, Schwartz, S., & Bookman, M. A. (1996). An interface agent for retrieval of patient-specific cancer information. In *Proceedings of the Human Factors and Ergonomics Society 40th Annual Meeting*. Santa Monica, CA: HFES.

Zachary, W.W., Ryder, J.M., Hicinbothom, J.H. (1998). Cognitive task analysis and modeling of decision making in complex environments. In J. Cannon-Bowers & E. Salas (Eds.), *Decision making under stress: Implications for individual and team training*. Washington, DC: American Psychological Association.

Zaff, B.S., McNeese, M.D., & Snyder, D.E. (1993). Capturing multiple perspectives: A user centered approach to knowledge and design acquisition. *Knowledge Acquisition, 5*, 79–116.

Zhang, J. (1992). The interaction of internal and external representations in a problem solving task. *Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society* (pp 954–958). Hillsdale, NJ: Erlbaum.

3 Discovering How Cognitive Systems Should Be Engineered For Aviation Domains: A Developmental Look at Work, Research, and Practice

MICHAEL D. MCNEESE
The Pennsylvania State University[1]

ABSTRACT

Pursuing Cognitive Systems Engineering (CSE) and understanding the implications of what this means for military aviation can be explored through a variety of perspectives and pathways. This review facilitates one perspective that could loosely be classified as a retrospective, developmental viewpoint that traces the author's own learning and discovery of how cognitive systems should be engineered. As part of this view, 15 years of work, research, and practice are examined for threads of consistency, continuity, and clarity as a foundation for learning what the future may hold. To make sense of the multiple directions and diversity within cognitive systems, this chapter is structured in the form of seven major queries to be answered. The answers developed for queries address many issues salient in the aviation domain in particular. The chapter also investigates what is important, valuable, and challenging by tracing the author's own experiential discovery of CSE through five progressive stages of development. As much of the author's experience in cognitive systems is within the domain of military aviation, this nexus is used to describe specific theories, frameworks, approaches, methods, tools, applications, and cases that have been uncovered and thought useful. The review concludes with potential challenges that practitioners need to consider in advancing CSE effectiveness—especially as applied to military aviation domains.

1. Michael D. McNeese was formerly a senior scientist working at the Crew System Interface Division of the U.S. Air Force Research Laboratory, Wright-Patterson AFB, OH. Much of the work upon which this chapter is based is extrapolated from over 19 years of experience with the Air Force in multiple venues.

3.1 INTRODUCTION

Work environments of the 21st Century place users in an information-rich world with little time to make sense out of events surrounding them, make decisions, or perform timely activities. In many cases, computational support and advanced interfaces for work activities have not been designed with cognition in mind. Unfortunately, this lack of "cognitive engineering" may produce what we refer to as *cogminutia fragmentosa*, where the user's cognitive world breaks down into small, isolated strands of thought as unanticipated events transpire (mental stove-pipes). People may experience a loss of meaning and control, become separated from the demands of their work, or fail to comprehend the emerging elements of a situation.

Cogminutia fragmentosa persists when there is no longer an interface between the user's cognitive world and the work he/she is responsible for. In other words, the user cannot properly adapt to the situation encountered (i.e., the user is in a maladaptive state). If this state continues, errors, failure, and even catastrophic disasters may be likely. This state may also contribute to affective and emotional responses by users (e.g., fear, anxiety, and rage) which further complicate agent-environment transactions. However, all is not lost. We are now at a point in history wherein it is not uncommon to observe human factors practitioners referring to Cognitive Systems Engineering (CSE) as their method or tool of choice to respond to work environments that produce cogminutia fragmentosa. As first-of-a-kind systems are proposed for complex environments (e.g., military aviation applications) CSE is now being utilized as a means to elicit various elements of expertise (e.g., cognitive skills, engagement rules, specific knowledge) from users, pilots, operators, or teams. As CSE is applied to real-world settings, expertise is qualitatively modeled (represented) and then used as a basis to predicate elements of a design (e.g., a human-computer interface, cockpit information system). In a typical application, practitioners engage users through a variety of methods that capture multiple facets of how work is transacted from agents to environment.

The preceding prelude functions as one guiding force for integrating chapters that appear in this State-Of-the-Art Report (SOAR). The overall goal is to take a broad overview of cognitive systems engineering. It is our desire to contrast/compare philosophies of use, goals, benefits, methods, tools, experiences, constraints and problems of use, lessons learned, and application examples as a means to generate new levels of understanding—especially as they relate to the specific constraints encountered in the military aviation community.

The hope is that this book provides a forum for what we know about CSE and addresses issues resident in understanding and applying CSE to cogminutia fragmentosa. The intent is to introduce multiple perspectives on this topic while pursuing integration of interests in the CSE area, especially as salient to the military aviation concerns. The international flavor of contributors is

expected to produce broadly defined points of view as different practitioners around the world apply CSE to selected domains of interest.

Given this backdrop of overall purpose and objectives of the review, this chapter delves into my own *developmental history* in discovering cognitive systems engineering in aviation (and other) environments.

As the old adage goes, *necessity is the mother of invention*. The last 15 years have afforded multiple opportunities to conceptualize, analyze, design, test/evaluate, and advocate for the role of the human in aviation-related domains. From early work involving field studies to the design of intelligent pilot-vehicle interfaces to current applications of supporting collaborative activity in Airborne Warning and Control Systems (AWACS) operations, there has been the necessity to reinvent terms of engagement. For example, the following questions have repeatedly emerged over the course of time: (1) What does *user-centered* design mean in the face of increasingly complex systems? (2) How do systems come to be more or less cognitive in nature? (3) How are *cognitive systems* requirements implemented as part of the design cycle of technological artifacts? As targets of opportunity change, these queries take on new meaning and may be answered in new ways with greater insight.

This chapter provides a 15-year retrospective history of the author's work in cognitive systems at Wright-Patterson Air Force Base, Ohio. Included are a look at aviation-related projects that provide the necessity for change which in turn have been the basis for innovative learning in the areas of *cognitive science, cognitive work analysis, cognitive modeling, cognitive field studies*, and *cognitive systems engineering*. A survey of various issues, frameworks, approaches, methods, tools, and application examples provides broad exposure to the overall question, "What is the use of cognitive systems engineering?" Using our own perspectives, developments, and case studies (as well as other practitioner approaches in CSE) a number of requirements, trends, and directions are discussed. The organization of the chapter is along the lines of seven systematic but interrelated queries designed to describe and evaluate work in CSE.

To capture a number of different foci and twists which have emerged across time in the development of cognitive system engineering, this chapter addresses "what's coming" with the seven specific objectives (stated as queries) as outlined in Table 3.1. These objectives form the fabric of the chapter and in turn are the waypoints that set up discussion topics. Each query also contains preeminent issues that are addressed as appropriate.

Table 3.1: Continuing Objectives

Considering 15 Years Worth of What

I	**What is it?**	**Defining Characteristics/Core Values**
II	**What are the formative conditions?**	**Background/History/Perspective**
III	**What are the objects of interest?**	**Use/Directions/Applications**
IV	**What are the representative approaches/examples?**	**Theories/Methods/Approaches**
V	**What has transpired?**	**Progress-Influences/Developing Stages**
VI	**What has evolved/ what has been learned?**	**Lessons Learned/Recombinate Themes**
VII	**What is next?**	**Emerging Issues/Future Directions**

3.2 QUERY I: WHAT IS IT?

3.2.1 Orientation

Put yourself in a real-world problem you may have recently experienced. You may have lost your billfold, locked your keys in your car, had your luggage misplaced, or even missed your flight.[2] Recently, I left my camera on board a flight simulator at an amusement park. Once the problem was "realized," I immediately tried to retrace my memory and define just where the camera was lost. Once the situation was defined well enough to assess the camera's predicted location, I had to decide how to get back into the simulator and retrieve it, which immersed me into a new planning situation. These problems may seem minor, but they present a person with an ill-defined, emerging dilemma where (1) understanding/remembering the context is crucial, (2) plans or actions previously considered routine start to break down, (3) problem solutions require innovation, risk taking, uncertainty, and even personal jeopardy.

As a problem solver you are accustomed to exploring solutions relevant to your own experience. Most of us are fairly good at working out our own cognitive engineering solutions in constrained settings. The dilemma arises when

2. This is an excerpt from the *CSERIAC Gateway* Newsletter (McNeese, 1995, Vol. VI, No. 5, pp. 1–4).

we are responsible for providing solutions for other people's problems that require us to know what they are thinking, or to be aware of the context under which they are replanning or taking new actions.

The difficulties experienced in the problem you just simulated can be magnified immensely when we encounter the intricacies of complex problems. When complexity increases, there is a much greater reliance on the cognitive, contextual, and interdependent collaborative factors in human-machine systems. Consequently, there has been an increasing role for cognitive engineering to make interrelated systems elements adaptable with the user's cognitive states. This is most evident in the changes necessitated by the introduction of computing systems into our workday environments.

Cognitive systems engineering is a technical specialty that affords different approaches for capturing users' multiple perspectives of knowledge, experience, and context for a given problem domain, and actively seeks user participation in transforming these elements into real-world design solutions (see McNeese, Zaff, Citera, Brown, & Whitaker, 1995; Zaff, McNeese, & Snyder, 1993). In this sense, the engineering of cognitive systems imparts "knowledge-as-design" for the user, by the user, and with the user. This is one explanation of CSE that I presented in the mid-1990s which takes a particular orientation. However, there are many other perspectives on the subject that one could adhere to. In the process of defining what CSE is and how it is used, it is instructive to assess the *defining characteristics*, *converging themes*, and *different examples* of how it is used. The following sections/queries provide a sampling of some of these items and give the reader a foundation of the more general aspects of CSE.

3.2.2 General Definitions

Before examining situated problems and work domains, and the objects of interest that have developed it is first necessary to explore what we have discovered in the way of defining CSE. Generating a common ground of definitions provides the first scaffold upon which the remainder of this chapter can build. One might think of CSE as a specialty area, a field of endeavor, a movement, or even a discipline. At many junctures CSE knits multiple specialties (e.g., psychology, engineering, and systems design) together to form an interdisciplinary enterprise that—in and of itself—is a work practice intended to generate *designs* that improve worker's effectiveness and well-being. No matter what you think the state of CSE currently is, there must be the realization that in many ways it is still emerging and developing. Best practices are heavily dependent on (1) CSE practitioners and their resident skills, knowledge, and capabilities, (2) the domains and workers which they study, (3) the objects of concern within these domains that workers use, and (4) the methods and tools

they use to discover domain-worker specificities and constraints. When one directs CSE towards the aviation *environment*, this provides new levels of influence as *practitioners* may have an aviation psychology background, *domains* involve new dynamic complexities in extreme conditions, *workers* to be studied (e.g., the fighter pilot) provide new kinds of affordances to observe, and innovative *methods and tools* may be developed in accordance with the constraints of these other elements. A main thrust then is for this chapter to look at the environments, practitioners, domains, workers, and the methods and tools I have encountered in the last 15 years with particular attention given to the aviation sector. Given this initial view, lets take a look at some major definitions from some of the top practitioners in CSE.

3.2.3 What Does Cognitive Systems Engineering Consist Of?

As a *systems philosophy*, cognitive engineering brings native user-and-use-centered knowledge to bear on the design of complex systems that subsequently require human interaction. User knowledge is elicited to integrate cognition, context, computation, and collaboration. Consequently, gaps between thinking, knowing, and doing are reduced to achieve more successful interfaces. Cognitive engineering is first and foremost centered on *individual differences* that shape expertise and knowledge as practiced within specific contexts. Integration among these elements systematically addresses functionality and compatibility, within and across different facets of the systems structure. As human-information systems become more involved, more complex, and more intelligent, the role of cognitive engineering looms as paramount first in the *front-end design process*. CSE could be thought of as a catalyst process to confront gaps of understanding among operators, their interfaces, and their contexts. (See Chapter 5 in this SOAR by Scott Potter and colleagues that explores these gaps in depth.)

3.2.4 Viewpoints and Converging Themes

A review of some early cognitive engineering efforts (Card, Moran, & Newell, 1983; Hollnagel & Woods, 1983; Norman, 1986, 1988; Woods & Roth, 1988; Rasmussen, 1986) found several overlapping, representative themes: (1) use-centered philosophies, (2) participatory approaches, (3) real-world problems, (4) elicitation/representations of knowledge, and (5) design processes.[3] Other efforts (Klein & Crandall, 1995; McNeese, et al. 1995; Sanderson, McNeese, & Zaff, 1994) reinforce these themes but suggest new advancements (e.g., mental simulation, collaborative design, and observational video analysis). Throughout the chapter these themes will be present (some will become more salient than oth-

3. Note that the efforts of Card, Moran, and Newell (1983) may be considered, and in fact are referenced, as a cognitive engineering approach although their effort does not exactly fit the confines of our themes. The GOMS model, the heart of their work, may be viewed as an *important bridge* that connects the theoretical principles of cognitive psychology with the practice of human factors engineering within the target area of human-computer interaction. For this reason it plays an important part in the history of cognitive engineering even though it is more of a research-centered approach. The use of GOMS methods today may still be considered one form of cognitive engineering.

ers for a given approach or method). Collapsing across a number of different viewpoints, general conceptualizations of the field can be informative.

As one of the founders of CSE, Jens Rasmussen believes that CSE is a highly interdisciplinary field that is concerned with the design of complex human-machine systems in which humans' cognitive needs are well supported (derived from Rasmussen, 1986). Woods and Roth (1988) suggest that the requirements and bottlenecks in cognitive task performance drive the development of tools to support the human problem solver. Some other views are as follows. CSE concerns design based on the discovery (and articulation) of first principles of how people interact with engineered systems in complex settings (Sanderson, McNeese, & Zaff, 1994). Vicente (1999) suggests CSE is about design that allows operators to adapt effectively and flexibly to unanticipated events. Hollnagel (1998) sees CSE as the basis for determining how we design joint cognitive systems so they can effectively control the situations where they have to function. Obviously these are just capsulated definitions, and one should go to each source for an expanded comprehension of each perspective. For even more definitions and defense of various views of the field of CSE, please refer to the 1998 special issue of the journal *Ergonomics* that contrasts different opinions and rationale of CSE.

As we look at definitions, it is evident that CSE has some shared, overlapping characteristics and core values that stand out. One characteristic suggested by Rasmussen—*well* supported cognitive needs—often gets secondary attention, but has utmost importance for the aviation sector. This is also an attribute that Vicente elaborates in his commitment to safety in work. Woods often addresses wellness through understanding how artifacts shape cognition and in turn become ways to avoid human error. When wellness and safety are not considered, then human errors can be probable. And much of how *well* comes to be is dictated by understanding the cognitive demands in the environment.

The challenges workers face in a domain must be supported by designs of cognitive tools that enable and reify their performance both alone and together. Hence, another characteristic evident is that "what" CSE does is *design with the human in mind*. The endpoint of CSE as a process is a better design for workers in a domain.

Another characteristic associated with these definitions is that designs support cognitive demands if they afford adaptive responses on the part of human and artificial agents (McNeese, 1986). This is what Norman (1993), another early pioneer of cognitive systems engineering refers to as "designing the things that make us smart." Highly brittle and rigid designs do not shape cognition (or collaboration) in ways that make work more effective, efficient, and safe but create a tunnel vision effect that limits human adaptation when emerging and uncertain situations arise in the complexities of work life. In summary, we might conclude that CSE can be defined as supporting distributed cognition through engineering design, designing cognitive technologies to enhance cog-

4. These are definitions I presented at the CSE International Workshop, Dayton, OH, May 19, 2000 that this chapter is predicated on.

Is about design that allows operators to adapt effectively and flexibly to unanticipated events (Vicente, 1999)

Is determining how we design joint cognitive systems so that they can effectively control the situations where they have to function (Hollnagel, 1998)

-- supporting distributed cognition through engineering design
-- designing cognitive technologies to enhance cognitive readiness
-- understanding the workplace to design adaptive support systems for workers

Figure 3.1: Capturing cognitive systems engineering.

nitive readiness of workers, and exploring the workplace to design adaptive support systems that mutually enable practitioners in their field of practice[4] (see Figure 3.1).

3.3 QUERY II: WHAT ARE THE FORMATIVE CONDITIONS?

A basic issue to reconcile is how cognitive engineering is different from human engineering and/or knowledge engineering. Cognitive engineering can be thought of as a middle ground existing between human factors and knowledge engineering. As such one might consider it as in service of two masters. It is informative to look at the traditions that led to the formation of cognitive engineering.

3.3.1 Human Factors Engineering

Human factors engineering has typically had a preeminent goal of serving the user but has generally treated the (1) context of work and (2) socio-organizational factors with much disdain (Bannon, 1992). Exceptions to this have played out in the areas of macro-ergonomics (Hendrick, 1986) and ecological

interfaces (Flach, 1990). Traditional human factors focuses on *human performance* and emphasizes issues such as workload, anthropometry, control-display integration, lighting, noise, and other factors that highlight design/human compatibility issues. Experiments and models may measure system states and human capabilities-limitations, in the context of ensuing mission-task performance, and vary a number of human-machine interface elements. Although this tradition is vastly important, and has its place in improving users' needs, it may not tell the whole story.

Within human factors practice, *gaps in understanding* occur while reconciling discrepancies among *what users think, what users know, what users do,* and *what users want*. These gaps are evidenced by failures in designs (unfortunately often attributable as "human errors"), clumsy interfaces (see Wiener, 1989; Reason, 1990), or brittle knowledge bases (Lenat & Guha, 1990). As complex systems emerge (e.g., the glass cockpit, nuclear power plants, intelligent highway systems), there is a necessity to modify the old order of human factors business. A reinvention of human factors engineering must accrete more investment in topics involving cognitive engineering, expertise, knowledge, naturalistic decision making, and real-world context to respond to gaps in understanding that are initiated in ill-defined, emergent, uncertain, multi-operator environments.

These gaps are amplified when people interact with each other, with information, or technological systems via various types of interfaces. When the interface involves a computer, the gaps often are irreconcilable as the level of knowledge necessary to complete intelligent interaction increases. And when the computer software comprises (1) knowledge-based, (2) groupware, or (3) evolutionary computing there is an element of "artificial intelligence" cast into the complexity.

3.3.2 Knowledge Engineering

Real-world problems involving human-computer interaction provided one of the developmental threads that gave rise to cognitive engineering. From this end of the spectrum, knowledge engineering was required to "propagate" knowledge (in various forms such as rules, frames, and predicates) within advanced computer technologies designed to perform as if they could reason like humans.

Like human factors engineering, there is a common thread with information requirements analysis, but in this case the analysis typically highlights information as it relates to computer system requirements. Buchanan & Shortliffe (1984) suggest that knowledge engineering is "the process of mapping an expert's knowledge into a program's knowledge base" (p. 5). They trace the term *knowledge engineering* as one coined by Edward Feibenbaum after Donald Michie's phrase *epistemological engineering.* This required engineers to elicit knowledge from subject-matter experts in an attempt to replicate that knowledge in expert systems.

Knowledge engineering practice tries to replicate an expert's knowledge as input into a workable and predefined knowledge structure. Knowledge building tools or even automated knowledge acquisition systems have been developed to streamline this process. Unfortunately, this process became a bottleneck as the knowledge engineer often required an expert to elicit native knowledge in a form (e.g., "if-then" rules) directly exportable to the requirements to knowledge structure. This resulted either in (1) brittle computer systems (what Woods calls clumsy automation) or (2) experts contemptuously rejecting the process or the means by which they were required to "heed knowledge" for a project. How to elicit *user-centered* knowledge from the operator, and how to transform that knowledge to fit information system requirements, initiated the formation of cognitive engineering. This still remains a core issue for the discipline today.

Knowledge engineering generally derives from computer science concerns and is not subject to principles such as "know thy user." Early techniques applied to limited, toy domains wherein "accessed knowledge" was relatively easy to come by. The process became much more difficult for real world systems (e.g., medical diagnosis, see Buchanan & Shortliffe, 1984; pilot aiding, see McNeese et al., 1990). Knowledge engineering usually progressed like most other engineering disciplines in the sense that processes were engaged for the sole intent of building an end product with little regard for how the end product is compatible with human interaction.[5]

When thinking about cognitive engineering, it is informative to point to a common ground that effectively looks at the converging principles and the fundamental ideas that span across different approaches in cognitive engineering. This section is designed to begin conceptualizing the domain along these lines. Specific approaches have much in common but also diverge at various points as we will see later.

3.4 QUERY III: WHAT ARE THE OBJECTS OF INTEREST OF CSE?

Success or failure of joint cognitive systems is inextricably tied to the situation context, conditions, processes, and measures that compose any given orchestration. In particular, *situated context* is increasingly becoming a salient foundation to understand real-world problem solving in a variety of domains (education, design, medicine, aircraft operations, see Hutchins, 1995; Young & McNeese, 1995; Zsambok & Klein, 1997). Such situated problems and envisioned designs to support workers context are the objects of interest for CSE. A more practical way of looking at this is stated by Woods (1998) where he indicates that "designs are hypotheses about how artifacts shape cognition/collaboration."

Situated problems and socio-cognitive factors are prevalent in many complex systems the U. S. Air Force is engaged in (e.g., space operations, information dominance, unmanned air vehicles, mission planning). These contexts are

5. Most expert systems were designed to be used by operators engaging the system—therein the end products were human-computer interfaces. But relatively little credence was given to this perspective in the early design of these systems. Exceptions were Buchanon's and Shortliffe's (1984) chapter on human engineering medical expert systems. Today, this has changed as the evolution of human-computer interaction and computer supported cooperative work has essentially redefined our vision of information systems in contrast to the notion of 'artificial intelligent' systems from the early-to-mid 1980s. Still, the engineering of knowledge within human-computer interfaces may proceed from traditional knowledge engineering standards resulting in impoverished or problem-

highly interdependent, collaborative in nature, highly integrated with multiple aviation concerns, and require global awareness in response to changing conditions and multiple uncertainties. The context most recently targeted in our studies involves Command, Control, Communications, and intelligence (C^3I) and planning operations (McNeese, Rentsch, & Perusich, 2000). Within the AWACS command and control domain, we have been applying cognitive engineering and modeling to the study operator intentionality, interrelated causality, contextual variation, boundary constraints, and emergent contingencies.

The vision of making situated problems (such as AWACS command and control) and envisioned designs the objects of CSE is historically related to the ideas and research approach of Suchman (1987) who suggests that cognition and collaboration come about—not by symbol systems—but through situated actions arising during the course of events that occur in a context. This is about cognition—not bounded by the individual brain or mind—but cognition constructed by social processes (Resnick, Levine, & Teasley, 1991). This view highlights "qualitative and naturalistic" components of what people actually do when they work together. In this sense, CSE is bound to approaches that are described as *situated cognition*, *ecological*, *participatory*, and *ethnographic* in nature. Inherently, this vision prescribes to the same basic view elaborated by Greenbaum and Kyng (1991) as it highlights situations and breakdowns, social relationships, knowledge, tacit skills, mutual competencies, group interaction, and experienced-based work. However, the vision also includes using the inductive insights gained from naturalistic views of cooperation to inform (1) experimental research paradigms that focus on quantitative, empirical studies and (2) design prototypes/infrastructures that may be evaluated through the use of both qualitative and quantitative tools.

3.4.1 Understanding Collaboration In Context

Much of the work that occurs today in the military, government, or private sector is done in teams or teams of teams. Teams may be continuously emergent, ad hoc and transient, and may contain smaller units that we term "multi-operator enclaves." Individuals who form a team may simultaneously be members of several other enclaves that disperse time, resources, and relationships in unexpected ways. Hence, teams may operate in layers of complexity and change. There is evidence that teams can fail in their endeavors (e.g., the Vincennes incident, the O-ring problem in the space shuttle disaster). Such failure exists as targets of opportunity for the CSE profession. CSE must assess the social-organizational, psychological-cognitive, and technological components of collaboration. Too often human factors fails to take a broad, systems approach to a problem and encounters a nearsightedness in research and guidelines. When the artifact of concern involves complex systems (e.g., computer-supported cooperative

work) it is imperative that a broader approach be utilized. In response to this concern, we address collaboration problems through the lens of ecological psychology and naturalistic decision making. Attempts to view collaborative systems more broadly have been made by Hinsz, Tindale, and Vollrath (1997) and others (Wegner, 1987) but primarily look at team function in terms of information processing or memory systems.

In addition to Suchman's (1987) situated cognition research, Thordsen and Klein's (1989) work on team mind is representative of early approaches that put more weight on the naturalistic components of collaboration. Even more salient is the work on "cognition in the wild" (Hutchins, 1995) which provides extensive knowledge on ship navigation from the vantage point of social-cultural practice. Young and McNeese (1995) define situated cognition as representative of real-world problem solving wherein group members spontaneously generate knowledge in the context of a situation; coordinate multiple cognitive processes, applied through multiple paths; and pick up critical perceptual cues for potential solutions. Collaboration directed towards solving real-world problems is often interpersonal, ill-structured, involves interwoven problems, extended timeframes, requires discovering problems and subproblems, and invites the social construction of knowledge. Simply put, an ecological psychology perspective emphasizes the interaction between an agent(s) and an environment wherein the attributes of each constrain the "interaction." This reciprocity is often referred to as agent-environment mutuality (Gibson, 1979; McNeese, 1996a). This may take the form of what an agent can effect (effectivities) or what the environment can afford (affordances). Affordances and effectivities are always in terms of each other. This systems-based view can be extrapolated to interpret cognition and collaboration in various forms

3.5 QUERY IV: WHAT ARE REPRESENTATIVE APPROACHES/PREMIER EXAMPLES OF CSE?

Theories, methods, and tools are often portrayed as consistent with (or even distinguished as) cognitive systems engineering approaches including knowledge acquisition (Cooke, 1994), cognitive task analysis (Gill & Gordon, 1997), naturalistic decision making (Klein, Orasanu, Calderwood, & Zsambok, 1993; Zsambok & Klein, 1997), operator modeling (Card, Moran, Newell, 1983), systems engineering (Rouse & Boff, 1987), field studies (Xiao, 1994), and ethnographic studies (Hutchins, 1995) to name a few. Within these areas, specific methods have migrated into the CSE focus. This includes, but is not limited to, repertory grids, protocol analysis, exploratory sequential data analysis, process tracings, direct observation, weighted networks, concept maps, design storyboards, Ishikawa diagrams, the decision ladder, means-ends hierarchies, retrospective reports, cognitive walkthroughs, and critical event logs. Various

approaches and methods evolved into different tools (e.g., the TAKE/COGENT tool; see Sanderson, McNeese, & Zaff, 1994; emerged from our AKADAM techniques and continued to evolve on the basis of both our use of the method and then subsequent use of the tool in practice). In many instances, CSE approaches may form through an amalgamation of theories, methods, and tools in these pre-existent areas. On the other hand, approaches are distinctively defined by the CSE label from their inception. As customary in science, there is a progression from theory-to-method-to-practice-to-tools-to-theory and so on. By looking at the roots of premiere CSE programs and tracking evolutionary paths, one can gain insights. Early practitioners (Card, Moran, & Newell; Hollnagel; Norman; Rasmussen; Woods;) may trace their roots to human information processing theory which subsequently influenced their conceptualization of cognitive systems engineering.

Yet, in every case the impetus came from real-world problems or application domains (e.g., process plant dynamics, Rasmussen, et al., 1994; robotics, Sheridan, 1992; the Pilot's Associate, Zaff, McNeese, & Snyder, 1993). This early emphasis suggests a lineage that is inherently related to ecological psychology (see Gibson, 1979) or more recently to naturalistic decision making (Zsambok & Klein, 1997). Early practitioners mainlined their own research in cognitive psychology as a scientific basis for CSE, but still paid attention to contextual issues. Perhaps in the book *The Psychology of Everyday Things*, Norman (1988) created mergers among human information processing, ecological psychology, and design issues. Rasmussen (1988) states that "human abilities and capabilities with respect to information processing behavior are closely related to the symbolical information features of the environment, and cognitive science will, therefore have to be akin to Brunswikian ecological psychology" (p. 332).

A major theoretical shift that occurred in the 1990s put even more emphasis on ecological precedence (e.g., situated cognition, Suchman, 1987; Young & McNeese, 1995). This shift consequently also influenced the development of new methods, tools, interfaces, and support systems, for example, computer-supported cooperative work (Whitaker, 1994), ecological interface design (Vicente & Rasmussen, 1992), naturalistic decision making (Zsambok & Klein 1997; Xiao, 1994), Scandinavian design work (Greenbaum & Kyng, 1991), and learning science (Rogoff & Lave, 1984).

Consequently, evolution in cognitive science has influenced cognitive systems engineering as well. Norman takes up the cause of "affordances" in design work. Rasmussen argues for the ecology of human-machine systems. Woods studies cognition in the wild. Sanderson, McNeese, and Zaff (1994) reference multiple data streams in sociotechnical systems. This shift also influences proliferation of CSE methods and tools. Klein employs the critical decision method/cognitive task analysis tools. McNeese applies the AKADAM method/COGENT tool to military situations. Sanderson explores new settings with the MacSHAPA tool. As new ideas perpetrate the "old guard" of cognitive

science, CSE continues to evolve and leave contrails for new theory, methods, and tools.

The following descriptive sections provide a limited, yet broadly specified perspective of different approaches to CSE. In many instances, a practitioner's designation under one of the constructs may seem arbitrary as they have gone on to develop many avenues of their programs that in fact propagate across many different levels of theory-methods-tools-practice-design. The examples presented herewith are classified with respect to *initial starting points* and/or the *primary area of influence* emanating from their work.

3.5.1 Theory-Driven Examples

3.5.1.1 The Work of Don Norman and Colleagues. Although Norman's work certainly flows into method and suggests ideas for tools, it primarily is theoretical and was foundational in establishing the field. Classifying Norman's work in the theoretical category is based upon his seminal work in cognitive systems engineering (specifically, Norman, 1986; 1988). Since then his work has actually been more attuned to ecological psychology, real-world concerns, and the design of artifacts (see Norman, 1993). It is also interesting to note that much of his early work in CSE evolved from his research in cognitive psychology. In fact, Lindsay and Norman (1977) wrote a classic book on the human information processing approach to psychology.

One way to trace Norman's view on CSE is to make an analogy to mechanical engineering, wherein theory, first-order principles, and laws (e.g., statics and dynamics) contribute as a science to the design of machines. Likewise, CSE is an approach to the design of human-information systems predicated upon the statics and dynamics of human behavior within an environment of action. The "act-in" aspect is important as a basic indicator of Norman's theory of interaction. At the heart of this theory is the principle that the intentions of humans must be translated into physical actions in the environment. The interpretation and translation required to do this involves a complex mapping problem; in other words, maps between the physical mechanisms (e.g., human-computer interfaces) and system states, and system states and correspondent psychological interpretation.

A theory of action focuses on doing things and the ensuing discrepancies between the psychological and physical. Discrepancies thus become anchors that open inquiry into major issues regarding design, analysis, and use of systems. Norman represents discrepancies as execution and evaluation "gulfs." Gulfs can be bridged by moving a system state closer to user's needs through the application of design, or by moving the psychological state closer to the system

by use of plans, action sequences, and interpretation that move goals and intentions in alignment with system states. Literally, the action taken to "move" towards intention is determined by what a person can do through his/her input alternatives. Hence, the degree of match/mismatch transpires through this gulf of execution. On the other hand, the gulf of evaluation requires a match between user goals and intentions through interpretation of the system state by the user. This gulf is revealed by output displays and is bridged through seeing the display, perceptually processing of display objects, interpretation, and evaluation in terms of how well initial intentions are carried out. As Norman indicates, complexity in the form of many levels of outcome and lack of immediate feedback play havoc with bridging these gulfs and eliminating discrepancies. Within this theory, behavior may be goal or event driven.

In addition to the theme of gulfs, Norman places emphasis on the user's *mental model* of the system, and the designer's model of the system and how these views interact to impact artifacts and discrepancies. Indeed, he notes that the user's model helps guide human behavior and thus transform confusing, difficult tasks into simple ones.

It is interesting to note that the evolutionary stance provided by Norman's work over the last 25 years represents the forces present in cognitive psychology, ecological psychology, human factors, and engineering design. As previously mentioned all these areas contribute as tributaries to what we have come to know as *cognitive systems engineering*. Norman's work is heavily influenced by cognitive psychology, human-computer interfaces, and design practice but does not really transition into the knowledge engineering/artificial intelligence areas as much. In this sense, it is very similar to the early Card, Moran, and Newell (1983) work. Yet, their work has progressed more as a method (the Goals, Operators, Methods, and Section Rulers model) which has been used and adapted by a variety of groups.

It is also informative to note that the theory of interaction historically is a cousin to basic human factors models related to control theory, and as Norman points out, to theories related to servomechanisms and cybernetics. So as tradition has it, Norman's approach represents a logical progression from psychology to design, from theory to practice. The influence of this approach has been voluminous in a variety of venues but one of the lasting values is that the work itself is elaborated based on first-order principles of human interaction.

3.5.1.2 The Work of David Woods and Colleagues. Like Norman, Woods' work has been very substantial in defining and developing the scope of CSE as a discipline. The "New Wine in New Bottles" paper (Hollnagel & Woods, 1983) is probably referenced more often than any other paper in the CSE discipline, and is considered a preeminent reference on the topic of CSE. Woods and his colleagues have had a continuing and major impact on the applications of CSE in

real-world environments (e.g. flight, medicine, information analysts, etc.). Perhaps the best example of this is in the field of human factors. For the first time at the 1995 Annual Meeting of the Human Factors and Ergonomics Society, a number of sessions, panels, and papers were approved by a newly formed technical group in Cognitive Engineering and Decision Making. Through the efforts of David Woods, Gary Klein, and many other practitioners, the dream of CSE is finally becoming a force in the way human factors is practiced. Woods has been very active in translating initial investments into actual methods and cognitive tools while continuing to wave the "cognition in the wild" flag to signify the importance of dynamic worlds.

In some ways, because of the early focus on dynamic worlds as a critical piece of the cognitive engineering scenery, Woods perhaps began with more of an ecological appreciation than some of his peers. Therein, we see impressions of this through such characterizations as dynamic worlds, distributed cognition, cognition in the wild, and natural environments surface to the forefront in Woods' publications. Although this is not necessarily indicative of a strict Gibsonian approach, it is certainly predicated on the role of context. As Woods refers to dynamic worlds there is a point that research in CSE is subject to the interdependencies/symbiotic nature of real-world problems. In this sense, Woods classifies CSE as being "problem driven."

From the problem comes the basic elements by which cognitive systems engineering unfolds—challenges, descriptions, agents, inadequacies, successes, demands, performance, computation, available resources—that in turn specify requirements for a cognitive engineer to follow. Often these elements are explored and defined by use of cognitive task analysis (Gill & Gordon, 1997) or cognitive work analysis (Vicente, 1999). By this attraction then, it is not surprising that Woods has also made much progress in the study of human error (see Woods, Johannesen, Cooke, & Sarter, 1994 for review), an area that could be considered either a part of or a cousin to CSE. The Woods approach then at least conceptually is related to Norman's (1981) work on action slips and errors as well as Reason and Mycielska's (1982) work in this area. Human error may very well be the lens that allows theory and practice to meld together, wherein cognition meets up with design.

One theme that resonates at the heart of Woods' approach (Woods & Roth, 1988) is the idea that the impact of computers in dynamic worlds results in operators experiencing more mental work and more complexity in general. As a consequence, these demands require creation of the appropriate cognitive descriptions and environments to avoid pending failures/errors. Unfortunately, this is easier said than done. Descriptions and representations frequently may produce brittle rule-based systems that cannot adapt to the dynamism experienced in the real world. Woods captures this dilemma through use of a tri-factor model that looks at problem-solving as interaction among dynamic worlds, agents that act upon that world, and the representations the agent uses to expe-

rience that world. Hence, CSE must put *cognitive tools* in the hands of problem solvers to cope within their natural problem solving habitat and avoid mismatches that can occur between the world, agent, and representation. This occurs by focusing on what is missing (or incomplete) in a habitat that could lead to patterns of errors.

Another important issue that Woods refers to in his work (not utilized as much as other aspects of his position) is the problem of *activating knowledge* under natural conditions of "uninformed access." Informed access conditions occur when operators are "told what to do" without activating knowledge on their own (uninformed access). When cognitive tools are only designed from an "informed access" perspective, the operator may not understand what the tool is doing and consequently be led into situations where error is likely. However, if the designer is aware of the conditions that lead an operator to overcome inert knowledge, then the cognitive engineering of the tool is more likely to lead to success in a real-world setting. This view of CSE derives from the tradition of research by John Bransford on spontaneous access of knowledge (see Bransford, Sherwood, Vye, & Rieser, 1986) and serves as a theoretical link in knowledge acquisition. In fact, the work by McNeese and colleagues (McNeese et al., 1995; Young & McNeese, 1995) is strongly predicated from this tradition and in this sense, historically similar to Woods' theoretical stance.

There is one other thread that co-occurs with the discussion of dynamic worlds that does not appear as much in other approaches. CSE is not simply a single operator endeavor. It must take into consideration the effects of multiple cognitive agents, distributed decision making, group processes, and organizational constraints. If one is true in allegiance to dynamic worlds then it is difficult to avoid contact with collaborative work settings. Because Woods' approach highlights this component of emerging cognitive systems, this is a testament to his early insights in this area. This focus has been carried through as many of his colleagues' work has taken place in collaborative, naturalistic settings. The focus on collaborative agents resonates quite well with Klein's and McNeese's work in military environments involving group processes (McNeese et al., 1995; McNeese, Rentsch, & Perusich, 2000). In fact, much of my own work from the last five years involves complex worlds that include multiple, active agents (e.g., battlefield management and planning, AWACS command and control, design teams, crew station interaction, cooperative learning teams).

Even though Woods' work is heavily focused toward creating cognitive tools, the work is considered here as a theoretically driven, principled approach to design which in turn the practitioner must pursue. In contrast (for this chapter) tool-driven approaches refer more to actual products that enhance CSE application. They are still cognitive tools in the Woodsian sense, but cognitive aids for cognitive engineers. The presumed application of Woods' approach will in fact lead to actual tools in a specified environment (Woods, 1998).

3.5.2 Method-Driven Examples

3.5.2.1 The Work of Gary Klein and Colleagues. Klein (1990) suggests that cognitive engineering is the attempt to design systems that are better adapted to the thought processes of users. His work has been very influential in defining how people make decisions in everyday naturalistic environments. From that perspective, decision systems and human-information interfaces may be designed to enhance the power of the user. Outcomes associated with Klein's work (see Klein, Woods, & Orasanu, 1993) have strengthened the ties between the cognitive engineering and naturalistic decision making areas while emphasizing (1) inherent demands and strategies in temporally evolving events, (2) the role of action in cognition, (3) the coupling of perceptual recognition with action, (4) the role limited resources play in cognitive effort, (5) how experience and reasoning strategies affect courses of action, and (6) the competence rather than the failure of decision makers. Klein (1993) developed a recognition model of decision making, highlighting cognitive task analysis methods/tools, all of which derive from study of real-world situations and subject-matter experts. More recent work poses how team processes and mental simulation are important to engineering cognitive systems. Examples of application include assessment of neonatal nurses, designer support analyses, hacker profiles, systems required for battlefield operations, firefighter strategies, and team coordination in aircraft crews.

Obviously, given space considerations, these approaches and applications could be significantly broadened and described in more depth. Yet, one can begin to get a feel for the definitions, features, differences, and similarities that mutually establish what is meant by a cognitive systems engineering approach to human-information systems.

Although the work of Woods and Klein both employ contextualist views that harbor the advantages of engaging expertise, making cognitive descriptions based on problems encountered in real-world settings, and designing tools that are participatory in nature, the approaches are laced with cognitivist references such as mental simulation, attention, knowledge activation, cognitive representation, etc. that signal crossover effects from cognitive psychology rather than carrying the ecological psychology banner of Gibson. The approaches may bear more lineage to the work provided by Brunswick than Gibson, and therein are not oppositional per se but somewhere between cognitive and Gibsonian-based ecological psychology.

3.5.2.2 The Work of Jens Rasmussen and Colleagues. Rasmussen, Pejtersen, and Goodstein (1994) define the scope of cognitive engineering as (1) *cognitive*

because thinking, problem solving, and decision making maintain a greater level of influence in our everyday lives than physical strength and dexterity; and (2) *engineering* because the development of concepts, methods, and tools can positively impact system designs that help users carry out daily activities. Historically, the work of Jens Rasmussen has contributed more to the field of CSE than that of any other person.

His activities in CSE encompass broad dimensions covering such topics as work domain representation, mental strategies, distributed decision making, social-organizational processes, and the division/coordination of work. These dimensions frequently analyze domains at several levels of abstraction by looking at parts that form a whole, or means that establish ends. Decomposing worker domains allows engineers to design in response to complexities inherent in the context. Rasmussen et al. (1994) suggest a function (*what* is used) can be seen both as goal (*why* it is relevant) or as a means for a lower level function (*how* it is realized). The model of expertise and decision making (Rasmussen, 1986) portrays an adaptive use of skills, rules, and knowledge to govern different types of behavior necessary while experiencing demands in a complex setting. His approach has been applied to the design of nuclear power plants, library retrieval systems, electronic trouble shooting, unmanned air vehicles, and medical decision making in hospitals, to name just some of practice he has investigated.

A few of the leading luminaries' approaches have been briefly described to show examples of how CSE has developed in similar yet different ways. The chapter will now change from discussing other practitioner perspectives to outline how we have discovered and developed CSE through work, research, and practice over the last 15 years, primarily in military and aviation contexts.

3.6 QUERY V: WHAT HAS TRANSPIRED?

3.6.1 A General View

One of the ways to capitalize on understanding CSE is to create a general framework that covers some of what we have reported already. Figure 2.2 shows a framework in which there are four major components that practitioners must consider in consort to practice CSE: (1) work in context, (2) analysis of cognitive activities, and (3) engineering of cognitive systems, and (4) first-of-a-kind artifacts. A cascading flow of activity is spawned from the study of work in context and how artifacts influence the context and the work. Using various methods to analyze different cognitive activities complements observing a field of practice. As analysis of cognitive activities ensue they must feed and be the foundation for engineering designs of cognitive systems.

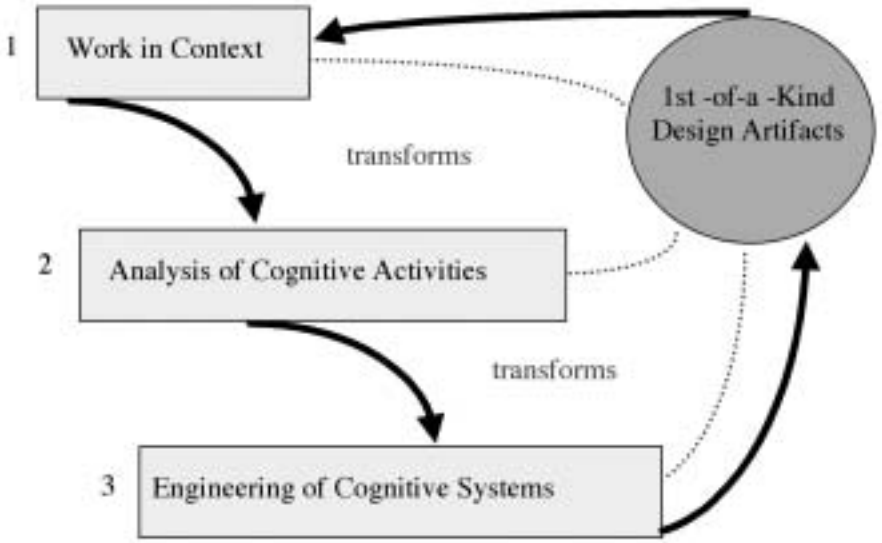


Figure 3.2: A general framework of cognitive systems engineering.

Engineering cognitive systems is the basis for deriving various first-of-a-kind artifacts which in turn are tested as *in situ* interventions. The relationship among these components is governed by multiple transforms that allow various versions of artifacts to emerge as solutions to solve situated problems.

In considering this general framework as we look over 15 years of activity, it is clear that at times the research has clearly not been balanced with multiple transforms but has been directed at examining different components at different times. One major challenge and an issue that have arisen is how to create and maintain balance among cognitive science, cognitive systems engineering, cognitive modeling, cognitive ergonomics, and software engineering while still keeping the field of practice as a primary object of interest. More often than not these specified areas can be at odds rather than showing any sense of integration. As we review different threads of activities it is clear that there needs to be a crosswalk to bridge these voids. However, such a crosswalk at this time is just beginning to unfold.

The following stages represent distinct growth patterns in my own discovery of CSE. They are used to communicate what I have learned and value as important, and provide a trajectory of development with varying sources of influence.

3.6.2 The Neonatal Stage

Upon reviewing my first ventures in CSE there is a clear sense of connectedness to some of the representative perspectives and converging themes presented earlier. My first venture into this area was a field study conducted at the University of Dayton in 1977 as part of a senior thesis project. The study entitled, *How People Space Themselves Out in University Places* (McNeese, 1977), used ethnographic methods and falls into the work in context component of the framework. This study was conducted with streams of influence from Barker (1968) relating to designed ecologies, and Gibson (1979) relating to perceptual learning. The gist of this early work was the nature of the design of spaces. Ironically, many of the issues of interest 25 years ago—how people produce spaces and effective space design as a function of social, cultural, and political use of space—are still a cogent topic of concern today.

My study investigated different people in different settings engaged in different activities (from an observational approach) and posed innovative design ecologies for given intents. At the time, "systems engineering" pointed more towards the *physical use* of places. Today, the study of *information spaces* and how people share information spaces at various places is a hot topic for CSE in general.

One other early direction ensued during my first five years of employment at Wright-Patterson Air Force Base (WPAFB), Ohio. I was placed in charge of developing the Aeronautical Systems Division's *Human-Computer Interface Mil-Prime Standard* (a set of guidelines for implementing and managing HCI for aviation concerns). During the early 1980s I was also assigned to two programs (based at the Aerospace Medical Research Laboratory at WPAFB) as a liaison for human factors and cognitive systems to work on the systematic development of different kinds of automation to be integrated into the cockpit (the *RAM/ACE* and early *CAT* programs; see McNeese, Warren, & Woodson, 1985).

This was a time when cognitive science, artificial intelligence, expert systems, situation awareness, and Human-Computer Interaction (HCI) were beginning to build with momentum and be incorporated into the plans and programs of the aeronautical/aviation systems of the future. The streams of influence for this time were first—Jens Rasmussen. I was greatly influenced by his "Human as a System Component" paper (Rasmussen, 1980), followed by other papers of his in the early-to-mid 1980s (e.g., Rasmussen, 1983). Another influence during this neonatal period was recognition of early human performance modeling efforts in aviation systems (Pew, Baron, Feehrer, & Miller, 1977) and the potential role it could have in simulating complexities. This was looked upon in terms of technology assessments of automation but directed towards how the technology supported cognitive abilities. In sum, this period set the stage for much of my work in CSE even though I was not fully aware of this at the time.

3.6.3 The Toddler Stage

In 1984 I transferred from the Air Force Aeronautical Systems Division to the Air Force Aerospace Medical Research Laboratory (AAMRL) while still on-site at WPAFB. Work began as part of a research group called COPE—C^3 Operator Performance Engineering. The goal of the COPE program was to study and produce user-centered command posts that incorporated human information technologies. COPE as the name indicates was an early form of CSE but with a unique vision of working with teams and collaborative units in real-world operative domains. Like the neonatal stage, the importance of situated problems in context was at the heart of a lot of activities; however, work under COPE provided many innovative facets of research that were both qualitative and quantitative, and pointed towards a true balance of work-analysis-engineering-design-intervention transforms indicated in the general framework. COPE work also generated exposure to some very innovative information technologies (e.g., large group displays, speech recognition systems, adaptive interfaces, and video teleconferencing).

Cognitive and collaborative system work in those days provided exposure to informational and organizational analysis, fieldwork to redesign real command posts, interactive and scenario-driven simulations, conflict resolution in cognitive systems, decision aids, and knowledge acquisition. In the COPE program team analysis was used to develop new large-group display technologies. In turn, these technologies were used for in-house experiments to assess theories of group process and team performance. This would be my first experience with what I would later refer to as a "living laboratory" approach (McNeese, 1996a).

Major streams of influence during the COPE experience were Jens Rasmussen and his way of modeling systems, Gary Klein and his preeminent work with expertise and field research (Klein, Calderwood, & Clinton-Cirocco, 1986; Klein, Calderwood, & MacGregor, 1989), William Rouse and his research on mental models and diagnosis in natural settings (Rouse & Morris, 1986), and Kenneth Hammond and his cognitive continuum theory (Hammond, McClelland, & Mumpower,1980) that suggested cognition lies on a continuum between analytical and intuitive poles, and varies according to situations. COPE hence became a time of grounding myself in various venues, with new theories and methods, while working in real world domains of (1) command and control, and (2) aviation support systems.

3.6.4 The Formative Years

As I continued to work at WPAFB in aviation and military-based C^3 settings, and then continued my graduate education at Vanderbilt University, several new streams of influence extended my understanding of CSE. One of the key

elements that proved to be a useful method for modeling agent-environment transactions was protocol analysis. As modeling continued to be valued, protocol analysis became one methodological bridge between cognitive science and CSE. The pioneering work of Herb Simon in analyzing and modeling cognition using protocols (Ericsson & Simon, 1984) informed my own dissertation work which was also enhanced by a protocol analysis tool, SHAPA (Sanderson, James, & Seidler, 1989) designed by Penny Sanderson to analyze and encode transcripts. One of the key insights at this time centered on creating a situated model of activity from the data itself using the protocol analysis tool as the basis to explore real-world transactions.

A very strong influence during this time was the work of my Ph.D. dissertation advisor, John Bransford, and his research on learning, problem solving, and the role perception plays in how people acquire and use knowledge (Bransford et al., 1986). The more theoretical work of the late 1980s became the basis for our cognitive task analyses of the 1990s.

This time also provided opportunities at Vanderbilt to work on macro-contexts and scaled world problems. Macro-contexts (e.g., the Jasper paradigm, see Young & McNeese, 1995) provided much of the affordances and perceptual learning of real worlds but within an experimental environment. This period continued to place much emphasis on distributed collaborative activities both at WPAFB (Snyder, Wellens, Brown, & McNeese, 1989; Wellens & McNeese, 1987; Wilson, McNeese, & Brown, 1987) and at Vanderbilt. Also the work of David Woods on cognitive demands and process tracings provided evidence of what CSE was coming to mean to me. As far as the components of the general framework, this period opened up new possibilities in the analysis of cognitive activities both with tools and with scaled worlds.

3.6.5 The Preteens

The early-to-mid 1990s was really a time when much of our solid work in CSE was established and where my group carved a niche as part of the whole CSE movement. At this point I was back at WPAFB and was working on a new program, the Pilot's Associate (PA). This program had as an intent to integrate much of the new research and technology into the fighter cockpit. However, one of the major problems experienced in knowledge engineering was that of the knowledge bottleneck. Our group at AAMRL was employed to advance new ways to develop intelligent cockpits from a user-centered philosophy. As I mentioned, the way people learned, the process by which individuals acquired, constructed, and accessed knowledge in natural, situated problems significantly influenced the methods we created for the Pilot's Associate, (see Zaff, McNeese, Snyder, & Lizza 1991). The set of techniques we used for the PA—termed the Advanced Knowledge And Design Acquisition Methodology

(AKADAM)—would be used in many other domains and become the anchor for our work throughout the 1990s (McNeese et al., 1995).

AKADAM (McNeese, Zaff, Peio, Snyder, Duncan, & McFarren, 1990; Zaff, McNeese, & Snyder, 1993) first and perhaps foremost utilized concept mapping as a user-centered knowledge elicitation method that employed different elements of cognitive task analysis to represent the complexities of user beliefs about their work setting, advanced technologies, and the context of work itself. AKADAM, as a CSE technique, was used to elicit knowledge and convert that knowledge into useful design artifacts. AKADAM covered all the elements of the general framework and highlighted the principle of "knowledge as design." Although very participatory in nature, AKADAM was very much "off-line." That is, users heeded knowledge about past experiences, cases, or stories which would be relevant for a given mission area (e.g., pilot target acquisition).

The AKADAM approach employed use of task networks (such as Integrated computer-aided manufacturing DEFinitions [IDEF]), concept mapping, and design storyboarding as multiple perspectives that yielded user-centered representations that could evolve into interface designs. Concept maps were developed in consort with experts (pilots, engineers, and designers) as a mapper explicitly represented expert knowledge drawn out as a concept map on a whiteboard. In essence, it was an interactive, graphic diagram of an expert's mental model about a situation—developed over various timeframes. We would do at least two interviews with an expert at different times to draw out different takes on knowledge. The experts—after they were done with their map—would take the map home and review it some more whereupon they could change it even further. Once we completed interviewing an expert we would summarize all the experts' maps as part of a summary map which often consisted of over 1000 concepts and relationships. Concept maps could be adapted to represent both declarative and procedural knowledge.

In addition to the concept mapping element of AKADAM we also utilized a functional decomposition/task network modeling technique (IDEF) to capture detailed elements of a mission. The third element, design storyboarding, placed experts in the role of designers to translate concepts into design interfaces for each leg of the mission geometry. These three techniques, concept mapping, IDEF, and design storyboarding, formed AKADAM. We used and adapted AKADAM in many different ways over the years and have used it successfully with different kinds of experts in unique fields of practice.

One other important aspect of AKADAM was that it became the impetus for us to develop a toolset (TAKE—later named COGENT) based on the concept mapping element (Sanderson, McNeese, and Zaff, 1994). Part of my group at WPAFB used TAKE to develop some significant design work for helicopters and other domains. This was part of the overall AKADAM philosophy—to develop CSE tools as we applied them in the field. This time was also prime for discussion of new tools both with Penny Sanderson and John Flach

to generate effective new approaches to work domain analysis—predicated on Rasmussen's abstraction hierarchy. The intent behind all these tools was to test and apply them in the field.

During this "preteen" period the lab also assisted in funding, and helped develop another significant CSE tool, MacSHAPA, through the expertise and guidance of Penny Sanderson (Sanderson, McNeese, & Zaff, 1994; Sanderson, Scott, Johnston, Mainzer, Watanabe, & James, 1994). This was a tool from the tradition of exploratory sequence data analysis and utilized comprehensive video analysis—coupled with the protocol analysis functions in SHAPA. It significantly enabled videotapes of human activity in context to be analyzed in new ways. It allowed efficacious models to be built based on in-situ data. By juxtaposing MacSHAPA with the AKADAM methods and the TAKE/COGENT tools we had an effective observer-participant approach to CSE (Sanderson, McNeese, & Zaff, 1994).

These various methods/tools formed a core for conducting a comprehensive cognitive engineering analysis of complex systems (McNeese, 1996b). They placed much of the responsibility for design on the user and were focused on acquiring user knowledge that spanned the gap to performance in context.

One final major event during the 1990s which significantly empowered our CSE approach in a specific domain was the establishment of the Collaborative Design Technology Laboratory (CDTL) (seeWhitaker, Selvaraj, Brown, & McNeese, 1995). This lab really covered all the elements of the general framework and practiced a "living lab" philosophy. The intent was to develop user-centered collaborative design technologies (Whitaker, Longinow, & McNeese, 1995) to enhance design teams in real-world settings (Citera, McNeese, Brown, Selvaraj, & Whitaker, 1995). The CDTL afforded multiple opportunities to put into practice many ideas of CSE and allowed ethnographic study, reconfigurable testbeds, as well as experimental lab projects (see Brown, Selveraj, Whitaker, & McNeese, 1995; Citera, Selveraj, McNeese, Brown, & Zaff, 1995). True to the spirit of the 1980s, the focus was absolutely on situated, complex problems, distributed collaboration, and developing technology to support teamwork. The thread relating CSE to teams continued to be assuaged but in new ways with new fields of practice.

3.6.6 The Restless Teenager

From the mid-1990s to 2000 many events coupled with CSE took place at the Air Force Research Laboratory. Methods certainly matured but there were other questions, issues, and challenges that have remained unanswered. First, let's take a look at some of these events that composed my continued discovery of the meaning of CSE from my own personal perspective.

One of the main events was the intent to use cognitive models and develop models as informed from cognitive task analysis. Task network modeling continued (informed by the use of concept mapping as a *cognitive task analysis*) as we developed models of fighter pilots engaged in target acquisition (Bautsch, Narayanan, & McNeese, 1997). This research used a Maverick launch mission task in a synthetic task environment to compare actual performance results of pilots (given different experimental conditions) with (1) an extended air defense simulation (a model without "user" parameters), (2) cognitive models as informed and built as a function of HCI guidelines and behavioral-based task analysis, (3) cognitive models informed and built as a function of cognitive task analysis, and (4) a cognitive architecture-based simulation, SOAR (Darkow, Marshak, Woodworth, & McNeese, 1998).

Modeling cognition in applied contexts is informative for knowledge acquisition, establishing a basis for design, and for exploring constraints of the problem space itself. Our results found that the type of task analysis performed influences the performance of the model created in reference to the baseline pilot performance. The research demonstrated that adaptive cognitive capabilities (e.g., in SOAR) were really not needed to capture some of the well-defined procedural sequences of the mission. However, on more advanced, ill-defined requirements these capabilities would absolutely be necessary to capture activities.

More current work in cognitive systems engineering/modeling focuses on addressing one of the wicked problems we encountered in our early work. It is coupling the knowledge elicitation processes and model representation to transform early concept maps into dynamic models (McNeese, Bautsch, & Narayanan, 1999). Early work was beneficial in transforming conceptual knowledge of the user (and his/her context) into design artifacts for a given mission (Zaff, McNeese, & Snyder, 1993). However, the concept mapping component of AKADAM—although very good for eliciting early issues, constraints, and design requirements—was too *loosely coupled* (as a knowledge representation typology) to directly transform knowledge into dynamic models. This is especially the case for situated contexts that require a high degree of interdependent, collaborative activity.

One recent project (McNeese, Rentsch, & Perusich, 2000) focuses on modeling C^3I operations using the *fuzzy cognitive map (*Perusich et al., 1999) method. Note the field of practice of interest has now switched to the area we initially started with in the original COPE projects—one of the ironies of this developmental process. However, the domain is much changed, is more global and multi-team focused, and includes use of many new information technologies (not the least of which is the internet and associated telecommunication technologies). Perhaps the term "battle management" connotes the broader level of coverage. The battle management domain absolutely incorporates many aviation-level contingencies (e.g., addressing new cognitive system concerns attributed to remotely controlled, unmanned air vehicles).

The initial assessment of the fuzzy cognitive map technique indicates it is an efficacious way of eliciting knowledge in a form that is easily and directly transformed into models of team processes/team decision making. Our assessment is based on contrasts with previous cognitive task analyses and cognitive engineering techniques we have applied to various fields of practice. The method can be generalized to many "collaboration in the wild" problems and is useful for simulating the differential effects of socio-cognitive and factors as they influence team effectiveness.

Our intent, given the tradition of the living lab philosophy, is to develop models (individual or team-level) that are used as the basis to (1) design decision support systems, cognitive aids, human-computer interfaces, or intelligent agents embedded in complex systems and, (2) be a way to easily test interface designs, support systems, and agents for how well a human (or a team) interacts and performs given certain constraints. Part 1. allows CSE to incur as it takes the "knowledge as model as design" view whereas part 2. exploits the model as a tool to be used to test/evaluate designs without having to do full-blown, timely experiments. One may also use these test/evaluations as preliminary data before the "intervention" part of the living lab cycle. That is, the model can help assess problems in a proposed design prototype and therein suggest improvement prior to actually embedding it in a field of practice for testing and usability analysis. Fuzzy cognitive maps have been the main technique and advancement to the early 1990s work on concept mapping and allow us to use models to capture "knowledge as design."

Fuzzy cognitive maps are directed di-graphs that contain (1) fractional edge strength values among variable concept states, and (2) feedback that affords a qualitative model of a complex system. They may be used to build a model of team— environment transactions wherein goals, information, physical attributes, value judgments, behaviors, decisions, and quality valuations are all needed to represent boundary constraints and emergent complexity (Perusich & McNeese, 1998a). They capture the causal reasoning of the expert constructing a system (or decomposing a problem) and are especially useful in situations where a variety of variables must be compared that lack a common numerical metric. Therein, they can be used to model decision making in teams that involve complex tradeoffs between disparate causes and effects because "apples and oranges" comparisons can be seamlessly made (Perusich & McNeese, 1997). Rather than forcing comparisons by transforming the measures of each variable to some artificial numerical scale, such as utility, each is represented by its states.

In contrast to traditional concept maps (McNeese et al., 1995) which tend to facilitate more surface-level maps of knowledge, fuzzy cognitive maps enable a deeper structure of knowledge (within the knowledge elicitation/data abstraction process and the subsequent qualitative modeling activities). Intentions, causes, and effects are represented as nodes in the map. A cause-

and-effect relation ship is represented in the map by a directed edge from one node to another. In a cognitive map, the strength of these edge strengths is limited to the values 1, –1, and 0, representing "A causally increases B," "A causally decreases B," and "A has no effect on B," respectively. In a fuzzy cognitive map, degrees of causality are incorporated by allowing edge strengths of any fractional value in the closed interval [–1,1]. Fuzzy causality, such as "A somewhat causes B" or "A causes B a little," is incorporated in the map by letting the edge strengths have an appropriate fractional value, such as 0.4 or –0.6.

As an "associate memory," inference in FCM is conducted by applying a set of input variables and allowing the map to equilibrate a stable output or observing a limit cycle. The presence of feedforward and feedback mechanisms in the associative memory creates causal loops that help understand and model the dynamics of complex systems (Perusich & McNeese, 1998b).

One of the advantages of the FCM representation was that it was fluid enough to capture many of the front-end constraints in problem finding but sophisticated enough to transform knowledge representations (captured from the elicitation process) into active, dynamic models. FCM can be applied to individuals but more importantly it has a unique capability to integrate many of the qualitative judgments that multidisciplinary team members need to make. The method can make conflicts explicit (as part of the representation) and equilibrate the consequences of conflicts in the overall problem space. Likewise, the multiple constraints of interdependent collaborative actions, the modeling of emergent complexity, and the effects of expectation and causality can be represented (using aspects of fuzziness) and equilibrated to simulate teamwork and socio-cognitive factors given a specific situated context.

Our initial assessments of FCM have been very positive. In contrast to many of our previous methods of cognitive engineering, the FCM approach provides a dynamic model of collaborative activity—directly instantiated from knowledge elicitation activities while maintaining user-centered, participatory principles. It also affords users the opportunity to directly experience (and incrementally change) their mental model in a way that resonates with their expectations.

The goal of our research is to understand socio-cognitive factors in situated contexts with specific application to military or other operational domains. As mentioned, FCM is just one component of our overall vision. Future work will integrate results of our team models with team schema similarity measurements (Rentsch, McNeese, Pape, Burnett, Menard, & Anesgart, 1998) using the living lab perspectives (McNeese, Perusich, & Rentsch, 2000). These models are envisioned to evolve into *embedded software mediators* that can assist in managing global, battlespace, and shared situation awareness, that is, can systematically be inserted as team members as part of a crew. With the reduction in crew size (a major issue to be addressed in the military), the capability of having software mediators available to a reduced-size crew will be very important for enhancing shared situation awareness in distributed environments. The

use of FCM-based software mediators to assist in C^3I teamwork with a reduced crew size is the next phase of our research. Obviously, we are interested in how the teamwork schemas of crew members will vary when software mediation is present, and as a consequence, how this can feed back into the development of fuzzy models.

As work continues there is recognition of a need for various kinds of collaborative task analyses within CSE (see McNeese & Rentsch, 2001). To this end, we have been eliciting and assessing team schemas/team member models as new kinds of knowledge to model and in turn to use for design.

One other major development has been the need to develop a Computer-Aided Cognitive Systems Engineering (CACSE) tool that acts as a support system to the cognitive systems engineer. Through the use of a Small Business Innovation Research-based development operating under the program leadership of Scott Potter, Emily Roth, and David Woods, we developed CACSE for that purpose (Potter, Roth, & Woods, 2000; Potter, Roth, Woods, & Elm, 1998). A main intent in this project was to take the knowledge developed in cognitive task analysis and make it useful and relevant for software engineers using their CASE tools. The project was very successful and produced a prototype tool which in turn was tested in a real-world application. Although the tool itself is a design hypothesis about how as an artifact it shapes CSE activities (Woods, 1998) (and requires revisions to go to the next level of development), a new path has been established.

Tools such as CACSE are perhaps the future as to where CSE will lead. Tools are necessary to make CSE work easier and doable given various constraints in place. The CACSE tool also provided a culmination—personally—as it is predicated on Rasmussen's abstraction hierarchy and much of Wood's understanding and contributions to CSE. However, by creating a real tool to be used by the software engineers who actually create designs for human-computer interfaces and support systems, much new knowledge about how to advance and use the abstraction hierarchy was produced (see Potter, et al., 1998; Potter et al., 2000 for more information on CACSE).

The development of CACSE reveals a future requirement for the CSE field. It is that CSE is not an island unto itself. As a continually evolving profession there is much necessity to migrate into work with other specialties such as was demonstrated with software engineering. One must continually ask the question of what CSE, cognitive modeling, and cognitive field studies point to and what is the "plan" to get from here to there(see McNeese, Baustch, & Narayanan, 1999). The degree of relationship and utility to other tangential specialties will establish credence and validity for the field. As has been told in this query—there is an absolute need to make "design" a core value in CSE.

3.7 QUERY VI: WHAT HAS EVOLVED/WHAT HAS BEEN LEARNED?

Obviously, the last query had many peaks and valleys, traversed uncharted territories, and some cases ended up in vast wastelands rather than productive fields of practice. Still, the value of discovery is that you come to recognize limits, failures, and boundary constraints of practice. Through discovery you make right and left turns to go in new directions. This query and the next assess what exactly evolved over the last 15 years and what has been learned as a function of the journey.

One way to examine the extent of evolvement is to look at different kinds of outputs and products as the lasting contributions in the field of cognitive systems engineering. The research and development in CSE during my tenure with the U.S. Air Force focused on the mutual interplay of understanding, modeling, and measuring teamwork and individual activities within complex systems. At the heart of this research has been the desire to apply innovative *theoretical orientations* (humane intelligence, McNeese, 1986; situated cognition, Young & McNeese, 1995; group sensemaking, Nosek & McNeese, 1997; socio-cognitive factors, McNeese, 2000); *cognitive systems engineering frameworks/techniques* (AKADAM framework, Zaff, et al., 1993; McNeese et al., 1995; cognitive fieldwork, McNeese et al., 1999; fuzzy cognitive mapping, Perusich & McNeese, 1997; collaborative task analysis, McNeese & Rentsch, 2000; Living Lab, McNeese, Perusich, & Rentsch, 2000); *CSE tools* (COGENT/MacSHAPA, Sanderson, McNeese, & Zaff, 1994; CACSE, Potter et al., 2000); *cognitive modeling methods* (Bautsch & McNeese, 1997, Perusich & McNeese, 1998a); and *scaled world research paradigms* (Automate, Citera et al., 1995; Maverick Mission, Bautsch et al., 1997; Jasper, McNeese, 1992, DDD, McNeese, Rentsch, & Perusich, 1999; Patriot, McNeese & Perusich, 2000; TRACES, Brown, et al., 1995; CITIES, Wellens, 1993; TRAP, Wilson, McNeese, & Brown, 1987); *designs* (team display-group interface, McNeese & Katz, 1987; McNeese & Brown, 1986; Whitaker, 1994; PA interface, Fraser, Hipel, Kilgore, McNeese, & Snyder, 1989, McNeese et al., 1990). Other outputs exist as well but these are the major objects that have evolved.

Upon looking at these various objects there are some "recombinant themes" that have been learned that may be critical for addressing challenges and issues for the future of CSE (to be discussed shortly). As we search for invariance across the evolutionary development of CSE from the first field study of the student union to the most current model employing fuzzy cognitive maps, there are some insights that stand out. First, the *design of spaces* is a key element of CSE—whether it be physical, informational, collaborative, or global-oriented. CSE is about knowing the constraints of a space and how a space is used to enable and advance human activities. Second, studying *fields of practice* with new tools always leads to new insights about what is wrong or what could make a practitioner more effective. So much can be learned from field-

work studies by use of a different—yet integrated—set of CSE tools. *Protocol analysis* is very important to capture different kinds of perspectives and acts to bridge some of the gaps we have mentioned. One of the newer advances that is showing an increasing rate of return for research is experimental scaled worlds (predicated on knowledge from CTAs that inform with respect to the context and user constraints). We are seeing more and more examples of scaled worlds that emulate real-world problem spaces but are to scale for efficiency and effectivity. Finally, we have learned that a loosely coupled, eclectic set of techniques without the ability to tie analysis to design, and without the ability to integrate with each other in a meaningful way, can leave the CSE specialist asking more questions than she/he answers. This can also lead to sloppy approaches to fieldwork and participant-observer methods. With respect to these various themes, there are two summary ideas that I would like to posit.

In the introduction to the query on what I have done over 15 years I posed a generic framework to simply capture the many directions I have pursued in the different periods of my CSE life. That framework has been specifically reified to be known as the living laboratory approach to CSE. We first introduced it in the mid-1990s (Whitaker et al., 1995; McNeese, 1996a) but have been using it quite a bit as a shield representing our beliefs concerning CSE. At the heart of this approach is a pulse which is repeated throughout this paper—the value of learning, discovery, and improvement. The living lab is a way of research life that places value on discovery through different venues, concurrency, ecological validity, feedback, mutually informative processes, technological intervention, and the willingness to broadly approach complex problems without dogmatic, doctrinaire biases.

As reflected upon CSE—the goal of the living lab is to enable CSE practitioners (working with specified fields of practice) to become a community of learners. As an active research community there is always the presence of the specific and how to attune support in that way. However, a more challenging goal is to recognize how the particular folds into the universal (Woods, 1998). CSE can make more effective and safer interfaces for a given project and domain, but the real goal is to learn principles across many fields of practice that point to universal findings about cognitive systems. One might say it is necessary to go from a field of practice to a *field of knowledge*. If we only address the particular and specific, CSE as a field of knowledge will suffer. This is not only true for a domain but may be applicable to other elements as well (e.g., tools used, technology intervention, scaled worlds, etc.). By spanning specific targets to abstract what is similar across contexts, learning, discovery, and the meaning of CSE can rise to new heights.

The most recent manifestations of a living lab view are the CACSE framework/toolset (Potter et al., 2000) and a new systematic approach for conducting cognitive field studies (McNeese, Bautsch, & Narayanan, 1999). Although

there is not enough space in this chapter, these new directions show how we have used what we have learned to advance the state of CSE.

3.7.1 Summary

In concluding this query, I wanted to state explicitly specific things learned along the way—that I consider of value for other CSE practitioners. These thoughts are not unique per se and I am sure they have been discovered by others in this field as well:

- Nothing is perfect—especially in a complex world
- Several integrated, bootstrapped approaches are better than single, or piecemeal approaches
- There are various gaps that CSE can help to fill, (e.g., What people intend vs. what people know vs. what people do which are all complicated by teamwork settings, in other words, CTA →cognitive modeling →cognitive design →software engineering →decision-based test & evaluation)
- Gaps are filled by people who pick up and transfer context in spaces (physical, perceptual, informational, social, cultural, emotional)
- Context rules! Constraints have to be known to control adaptive process but they often are perceptually bound
- Individual differences are very active and must be considered but so too are social norms, cohesiveness, and teamwork activities
- It takes time to develop accurate models of users at work and to develop the underlying engineering of joint cognitive systems
- CSE as a scientific endeavor is lagging behind but as an engineering specialty is moving forward
- Time in a field of practice is usually constrained—time with practitioners is short and often restricted—plan accordingly, practical, proprietary, security, safety concerns
- Practice of CSE often is not integrated with ongoing design engineering practice—CSE becomes an island
- Reliability, validity, and verification issues still nag practitioners but answers are slow to come forth—there often is a bias explosion on the part of the approach and the practitioner conducting a study
- Science/Philosophy/Engineering faultlines—"turn the crank" mentality pervades often
- Too much CTA—not enough design
- How to predict beyond what is given based on cases/domains not known—the envisioned world problem
- "What is the meaning of this" issue—multiple paths, beliefs, percep-

tions may be difficult to capture and then represent as models.

These are a few of the discoveries made that have informed continuing evolution.

3.8 QUERY VII: CONCLUSIONS—WHAT'S NEXT?

Getting beyond where we are going or where we are heading is a nontrivial task. It requires reaching into the future on the basis of what we have learned from the past. If we look at CSE as a whole, it is evident that the field has multiple personalities from multiple levels of inheritance that result in multiple meanings of where it is headed. Many examples of CSE are put forth that may only fit subsets of the definitions, themes, and example approaches provided in earlier queries. On the one hand perhaps this results in a dilution of CSE as field in its own right. For example, those who say that it is nothing more than "glorified human factors" or a subset of knowledge engineering (see McNeese, 1996b, that describes how CSE is different from these areas). Alternatively, there are those who want such a narrow definition that perhaps many current practitioners would be excluded. In the final hour we simply need to remember to ask ourselves several things that keep us on the right road to discovery as a reason for what is next:

- What is the use of CSE?
- What specific directions should CSE head towards (places where it has not been previously)?
- What inherent weaknesses require the greatest attention?
- What have we learned as an evolving field from given standpoints
- What have we learned in given domains across multiple domains?
- Is CSE science, engineering, anthropology? All or neither?
- How shall we teach and communicate with others the inherent worth, value, and impact of CSE?

Too, the question of how and how well the field addresses the universal principles—as derived from multiple domains of practice—determines the quest of future evolution. As we consider these basic questions and reflect on the queries throughout this chapter it is instructive to end with what I think are legitimate challenges for CSE to pursue to strengthen its position and to look for what is next. The following are an initial set of challenges—certainly incomplete—to begin considering and answering as a community of learners. Because the intent of this book aims toward the aviation setting as a target, many of the challenges are clothed with that context.

3.8.1 The Challenges

3.8.1.1 Philosophy/Value.

- For aviation projects, how is cognitive systems engineering different from human factors engineering—what is the value added for applying CSE to aviation?
- What is the inherent relationship of cognitive science to cognitive task analysis to cognitive modeling to cognitive systems engineering? How can these areas all be used systematically in addressing prime military aviation concerns?
- How does one address collaborative, team level issues with CSE?
- In your practice of CSE, what are the current weaknesses and limitations? How would you improve these for the next generation of CSE methods and tools?

3.8.1.2 Methodology/Tool-Use.

- What is the nature of translating the output of cognitive task analysis into design visualizations? Is this creativity? Does CSE suggest a specific process for doing this?
- What specific tools do you use as a practitioner in applying CSE to specific contexts? What computer-based tools (or other tools) would you like to see come on to the horizon?
- What do verification and validation mean in the context of CSE applied to aviation domains—for cognitive task/work analysis; for designs created from CSE process?
- What is the form and basis for the kind of modeling you do in CSE? How do you assess/validate/test models?

3.8.1.3 Aviation Application & Field of Practice.

- Is there anything unique about military aviation contexts that changes the application of cognitive system engineering principles?
- What are the crux of the issues involving specificity of context (e.g., a given military aviation domain such as the F–22) and generalizability of findings to other domains—what transfers and how do you know this?
- What are the objects of your study in aviation-based CSE? How do you begin a project? How do you deal with first-of-kind, unfielded sys-

tems design? How do you address uncertainty of data, info, knowledge, etc. with the CSE process?

- In conducting aviation-based fieldwork as part of the CSE process, what lessons learned could you share that would be informative to other practitioners?

REFERENCES

Bannon, L. J. (1992). From human factors to human actors: The role of psychology and human-computer interaction studies in system design. In J. Greenbaum & M. Kyng (Eds.), *Design at work: Cooperative design of computer systems* (pp. 25–44). Hillsdale, NJ: Erlbaum.

Barker, R. (1968). *Ecological psychology*. Stanford, CA: Stanford University Press.

Bautsch, H. S., & McNeese, M. D. (1997). Assessing the value of human performance modeling in exploring pilot-system dynamics. *Proceedings of the 41st Annual Meeting of the Human Factors and Ergonomics Society, 2,* (pp. 1032–1036). Santa Monica, CA: HFES.

Bautsch, H.S., Narayanan, S., & McNeese, M.D. (1997). Development and evaluation of a cognitive model of human performance in fighter aircraft. *Proceedings of the IEEE Systems, Man , & Cybernetics International Conference* (pp. 2109-2113). New York: IEEE.

Bransford, J., Sherwood, R., Vye, N., & Rieser, J. (1986). Teaching and problem solving: Research foundations. *American Psychologist*, *41*, 1078–1089.

Brown, C. E., Selvaraj, J. A., Whitaker, R. D., & McNeese, M. D. (1995). *The development of TRACE: An integrative bargaining paradigm for investigating multidisciplinary design tradeoffs* (AL/CF–TR–1995–0073). Wright-Patterson Air Force Base, OH: Armstrong Laboratory.

Buchanan, B. G., & Shortliffe, E. H. (1984). *Rule-based expert systems: The MYCIN experiments of the Stanford heuristic programming project*. Reading, MA: Addison-Wesley.

Card, S., Moran, T., & Newell, A. (1983). *The psychology of human-computer interaction.* Hillsdale, NJ: Erlbaum.

Citera, M., McNeese, M. D., Brown, C.E., Selvaraj, J. S., Zaff, B.S., & Whitaker, R. (1995). Fitting information systems to collaborating design teams. *Journal of the American Society of Information Science, 46* (7), 551–559.

Citera, M., Selveraj, J. A., McNeese, M. D., Brown, C. E., & Zaff, B. S. (1995). *Automate: A research paradigm to study collaboration in multidisciplinary design teams* (AL/CF–TR–1995–0087). Wright-Patterson Air Force Base, OH: Armstrong Laboratory.

Cooke, N. J. (1994). Varieties of knowledge elicitation techniques. I*nternational Journal of Human-Computer Studies, 41*, 801–849.

Darkow, D. J., Marshak, W. P., Woodworth, G., & McNeese, M. D. (1998). *Low-level cognitive modeling of aircrew function using Soar artificial intelligence architecture* (AFRL–TR–98). Wright-Patterson Air Force Base, OH: Air Force Research Laboratory.

Ericsson, K. A., & Simon, H. A. (1984). *Protocol analysis: Verbal reports as data.* Cambridge, MA: MIT Press.

Flach, J. (1990). The ecology of human-machine systems I: Introduction. *Ecological Psychology, 2* (3), 191–205.

Fraser, N. M., Hipel, K. W., Kilgore, D. M., McNeese, M. D., & Snyder, D. E. (1989). An architecture for integrating expert systems. *Decision Support Systems, 5*, 263–276.

Gibson, J. J. (1979). *The ecological approach to visual perception*. Boston, MA: Houghton Mifflin.

Gill, R. T., & Gordon, S. E. (1997). Cognitive task analysis. In C. E. Zsambok & G. Klein (Eds.), *Naturalistic decision making*. Mahway, NJ: Erlbaum.

Greenbaum, J., & Kyng, M. (Eds.). (1991). *Design at work: Cooperative design of computer systems*. Hillsdale, NJ: Erlbaum.

Hammond, K. R., McClelland, G. H., & Mumpower, J. (1980). *Human judgment and decision making: Theories, methods, and procedures*. New York: Praeger.

Hendrick, H. W. (1986). Macroergonomics: A conceptual model for integrating human factors with organizational design. In O. Brown & H.W. Hendrick (Eds.), *Human factors in organization design and management II* (pp. 467–477). Amsterdam: North Holland.

Hinsz, V. B., Tindale, R. S., & Vollrath, D. A. (1997). The emerging conceptualization of groups as information processors. *Psychological Bulletin*, *121*, 43–64.

Hollnagel, E., & Woods, D.D. (1983). Cognitive systems engineering: New wine in new bottles. *International Journal of Man-Machine Studies*, *18*, 583–600.

Hollnagel, E. (1998). Comments on "Conception of the cognitive engineering design problem" by John Dowell and John Long. *Ergonomics, 41* (2), 160–162.

Hutchins, E. (1995). *Cognition in the wild*. Cambridge, MA: MIT Press.

Klein, G. A. (1990). Knowledge engineering: Beyond expert systems. *Information and Decision Technologies, 16* (1), 27–41. Holland: Elsevier.

Klein, G. (1993). A recognition-primed decision (RPD) model of rapid decision making. In G. A. Klein, J. Orasanu, R. Calderwood, & C. E. Zsambok (Eds.), *Decision making in action: Models and methods* (pp. 138–147). Norwood, NJ: Ablex.

Klein, G., Calderwood, R., & Clinton-Cirocco, A. (1986). Rapid decision making on the fireground. *Proceedings of the Human Factors Society 30th Annual Meeting* (pp. 576–580). Santa Monica, CA: HFS.

Klein, G., Calderwood, R., & MacGregor, D. (1989). Critical decision method for eliciting knowledge. *IEEE Transactions on Systems, Man, and Cybernetics*, *19* (3), 462–472.

Klein, G., & Crandall, B. W. (1995). The role of mental simulation in problem solving and decision making. In P. Hancock, J. Flach, J. Caird, & K. Vincente (Eds.), *Local applications of the ecological approach to human-machine systems*. Hillsdale, NJ: Erlbaum.

Klein, G. A., Orasanu, J., Calderwood, R., & Zsambok, C. E. (Eds.). (1993). *Decision making in action: Models and methods*. Norwood, NJ: Ablex.

Klein, G. A., Woods, D. D., & Orasanu, J. (1993). Conclusions. In G. A. Klein, J. Orasanu, R. Calderwood, & C. E. Zsambok (Eds.), *Decision making in action: Models and methods*. Norwood, NJ: Ablex.

Lenat, D. B., & Guha, R. V. (1990). *Building large database systems*. Reading, MA: Addison-Wesley.

Lindsay, P. H., & Norman, D. A. (1977). *Human information processing: An introduction to psychology*. New York: Academic Press.

McNeese, M. D. (1977). *How people space themselves out in the JFK union*. Unpublished senior honors thesis; University of Dayton, Dayton, OH.

McNeese, M. D. (1986). Humane intelligence: A human factors perspective for developing intelligent cockpits. *IEEE Aerospace and Electronic Systems, 1* (9), 6–12.

McNeese, M. D. (1992). *Analogical transfer in situated cooperative learning*. Unpublished doctoral dissertation, Vanderbilt University, Nashville, TN.

McNeese, M. D. (1996a). An ecological perspective applied to multi-operator systems. In O. Brown & H. L. Hendrick (Eds.), *Human factors in organizational design and management*—VI (pp. 365–370). The Netherlands: Elsevier.

McNeese, M. D. (1996b). A comprehensive cognitive engineering process for human-information systems. *Proceedings of the 3rd Annual Symposium on Human Interaction with Complex Systems, 1,* (pp. 260–267). Los Alamitos, CA: IEEE Computer Science Society.

McNeese, M. D. (2000). Socio-cognitive factors in the acquisition and transfer of knowledge. *Cognition, Technology, and Work, 2,* 164–177.

McNeese, M. D., Bautsch, H. S., & Narayanan S. (1999). A framework for cognitive field research. *International Journal of Cognitive Ergonomics, 3* (4), 307–332.

McNeese, M. D., & Brown, C. E. (1986). *Large group displays and team performance: An evaluation and projection of guidelines, research, and technologies* (AAMRL–TR–86–035). Wright-Patterson Air Force Base, OH: Armstrong Aerospace Medical Research Laboratory.

McNeese, M. D., & Katz, L. (1987). Legibility evaluation of a large screen display system under medium ambient illumination. *Journal Proceedings of The Society For Information Display, 28* (1), 59–65.

McNeese, M. D., & Perusich, K. (2000). Constructing a battlespace to understand macroergonomic factors in team situational awareness. *Proceedings of the Industrial Ergonomics Association/Human Factors and Ergonomics Society (IEA/HFES) 2000 Congress* (pp. 2-618–2-621), Santa Monica, CA: HFES.

McNeese, M. D., Perusich, K., & Rentsch, J. (1999). What is command and control coming to? Examining socio-cognitive mediators that expand the common ground of teamwork. *Proceedings of the 43rd Annual Meeting of the Human Factors and Ergonomic Society* (pp. 209–212). Santa Monica, CA: HFES.

McNeese, M. D., Perusich, K., & Rentsch, J. R. (2000). Advancing socio-technical systems design via the living laboratory. *Proceedings of the Industrial Ergonomics Association/Human Factors and Ergonomics Society (IEA/HFES) 2000 Congress* (pp. 2-610–2-613), Santa Monica, CA: HFES.

McNeese, M. D., & Rentsch, J. R. (2001). Identifying the social and cognitive requirements of teamwork using collaborating task analysis. In M. D. McNeese, E. Salas, & M. Endsley (Eds), *New trends in activities: Understanding system dynamics in complex environments.* Santa Monica, CA: HFES.

McNeese, M. D., Rentsch, J. R., & Perusich, K. (2000). Modeling, measuring, and mediating teamwork: The use of fuzzy cognitive maps and team member schema similarity to enhance BMC³I decision making. *IEEE International Conference on Systems, Man, and Cybernetics* (pp. 1081–1086). New York: Institute of Electrical and Electronic Engineers.

McNeese, M. D., Warren, R., & Woodson, B. (1985). Cockpit automation technology—A further look. *Proceedings of the 29th Annual Meeting of the Human Factors Society* (pp. 884–888), Santa Monica, CA: HFS.

McNeese, M.D., Zaff, B.S., Citera, M., Brown, C. E., & Whitaker, R. (1995). AKADAM: Eliciting user knowledge to support participatory ergonomics. *The International Journal of Industrial Ergonomics, 15* (5), 345–363.

McNeese, M. D., Zaff, B.S., Peio, K. J., Snyder, D E, Duncan, J. C., & McFarren, M R. (1990). *An advanced knowledge and design acquisition methodology: Application for the pilots associate* (AAMRL–TR–90–060). Wright-Patterson Air Force Base, OH: Armstrong Aerospace Medical Research Laboratory.

Norman, D. A. (1981). Categorization of action slips. *Psychological Review, 88*, 1–15.

Norman, D. A. (1986). Cognitive engineering. In D. A. Norman & S. W. Draper (Eds.), *User Centered System Design.* Hillsdale, NJ: Erlbaum.

Norman, D. A. (1988). *The psychology of everyday things.* New York: Basic Books.

Norman, D. A. (1993). *Things that make us smart*. New York: Addison-Wiley.

Norman, D. A., & Draper, S. W. (Eds.). (1988). *User-centered system design: New perspectives on human-computer interaction*. Hillsdale, NJ: Erlbaum.

Nosek, J., & McNeese, M. D. (1997). Augmenting group sensemaking in ill-defined, emerging situations. *Information Technology and People, 10* (3), 241–252.

Perusich, K., & McNeese, M. D. (1997). Using fuzzy cognitive maps for data abstraction. *Proceedings of the Annual Meeting of the North America Information Processing Society* (pp. 5–9). New York: IEEE Press.

Perusich, K., & McNeese, M. D. (1998a). *Understanding and modeling information dominance in battle management: Applications of fuzzy cognitive maps.* (AFRL–TR–98–0040). Wright-Patterson Air Force Base, OH: Air Force Research Laboratory.

Perusich, K., & McNeese, M. D. (1998b October). *Modeling collaborative activities in complex systems.* Poster presentation the 41st Annual Meeting of the Human Factors and Ergonomics Society, Chicago, IL.

Perusich, K., McNeese, M. D., & Rentsch, J. (1999). Qualitative modeling of complex systems for cognitive engineering. In A.F. Sisti (Ed.), *Proceedings of the SPIE, (Ed.) 3996*, 240–249.

Pew, R. W., Baron, S., Feehrer, C. E., & Miller, D. C. (1977). *Critical review and analysis of performance models applicable to man-machine system evaluation*. Cambridge, MA: Bolt, Beranek, & Newman.

Potter, S. S., Roth, E. M., & Woods, D. D. (2000). *Computer-aided cognitive systems engineering* (Unpublished Technical Report). Pittsburgh, PA: Logica Carnegie Group.

Potter, S. S., Roth, E. M., Woods, D. D., & Elm, W. C. (1998). *Toward the development of a computer-aided cognitive engineering tool to facilitate the development of advanced decision support systems for information warfare domains* (AFRL–HE–WP–TR–1998–0004). Wright-Patterson Air Force Base, OH: Air Force Research Laboratory.

References

Rasmussen, J. (1980). The human as a system component. In H. T. Smith & T. R. G. Green (Eds.), *Human interaction with computers* (pp. 67–96). London: Academic Press.

Rasmussen, J. (1983). Skills, rules, and knowledge; signals, signs, and symbols, and other distinctions in human performance models. *IEEE Systems, Man, & Cybernetics, 13,* 257–266.

Rasmussen, J. (1986). *Information processing and human-machine interaction: An approach to cognitive engineering*. New York: North-Holland.

Rasmussen, J. (1988). Cognitive engineering: A new profession. In L. P. Goodstein, H. B. Anderson, & S. E. Olsen (Eds.), *Tasks, errors, and mental models.* Philadelphia: Taylor & Francis.

Rasmussen, J., Pejtersen, A. M., & Goodstein, L. P. (1994). *Cognitive systems engineering*. New York: Wilcy.

Reason, J. (1990). *Human error*. Cambridge, England: Cambridge University Press.

Reason, J., & Mycielska, K. (1983). *Absent minded? The psychology of mental lapses and everyday error.* Englewood Cliffs, NJ: Prentice Hall.

Rentsch, J. R., McNeese, M. D., Pape, L. J., Burnett, D. D., Menard, D. M., & Anesgart, M. (1998). *Testing the effects of team processes on team member schema similarity and task performance: Examination of the team member schema similarity model* (AFRL–TR–98–0070). Wright-Patterson Air Force Base, OH: Air Force Research Laboratory.

Resnick, L., Levine, J., & Teasley, S. D. (1991). *Perspectives on socially shared cognition*. Washington, DC: American Psychological Association.

Rogoff, B., & Lave, J. (Eds.). (1984). *Everyday cognition: Its development in social context*. Cambridge, MA: MIT Press.

Rouse, W. B., & Boff, K. R. (Eds.). (1987). *System design: Behavioral perspectives on designers, tools, and organizations*. New York: Elsevier.

Rouse, W. B., & Morris, N. M. (1986). On looking into the black box: Prospects and limits on the search for mental models. *Psychological Bulletin, 100* (3), 349–363.

Sanderson, P. M., James, J., & Seidler, K. S. (1989). SHAPA: An interactive software environment for protocol analysis. *Ergonomics, 32*, 1271–1302.

Sanderson , P. M., Scott, J. P. P., Johnston, P., Mainzer, J., Watanabe, L. M., & James, J. M. (1994). MacSHAPA and the enterprise of exploratory sequential data analysis (ESDA). *International Journal of Human-Computer Studies*, *41*, 633–668.

Sanderson, P. M., McNeese, M. D., & Zaff, B. S. (1994). Handling complex real world problems with two cognitive engineering tools: COGENT and MacSHAPA. *Behavior Research Methods, Instruments, & Computers*, *17* (2), 117–124.

Sheridan, T. B. (1992). *Telerobotics, automation, and human supervisory control.* Cambridge, MA: MIT Press.

Snyder, D E, Wellens, A R, Brown, C.E., & McNeese, M. D. (1989). Three paradigms for the study of multi-person and human-machine interaction. *Proceedings of the IEEE Systems, Man, & Cybernetics Conference* (pp. 480–481), Boston, MA: IEEE.

Suchman, L. (1987). *Plans and situated actions: The problem of human-machine communication.* New York: Cambridge University Press.

Thordsen, M. L., & Klein, G. (1989). Cognitive processes of the team mind. *IEEE Conference on Systems, Man, and Cybernetics* (pp. 46–49). New York: IEEE.

Vicente, K. (1999). *Cognitive work analysis*. Mahway, NJ: Erlbaum.

Vicente, K., & Rasmussen, J. (1992). Ecological interface design: Theoretical foundations. *IEEE Systems, Man, and Cybernetics*, *22* (4), 589–607.

Wegner, D. M. (1987). Transactive memory: A contemporary analysis of the group mind. In B. Mullen & G. R. Goethals (Eds.), *Theories of group behavior* (pp. 185–208). New York: Springer-Verlag.

Wellens, A. R. (1993). Group situational awareness and distributed decision making: From military to civilian applications. In N. J. Castillian (Ed.), *Individual and group decision making: Current issues.* Hillsdale, NJ: Erlbaum.

Wellens, A. R., & McNeese, M. D. (1987). A research agenda for the social psychology of intelligent machines. *Proceedings of the IEEE National Aerospace and Electronics Conference* (NAECON), Vol. 3, (pp. 944–949). Dayton, OH: IEEE Aerospace and Electronics Systems Society.

Whitaker, R. (1994). GDSS' formative fundaments: An interpretive analysis. *Computer Supported Cooperative Work: An International Journal, 2*, 239–260.

Whitaker, R. D., Longinow, N. E., & McNeese, M. D. (1995). *Group interfaces: A profile and a prototype* (Tech. Rep. AL/CR–TR–1995–0069). Wright-Patterson AFB, OH: Armstrong Laboratory.

Whitaker, R. D., Selvaraj, J. A., Brown, C. E., & McNeese, M. D. (1995). *Collaborative design technology: Tools and techniques for improving collaborative design* (AL/CF–TR–1995–0086). Wright-Patterson Air Force Base, OH: Armstrong Laboratory.

Wiener, E. (1989). *Human factors of advanced technology ("glass cockpit") transport aircraft*. (Tech. Rep. 117528). Moffett Field, CA: NASA Ames Research Center.

Wilson, D., McNeese, M. D., & Brown, C.E. (1987). Team performance of a dynamic resource allocation task: Comparison of shared versus isolated work setting. *Proceedings of the 31st Annual Meeting of the Human Factors Society*, Vol. 2, (pp. 1345–1349). Santa Monica, CA: HFS.

Woods, D. D. (1998). Designs are hypotheses about how artifacts shape cognition and collaboration. *Ergonomics, 41* (2), 168–173.

Woods, D. D., Johannesen, L. J., Cooke, R. I., & Sarter, N. B. (1994). *Behind human error: Cognitive systems, human error, and hindsight.* Wright-Patterson Air Force Base, OH: Crew Systems Ergonomics Information Analysis Center.

Woods, D. D., & Roth, E. M. (1988). Cognitive engineering: Human problem solving with tools. *Human Factors, 30*, 415–430.

Xiao, Y. (1994). *Interacting with complex work environments: A field study and a planning model.* Unpublished doctoral dissortation, University of Toronto, Toronto, Canada.

Young, M. F., & McNeese, M. D. (1995). A situated cognition approach to problem solving. In P. Hancock, J. Flach, J. Caird, & K. Vincente (Eds.), *Local applications of the ecological approach to human-machine systems*. Hillsdale, NJ: Erlbaum.

Zaff, B. S., McNeese, M. D., & Snyder, D. E. (1993). Capturing multiple perspectives: A user-centered approach to knowledge acquisition. *Knowledge Acquisition, 5* (1), 79–116.

Zaff, B. S., McNeese, M. D., Snyder, D. E., & Lizza, C. S. (1991). An integrated knowledge representation: Developing a large scale knowledge base for the pilots associate. *Proceedings of the Associate Technology Symposium* (pp. 244–255). Fairfax, VA: George Mason University Press.

Zsambok, C. E., & Klein, G. (1997). *Naturalistic decision making*. Mahway, NJ: Erlbaum.

4 Balancing Practice-Centered Research and Design

DAVID WOODS
The Ohio State University

KLAUS CHRISTOFFERSEN
The Ohio State University

ABSTRACT

Many are interested in Research and Development (R&D) at the intersection of people, technology and work. R&D is a world divided and hobbled. Innovation is tantalizing yet elusive. In a rush, we achieve only a cumbersome process of trial and error (publicizing the extent of design errors and failures would be bad for investment). The standard metaphor and organizational construct of the pipeline has failed given the possibilities for change and the predilection for new technology to demand connections across disciplinary boundaries. R&D in this area is a world too often without effective interconnections and cross-stimulation. This piece provides an alternative model at two levels. First, it presents complementarity as a strategy for practice-centered research and design. The strategy is foundational for the intent behind the label *cognitive systems engineering* (and related labels like *distributed cognition, naturalistic decision making*, and on and on) as an alternative to traditional disciplinary approaches. Second, it replaces the shopworn cliché of an R&D pipeline (a metaphor that may never have had substance) with synchronization of multiple parallel cycles of learning and development that operate at different time scales. Interlocking these cycles is a difficult challenge—a challenge in producing an organizational framework and supporting mechanisms to create and extend distributed innovation.

4.1 BROKEN PIPES

Many are interested in Research and Development (R&D) at the intersection of people, technology, and work. Developers and technologists make claims about how a prospective new capability or new system development project will impact on performance in one or many settings. Sponsors are caught up in the sweeping dreams permitted by technology unfettered from harsh contexts of use, yet they fear software development projects that fail to provide useful tools or that create unanticipated negative effects. Practitioners and observers of practitioners at work note repeated forms of clumsiness in the technology deployed and unanticipated side effects of change. Researchers, blinded by the glare of disciplinary labels, drastically reduce situations to fit into a lab one variable at a time yet claim priority in the search for generic regularities. Human factors practitioners and usability engineers are called in too late to repair the connection between systems and use. Research results seem irrelevant to design. Design seems local and unique.

R&D at the intersection of people, technology, and work is a world divided and hobbled. Innovation is tantalizing yet elusive. In the rush, we achieve only a cumbersome process of trial and error (publicizing the extent of design errors and failures would be bad for investment). The standard metaphor and organizational construct of the pipeline has failed given the possibilities for change and the predilection for new technology to demand connections across disciplinary boundaries. R&D in this area is a world too often without effective interconnections and cross-stimulation.

We provide an alternative model at two levels. The first attribute is *complementarity* as a strategy for practice-centered research and design. This is the foundational strategy behind the label *cognitive systems engineering* (and related labels like *distributed cognition* and *naturalistic decision making*) that makes it a substantive alternative to traditional disciplinary approaches. In other words, all the new labels about the syntheses required to study and shape the intersection of people, technology, and work are only superficial exercises in career enhancement unless they provide substance to complementarity.

Second, the model here replaces the shopworn cliché of an R&D pipeline (a metaphor that may never have had substance) with synchronization of multiple, parallel cycles of learning and development that operate at different time scales. Interlocking these cycles is a difficult challenge—a challenge in producing organizational framework and supporting mechanisms to create and extend *distributed innovation*.

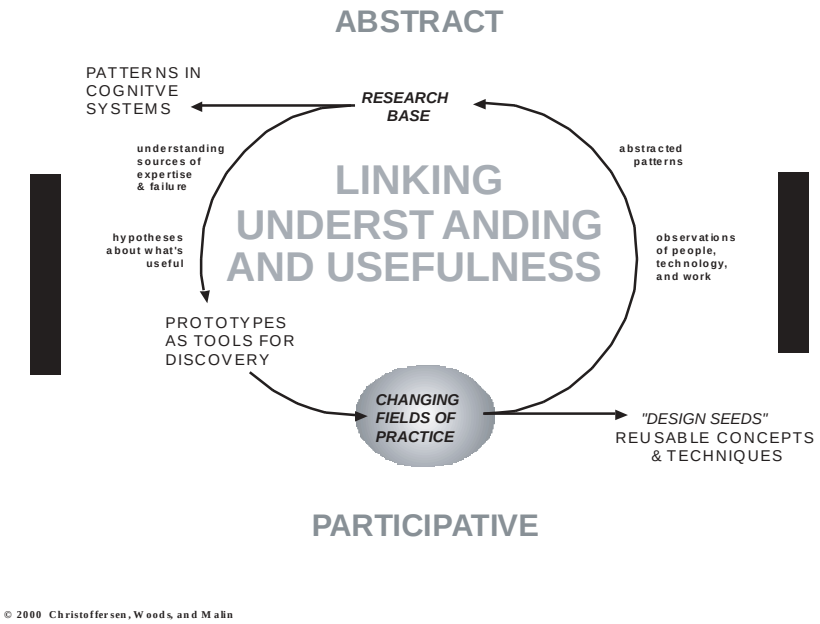


Figure 4.1: A practice-centered approach to research. Complementarity defines cognitive systems engineering in the innovation process.

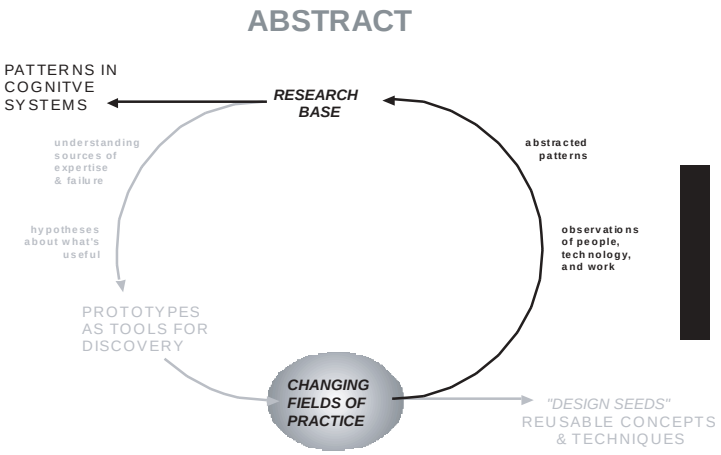
4.2 COMPLEMENTARITY AS A STRATEGY TO BALANCE RESEARCH AND DESIGN

Two coordinated strands define complementarity (see Figure 4.1). In one strand (see Figure 4.2), inquiry is directed at capturing phenomena, abstracting patterns, and discovering the forces that produce those phenomena despite the surface variability of different technology and different settings. In this sense effective research develops a book of "patterns" as a generic but relevant research base.

But the challenge of stimulating innovation goes further. A second strand of processes are needed that link this tentative understanding to the process of discovering what would be useful (see Figure 4.3). Success occurs when "reusable" (that is, tangible but relevant to multiple settings) design concepts and techniques are created to "seed" the systems development cycle.

Discovery of what would be useful occurs in the research cycle because

- Development also functions as opportunities to learn. Artifacts are not just objects; they are hypotheses about the interplay of people, technology, and work. In this cycle prototypes function as tools for discov-



© 2000 Christoffersen, Woods, and Malin

Figure 4.2: Discovering patterns in cognition at work. Observing, abstracting, explaining phenomena at the intersection of people, technology, and work.

ery to probe the interaction of people, technology, and work and to test the hypothesized, envisioned impact of technological change.

- The limited resource horizon of all development projects always places limits on learning and exploring. Inevitably in development projects, prototypes function only as partially refined final products, and the design mindset narrows in on critical paths toward and impasses that could block the realization of a tangible object in a particular setting. Seeding future development cycles with new reusable concepts about what would be useful requires a longer-term focus.

In the end, innovation is stimulated through both creation of possible futures and reflection about the effects of those changes while the commitment to any particular object is relaxed and the limited horizon of development cycles is stretched. The combination creates a complementary cycle of learning and development (see Figure 4.1). Advancing our understanding abstracts patterns and phenomena from observations of the interplay of people, technology, and work and develops explanations for the appearance of these patterns across different fields of practice. This cycle seeks to discover performance-related issues within each given setting and to develop hypotheses about what may be useful

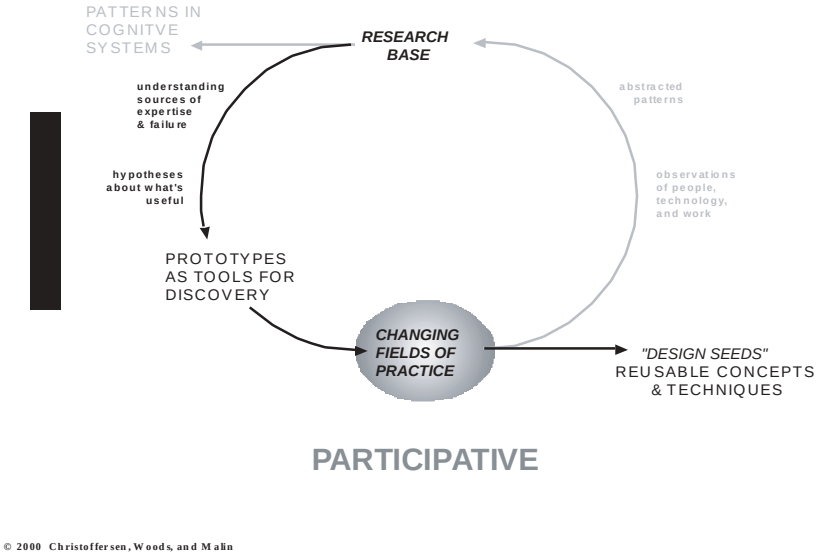


Figure 4.3: Leveraging research to generate useful design concepts. Generating reusable concepts about what would be useful to seed development.

in response to these issues. Aiding concepts are embodied in prototypes as part of a continuing learning and discovery process. Over time, the result is a generically defined set of concepts and techniques that can seed development in multiple specialized areas where the relevant performance issues play out.

An effective balance generates two types of advances, each as tentative syntheses of what we think we know about the interplay of people, technology, and work (Figure 4.1, A practice-centered approach to research). The research base is seen as patterns abstracted across different unique settings, patterns that are in need of explanation and concepts that could explain these observations. As Hutchins (1999) put it, "There are powerful regularities to be described at a level of analysis that transcends the details of the specific domain. It is not possible to discover these regularities without understanding the details of the domain, but the regularities are not about the domain specific details, they are about the nature of human cognition in human activity."

The second product of an effective balance would be the ability to capture and share design "seeds"—concepts and techniques about what would be useful to advance cognition and collaboration at work. These are seeds in the sense that they represent concepts that are sensitive to constraints that arise in multiple settings and they can stimulate development across different specific settings. "If we are to enhance the performance of operational systems, we need

conceptual looking glasses that enable us to see past the unending variety of technology and particular domains" (Woods & Sarter, 1993). To achieve this complementarity, usefulness (i.e., criteria that new systems enhance performance in context) becomes a criteria for research. How does the concept effectively seed and leverage development in more than a specific case?

In coordinating these processes, four values guide R&D activities. Fields of practice are the primary focus. *Authentic* samples of what it means to practice in that field of activity and how the organizational dynamics pressure or support practice stimulate the process of observation. However, observers will quickly become lost in the detail of particular settings at particular points in time with particular technological objects unless they can compare and contrast settings over time to *abstract* patterns and produce candidate explanations for the basic patterns. The third component is generative—in studying the interaction of people, technology, and work across fields of practice we must *generate* or discover new ideas, including explanations for the phenomena and patterns observed, but more critically, new hypotheses about what would be useful to probe the field of practice, test our tentative understanding, and to seed upcoming development cycles. In the final analysis the activity is *participative* as we work with practitioners in these fields of activities to understand how they adapt to the pressures and demands of the field of activity.

The two half cycles (Figures 4.2 and 4.3) are interdependent, not separate. The point of the processes of observation, abstraction, and explanation is to find the essential factors under the surface variability. In other words, the test of understanding is the ability to anticipate the impacts of technological change. The ultimate risk for the researchers is to acknowledge that they are part of the process under study. The researcher participates in the struggle of envisioning with other stakeholders. Researchers also must acknowledge their role as designers—the development of tools that make us smart or dumb. The ultimate test for the designers is to risk abstraction and acknowledge their prototypes as hypotheses at empirical jeopardy.

Thus, in a practice-centered process we face challenges related to the four basic values:

- Transcending *limits to authenticity* to capture how the strategies and behavior of people are adapted to the constraints and demands of fields of practice.
- Meeting the *challenge of abstraction* to find and explain patterns behind the surface variability.
- *Sparking inventiveness* to discover new ways to use technological possibilities to enhance human performance, to identify leverage points, and to minimize unanticipated side effects.
- *Creating future possibilities* as participants with other stakeholders and problem holders in that field of practice.

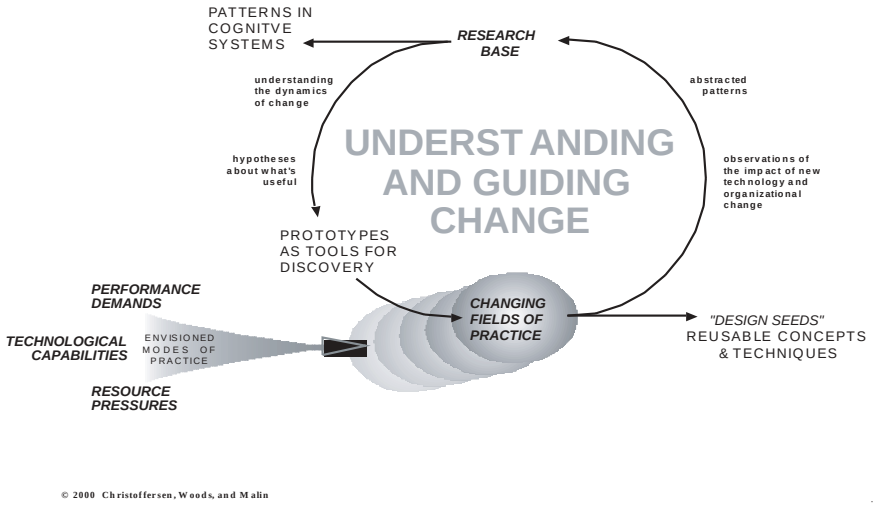


Figure 4.4: The envisioned world problem: A moving target for design.

These are generic, but relevant; finding in the particular the existence and expression of universal patterns—these are not contradictions or conflicts but creative tensions at the root of complementarity, harnessed for innovation.

4.3 MOVING TARGET

Achieving the complementarity captured in Figure 4.1 is in fact quite difficult for a number of reasons. One is captured in Figures 4.4 and 4.5, which creates a moving target for development. Fields of practice are not static; rather demands, pressures, and resources are changing. New possibilities are envisioned and advocates push their particular vision, but the introduction of new systems transforms the nature of practice through new roles, new judgments, new forms of coordination, and new paths toward and forms of breakdown.

We usually see new computerization as a solution to performance problems or limits. In others words, advancing the baseline of technology and focused development projects in specific areas require envisioning future operations. However, envisioned operation concepts have two basic properties:

- *Plurality*—there are multiple versions of how the proposed changes will affect the character of the field of practice in the future; and

- *Underspecification*—each envisioned concept is vague on many aspects of what it would mean to function in that field of practice in the future.

New technology is a source of change as performance demands and resource pressures change. New technology becomes wrapped up in organizational change as well. The question then is—can design anticipate the full range of effects of the change? Usually, technology change produces unintended and sometimes negative side-effects in addition to new capabilities. Thus, we are part of a dynamic process that we also wish to understand and influence—a dynamic process of technology change generating a new set of capabilities and complexities, leading to adaptations by stakeholders, producing a changing mix of success and failure.

In addition to plurality and underspecification, envisioned modes of operation are a *prediction* about the effects of change on people, technology, and work. As predictions, envisioned concepts can have two other properties:

- *Ungrounded*—envisioned concepts can easily be disconnected or even contradicted from the research base on the actual consequences of the changes on people, technology and work.
- *Overconfident*—advocates are miscalibrated and overconfident that, if the systems envisioned can be realized, the predicted consequences and only the predicted consequences will occur.

The envisioned world problem demands that we develop means to ground predictions on relevant empirical results abstracted from observations in context. Understanding the dynamic process of change and adaptation will lead to better control of the process—essentially an innovation process at the intersection of people, technology, and work. Armed with knowledge about the dynamics of change and adaptation, we can address potential side effects at a time when intervention is less difficult and less expensive (because the field of practice is already in a period of change and systems development is in the process of creating tangible objects).

4.4 SYNCHRONIZING PARALLEL, INTERLOCKING CYCLES

Figure 4.5 also broadens the context for practice-centered research and design to indicate the relationship to the systems development process.

In the added development cycle, teams develop systems to address problems that arise in a specialized field of practice (for example, topics of recent interest for the Air Force are distributed replanning and unmanned aerial vehicles). In this cycle, a resource horizon drives the scope of development, and there are constraints at multiple levels of realization which need to be balanced

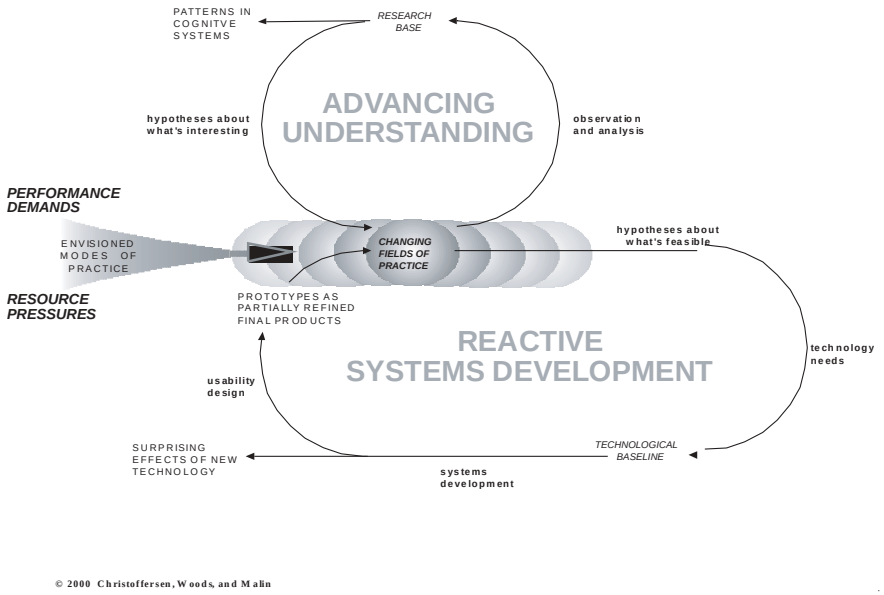


Figure 4.5: Mis-synchronized cycles.

in a successful product. As a result, (1) there is a process of matching requirements to technological possibilities, (2) prototypes function as partially refined final products, (3) usability testing refines the potential product to fit the field of practice, and (4) the end result is tangible systems that can be introduced into a specific field of practice.

Figure 4.6 adds the third cycle of learning and development to complete the picture. In the bottom cycle, research technologists advance the technological baseline either by expanding the capabilities of autonomous technology or by increasing the availability of technological capabilities (reducing cost, expanding access to technological capabilities).

The graphic illustrates the kind of balance and cross-stimulation across these traditional R&D activities that can enhance innovation in the search for useful systems. This coupling creates an organization that is sensitive and responsive to the needs and opportunities for improving performance in complex socio-technical systems.

The challenge is interlocking and *synchronizing* these multiple cycles. When these cycles are balanced, complementary processes, they drive the innovation process by which technological possibilities are harnessed to advance performance and reduce risks of development in specific fields of practice. Achieving

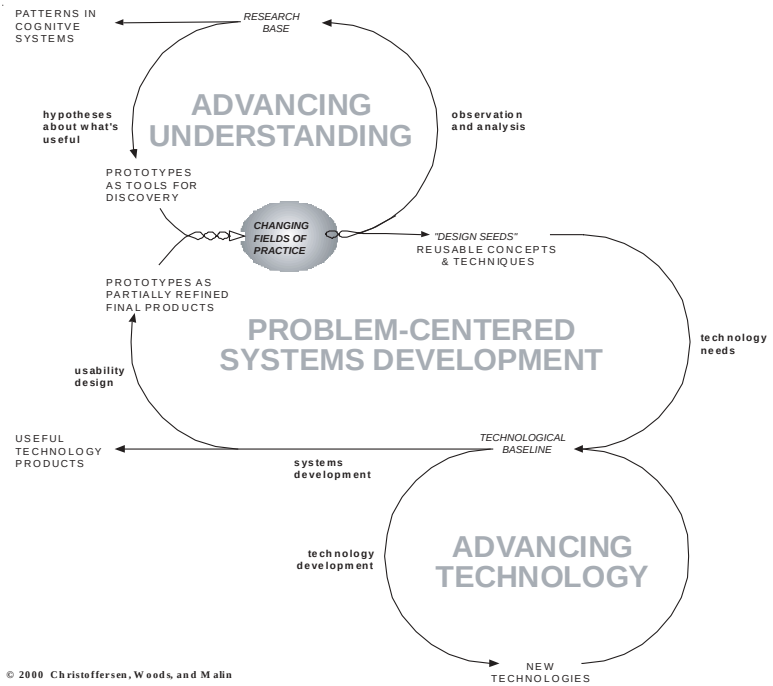


Figure 4.6: The engine of innovation: Balanced cycles. Synchronization of cycles as an alternate to the pipeline model of R&D.

balance requires *effort-after-coordination* as people working in each area step outside of their own activities to see the relationship to the other cycles.

4.5 IMBALANCE

But each of the cycles happens at different time scales, which makes synchronization difficult. Each of these cycles has its own criteria for success, which pushes personnel deeper into their own role and subgoals. Without organizational support and investment, the cycles spin apart, out of balance, and fail to mutually reinforce each other. The result is journal papers that provide no insight as to the real phenomena, lack of innovative uses of new possibilities, new technology that users see as clumsy in context, and the same problems ("classics") and questions re-appear with limited or no progress (Figure 4.7).

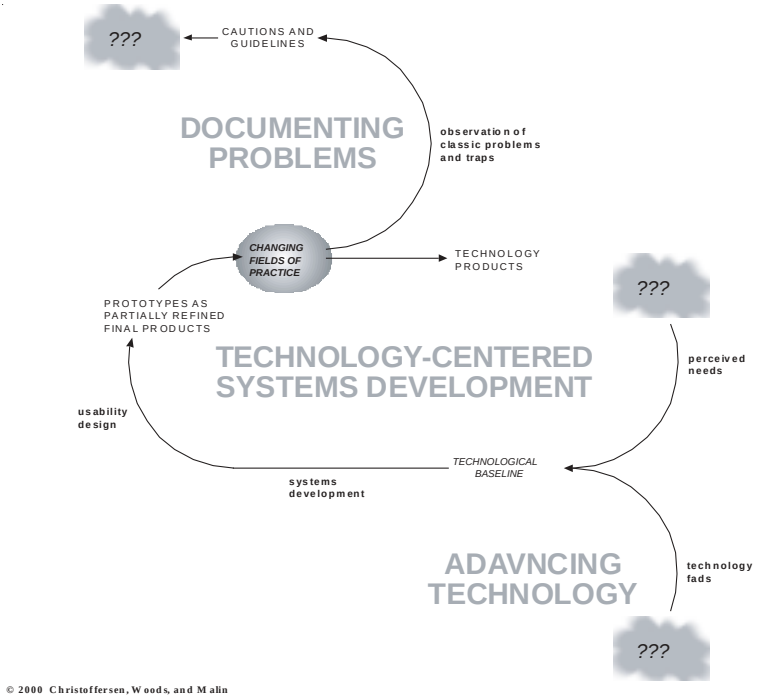


Figure 4.7: The engine of frustration: A recipe for design by trial and error. How the balance breaks down.

4.6 POINTS OF BALANCE

The key to balancing these cycles lies in their connections with one another. As interlocking cycles, each one stimulates and guides the others in a process centered around the dynamic processes evident in changing fields of practice. Coupling the uppermost and middle cycles ensures that research is relevant and representative, while providing a base of analytical perspectives and design concepts that is portable across domains and that can be leveraged in systems development. Research is integral and important when it can proactively suggest promising design directions, rather than simply critiquing poor designs retrospectively, when opportunities for modification are limited. The introduction of a prototype into a given setting functions as a way to test and refine it as a product (the middle cycle), but also represents a natural experiment (the top cycle but at a different time constant) on the behavior of the socio-technical system and the effectiveness of the aiding concepts embodied by the prototype (artifacts are hypotheses about the interplay of people, technology, and work). Recognizing and maintaining these separate but parallel statuses of

prototypes maximize the organization's informational return on investment during development. When these cycles aren't closely coupled, the result is a design process that progresses primarily by trial and error, and a human factors organization that is perennially behind the curve.

When in balance, there are means to relate artifacts (i.e., what is technologically possible) to cognitive systems (i.e., what would be useful) and to the limited resource horizons of real development. If the technology development process is disconnected from systems development, and from the research base on "artifacts, their uses and effects," the result can be a very low "hit rate" of useful new systems. Rather than just measure success in terms of autonomous machine capabilities, technology developers must look empirically at what will support effective performance within the larger socio-technical system.

The challenge is interlocking and synchronizing these multiple cycles. When these multiple cycles are balanced, complementary processes, they drive the innovation process by which technological possibilities are harnessed to advance performance and reduce risks of development in specific fields of practice.

4.7 DIFFERENT ROLES FOR HUMAN-FACTORS-RELATED WORK

Work related to themes on human factors can go on in each of the three cycles. Research that advances technology can be about technology for interacting with people, for example, advancing the capability of augmented reality or natural language technology. Research that advances our understanding of the interaction of people, technology, and work should generate concepts and techniques that could be used in different development projects; for example, patterns in data overload or supervisory control of automation lead to design concepts and techniques to make automated systems team players. Usability engineering plays a role in systems development, early by using seed concepts from the research base to identify leverage points, and later through usability design and testing.

4.8 COMPLEMENTARITY AND SYNCHRONIZATION

The graphics in this series do not distinguish research and application. Research does not flow down to application. Instead, we need to situate activities of practice-centered research and design in the larger process of systems development for specialized target areas and relative to the general expansion in technological possibilities.

Researchers are connected to the systems development process by observing and abstracting patterns about the interaction of people, technology, and work. These researchers must be able to contribute concepts and techniques about what would be useful in the initial stages of a development cycle, but

they are relaxed from the limited resource and time horizons that pressure development of real working systems.

Those working to advance technology for human interaction are connected to the systems development process by using or participating in studies of the actual effects of technology change. Rather than just measure success in terms of autonomous machine capabilities, they can look to the research base for empirically based patterns and models about how technology developments support effective collaboration with human practitioners.

Effective innovation in system development depends on having technological advances to draw on and on having concepts about what may be useful to support human performance available early in a development cycle to be able to identify leverage points and to anticipate side effects of change.

In effect, the balancing act needs mechanisms to support *distributed innovation.* This is an example of the area of human-machine systems called computer-supported collaborative work or CSCW. Usually this work is directed at practitioners, designers, or managers. Here the need is to use principles for collaborative work and the technology infrastructure for connectivity to support distributed innovation. Doing this, as building any kind of collaboration, requires energy and investment in coordinated activities across the multiple parallel cycles.

4.9 ACKNOWLEDGMENTS AND RECOMMENDED READING

The perspective developed here is derived from and continues a series of papers on design and productions on the role of cognitive systems engineering in research and design. In particular, it is based on and drawn from a plenary address at the 44th Annual Meeting of the Human Factors and Ergonomics Society and International Ergonomic Association, August 2, 2000 and the accompanying material in *Ergonomics* (D. Woods, K. Christoffersen, and D. Tinapple, Complementarity and Synchronization as Strategies for Practice-Centered Research and Design). Much of this chapter is taken directly from this production and paper.

The ideas were developed, first, as part of research projects on Cooperative Cognition sponsored by NASA and second as part of research projects on Data Overload sponsored by the Air Force Research Laboratory, Wright-Patterson Air Force Base (WPAFB). Jane Malin of NASA and Gil Kuperman of WPAFB were instrumental in stimulating, critiquing, and nurturing the development of these ideas in these projects.

The series of papers and multimedia productions on design that lead up to and buttress the framework of complementarity and synchronization include:

- Woods, D.D. (1998). Designs are Hypotheses about How Artifacts Shape Cognition and Collaboration. *Ergonomics, 41*, 168–173.

- Woods, D.D., Patterson, E.S., Corban J., & Watts, J.C. (1996). Bridging the Gap between User-Centered Intentions and Actual Design Practice. *Proceedings of the 40th Annual Meeting of the Human Factors and Ergonomics Society*. Santa Monica, CA: Human Factors & Ergonomics Society.
- Woods, D.D., & Tinapple, D.W (1999), *Watching Human Factors Watch People at Work*. Presidential Address, 43rd Annual Meeting of the Human Factors and Ergonomics Society. Houston, TX. [Multimedia production at http://csel.eng.ohio-state.edu/hf99/]
- Potter, S.S., Roth, E.M., Woods, D.D., & Elm, W. (2000). Bootstrapping Multiple Converging Cognitive Task Analysis Techniques for System Design. In J.M.C. Schraagen, S.F. Chipman, & V.L. Shalin (Eds.), *Cognitive task analysis*. Mahwah, NJ: Erlbaum.
- Dekker, S., & Woods, D.D. (1999). Extracting Data from the Future: Assessment and Certification of Envisioned Systems. In S. Dekker & E. Hollnagel (Eds.), *Coping with computers in the cockpit* (pp. 7–27). Aldershot, UK: Ashgate.
- Hoffman, R., & Woods, D.D. (2000). Studying Cognitive Systems in Context. *Human Factors, 42*(1), 1–7.

REFERENCES

- Hutchins, E. (personnel communication, 1992).
- Woods, D.D. & Sarter, N.B. (1993) Evaluating the impact of new technology on human-machine cooperation. In J. . Stager (Eds.), *Verification and validation of complex and integrated human machine systems.* Berlin: Springer-Verlag.

5 Using Intermediate Design Artifacts to Bridge the Gap Between Cognitive Analysis and Cognitive Engineering

SCOTT S. POTTER
Aegis Research Corporation

WILLIAM C. ELM
Aegis Research Corporation

EMILIE M. ROTH
Roth Cognitive Engineering

JAMES W. GUALTIERI
Aegis Research Corporation

JAMES R. EASTER
Aegis Research Corporation

ABSTRACT

There has been a growing interest in using Cognitive Task Analysis (CTA) to understand the requirements of cognitive work and to provide a foundation for design of new decision-support systems. While CTA techniques have proved successful in illuminating the sources of cognitive complexity in a domain of practice and the basis of human expertise, the results of the CTA are often only weakly coupled to the design of support systems. A critical gap occurs at the transition from CTA analysis to system design, where insights gained from the CTA must be translated into design requirements. We describe an approach to bridging this gap. Our approach uses intermediate design artifacts to create a design thread that links the demands of the domain as revealed by the CTA, to the cognitive and collaborative processes that require support, through the elements of the decision aid that explicitly address those support requirements.

We present a visualization that was developed using this methodology to illustrate the approach. The visualization was developed to support military commanders in selecting among alternative courses of action.

5.1 INTRODUCTION

There is a tremendous need for powerful Decision Support Systems (DSS) to support humans in the increasingly complex work domains they face. The methodology presented in this chapter to "bridge the gap" between cognitive analysis and the design of effective decision aids is based on some fundamental premises chaining back from the fundamental goal of building effective DSSs—ones that "make the problem transparent to the user" (Simon, 1981). These premises constitute the underlying basis for the methodology described in this chapter. By understanding these underlying assumptions, the reader will have the necessary context to understand why the methodology evolved to its current state: an adaptation of fundamental research into a pragmatic engineering approach. The ultimate goal has been to develop DSSs that complement the human decision maker to form an integrated human-machine team capable of solving difficult, real-world problems more effectively than either individually.

Behind this goal is *Premise 1—Humans form a mental model of the domain as part of their understanding and problem solving*. In addition, they employ a variety of problem-solving strategies to reason on that mental model and make decisions, varying both by situation and by variations between human decision makers. These problem-solving strategies will be a situationally varying mix of sensory-motor responses (skill-based), actions based on stored rules and experience (rule-based), and behavior based on an internal representation of underlying, fundamental behavioral characteristics of the work domain (knowledge-based). Rouse and Morris' (1986) definition of mental models is functional in nature and will be the one utilized in this chapter. They define mental models as "the mechanism whereby humans are able to generate descriptions of system purpose and form, explanations of system functioning, observe system states, and prediction of future system states" (Rouse & Morris, 1986; p. 351). Experts employ "better" mental models—models with richer domain knowledge, more structure and interconnections, and a basis in the underlying principles of the work domain (Woods & Roth, 1988).

The requirement for the DSS to be "transparent" to the user, that is, to allow an understanding of the problem to flow into the human's decision making with virtually no cognitive effort results in *Premise 2—The DSS must itself embody a "knowledge model" of the domain that closely parallels mental models representative of expert human decision-making.* This is consistent with the notion of common frame of reference (Woods & Roth, 1988). The more close-

ly the models parallel, the less cognitive effort it will take on the part of the human to "transform" the data of the DSS into the information needed to mentally solve the problem. As the degree of match approaches one, the concept of "transparent" becomes a reality. In addition to providing an effortless match for the expert operator, the knowledge model within the DSS can be viewed as an analytically derived *a priori* starting point for the novice user's own embryonic mental model of the domain. Also, the DSS's knowledge model can serve as a constant reminder of infrequently used concepts and knowledge even to experts.

To engineer such a knowledge model within the DSS results in *Premise 3—An effective DSS knowledge model is composed of functional nodes and relationships intrinsic to the work domain.* Explicit denotation of the cognitive tasks associated with nodes and relationships forms the basis of the support requirements of the DSS, which then must support each of the cognitive tasks across a variety of strategies. This DSS knowledge model therefore defines the basis for both the structure and content of the DSS itself; it becomes the overall design specification (Woods & Roth, 1988).

To deliver such a knowledge model requires *Premise 4—An adaptation of Rasmussen's abstraction hierarchy provides the needed representation of the abstract concepts and relationships applicable across all situations and strategies.* Functional nodes representing goals and functional processes to achieve those goals, at varying levels of abstraction, are a critical element differentiating expert (high-quality) problem solving from novice problem solving. In addition, the explicit relationships between those nodes provide the natural structure of the work domain. The functional abstraction hierarchy described in this paper is a pragmatic adaptation of that approach, but nonetheless based on the same premise: An abstract model of the functional nature of the domain provides the necessary exploration of highly abstract concepts of the domain while being sufficiently robust (independent of the physical particulars of the situation) to avoid brittleness in unexpected problem solving situations (Vicente, 1999). With this premise, the problem has come full circle—the domain defines the fundamental skeleton of the DSS's knowledge model to support a human dealing with that same domain.

These premises were organized as a reverse chain of logic from the starting requirement of "build a DSS to deliver improved human-machine decision making effectiveness," which exactly represents the evolution of the methodology described below. It provides a sense for the rationale underlying the various decisions that created the methodology. The remainder of the chapter is presented in a forward chain more representative of how the methodology is actually practiced for development of a particular DSS: When presented with a complex, real-world (naturalistic decision making) domain, how can a truly effective (transparent) DSS be developed?

To answer that question, there has been growing interest in using Cognitive Task Analysis (CTA) to understand the requirements of cognitive work and to provide a foundation for design of new decision-support systems (Schraagen, Chipman, & Shalin, 2000; Vicente, 1999). While CTA techniques have proved successful in illuminating the sources of cognitive complexity in a domain of practice and the basis of human expertise, the results of the CTA are often only weakly coupled to the design of support systems (Potter, Roth, Woods, & Elm, 2000). A critical gap occurs at the transition from CTA analysis to system design, where insights gained from the CTA must be translated into design requirements.

In this chapter we will describe our approach to bridging the gap between analysis and design. Our approach uses intermediate, decision-centered artifacts to create a bridge that links the demands of the domain as revealed by the cognitive analysis, to the cognitive processes (both individual and collaborative) that require support, through the elements of the decision aid that explicitly address those support requirements. Following the methodological discussion we will present an illustrative visualization that was developed using this methodology. The visualization was developed to support military commanders in selecting among alternative courses of action in applying combat power.

5.2 DEVELOPING A DESIGN THREAD FROM COGNITIVE ANALYSIS TO DECISION AIDING

Our approach is predicated on the premise that the design of advanced visualizations and decision-aids must explicitly reflect the fundamentals of the domain of practice and the demands it imposes on domain practitioners. We employ a structured, principled methodology to systematically transform the problem from an analysis of the demands of a domain to identifying visualizations and decision-aiding concepts that will provide effective support. The steps in this process include:

- Capturing the essential domain concepts and relationships that define the problem-space confronting the domain practitioners
- Identifying the cognitive demands/tasks/decisions that arise in the domain and require support
- Identifying the Information Requirements (IRs) to successfully execute these cognitive demands/tasks/decisions
- Defining the relationships between Decision Requirements (DRs) and user interface design concepts
- Exploring techniques to implement these design concepts into powerful visualization and decision support concepts.

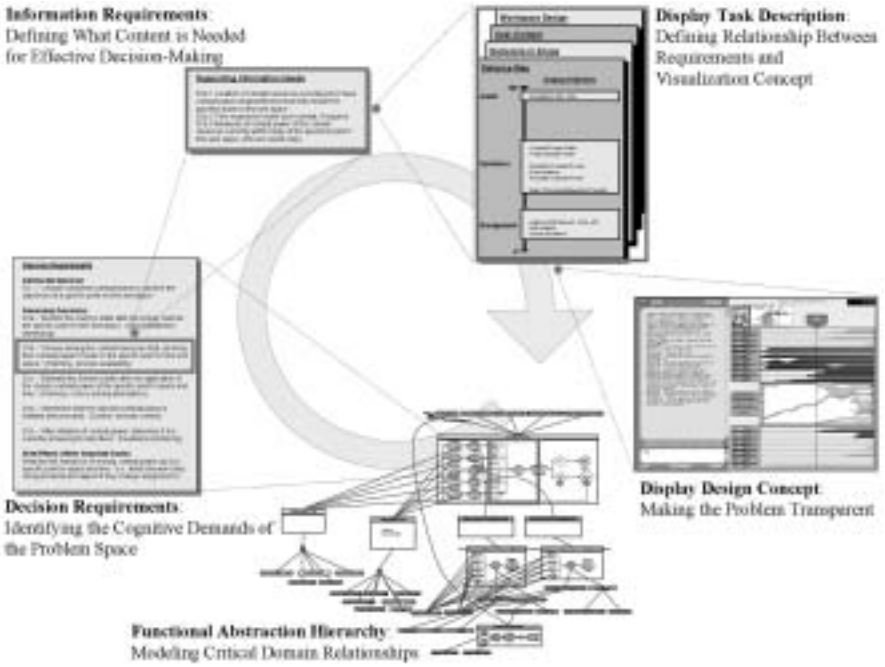


Figure 5.1: A sequence of analysis and design steps creates a continuous design thread that starts with a representation of domain concepts and relationships through development of decision support requirements to creation of visualization and aiding concepts and rapid prototypes with which to explore the design concepts.

In our analysis and design approach, we create design artifacts that capture the results of each of these intermediate stages in the design process. These design artifacts form a continuous design thread that provides a principled, traceable link from cognitive analysis to design. However, the spans of the bridge are constructed by the process of generating these artifacts, not the artifacts themselves. The artifacts serve as a mechanism to record the results of the process. Figure 5.1 provides a visual depiction of the sequence of methodological activities and associated design artifacts. The remainder of the paper describes and illustrates the approach.

5.2.1 Representing the Way the World Works—Building a Functional Abstraction Hierarchy

This methodology begins with a function-based goal-means decomposition of the system. This methodology has roots in the formal, analytic goal-means decomposition method pioneered by Rasmussen and his colleagues as a for-

malism for representing cognitive work domains as an abstraction hierarchy (e.g., Lind, 1993; Rasmussen, 1986; Rasmussen, Pejtersen, & Goodstein, 1994; Roth & Mumaw, 1995; Vicente, 1999; Vicente & Rasmussen, 1992; Woods & Hollnagel, 1987). A work domain analysis is conducted to understand and document the goals to be achieved in the domain and the means available for achieving them (Vicente, 1999). The objective of performing this analysis is to develop a structure that links the purpose(s) of individual controllable entities with the overall purpose of the system. This includes knowledge of the system's characteristics and the purposes or functions of the specific entities. The result of the first phase is a Functional Abstraction Hierarchy (FAH)—a multilevel recursive means-ends representation of the structure of the work domain—that anchors the first span of the bridge.

The work domain analysis is performed based on extensive interactions with expert practitioners in the domain and includes face-to-face interviews with the experts, watching the experts work in the domain, verbal protocol techniques, and other CTA and Cognitive Work Analysis (CWA) methods (see Potter, Roth, Woods, & Elm, 2000; Vicente, 1999). In practice, building an FAH is an iterative, progressively deepening process. It starts from an initial base of knowledge (often very limited) regarding the domain and how practitioners function within it. Then, complementary techniques are used to expand and enrich the base understanding and evolve a function-based model from which ideas for improved support can be generated. This process is highly opportunistic. Whether one starts by focusing on understanding the domain or by focusing on the knowledge and skills of domain practitioners depends on the specific local pragmatics. The key is to focus on progressively evolving and enriching the model so as to ultimately discover an understanding of the goal-driven characteristics of the domain that will lead to an understanding of the decisions practitioners are faced with in the domain.

The phrase "bootstrapping process" has been used to describe this process and emphasize the fact that the process builds on itself (Potter et al., 2000). Each step taken expands the base of knowledge providing opportunity to take the next step. Making progress on one line of inquiry (understanding one aspect of the field of practice) creates the room to make progress on another. For example, one might start by reading available documents that provide background on the field of practice (e.g., training manuals, procedures); the knowledge gained will raise new questions or hypotheses to pursue that can then be addressed in interviews with domain experts, and it will also provide the background for interpreting what the experts say. In turn, the results of interviews or exercises may point to complicating factors in the domain that need to be modeled in more detail in the FAH. This provides the necessary background to create scenarios to be used to observe practitioner performance under simulated conditions or to look for confirming example cases or interpret observations in naturalistic field studies.

The resulting FAH specifies the domain objectives and the functions that must be available and satisfied to achieve their goals. In turn, these functions may be abstract entities that need to have other, less abstract or less aggregated functions available and satisfied so that they might be achieved. This creates a decomposition network of objectives or purposes that are linked together from abstract goals to specific means to achieve these goals. For example, in the case of engineered systems, such as a process control plant, functional representations are developed that characterize the purposes for which the engineered system has been designed, and the means structurally available for achieving those objectives. In the case of military command and control systems, the functional representations characterize the functional capabilities of individual weapon systems, maneuvers, or forces and the higher-level goals related to military objectives.

The FAH provides a framework for making explicit the goals to be achieved in the domain and the alternative means available for achieving those goals. High-level goals, such as impacting a critical function, are decomposed into supporting lower-level subgoals. This provides the basis for identifying—through subsequent steps in the analysis and design process—the cognitive activities that arise in the domain and the information needed to support those decisions. The FAH enables the designer to determine where decision making is likely to be difficult due to the fundamental characteristics of the domain. For example, the FAH helps convey places in problem space where objectives compete with each other (e.g., where choices have to be made that require some level of sacrificing of one objective to achieve another, perhaps more heavily weighted, objective), or otherwise constrain each other (e.g., where the satisfaction of multiple goals needs to be considered in determining the best course of action).

Figure 5.2 depicts an example that illustrates these essential characteristics of an FAH:

- Processes may affect more than one goal—these side effects govern the operation of a process to achieve the goal of interest
- Each process can be modeled qualitatively to represent how it works to achieve a goal
- Relationships within the model can be recursive—processes can have requirements that are supported by more "abstract" process
- The term "hierarchy" is actually a misnomer; the structure of the model is actually a network
- Moving up through the network defines supported processes and impact on goal achievement; moving down defines supporting processes and requirements for goal achievement.

There are a growing number of examples of successful systems that have been developed based on a work domain analysis. Examples of functional

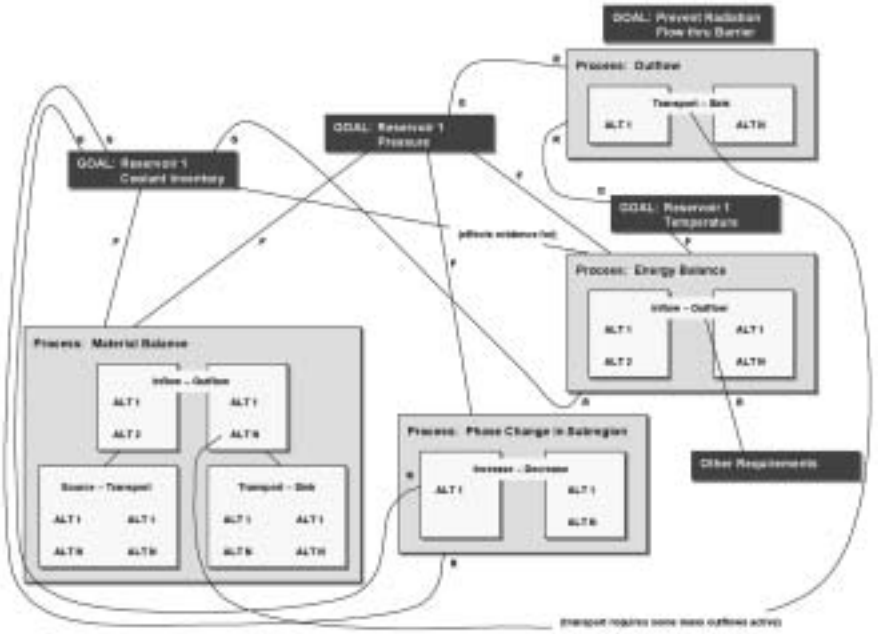


Figure 5.2: A sample goal-means decomposition from Woods and Hollnagel (1987) for one port (primary-system thermodynamics) of a nuclear power plant.

abstraction hierarchies and how they were used to design new visualizations and DSSs can be found in Roth, Lin, Kerch, Kenney, and Sugibayashi (in press) and Potter, Roth, Woods and Elm (2000). Examples of the application of this approach to model cognition and collaboration and to develop new online support systems in time pressured tasks such as situation assessment, anomaly response, supervisory control, and dynamic replanning include domains such as military intelligence analysis (Potter, McKee, & Elm, 1997), military aeromedical evacuation planning (Cook, Woods, Walters, & Christoffersen, 1996; Potter, Ball, & Elm, 1996), military command and control (Chalmers, Easter, & Potter, 2000), railroad dispatching (Roth, Malsch, Multer, Coplen, & Katz-Rhoads, 1998), and nuclear power plant emergencies (Roth, Lin, Thomas, Kerch, Kenney, & Sugibayachi, 1998).

5.2.2 Modeling Cognitive Demands—Deriving DRs

With the FAH representation of the work domain as the underlying framework, it is possible to derive the cognitive demands for achieving domain goals. In our methodology, we refer to these demands as DRs. Thus, the term "decision" is used in a broad sense. Based on the underlying premises of the modeling

methodology, these decisions center around goal-directed behavior, such as monitoring for goal satisfaction and resource availability, planning and selection among alternative means to achieve goals, and controlling activities (initiating, tuning, and terminating) to achieve goals (Roth & Mumaw, 1995) as well as collaboration activities in team settings (Gualtieri, Roth, & Eggleston, 2000). By organizing the specification of operator DRs around nodes in the goal-means structure, rather than organizing requirements around predefined task sequences (as in traditional approaches to task analysis), the representation helps ensure that the resulting design concepts (on the system design side of the gap) reflect a decision-centered perspective. The resulting displays and decision-aids will thus support domain practitioners in understanding the goals to be achieved and what decisions and actions need to be taken to achieve these goals.

The cognitive demands that are derived from a cognitive analysis of the work domain constitute a second span in the bridge—DRs. DRs are tied directly to nodes in the FAH and provide an intermediate artifact that forms the essential part of the design thread, eventually providing an end-to-end connection from goal nodes in the FAH to supporting visualization and decision support concepts.

The FAH forms the basis for the structure of the decision-making activities that will be reflected in the DRs. For example, every goal node in the FAH has associated "goal monitoring" types of decisions. Likewise, processes have associated "process monitoring" decisions. Similarly, there will always be "feedback monitoring" types of decisions related to assessing whether actions are achieving desired results. Depending on the relationships between nodes in the FAH, there will be decisions related to prioritization of goals, selection of alternative means to achieve a particular goal, and monitoring side effects of actions.

An underlying "template" for this step in the analysis is presented in Table 5.1.

The key issue here is that this template is not meant to be a rote, "turn the crank" type of process. Rather, these questions are meant to be a guide to stimulate thinking about relevant decision-making in the context of a FAH model of the target work domain. Each domain is unique in the decision-making demands imposed on the human operators. As such, each work domain will require slightly different variants of these questions. Successful elucidation of DRs will also depend on corroboration from multiple data sources, including case studies, interviews, observations, etc. In addition, guiding insights can come from research on similar work domains as well as basic research on human cognition, decision-making, biases, and errors. For example, previous work on decision making in dynamic, high-risk worlds can guide analysis and interpretation of analogous worlds in terms of potential points of complexity, typical decision making difficulties and strategies, and critical characteristics of difficult problem-solving scenarios.

Table 5.1: Template of Typied Decision Types and Associated Knowledge Acquisition Questions. (Adapted from Roth & Mumaw, 1995.)

- Monitoring/Situation Awareness:
    - Goal Monitoring:
        - Goal Satisfaction: Are the function-related goals satisfied under current conditions?
        - Margin to Dissatisfaction: Are goal limits/restrictions being approached?
    - Process Monitoring:
        - Active processes: What processes are currently active? What is the relative contribution of each of the active processes to goal achievement? Are the processes performing correctly?
        - Process element monitoring: Are the individual processes and their components working as they are supposed to?
        - Automation monitoring: Are automated support systems functioning properly? What goals are the automated support systems attempting to achieve? Are these appropriate goals?
    - Feedback Monitoring:
        - Procedure adequacy: Is the current procedure achieving the desired goals?
        - Control action feedback: Are the operator control actions achieving their desired goals?
- Planning:
    - Goal priority: Which goal has the highest priority?
    - Process availability: What alternative processes are available for achieving the goals?
    - Choices among alternatives: Can an alternative process be deployed?
    - Consequences and side-effects of actions: What other processes or functions are affected by the current actions?
- Control:
    - Process control: How is the process controlled for process deployment, tuning for optimum performance, termination?
    - Manual take-over: If intervention is required, what actions must be taken?

5.2.3 Capturing the Means for Effective Decision-Making—Identifying Information Requirements

The next step in the process is to identify and document the information required for each decision to be made. Information requirements are defined as the set of information elements necessary for successful resolution of the asso-

1. This IR supports a DR of "monitor flow coefficient within operational limits."

ciated decision requirement. This set of information constitutes a third span in the bridge—IRs. The focus of this step in the methodology is on identifying the ideal and complete set of information for the associated decision-making.

IRs specify much more than specific data elements; it is data in context that becomes information (Woods, 1988, 1995). The data-to-information relationship can be complex and require a significant amount of computations and/or transformations. For example, in the case of a thermodynamic system, an IR might be "flow coefficient with respect to appropriate limits."[1] This requires the estimation of the parameter "flow coefficient" derived from model-based computations and sensor values and the comparison of that parameter against a limit referent. The degree of transformation required can vary from simple algebra to complex, intelligent algorithms. Potter et al. (1996) provide an example of IRs that were only able to be satisfied by an advanced planning algorithm and significant data transformations.

In addition, identifying IRs is focused on satisfying the DRs and is *not* limited by data availability in the current system. In cases where the required data is not directly available, this approach provides a rationale for obtaining that data (e.g., pulling data from a variety of previously "stove-piped" databases, adding additional sensors, or creating "synthetic" values). This is a critical change from the typical role that human factors engineers have had in the past (designing an interface after the instrumentation has been specified). Consequently, this type of an approach is fundamentally broader in scope than other approaches to interface design that do not consider the impact of IRs on system architecture specifications (Vicente, Christoffersen, & Hunter, 1996).

An interesting anecdote of this occurred in an interface design effort for a thermodynamic system (Potter et al., 1992). At this point in the process, the IR of "predicted liquid level in the accumulators versus current level over time" was identified to compensate for significant lags in the system in monitoring for system integrity. One of the engineers argued "but we don't have any way to sense 'predicted level'—that's our fundamental problem!" Slowly, another engineer in the room raised his hand and offered "my high-fidelity simulation of the system calculates that exact thing…I've just never talked about it because I didn't think it was of any value to anyone except me."

Just as the FAH representation provided the framework for the derivation of DRs, the DRs provide the essential context for the IRs because they indicate the factors (and thus information) that will need to be considered in making decisions. For example, in a "choice among alternative resources" type of decision, the choice requires information about the availability of the alternatives (supporting relationships), current tasking of those alternatives and the impact of selecting it for the task under consideration (side-effects), and specific performance capabilities of the alternatives (lower level functional properties of the alternatives).

5.2.4 Linking DRs to Aiding Concepts—Developing and Documenting a "Model of Support"

Once the FAH has been augmented with critical decision and information requirements, it becomes a solid foundation for the development of aiding concepts. The objective is to design visualization and decision support concepts to reflect the IRs that, as a result of the linkage back to the FAH, are organized into a virtual "information space" explicitly replicating the domain structure captured in the FAH. To accomplish this objective, this task develops the mapping between information on the state and behavior of the domain (i.e., decision and information requirements uncovered) and the syntax and dynamics of the visualization or decision aid being developed. From an interface design perspective, the goal is to reveal the critical IRs and constraints of the decision task through the user interface in such a way as to capitalize on the characteristics of human perception and cognition. This approach is consistent with cognitive engineering principles that have variously been called *Representational Aiding* (Bennett & Flach, 1992; Roth, Malin, & Schreckenghost, 1997, Woods, 1995) and *Ecological Interface Design* (Vicente & Rasmussen, 1990, 1992; Reising & Sanderson, 1998).

The display concept and how it supports the cognitive tasks is then captured in a Display Task Description (DTD). The DTD defines the goals and scope of a display in terms of the cognitive tasks it is intended to support (and thus a defined target region of the FAH). It also provides a specification of the supporting information and graphic elements required to support the cognitive tasks. A DTD is another span of the bridge that helps to link the decisions within the work domain to the visualization and decision support concepts intended to support those decisions. In many cases, multiple design concepts may be generated that support a given set of decisions. These alternative solutions can be captured in a DTD.

An advantage of a DTD is that it requires designers to be more explicit about the specific cognitive activities that a given visualization or decision support concept addresses. DTDs specify the decisions to be supported and the cognitive performance objectives that the display or decision aid is intended to achieve and the information (not just data) that must be conveyed. Explicit links are made between particular aspects of the display concepts and specific cognitive demands they are intended to support. As such they constitute explicit hypotheses—a model of support—that can be empirically evaluated. As a consequence, DTDs enable more informed and pointed testing of the effectiveness of the proposed aiding concepts.

Another advantage of DTDs is that they enable designers and evaluators of designs to clearly distinguish and independently evaluate the objectives of a display in terms of intended support from the particulars and aesthetics of its implementation. One can ask "are the support objectives of this display correct

and complete?" as well as "does a particular embodiment of the display concept achieve the intended support objectives?" This is a key aspect of bridging the gap between analysis and design.

The DTD also represents a configuration management tool, critical for ensuring coverage of the functional decision space across all displays and display elements. The DTD represents a shift in focus from "what" is to be displayed to "how," including annotations on relative importance that maps to relative salience on the visualization, etc. The DTD is not only a compilation of information developed earlier, it has the added value of a more complete description of the behaviors and features needed to communicate the information effectively as well as an allocation of the IRs across the entire set of displays within the workspace. When done correctly it is still in the form of a "requirement" and not an implementation. This artifact becomes a key transition between the cognitive system engineer, the system developer, and the system tester.

5.2.5 Developing Prototypes That Instantiate the Aiding Concept

The introduction of new technology inevitably transforms the work domain and the demands placed on domain practitioners, often in unanticipated ways. New technology introduces new error forms; new representations change the cognitive activities needed to accomplish tasks and enable the development of new strategies; new technology creates new tasks and roles for people at different levels of a system. Changing systems change what it means for someone to be an expert and change the kinds of errors that will occur.

Given this transformation, developers face the "envisioned world" problem of the unforeseen impacts of the introduction of new technology (Dekker & Woods, 1997; Smith, Woods, McCoy, Billings, & Sarter, in press; Woods, 1998). A similar phenomenon has been noted in the computer-human interaction literature where it is referred to as the "task-artifact cycle" (Carroll, Kellogg, & Rosson, 1991). Concepts for new visualizations and DSSs (aiding concepts) represent hypotheses about what will provide effective support to domain practitioners in the envisioned world. Rapid prototypes of aiding concepts that implement the DTDs become tools for discovery (Potter, Roth, Woods, & Elm, 2000). By exploring the impact of prototypes that embody aiding concepts, it becomes possible to evaluate the effectiveness of the envisioned support systems (i.e., test the hypotheses) as well as to identify additional support requirements and unanticipated consequences of the introduction of the new technologies.

The envisioned world problem means that system developers must face a challenge of prediction:

- What will be the cognitive demands of the envisioned world?
- How will the envisioned support concepts shape cognition

and collaboration?

- How will practitioners adapt artifacts to meet their own goals, given mismatches to the actual demands and the pressures they experience?
- How will the new technology impact a domain that doesn't yet exist or is evolving?

The goal of such predictions is to influence the development process so that new decision aids provide effective, robust decision support. This can be accomplished by developing prototypes of the visualization and decision support concepts as specified in the DTD. These prototypes provide a concrete instantiation of the aiding concepts specified in the DTD. They can be used to explore the viability of the aiding concept. Then, each opportunity to assess the utility of the prototype can also provide additional understanding of the requirements for effective support. Thus, these assessments can serve to enrich and refine the initial FAH and identify additional decision and information requirements that were missed in the original analytic process. Note that extending the analysis to encompass exploration of the envisioned world in a closed-loop, iterative manner contrasts with the narrow view of cognitive analysis as an initial, self-contained technique whose product is handed-off to system designers in a waterfall model approach.

The methodology and resulting design artifacts described in these sections provide the blueprint for the development of DSSs designed around the fundamental, underlying demands of the work domain to deliver significantly improved human-machine decision-making effectiveness. The remainder of this chapter will provide an instantiation of these artifacts for a component of a DSS prototype designed to support military commanders in selecting among alternative courses of action in applying combat power.

5.3 AN ILLUSTRATIVE CASE: AN ECOLOGICAL INTERFACE SUPPORTING COMMAND DECISION MAKING

A decision aid that was recently developed to support military command decision-making can be used to illustrate the breakthroughs achieved from this methodology. The display was developed as part of the DARPA Command Post of the Future program, as a means of illustrating the use of this approach to support the development of powerful new visualizations (Logica Carnegie Group, 2000a). The discussion of this case study will follow the intermediate artifacts that provide a design thread from cognitive analysis through prototype implementation.

2. The electronic tactical decision game (eTDG) was developed by John Schmidt as part of the Command Post of the Future program.

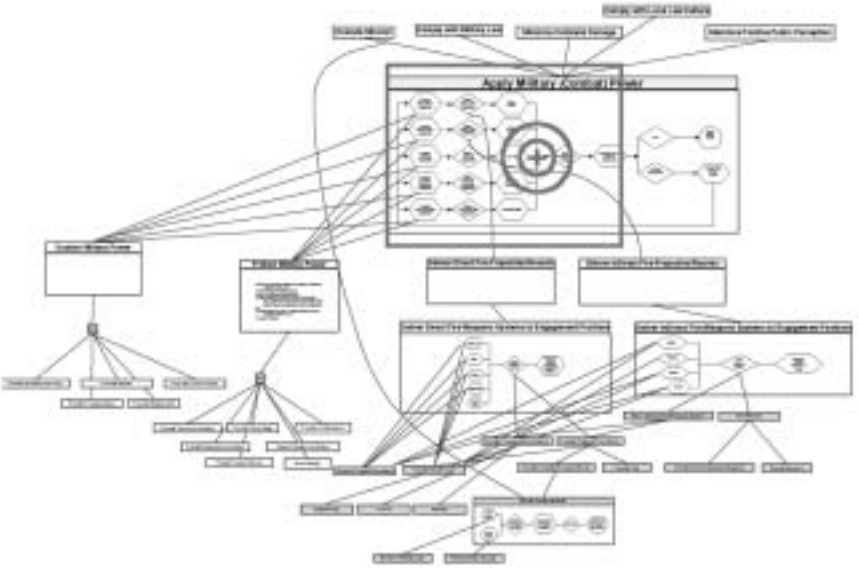


Figure 5.3: Functional abstraction hierarchy of military command and control with the "Apply Military (Combat) Power" portion highlighted.

5.3.1 Functional Abstraction Hierarchy (FAH)

As part of the program an FAH was developed for military command and control. The FAH is presented in Figure 5.3. It depicts the goals, functional processes necessary to achieve the goals, and the subgoals that result from the need to support the functional processes.

A variety of knowledge elicitation techniques were used to bootstrap an initial understanding of the decision problems faced by military commanders, how military commanders conceptualize the problem space, and the complications that arise in the domain that increase the difficulty of decisions.

Knowledge acquisition activities included:

- Reviewing military documents
- Conducting structured interviews with military commanders at different levels in the command chain, including recently retired general officers with combat experience
- Participating in electronic tactical decision games that were conducted as part of the Command Post of the Future program and included recently retired general officers as game participants[2]
- Presenting draft versions of the FAH to military commanders and revising them based on their feedback. This was an iterative process that occurred over several cycles.

One of the findings of the knowledge elicitation process is that military commanders think about military units in terms of the abstract concept "combat power" that military units possess and can generate. This abstract concept includes not only the number and type of military equipment that the unit possesses, but also less tangible factors such as the unit's morale, fatigue, etc., as well as a number of external factors including terrain. In Figure 5.3, "Apply Combat Power," has been generalized to "Apply Military (Combat) Power" to reflect the fact that the modern day military is often asked to engage in non-combat operations (e.g., providing humanitarian aid).

The function of applying combat power (or more generally military power) is performed in the context of meeting several higher-level goals. The most direct higher-level goal is to satisfy mission objectives. There are other higher-level goals, however, that must be taken into account. These include "complying with military law," "minimizing collateral damage," "complying with local laws and cultures," and a need to attain "positive public perception" of the operation. These additional goals can place constraints on achieving mission goals. The fact that high-level goals can compete with each other is reflected in the FAH.

The military command and control FAH illustrates one of the important benefits of developing a functional abstraction hierarchy representation of the domain. The exercise of developing an FAH enables the cognitive analyst to see beyond the physical level of description of the domain and to begin to understand and represent the domain at higher levels of abstraction. The concept of "combat power" that is represented in the military command and control FAH is a good example. Combat power can be thought of as a "commodity." The commander is given resources (troops, planes, tanks, logistics support) that have many complex functional properties and interdependencies that affect the amount of combat power that can be delivered at a given point in space and time. As such, the commander can be thought of as the manager of a very precious commodity, seeking ways to maximize the combat power that can be brought to bear to achieve a particular mission.

Military commanders routinely think in terms of abstract concepts such as "relative combat power" in making decisions about movement of troops and equipment to achieve a mission objective. Yet the current tools available to them (e.g., physical maps of the terrain with icons representing placement of troops and equipment) provide a much more physical representation of the problem space, making the estimation of relative combat power difficult and prone to error. Anecdotally, the IR for this domain included "relative combat power over time" a very untraditional concept that created significant excitement since it solidified some doctrinal writings about controlling the "tempo" of the conflict.

A second important attribute illustrated by the military command and control FAH is the value of the conceptual structure it provides. The focus of effort in developing a FAH is on constructing a representation that captures

the important concepts and interrelationships in the domain at appropriate levels of abstraction and makes the goals, means, relationships, and constraints explicit. Initially, the emphasis is placed on shaping the representation around key concepts (goals and processes) and relationships. Once the right structure is in place, supporting details (i.e., less abstract concepts such as procedures, and functional capabilities) can be added to the FAH to flesh out the model.

It has often been our experience that when we finally get the representation "right" (after many iterations) the resulting FAH looks simple and is readily understood and accepted by domain Subject-Matter Experts (SMEs). We have coined the term "of course test" to describe the typical reaction of domain practitioners to an FAH that successfully crystallizes the most important abstract concepts and inter-relationships in the domain. It should be pointed out, however, that while the FAH may look simple and obviously true to the SMEs once it is presented to them, the conceptual structure represented in the FAH is not something that the domain experts could have spontaneously generated on their own. Often the concepts represented in the FAH are things that the domain experts understand implicitly, and readily resonate to once shown, but could not generate themselves.

The power of this abstraction hierarchy is that it allowed the cognitive analysts to recognize the need to create displays and decision-aids that allow the commander to visualize and control the domain at this higher, more abstract level of goal achievement. As part of the program a display was created that illustrates this point.

The objective of the display was to support commanders in choosing the appropriate military power to achieve mission objectives. The display was targeted at combat applications where the mission objective is to engage and defeat an enemy. In this combat context the probability of meeting mission objectives is a function of the ratio of friendly to enemy combat power at the point in time and space where the engagement is to take place.

The "choose combat power" display was designed to support commanders in deciding: (1) at what location to engage the enemy; (2) when to engage the enemy; and (3) what combat resources to deploy to achieve an acceptable potential to defeat the enemy. While this paper focuses on the "choose combat power" display, the display is intended to be only one of a suite of displays that would encompass a complete battle command decision-aid.

The FAH, shown in Figure 5.3, provided the starting point for developing the visualization to support commanders in choosing the appropriate combat power to carry out the mission objective(s). The starting point is the "apply military (combat) power" node in the FAH. This entails choosing the combination of resources that the commander believes will effectively carry out the mission.

Table 5.2: DRs Associated With the "Choose Combat Power" Visualization Concept

DRs
DR1a—"Determine the point in time when the enemy will reach a specified point in space and will monitor the enemy's combat power over time." (*This decision is driven by the need to establish the mission requirements (i.e., evaluate enemy combat power) so as to understand the friendly combat power that will be required to achieve mission goals (i.e., defeat the enemy)).*
DR1b—"Choose among the friendly combat resources that can bring their combat power to bear at the specific point in space and time." (*This decision considers the availability of resources to achieve mission goals.)*
DR1c—"Estimate the potential to defeat the enemy after the application of the chosen power at the specific point in time and space." (*This decision is related to the choice among the various alternatives.)*
DR1d—"Determine the impact of moving combat power to a specific point in space and time, i.e., what else were these resources committed to and what will happen if they change their assignments/committments?" (*This decision considers the side effects of choosing various alternatives.)*

5.3.2 Decision Requirements

The next step in the analysis and design process was to utilize the structure of the problem space map (i.e., the FAH) and derive the supporting decisions for accomplishing the objective in question. In this specific context, the "apply military (combat) power" node in the FAH suggested the decision of "choose combined combat power to achieve the objectives at a specific point in time and space." The decision is written in a generic form (i.e., independent of the particular battle or terrain that originally exposed the need for the decision). This is part of the process of making the DSS not situationally dependent. Then, the specific set of DRs were derived in part by imposing the template of generic questions (see Roth & Mumaw, 1995). The set of DRs that was derived through this process is shown in Table 5.2.

5.3.3 Supporting Information Requirements

Associated with each of these decisions is requisite domain-specific information needed to inform the decision. A description, in generic terms, of the type of information necessary to support the "choose combat power" decision is given in Table 5.3. The identification numbers for IRs cross-reference the DRs they support.

Table 5.3: Supporting Information Requirements Associated With the "Choose Combat Power" Decision Requirement

Information Requirements
DR 1a:
IR 1a.1 – "Expected arrival time of enemy combat resources at the specified point in space," (i.e., the lead unit, as well as other follow-on units).
IR 1a.2 – "Estimated measure of combined enemy combat power at the specified point in space, beginning at the arrival time of the first enemy unit and extending through follow-on units."
DR 1b:
IR 1b.1 – "The time required for selected friendly combat resources to reach the specified point in space."
IR 1b.2 – "Estimated measure of combined combat power of the selected friendly combat power resources once they reach the specified point in space."
DR 1c:
IR 1c.1 – "Measure of combat power ratio of friendly to enemy combat power beginning with the arrival of the first unit (friendly or enemy) over time."
IR 1c.2 – "Indication of combat power ratios required to defeat the enemy under different battle conditions (i.e., doctrinal / procedural referent information)."
IR 1c.3 – "Location of alternative resources of both friendly and enemy combat power that could be brought to bear."
IR 1c.4 – "The time required to bring to bear the combat power of these alternative friendly and enemy combat resources."
IR 1c.5 – "Measures of cumulative combat power of both friendly and enemy resources as additional friendly and enemy resources are selected (over a specified window in time)."

5.3.4 Display Task Description

Based upon the FAH, the DRs, and the IRs, a DTD was created to be used to initiate the visualization design. As mentioned earlier, the DTD provides a specification of the specific supporting IRs for the resulting graphic elements to support the cognitive tasks/decisions. If the graphic elements and visualizations effectively convey the IRs identified during the IRs analysis, we have successfully bridged the gap between the essential demands of the work domain and the resulting decision support concepts. In most cases, this also requires consideration of allocation of functional/decision/information scope for each display. However, for the present example, we have not included that broader scope issue in this discussion. Table 5.4 presents the display task description for the "choose combat power" display.

Table 5.4: Display Task Description for the "Choose Combat Power" Display

Display Task Description	
Context	
This display is intended for combat applications where the mission objective is to engage and defeat an enemy. It should assist the commander in choosing combined combat power and managing the space/time tradeoff to control the battlespace. Specifically it is to assist in deciding: (1) at what location to engage the enemy; (2) when to engage the enemy; and (3) what combat resources to deploy to maximize the potential to defeat the enemy.	
Decision and Information Requirements	**Visualization Needs**
DR 1a—"Determine the point in time when the enemy will reach a specified point in space and will monitor the enemy's combat power over time." **IR 1a.1—**"Expected arrival time of enemy combat resources at the specified point in space." (i.e., the lead unit, as well as other follow-on units). **IR 1a.2—**"Estimated measure of combined enemy combat power at the specified point in space, beginning at the arrival time of the first enemy unit and extending through follow-on units."	Visualize the time required for enemy unit(s) to reach location designated by commander, and visualize cumulative enemy combat power at that location as a function of time.
DR 1b—"Chose among the friendly coombat resources that can bring their combat power to bear at the specific point in space and time." **IR 1b.1—**"The time required for selected friendly combat resources to reach the specified point in space." **IR 1b.2—**"Estimated measure of combined combat power of the selected friendly combat power resources once they reach the specified point in space."	Visualize the time required for friendly "combat power" unit to bring their combat power to bear on the designated point (in space and time) and visualize cumulative friendly combat power at that location as a function of time.
DR 1c—"Estimate the potential to defeat the enemy after the application of the chosen power at the specific point in time and space." **IR 1c.1—**"Measure of combat power ratio of friendly to enemy combat power beginning with the arrival of the first unit (friendly or enemy) over time." **IR 1c.2—**"Indication of combat power ratios required to defeat the enemy under different battle conditions (i.e., doctrinal/procedural referent information)." **IR 1c.3—**"Location of alternative resources of both friendly and enemy combat power that could be brought to bear." **IR 1c.4—**"The time required to bring to bear the combat power of these alternative friendly and enemy combat resources." **IR 1c.5—**"Measures of cumulative combat power of both friendly and enemy resources as additional friendly and enemy resources are selected (over a specified window in time)."	Visualize relative friendly to enemy combat power ratio at the designated point in space as a function of time, and compare to combat power ratio required to defeat enemy under different battle conditions. Visualize uncertainty of estimates of Combat Power Ratio. Visualize changes in enemy and friendly combat power and combat power ratio as selected enemy and/or friendly units are added or removed.

5.3.5 Rapid Prototype of "Choose Combat Power" Display

A rapid prototype of the "choose combat power" display was developed to provide a concrete instantiation of the concepts embodied in the DTD. Two versions of the rapid prototype were developed. First, a static "storyboard" version of the display was built to serve as a vehicle for communicating the display concepts. Later, a software prototype was developed that is called Joint Operations Environment (JOE) to explore the dynamic behavior when driven from a simulated dataset.

Figure 5.4 presents a "scenario" that illustrates the context in which a visualization such as JOE would be used. The commander would begin by viewing a physical representation of the terrain with enemy and friendly units and their positions identified. This is similar to the types of geographic displays that

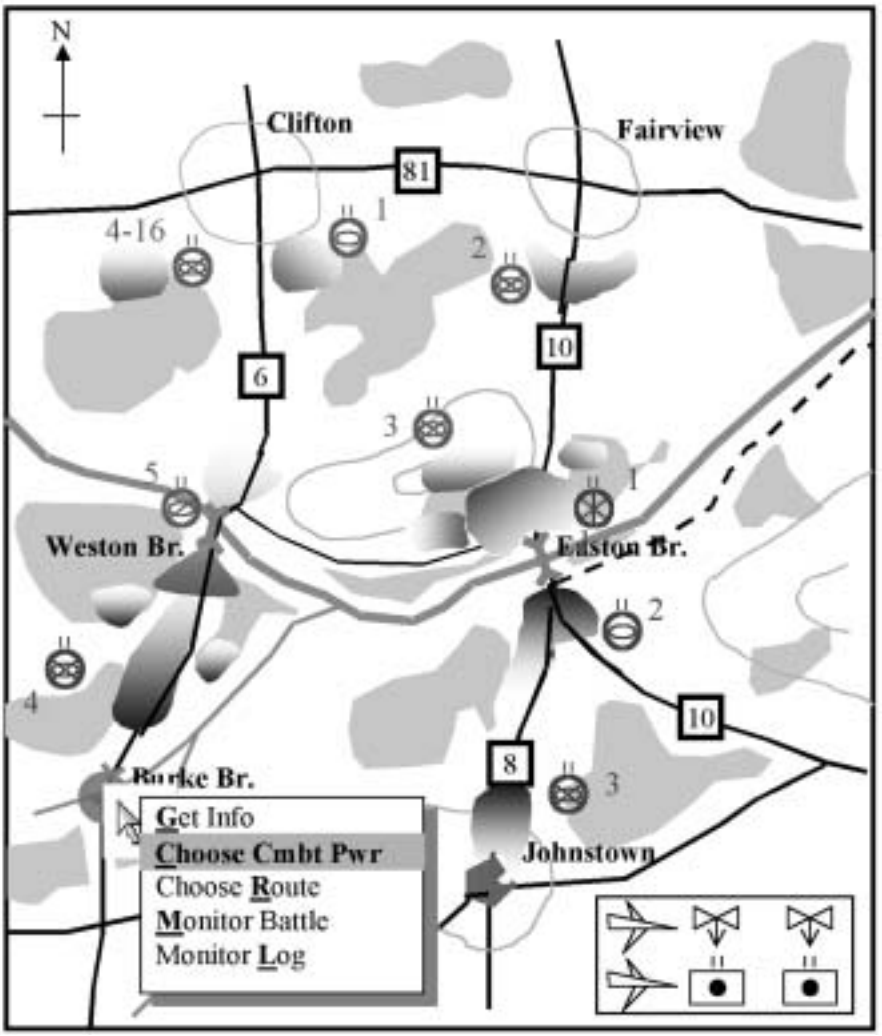


Figure 5.4: Geographic map providing the scenario context for the "choose combat power" visualization. This is classically the DSS a commander uses (along with some outboard math) to make the decision.

commanders use today. This scenario starts with the "Red" (enemy) forces attempting to deny the "Blue" (friendly) forces bridgehead at Weston and Easton bridge. Initially, Blue had decided to use the 2nd and 3rd battalions to support the advantage gained at Easton bridge. This would allow Blue to capture Fairview and possibly Clifton. However, at that point, Red's 5th battalion breaks through Blue's position at Weston bridge and threatens a flanking

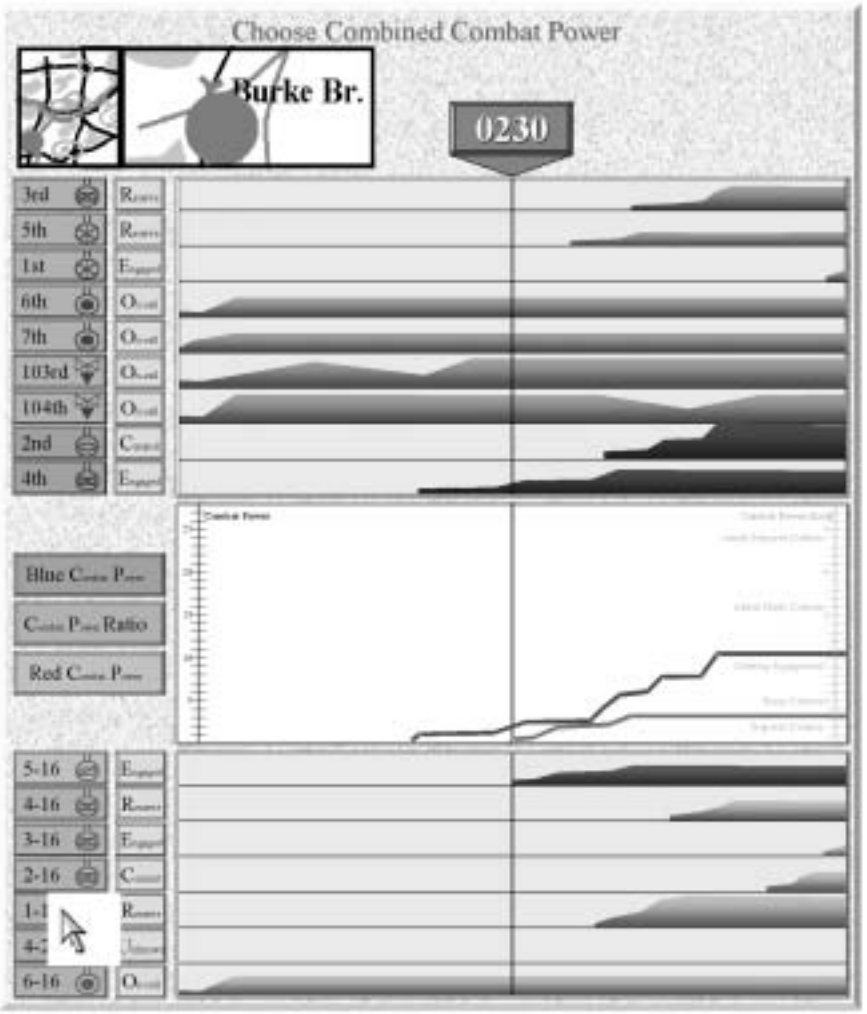


Figure 5.5: JOE visualization showing combat power of Red and Blue forces at the time that the 2nd and 4th Blue battalions and 5th Red battalion reach the "choke point" south of the Burke bridge.

attack from the west. At this point the Blue commander recognizes the need to interdict the Red 5th battalion that is pursuing the Blue 4th battalion. The commander considers using the 2nd Armor to interdict the Red 5th battalion at the "choke point" south of the Burke bridge.

To evaluate whether the 2nd Armor is sufficient to achieve the mission goal, the commander brings up the "choose combat power" display. The commander selects the choke point south of the Burke bridge as the designated location

Table 5.5: Mapping of Graphical Elements to DRs for the "Choose Combat Power" Display

Decision and Information Requirements	Visualization Needs	Graphical Elements
DR 1a – "Determine the point in time when the enemy will reach a specified point in space, and monitor the enemy's combat power over time." IR 1a.1 – "Expected arrival time of enemy combat resources at the specified point in space." (i.e., the lead unit, as well as other follow-on units) IR 1a.2 – "Estimated measure of combined enemy combat power at the specified point in space, beginning at the arrival time of the first enemy unit and extending through follow-on units."	Visualize the time required for enemy unit(s) to reach location designated by commander, and visualize cumulative enemy combat power at that location as a function of time.	
DR 1b – "Choose among the friendly combat resources that can bring their combat power to bear at the specific point in space and time." IR 1b.1 – "The time required for selected friendly combat resources to reach the specified point in space." IR 1b.2 – "Estimated measure of combined combat power of the selected friendly combat power resources once they reach the specified point in space."	Visualize the time required for friendly "combat power" unit to bring their combat power to bear on the designated point (in space and time) and visualize cumulative friendly combat power at that location as a function of time.	
DR 1c – "Estimate the potential to defeat the enemy after the application of the chosen power at the specific point in time and space." IR 1c.1 – "Measure of combat power ratio of friendly to enemy combat power beginning with the arrival of the first unit (friendly or enemy) over time." IR 1c.2 – "Indication of combat power ratios required to defeat the enemy under different battle conditions (i.e., doctrinal / procedural referent information)." IR 1c.3 – "Location of alternative resources of both friendly and enemy combat power that could be brought to bear." IR 1c.4 – "The time required to bring to bear the combat power of these alternative friendly and enemy combat resources." IR 1c.5 – "Measures of cumulative combat power of both friendly and enemy resources as additional friendly and enemy resources are selected (over a specified window in time)."	Visualize relative friendly-to-enemy combat power ratio at the designated point in space as a function of time and compare to combat power ratio required to defeat enemy under different battle conditions. Visualize uncertainty of estimates of Combat Power Ratio. Visualize changes in enemy and friendly combat power and combat power ratio as selected enemy and/or friendly units are added or removed.	

of the engagement and the 2nd and 4th Blue battalions and the 5th Red battalion as the units involved in the engagement. JOE computes the time it will take for these units to reach this designated location and the combat power that the Red and Blue sides will be able to bring to bear at that location over time.

Figure 5.5 presents the resulting JOE display. The display has three major areas. The top third of the display shows all the Blue units that could potentially be deployed, their current status, the time at which they would reach the designated location, and the combat power they could bring to bear over time. The shaded areas represent amount of combat power as a function of time. The horizontal axis represents time, with the left most representing the current time. The heights of the shaded areas represent the amount of combat power

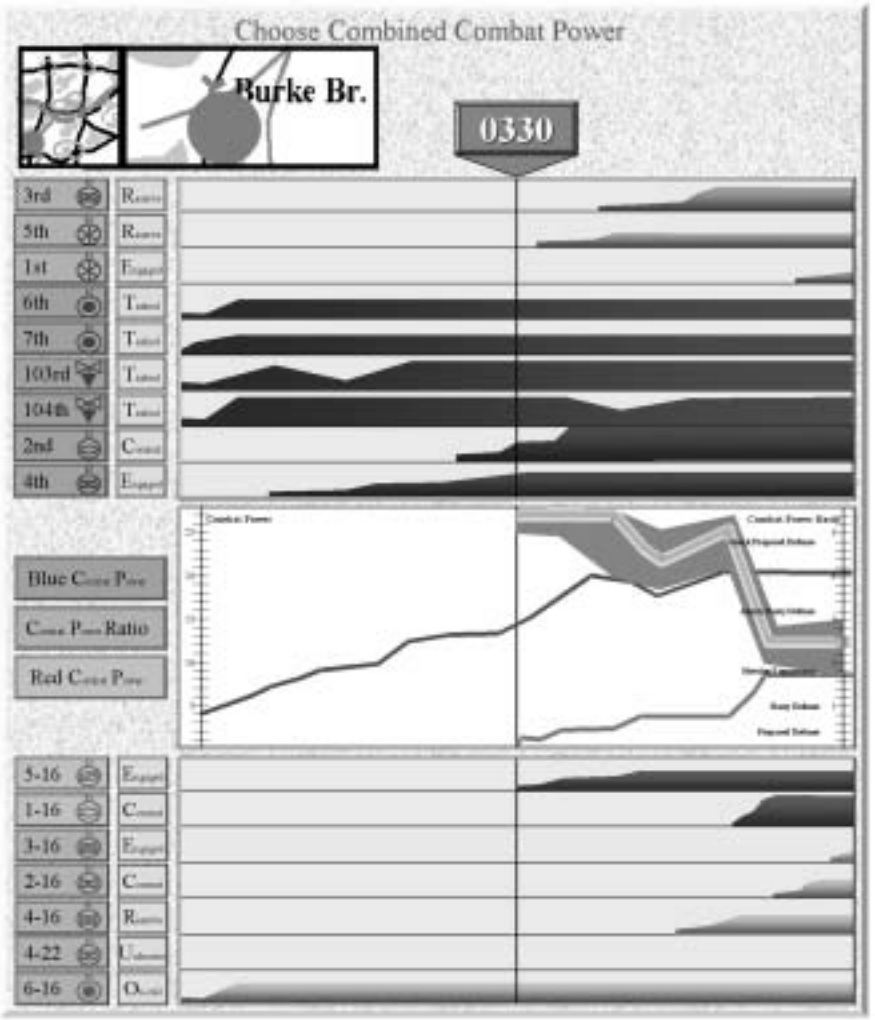


Figure 5.6: Updated JOE display reflecting the changes in relative combat power based on additional units selected. The thick line with bands around it illustrates a feature in JOE to display estimated combat power ratio (Blue/Red with bands of uncertainty around it).

at a given point in time. The units that are highlighted (appear darker) are the units that the commander has selected. Similarly the bottom third of the display shows the Red units and their combat power, with the highlighted unit (5th Battalion) indicating the Red unit selected by the commander. The middle third of the display is used to present the cumulative combat power of the com-

bined Red and (separately) combined Blue units selected as a function of time. Thus, the higher curve in the center area represents the combined combat power of the 2nd and 4th Blue battalions over time. Combat power increases significantly when the 2nd Battalion reaches the designated location. The vertical line indicates the estimated time of arrival of the first Red unit. These three major areas map directly to the information and DRs contained within the DTD, as indicated in Table 5.5.

The JOE display can be used to visually compare the combat power of the Blue and Red units at the time that the Red unit first reaches the designated location. The graphic presentation of combat power as a function of time makes it visually apparent that the Red 5th battalion will reach the designated point before the Blue 2nd battalion can get there. As a result, the difference in combat power between the Blue and Red units at the time of the initial engagement will be small. Therefore the goal of defeating the Red units is unlikely to be achieved.

Figure 5.6 illustrates two additional features of JOE. First, JOE can be used to explore alternative combat resource choices by adding or removing enemy and friendly units to assess the impact it would have on combat power at a given point in time and space. Second, Joe can be used to display combat power ratio of "friendly" to "enemy" forces directly as well as bands of uncertainty around the estimate.

Figure 5.6 illustrates the change to the display when the commander decides that the Red first armor battalion is also likely to join the engagement. The commander decides that additional Blue resources (artillery and helos) will need to be brought in. The commander selects these additional resources and the JOE display is updated to reflect the changes in Red and Blue combat power that occur as a result of introducing these additional units. The commander also requests that combat power ratio be visually displayed. The thicker line in the center of Figure 5.6, an estimate of combat power ratio (Blue/Red) also represents the uncertainty band around the estimate of combat power ratio. A combat power ratio scale appears on the right with indications of combat power ratios recommended for different types of engagements based on conventional military guidance. As JOE makes visually apparent, in the new set of choices, there is enough Blue combat power brought to bear to make it highly probable that the Blue forces will achieve their mission.

As mentioned earlier, while this paper has focused on the "choose combat power" display, the display is intended to illustrate one of a suite of displays that would encompass a complete battle command DSS. As is suggested by the menu that appears in Figure 5.3, additional displays would address other aspects of battle command such as selection of routes and monitoring of the battle.

In addition, display navigation mechanisms would be provided to enable the commander to more fully understand the basis for the combat power values. For example the commander should be able to view the factors and values that contributed to combat power calculations and make changes to the values and

factor weightings as judged appropriate. This is consistent with principles for design of effective decision aids that include the importance of making the basis of recommendations transparent to the user (the principle of decomposability) and the importance of enabling the user to direct the decision aid (the principle of directability) described in Roth, Malin, and Schreckenghost (1997).

Mechanisms would also be provided to enable the commander to assess the impact of selecting particular units to join an engagement on the achievement of other goals. Specifically, the commander should be able to select a unit being considered for a particular engagement, be shown the current (or other planned) commitments for this unit, and the potential impact on achieving those goals if the unit were to be reassigned. This supports the decision requirement to be able to assess the side effects of decisions, identified in Table 5.2.

5.4 DISCUSSION

The "choose combat power" visualization and the JOE rapid prototype of the display concept provide a concrete illustration of how a structured, principled methodology can systematically transform the problem from an analysis of the demands of a domain to identifying effective decision-aiding concepts. The process of generating intermediate design artifacts can be used to build a traceable bridge from cognitive analysis to design. The process of developing the JOE display concept began with cognitive analysis of the domain. A functional abstraction hierarchy was developed that captured the goals in the domain, the means available for achieving them, and the goal constraints and interactions inherent in the domain. The abstract concept of "combat power" and the central role it plays in choosing resources to bring to bear and accomplish a military mission emerged out of this analysis. A second insight that emerged out of the analysis was the importance of representing combat power as a function of time. It takes time for resources to be assembled and moved to a specified location. One of the breakthrough insights in developing the "choose combat power" display was the realization that such a visualization would allow commanders to explicitly manage the time of the culminating point of the engagement. These insights provided the basis for development of the "choose combat power" display concept and the JOE prototype. JOE enables domain practitioners to visualize "combat power" as a function of time and to manipulate it directly. It provides a clear example of an "ecological" interface in that it translates an abstract, functional concept (relative combat power as a function of time), into a concrete visualization that can be apprehended perceptually.

The "choose combat power" case study illustrates the value of performing a multilayered, work domain analysis based design process for generating powerful visualization concepts. First, the process of developing the FAH representation was central to the identification of the key abstract concept. This

concept would not have been recognized without developing the FAH representation, as evidenced by typical physical map-based displays in widespread existence. Second, the process of defining the DRs, IRs, and display task descriptions (intermediate artifacts created to establish a traceable link between the analysis of domain demands and the display concept that emerged) serve as critical thought/design steps in the process. They force the analyst to revisit the FAH and consider issues possibly overlooked, be explicit about the cognitive tasks associated with the nodes in the model, and consider the often complex transformation between data and information. As a whole, they successfully serve to "bridge the gap" between the underlying cognitive analysis and resulting DSS design.

In this paper, the steps in the cognitive systems engineering process are presented as if they are performed in a strictly sequential order. First domain analysis, then DRs analysis, then IRs analysis, etc. It is presented this way for expository simplicity. In practice the process is much more parallel, opportunistic, and iterative in nature than presented here. For example, it is not unusual for an initial visualization idea to emerge before the complete DRs it is intended to support or the supporting information needs have been clearly articulated. The order in which the artifacts are produced is not as important as the fact that all artifacts are eventually produced that provide a functional description of the cognitive and decision tasks that the display is intended to support, and the information and display elements that provide the required support. As mentioned previously, the multiple spans of the bridge are constructed by the process of generating these artifacts, not the artifacts themselves. The key point is that it is the process of generating these artifacts that forces the cognitive analysts to think about the problem in a systematically transformed manner and capture the evolving requirements in a manageable sequence of steps.

The generation of intermediate artifacts that model the structure of the work domain, the demands and cognitive activities to be supported, the IRs for these cognitive activities, and decision support concepts designed to provide this support are needed to provide a traceable link from analysis, to design requirement, to display concept. The approach outlined in this paper offers a means for using a model of the underlying, fundamental behavioral characteristics of the work domain in a principled manner to generate well-grounded decision support concepts for the cognitive demands facing the human-machine decision-making team. This type of approach is essential to bridge the gap between a cognitive analysis of the work domain and the development of innovative decision aids for "envisioned world" types of problems to provide highly effective and robust decision-making performance.

5.5 ACKNOWLEDGMENT

The authors would like to recognize the support of DARPA's Command Post of the Future project for funding the design and development of the Joint Operational Environment under a contract to Logica Carnegie Group (D88D–17–99–C–0047). Specifically, we would like to thank Ward Page and Tom Cook for their vision of the utility of cognitive systems engineering to the development of command and control systems. We would also like to recognize Bridagier General Keith Holcomb, Colonel Larry Shattuck, and Major Jeff From for their support and military expertise in the development of JOE. Finally, we would like to acknowledge the intellectual contribution of Emily Patterson and David Woods with whom we have had many stimulating discussions. We also wish to thank Gary Klein and Kim Vicente for their thoughtful review and constructive comments of an earlier draft of this chapter.

REFERENCES

Bennett, K. B., & Flach, J. M (1992). Graphical displays. Implications for divided attention, focused attention, and problem solving. *Human Factors*, *34*, 513–533.

Carroll, J. M., Kellogg, W. A., & Rosson, M. B. (1991). The task-artifact cycle. In J. M. Carroll (Ed.), *Designing interaction: Psychology at the human-computer interface* (pp. 74–102). Cambridge, England: Cambridge University Press.

Chalmers, B. A., Easter, J. R., & Potter, S. S. (2000, June). Decision-centred visualisations for tactical decision support on a modern frigate. In *Proceedings of 2000 Command and Control Research and Technology Symposium, Making Information Superiority Happen*. Monterey, CA: Naval Postgraduate School.

Cook, R. I., Woods, D. D., Walters, M., & Christoffersen, K. (1996, August). Coping with the complexity of aeromedical evacuation planning: Implications for the development of decision support systems. In *Proceedings of the 3rd Annual Symposium on Human Interaction with Complex Systems*, Dayton, OH: IEEE.

Dekker, S., & Woods, D. D. (1997, October). *The envisioned world problem in cognitive task analysis*. Paper presented at the ONR/NATO Workshop on Cognitive Task Analysis, Washington, DC

Gualtieri, J. W., Roth, E. M., & Eggleston, R. G. (2000). Utilizing the abstraction hierarchy for role allocation and team structure design. In *Proceedings of HICS 2000—5th International Conference on Human Interaction with Complex Systems* (pp. 219–223). Urbana-Champaign, IL: IEEE.

Lind, M. (1991). Representations and abstractions for interface design using multilevel flow modelling. In G. R. S. Weir & J. L. Alty (Eds.), *Human-computer interaction and complex systems.* London: Academic Press.

Lind, M. (1993, July). Multilevel flow modeling. *AAAI93 Workshop on Reasoning about Function*, Washington, DC.

Logica Carnegie Group: (2000a). *Command Post of the Future: Cognitive modeling and engineering* (1999 Year End Report). Aberdeen Proving Ground, MD: U.S. Army Research Laboratory, Human Research and Engineering Directorate.

References

Potter, S. S., Ball, R. W., Jr., & Elm, W. C. (1996, August). Supporting aeromedical evacuation planning through information visualization. In *Proceedings of the 3rd Annual Symposium on Human Interaction with Complex Systems* (pp. 208–215). Dayton, OH: IEEE.

Potter, S. S., McKee, J. E., & Elm, W. C. (1997). *Decision centered visualization for the military capability spectrum project* (Unpublished technical report). Pittsburgh, PA: Carnegie Group.

Potter, S. S., Roth, E. M., Woods, D. D. & Elm, W. (2000). Bootstrapping multiple converging cognitive task analysis techniques for system design. In J. M. C. Schraagen, S. F. Chipman, & V. L. Shalin (Eds.), *Cognitive task analysis*. Mahwah, NJ: Erlbaum.

Potter, S. S., Woods, D. D., Hill, T., Boyer, R. L., & Morris, W. S. (1992, October). Visualization of dynamic processes: Function-based displays for human-intelligent system interaction. In *Proceedings of the 1992 IEEE International Conference on Systems, Man, and Cybernetics* (pp. 1504–1509). Chicago, IL: IEEE.

Rasmussen, J. (1986). *Information processing and human-machine interaction: An approach to cognitive engineering.* New York: Elsevier.

Rasmussen J., Pejtersen, A. M., & Goodstein, L. P. (1994). *Cognitive systems engineering.* New York: Wiley.

Reising, D. V., & Sanderson, P. (1998). Designing displays under ecological interface design: Towards operationalizing semantic mapping. In *Proceedings of the Human Factors and Ergonomics Society 42nd Annual Meeting* (pp. 372–376). Santa Monica, CA: HFES.

Roth, E. M., Lin, L., Thomas, V. M., Kerch, S., Kenney, S. J., & Sugibayashi, Nubuo (1998). Supporting situation awareness of individuals and teams using group view displays. In *Proceedings of the Human Factors and Ergonomics Society 42nd Annual Meeting* (pp. 244–248). Santa Monica, CA: HFES.

Roth, E. M., Lin, L., Kerch, S., Kenney, S. J., & Sugibayashi, N. (In press). Designing a first-of-a kind group view display for team decision making: A case study. In E. Salas & G. Klein (Eds), *Linking expertise and naturalistic decision making.* Mahwah, NJ: Erlbaum.

Roth, E. M., Malsch, N., Multer, J., & Coplen, M. (1999). Understanding how train dispatchers manage and control trains: A cognitive task analysis of a distributed planning task. In *Proceedings of the Human Factors and Ergonomics Society 43rd Annual Meeting* (pp. 218–222). Santa Monica, CA: HFES.

Roth, E. M., Malin, J. T. , & Schreckenghost, D. L. (1997). Paradigms for intelligent interface design. In M. Helander, T. Landauer, & P. Prabhu (Eds.), *Handbook of human-computer interaction* (2nd ed., pp. 1177–1201). Amsterdam: North-Holland.

Roth, E. M., & Mumaw, R. J. (1995). Using cognitive task analysis to define human interface requirements for first-of-a-kind systems. In *Proceedings of the Human Factors and Ergonomics Society 39th Annual Meeting* (pp. 520–524). Santa Monica, CA: HFES.

Rouse, W., & Morris, N. (1986). On looking into the black box: Prospects and limits in the search for mental models. *Psychological Bulletin*, *100* (3), 349–363.

Schraagen, J.M.C., Chipman, S.F., & Shalin, V.L. (Eds.) (2000). *Cognitive task analysis*. Mahwah, NJ: Erlbaum.

Simon, H. (1981). *The sciences of the artificial*. Cambridge, MA: MIT Press.

Smith, P., Woods, D. D., McCoy, E., Billings, C., Sarter, N., Denning, R., & Dekker, S. (In press). Using forecasts of future incidents to evaluate future ATM system designs. *Air Traffic Control Quarterly.*

Vicente, K. (1999). *Cognitive work analysis.* Mahwah, NJ: Erlbaum.

Vicente, K. J., Christoffersen, K., & Hunter, C. N. (1996). Response to Maddox critique. *Human Factors, 38* (3), 546–549.

Vicente, K. J., & Rasmussen, J. (1990). The ecology of human-machine systems II: Mediating direct perception in complex work domains. *Ecological Psychology, 2,* 207–249.

Vicente, K., & Rasmussen, J. (1992). Ecological interface design: Theoretical foundations. *IEEE Transactions on Systems Man and Cybernetics, 22*, 589–606.

Woods, D. D. (1988). *The significance messages concept for intelligent adaptive data display* (Unpublished technical report). Columbus, OH: The Ohio State University.

Woods, D. D. (1995). Toward a theoretical base for representation design in the computer medium: Ecological perception and aiding human cognition. In J. Flach, P. Hancock, J. Caird, & K. Vicente (Eds.), *Global perspectives on the ecology of human-machine systems* (pp. 157–188). Hillsdale, NJ: Erlbaum.

Woods, D. D. (1998). Designs are hypotheses about how artifacts shape cognition and collaboration. *Ergonomics*. *41*, 168–173.

Woods, D. D., & Hollnagel, E. (1987). Mapping cognitive demands in complex problem-solving worlds. *International Journal of Man-Machine Studies, 26*, 257–275.

Woods, D. D., & Roth, E. M. (1988). Cognitive engineering: Human problem solving with tools. *Human Factors, 30* (4), 415–430.

6 Cognitive Work Analysis for Air Defense Applications in Australia

NEELAM NAIKAR
Defence Science and Technology Organisation

GAVAN LINTERN
Defence Science and Technology Organisation

PENELOPE SANDERSON
Swinburne University of Technology

ABSTRACT

Cognitive Work Analysis (CWA; Rasmussen, Pejtersen & Goodstein, 1994; Vicente, 1999) is most familiar to cognitive engineers from its successes in the area of interface design. In this report, we describe how we have used CWA in a variety of other contexts at the Defence Science and Technology Organisation (DSTO) in Australia. First, we describe the five analytic techniques of CWA, and we show how CWA can be used throughout a system's life cycle, from requirements definition to system retirement. Second, we provide specific examples of projects from the air defense domain in which we have used CWA. These projects include (1) evaluating alternative designs for Airborne Early Warning and Control (AEW&C) aircraft, (2) evaluating human-system integration solutions for AEW&C, (3) identifying training needs for F/A–18 pilots and developing functional requirements for a training system that meets those needs, and (4) designing information work spaces for command and control. These examples give strength to the argument that CWA can be used just as effectively in areas other than interface design where the professional contribution of cognitive engineers is required.

6.1 BACKGROUND TO CWA

6.1.1 Introduction

Cognitive Work Analysis (CWA) is a systems-based approach for analyzing, modeling, designing, and evaluating complex sociotechnical systems (Rasmussen, Goodstein, & Pejtersen, 1994; Vicente, 1999). CWA is therefore one of the many different techniques for analyzing work that cognitive systems engineering can call upon. The specific focus of CWA, though, is to support operator adaptation and flexibility in complex, real-time work domains, especially during unanticipated contingencies. CWA therefore appears particularly well suited to air defense domains, where an opponent will often wish to create unanticipated situations to achieve a tactical advantage.

CWA aims to support operator adaptation to unexpected circumstances by designing interfaces that are based, not on *typical* or *optimal* work patterns, but on the *constraints* that shape the work patterns in the first place. During novel situations, workers usually cannot rely on work procedures that were planned in the context of typical or expected situations. Rather, to prevent system failure, workers must adapt their behavior to the particular situation at hand, without crossing the boundaries on safe and effective operation in that work domain. Thus, in designing computer-based interfaces, designers should make visible the constraints or boundary conditions on safe and effective operation in that work domain. Workers are then more likely to respond appropriately when something unusual happens.

The concept of constraints is therefore critical to CWA. Constraints are factors that shape workers' activity by imposing limits as well as offering possibilities for safe and effective action. In Figure 6.1, activity is seen as a region of possible action trajectories in the center of the diagram while some of the key factors that place constraints on safe and effective action are shown around the edges. In most work systems, there are a large number of possible action trajectories that do not violate the constraint-based boundaries of that work space. Thus, an analysis of the specific sequences of behavior that should happen or typically happen in a work space is bound to be incomplete. Rather, activity is more robustly described by the constraints that shape the action trajectories.

CWA consists of five modeling techniques that are tailored for analyzing constraints. Figure 6.2 summarizes the five analytic techniques of CWA, the information each technique provides, and the form of the analytic product in which the information is delivered. In the following sections, we provide more detailed descriptions of the five analytic techniques of CWA.

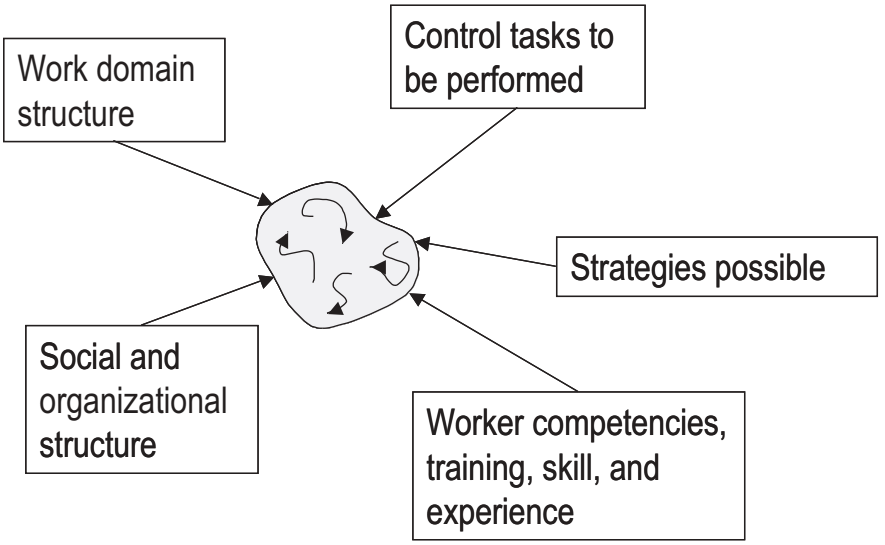


Figure 6.1: Five general areas of constraints (boxes) work together to shape possible and effective action trajectories (arrows in center area).

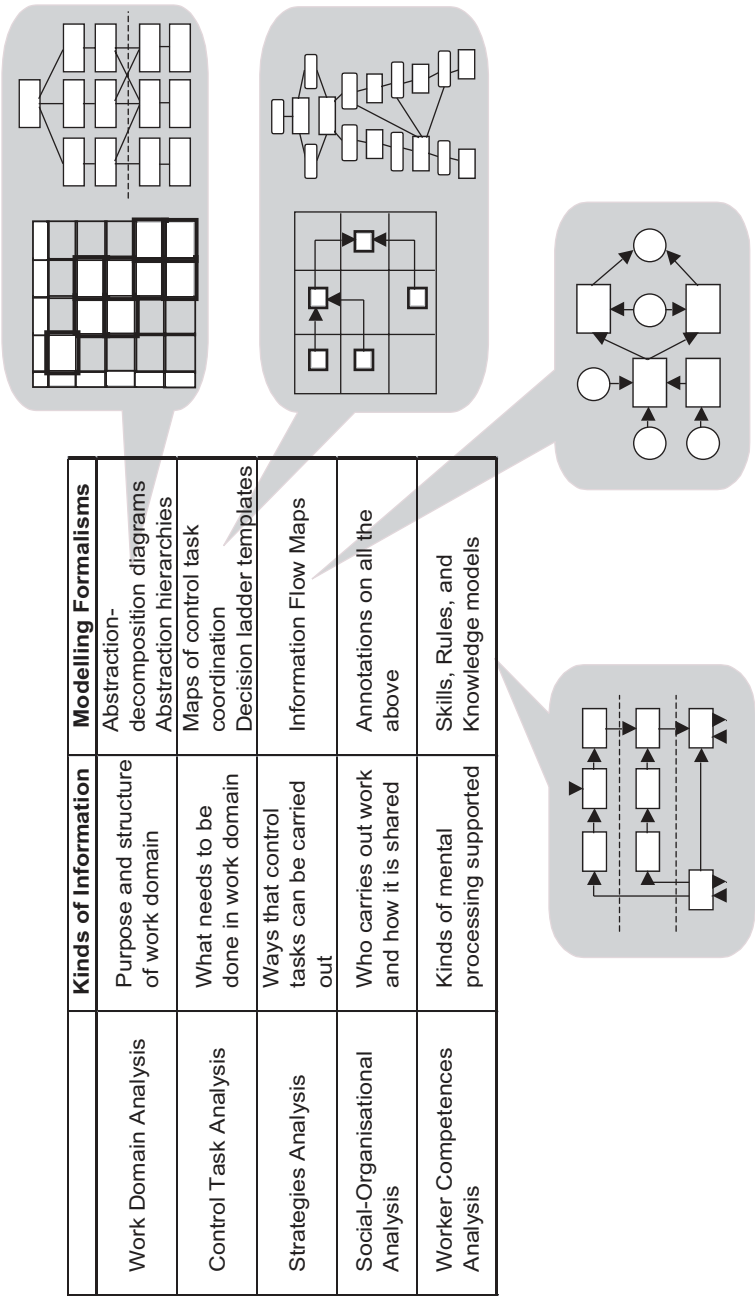
6.1.2 Work Domain Analysis

Work domain analysis models the purposive and physical constraints of the system in which activity take places. These constraints include the (1) functional purposes or high-level objectives of the system, (2) priorities and values that are preserved during system operation, (3) purpose-related functions that are executed and coordinated to achieve system goals, (4) physical functions, such as the functionality afforded by the physical devices of the system, and (5) physical form, such as the physical devices themselves. These constraints are typically described in an abstraction hierarchy or within an abstraction-decomposition space.

The constraints identified by work domain analysis are event-independent. This means that the functional purposes, priorities and values, and purpose-related functions of a system, as well as its physical resources (physical function and physical form), are relevant across a broad range of scenarios or situations, including unanticipated events.

6.1.3 Control Task Analysis

Control task analysis focuses on what needs to be done in a work domain for a system to achieve its functions and objectives. In particular, this analysis identifies activities having to do with information gathering and situation analysis,



	Kinds of Information	Modelling Formalisms
Work Domain Analysis	Purpose and structure of work domain	Abstraction-decomposition diagrams Abstraction hierarchies
Control Task Analysis	What needs to be done in work domain	Maps of control task coordination Decision ladder templates
Strategies Analysis	Ways that control tasks can be carried out	Information Flow Maps
Social-Organisational Analysis	Who carries out work and how it is shared	Annotations on all the above
Worker Competences Analysis	Kinds of mental processing supported	Skills, Rules, and Knowledge models

Figure 6.2: Five phases of CWA with iconic representations of their most familiar analytic products.

hypothesis generation and testing, planning, and execution. The decision ladder template is most often used for modeling control tasks. Later in this paper, we also discuss our extension of control task analysis, which we have called *Temporal Coordination Control Task Analysis*, because it shows the constraints on the coordination of activities over time.

6.1.4 Strategies Analysis

Strategies analysis focuses on ways of performing control tasks. Typically, there are several strategies for performing a single control task. For example, an electrician could use either a pattern-recognition strategy or a decision-table strategy to diagnose problems with faulty equipment. The two strategies require very different kinds of cognitive resources; the former strategy relies on the expertise of the electrician whereas the latter strategy relies on the expertise of the person who developed the decision table. The modeling template most often used for strategies analysis is an information flow map.

6.1.5 Socio-Organizational Analysis

Socio-organizational analysis is concerned with how work is allocated among human workers and intelligent agents. The allocation of work might be across different parts of the work domain, control tasks, or strategies. For example, the strategy of using a decision table in electronic troubleshooting might be distributed across human workers and automation; a human worker enters the symptoms of faulty equipment into a computer that responds with a list of potential faults and options for repair. The abstraction hierarchy or abstraction-decomposition space, decision ladders, and information flow maps can all be used as templates for work allocation.

6.1.6 Worker Competencies Analysis

Worker competencies analysis focuses on the competencies (e.g., knowledge, skills) that workers need for carrying out the work of the system. Hence, it is only at this phase of the CWA framework that the particular constraints of human workers are considered, because the constraints identified in the initial phases of CWA will affect the analysis of competencies. For example, if a control task is allocated purely to machine automation, then human workers will not require skills for performing this control task. Rasmussen's (1986) skills, rules, and knowledge taxonomy can be used for matching work demands to human capabilities and limitations.

6.2 CWA AND THE SYSTEM LIFE-CYCLE

Many researchers and human factors practitioners have commented that carrying out a full CWA is a daunting task. However, we argue, first, that there are many investigations where this level of effort is warranted and, second, that the products of CWA can be put to multiple uses throughout the system life-cycle. Hence, CWA ultimately provides benefits that are far greater than the initial investment in time and resources (Sanderson, Naikar, Lintern, & Goss, 1999). Another paper that echoes the same point is Leveson (2000) in which she discusses the application of CWA-based ideas to the construction of a design rationale for software systems.

Figure 6.3 illustrates some of the stages in the system life-cycle where one or more CWA analytic products may be useful. The figure shows the five phases of CWA in its columns and the different stages in the system life-cycle in its rows. Cells describe how the products of CWA modeling may contribute at each point in the system life-cycle. Some of the cells have been filled in with uses, but this does not mean that other uses might not be found at the same point. Other cells have been left empty, but this does not mean that they have nothing to contribute to the stage of the system life cycle indicated. We expect that with further use of CWA across different contexts, we will be able to flesh out this table with more examples.

In the remainder of this section we briefly review the stages of the system life-cycle, and we outline how CWA might inform each stage. Later in this paper, we provide specific examples of how we have used CWA at different points in the system life-cycle. We focus mostly on work domain analysis because that is where we have most experience, but this should not be taken to belittle the actual and potential contributions of other phases.

6.2.1 Requirements

At the outset of developing a system, work domain analysis is the primary framework for identifying requirements. It helps analysts think about why a new system should exist, what functions it should implement, and what physical devices are necessary. In addition, work domain analysis provides a framework for putting requirements into context, for example, by indicating whether the requirements relate to the purposes and priorities of the system, or whether the requirements relate to the physical devices of the system. For our work on evaluating designs for AEW&C, we developed a work domain analysis that was essentially constructed from requirements documents (Naikar & Sanderson, 2000a, b).

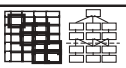
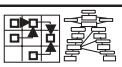
	WDA	CTA	SA	SOA	WCA
				*Annotation of other diagrams*	
	Purpose and structure	Control tasks and coordination	How control tasks carried out	Who does what and with whom	Mental processes entailed (SRK)
Requirements	develop				
Specifications	develop	develop			
Design					
Hardware, software	define				
Control tasks	given	define			
Dialog support		given	define		
Actor roles			given	define	
Interface formats				given	define
Simulation	means/ends	activities	options	agent(s)	ag. properties
Evaluation of design(s)	char/eval/comp	char/eval/comp	char/eval/comp	char/eval/comp	char/eval/comp
Implementation	guide				
Test	judge match	judge perf	judge process	judge roles	judge workload
Operator Selection				guide	guide
Operator Training	guide	guide	guide	guide	guide
Routine use	describe	describe	describe	describe	describe
Non-routine use	describe	describe	describe	describe	describe
Maintenance	describe				
Research (HF studies)	system ID	task contexts	tools, options	human context	support needs
Upgrades	model effects	model effects	model effects	model effects	model effects
System retirement	judge shortfall	judge shortfall			

Figure 6.3: Actual and potential uses of the five phases of CWA over the life-cycle of a complex sociotechnical system. The five phases of CWA have been abbreviated as follows: Work Domain Analysis (WDA), Control Task Analysis (CTA), Strategies Analysis (SA), Socio-Organizational Analysis (SOA), and Worker Competencies Analysis (WCA).

6.2.2 Specification

In the specifications stage, more detail is needed for design to proceed. A system developer needs to know what must be done in a work domain for the system to achieve its functions and purposes. Therefore, the knowledge that a control task analysis provides is important for building specifications for the system. In addition, Leveson (2000) has found that an extension of work domain analysis, to capture the design intent in complex software engineering projects, is useful for developing "intent specifications."

6.2.3 Design

As outlined in Vicente (1999), design has been classified into five general stages that reflect what each of the five phases of CWA offers. Hardware and software needs (models, databases, sensors, etc) can be described principally by work domain analysis, even though it may be guided by information from other phases. Control tasks can be identified by control task analysis, provided we know the work domain constraints (thus "given" under work domain analysis).

Dialogue support can be informed by strategies analysis, given we know what the control tasks are. The definition of actor roles, including allocation of function and coordination structures, can be described with socio-organizational analysis, given we know what the effective strategies are. Finally, the format of interfaces and visual displays can be informed by worker competencies analysis, given we know what the work allocation is.

6.2.4 Simulation

Simulation takes several forms. First, simulation refers to modeling events and the system's response to events. This will not be discussed in detail here but an example of control task analysis providing the foundation for agent-based simulation can be found in Sanderson et al. (1999). Second, simulation refers to the development of full- or part-task simulators for supporting training, further system development, etc. Later in this chapter, we will show how work domain analysis can be used for defining the requirements of large-scale training simulators (see also Lintern & Naikar, 2000; Naikar & Sanderson, 1999).

6.2.5 Evaluation of Designs

CWA provides us with tools for evaluating different designs during the development of a system. Work domain analysis, in particular, has a unique "summarizing" role in this respect. With this framework, alternative designs may be evaluated in terms of how well the technical solution (physical form and physical function) supports the purpose-related functions, priorities and values, and functional purposes of the work domain (Naikar & Sanderson, 2000a, b). In addition, control task analysis can be used as a backdrop for evaluating the human-system integration solutions of alternative design proposals (Sanderson & Naikar, 2000). Both of these projects will be discussed in more detail later in this paper.

6.2.6 Implementation

CWA products have the potential to guide the implementation process. In particular, they provide reference documents to consult while a design is being realized. For example, Leveson's (2000) addition of a "refinement" dimension to the abstraction-decomposition space is a recognition of how CWA products might guide implementation while at the same time evolving with it.

6.2.7 Test

As implementation proceeds, CWA can guide the testing process, particularly if a part- or full-scale simulator is available. Work domain analysis has an important role to play here. Groupings of system functions and physical elements, at their present stage of development, can be tested in terms of (1) whether they achieve their collective ends (the abstraction dimension), and (2) whether they function effectively as wholes (the decomposition dimension). Thus, work domain analysis allows testing of whether the functional structure of the system that emerges from implementation matches the analysis that was initially drawn up. Similarly, to the extent that a simulator or field context makes possible, CWA also allows testing of whether the system under development supports the necessary control tasks, strategies, role allocation and coordination structures, and operators' cognitive capabilities.

6.2.8 Operator Selection

Operator selection can be informed by the competencies that workers require to carry out the work of the proposed system. Thus, worker competencies analysis is highly relevant to this stage of the system life-cycle. However, because worker competencies are affected by how work has been allocated across human and machine agents, socio-organizational analysis also has a role to play here.

6.2.9 Operator Training

CWA can inform training in powerful ways that focus more on satisfying the functional purposes of a work domain by adapting behavior rather than by evoking procedures (Lintern & Naikar, 1999; Naikar & Sanderson, 1999; Naikar, Sanderson, & Lintern, 1999). Thus, work domain analysis identifies training needs in terms relating to the essential functional structure of the work domain rather than to specific trajectories of behavior. Similarly, control task analysis identifies training needs as a set of "problems to solve" rather than as specific steps for solving problems. strategies analysis identifies training needs as multiple strategies that workers can use for performing control tasks rather than as an idealized or one best strategy. Socio-organizational analysis identifies training needs relating to a configurable work allocation structure rather than to a fixed structure. Finally, worker competencies analysis identifies training needs in terms of different levels of mental processes rather than a single, fixed level.

6.2.10 Routine, Nonroutine, and Maintenance Activity

CWA can be used to describe routine, nonroutine, and maintenance activity within a system. Indeed, much of Rasmussen's work while developing the modeling techniques used in CWA was descriptive in nature (Rasmussen et al., 1994). Descriptions of different kinds of activity can be thought of as animations and annotations on the products of CWA modeling, to show, for example, how a problem-solving sequence might be traced as a trajectory over the work domain, a chain of control tasks, a choice of a particular strategy, an interaction between different agents, or an exercise of certain cognitive competencies.

6.2.11 Research (HF Studies)

CWA can inform the design and operationalization of research programs in important ways (see Vicente, 1999). We have started to use CWA to ensure the representative design of experiments examining crew coordination and display design. For example, temporal coordination control task analysis gives us a profile of the control tasks within a phase of operation, and the temporal, logical, and structural coordination between those tasks. The temporal coordination control task analysis therefore sketches the "ecology" of the work environment that must be recreated or simulated in a laboratory study so that the empirical investigation is conducted in a representative setting.

6.2.12 Upgrades

CWA provides a framework for describing and predicting the impact of technological change. Using examples from elevator system design and the introduction of automated charting in an anesthesia environment, Benda and Sanderson (1999a, 1999b) have demonstrated that both work domain analysis and control task analysis can be extended to show how changes in physical devices and physical functioning create new constraints or affordances for activity that may have the ultimate effect of changing the nature of the work domain itself. For example, the automated patient record may afford the function of relating patient outcomes to preoperative events. Similar uses of CWA have been found for the F/A–18 upgrade and this will be discussed later in this chapter.

6.2.13 System Retirement

Finally, we envisage that CWA would be useful in making the decision to retire or decommission a system. A work domain analysis of the broader work context may show that a particular system is no longer helpful or competitive in meeting the functional purpose of the work domain. In addition, control task analysis may be useful for revealing shortfalls in the relevance or effectiveness with which control tasks are carried out.

6.2.14 Examples of CWA in Context

Having provided a brief survey of the possible uses of CWA at all stages of the system life-cycle, we now turn to four examples from our own work over the last two years. The four examples describe (1) the use of work domain analysis for evaluating alternative designs for an AEW&C system, (2) an extension of control task analysis for analyzing human-system integration solutions for AEW&C, (3) the use of work domain analysis for identifying training needs of F/A–18 pilots, and for defining functional requirements for a training system that meets those needs, and (4) an outline of how all the phases of CWA can be used for specifying the information needs for command and control. Although we are unable to describe these projects in detail here, we cite our other publications for more information about this work.

6.3 EVALUATION OF DESIGNS

One of our most successful applications of CWA at DSTO involved supporting the Australian Defence Force during the acquisition of a fleet of AEW&C aircraft. This system, which is valued at $3 billion, is being manufactured by Boeing in the United States. When it is delivered to Australia, the primary role of AEW&C will be to conduct surveillance and to coordinate the activity of defense assets in an allocated area of operations. Each aircraft will be equipped with a suite of physical devices including onboard sensors, satellite intelligence links, communications systems, and a work station for up to ten crew members. AEW&C is, therefore, one of the most complex systems to which CWA has been applied to date.

Our use of CWA occurred during the early stages of procurement when a formal evaluation of alternative design proposals for AEW&C was being conducted (Naikar & Sanderson, 2000a, b). Three potential AEW&C designs had been submitted by Boeing, Raytheon E-Systems, and Lockheed Martin and, initially, the AEW&C Project Office had planned to use only two techniques to select the winning design. The two techniques, technical and operational, are

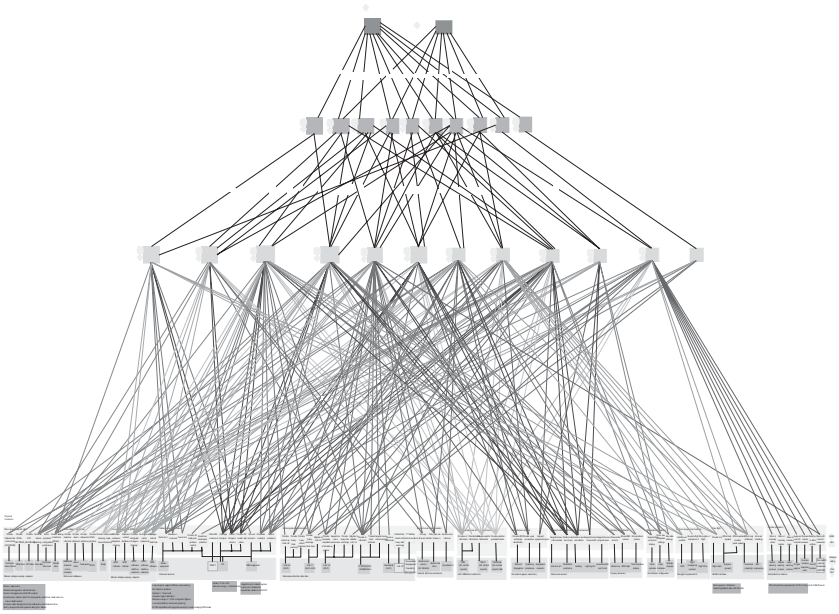
standard systems engineering techniques that are commonly used for procuring large-scale military systems (Department of Defence, 1995, 1999). For the technical evaluation, evaluators were to assess the technical strengths and limitations of each of the physical subsystems of AEW&C, for example, its radar, communications, and navigation systems. For the operational evaluation, evaluators were to use *Monte Carlo* simulation to test how computational models of each of the three designs would perform in six mission scenarios.

During a preliminary evaluation of the three designs, however, the AEW&C Project Office realized that the technical evaluation would produce a series of disparate reports about the physical devices of AEW&C. Thus, a radar report might indicate that design "A" was better than designs "B" and "C," whereas a communications report might indicate that design "B" was better than designs "A" and "C," and so on. The Project Office quickly became very concerned about how they would integrate the recommendations of several reports to reach a final decision about the best AEW&C design.

When this problem was presented to the evaluation team, of which we were members, it struck us that the reason for designing all these physical devices into a single system was to support a distinct set of functions, priorities and values, and purposes. Consequently, we could use work domain analysis to evaluate all physical-device solutions against these high-level functional properties. By comparing all the designs against the same set of functional criteria, it would be easier to select the best overall AEW&C design. After convincing the AEW&C Project Office that work domain analysis could be used to solve the "integration problem," we began to develop this approach to evaluation in the final year of tender evaluation (known as *source selection* in the U.S.).

Our first step was to develop an abstraction hierarchy for AEW&C. We were able to put together an initial representation from reviewing several AEW&C documents. For example, the AEW&C *concept of operations* provided information about the purposive functions of AEW&C (first three layers of the abstraction hierarchy) whereas the AEW&C *system specification* provided information about the physical functions and physical subsystems of AEW&C (last two layers). We then worked with several subject-matter experts, including military personnel and defense scientists, to revise and refine our initial characterization.

Figure 6.4 provides (1) a global view of the AEW&C abstraction hierarchy that we developed, and (2) a sample of functions from each layer of the AEW&C abstraction hierarchy. Using the abstraction hierarchy, the AEW&C evaluation team could trace the impact of physical-subsystem designs on the higher level functions of AEW&C. For example, evaluators could trace that a radar with a long range would allow AEW&C to gather information about entities at a greater distance from the platform, thus enhancing the timeliness with which entities are detected, tracked, and identified. On the other hand, a radar with a long range would also transmit electronic emissions over greater



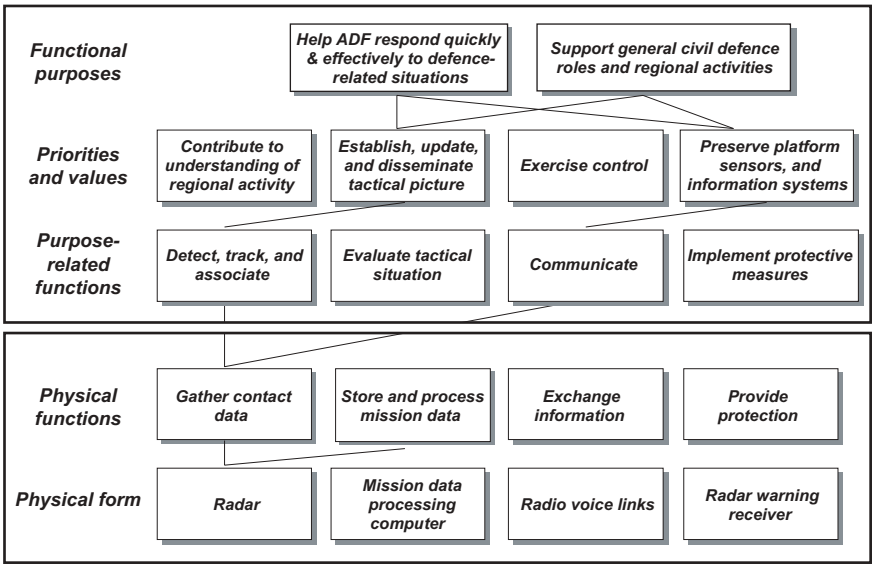


Figure 6.4: (a) A global view of the AEW&C abstraction hierarchy.
(b) A sample of functions from each layer of the AEW&C abstraction hierarchy.

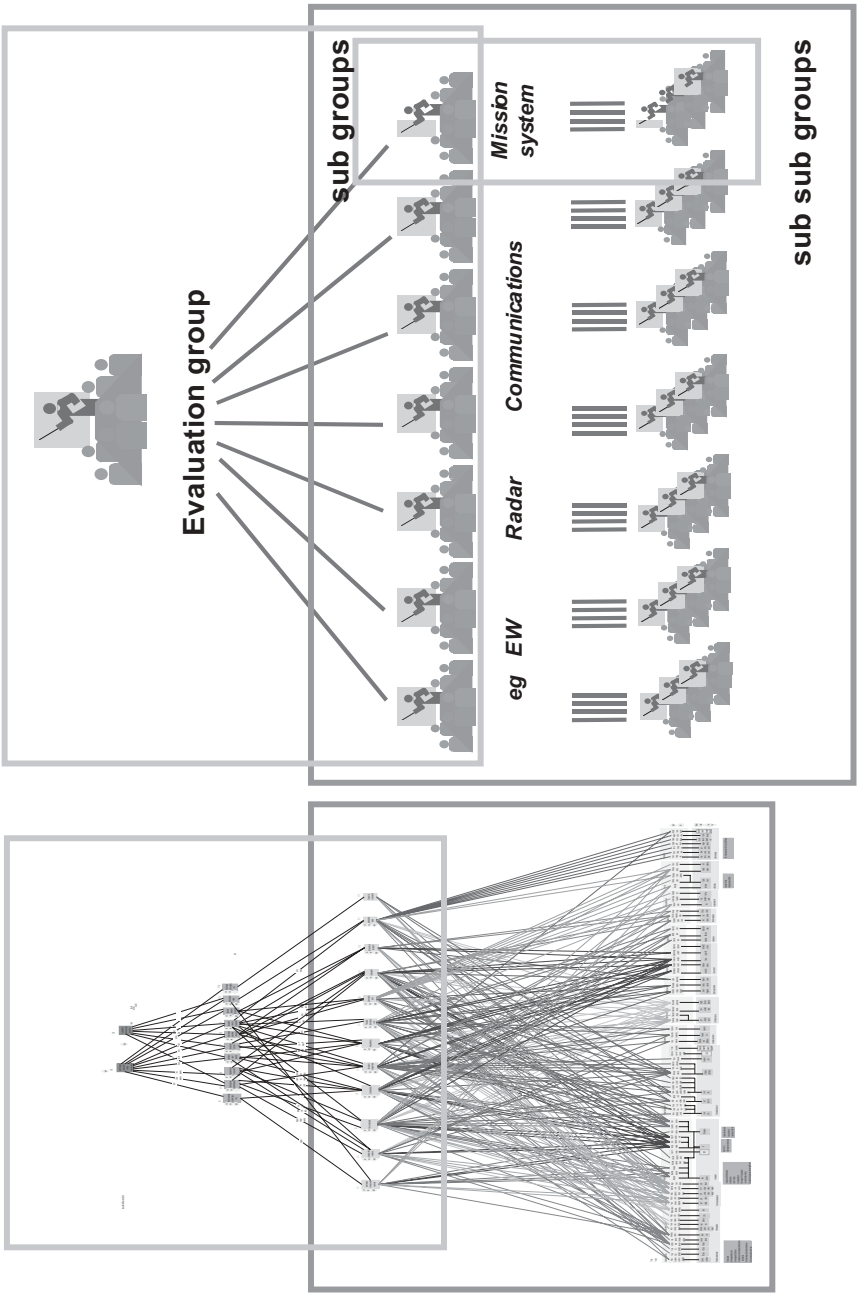


Figure 6.5: Structure of the group responsible for evaluating AEW&C designs.

distances, thereby communicating the presence of the AEW&C platform more broadly, which would compromise its survivability.

As well as constructing the AEW&C abstraction hierarchy, we also developed a process for using work domain analysis to evaluate AEW&C designs. This process took advantage of the structure of the evaluation team that had been set up for the technical evaluation (Figure 6.5). This team had been decomposed into subgroups that were responsible for carrying out a technical evaluation of a set of physical devices. For the work domain analysis-based evaluation, each subgroup evaluated how well the physical-device solutions supported the purpose-related functions of AEW&C. Following this, the leader of the evaluation team and his assistants from each of the subgroups evaluated the impact at the purpose-related functions layer against the priorities, and values, and functional purposes of AEW&C. We note that Sanderson has developed a Microsoft Excel™ spreadsheet for tracking and recording these types of judgments.

The evaluation team did not find this process overly taxing and in a briefing to the Deputy Secretary of the Department of Defence, the AEW&C Project Office singled out work domain analysis for its usefulness to the AEW&C acquisition. As well as solving the "integration problem," the AEW&C Project Office found it useful that they were able to express the results of the evaluation in terms of military utility, such as the purpose-related functions, priorities and values, and functional purposes of AEW&C (rather than in terms of technical properties). In addition, they thought that work domain analysis provided a good "sanity check" because it supported a systematic and explicit evaluation of the three designs.

The work domain analysis also provided a complementary perspective to the standard evaluation techniques that had been used for evaluating AEW&C designs. First, by focusing evaluation on the functional properties of AEW&C, work domain analysis promoted an understanding of how well the designs fulfilled the work requirements of AEW&C. In contrast, the technical evaluation focused on how well the designs would perform as individual technical units. Second, as the functional properties identified by work domain analysis are event-independent (Vicente, 1999), this approach promoted an understanding of how AEW&C designs would perform in a broad range of situations, including those that cannot be anticipated up front. The operational evaluation, on the other hand, focused evaluation on how the designs would perform in a small range of likely mission scenarios.

In conclusion, we recognize that it may be difficult to convince organizations, with well established policies and practices, to adopt novel approaches like work domain analysis. Therefore, we point out that work domain analysis can be used fruitfully on a smaller scale within a particular project. For example, human factors practitioners could use work domain analysis to conduct a comprehensive evaluation of the human engineering solutions of alternative

design proposals. Such bottom-up applications of work domain analysis may help to demonstrate its usefulness to senior managers and decision makers.

6.4 ANALYSIS OF HUMAN-SYSTEM INTEGRATION

In recent years, we have become all too familiar with stories of systems that were designed with little concern for the work of human operators. Thus, one of our key concerns on the AEW&C project was whether the human-system integration (HSI) solutions of competing designs would support the intended activity of AEW&C. Our approach was to develop a description of AEW&C activity that could be used as a background for examining HSI solutions, as details about the designs became available to us (Sanderson & Naikar, 2000).

The framework that we developed for analyzing AEW&C activity—Temporal Coordination Control Task Analysis (TCCTA)—combined the approaches of Rasmussen et al. (1994) and Vicente (1999) for control task analysis, and also included several important extensions. In essence, the TCCTA describes the activity of complex systems in terms of (1) the entire activity context in which control tasks occur; (2) the control tasks themselves, the temporal and logical relations between the control tasks, and the physical and purposive constraints (from the work domain analysis) acting on the control tasks; and (3) the mechanics of the HSI for each control task.

To conduct these analyses for AEW&C, we relied on the same sources of information as for the work domain analysis, namely, various AEW&C-related documents and subject-matter experts. The documents provided general descriptions of the operational role of the AEW&C system, the types of scenarios in which this platform would most likely participate, and the broad responsibilities of crew members. The subject-matter experts made projections of the activity that was necessary by the AEW&C system for it to achieve the goals of the work domain.

6.4.1 AEW&C Activity Context

Figure 6.6 shows that we described the activity context for AEW&C in terms of major classes of work functions (rows) and mission contexts (columns). The major classes of work functions are (1) mission planning and reporting; (2) system setup, configuration, and shutdown, (3) surveillance, and (4) asset control. The major classes of mission contexts are; (1) on ground, not in aircraft; (2) on ground, in aircraft; (3) on way to station; (4) on station; (5) returning to base; (6) on ground, in aircraft; and (7) on ground, not in aircraft.

The activity context for AEW&C captures the concerns of the crew at each phase of mission, and their changing preoccupations as the mission progress-

Figure 6.6: The activity context for AEW&C.

es. For example, mission planning is a preoccupation in the earlier phases of a mission, whereas mission reporting becomes a preoccupation at the later stages. In addition, the AEW&C activity context captures the changing background activity against which various control tasks are performed. For example, the background activity for mission planning when "on ground, in aircraft" involves system setup and system configuration control tasks. However, the background activity for mission planning when "on station" involves system configuration, surveillance, and asset control tasks. It is important to look at the entire activity context, and not just single control tasks or clusters of control tasks, because the same control tasks may require different kinds of HSI solutions depending on the context within which the tasks are performed.

6.4.2 AEW&C Control Tasks

Having identified the AEW&C activity context, we then focused on the control tasks that are necessary within that context for achieving the goals of the work domain. Figure 6.7 shows some of the control tasks for the class of surveillance activity for AEW&C; the control tasks are in the shaded boxes. Above each control task, we identified the priorities and values (from the work domain analysis) that the crew members must preserve as they execute the control tasks. Below each control task, we identified the purpose-related functions that the control tasks support or promote. To the right of each control task, we identified the set of actors that might be responsible for executing the control tasks. The connecting arrows illustrate the temporal and logical coordination of control tasks, and the set of initiating conditions for each of the control tasks. However, we are still developing a notation for capturing these types of constraints more effectively.

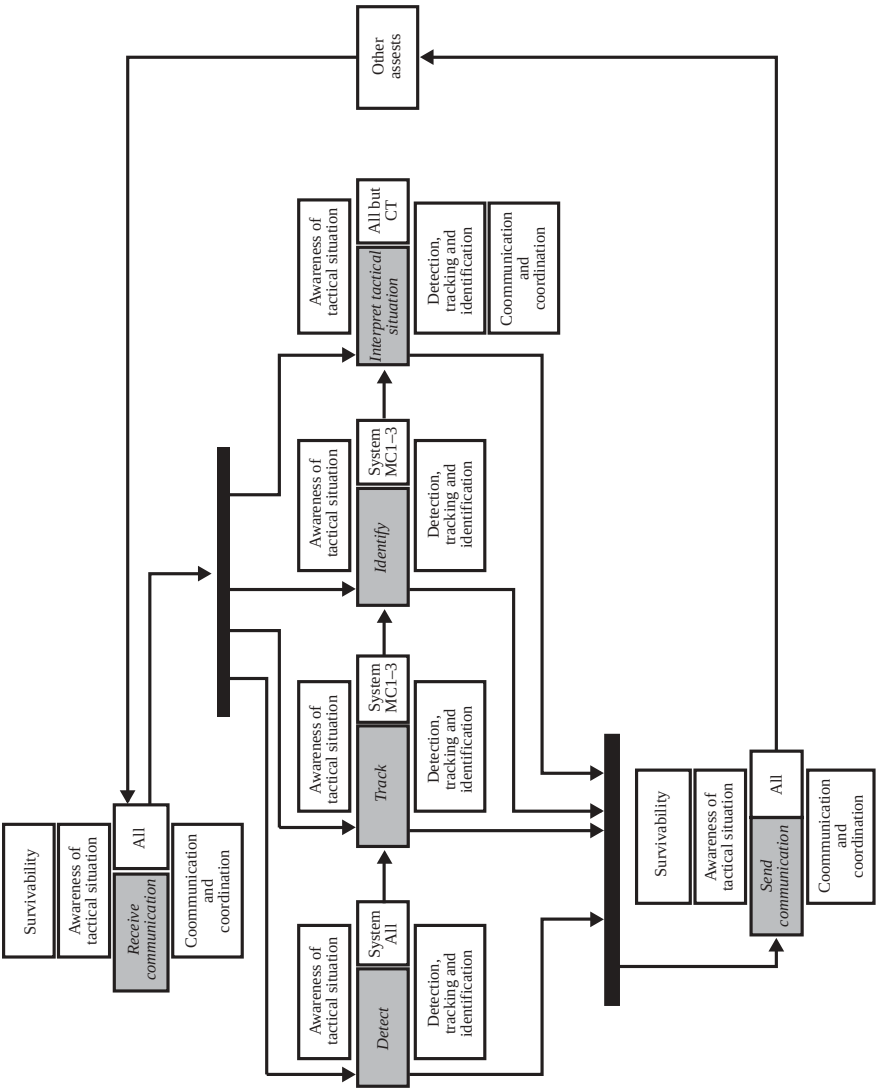


Figure 6.7: Some of the control tasks for the surveillance activity of AEW&C with connections to aspects of the underlying work domain analysis.

6.4.3 AEW&C Human-System Integration

Our third step in describing AEW&C activity was to identify the HSI mechanics for each of the control tasks. On the AEW&C project, our role was that of evaluators rather than designers of the HSI solutions. Thus, to perform this

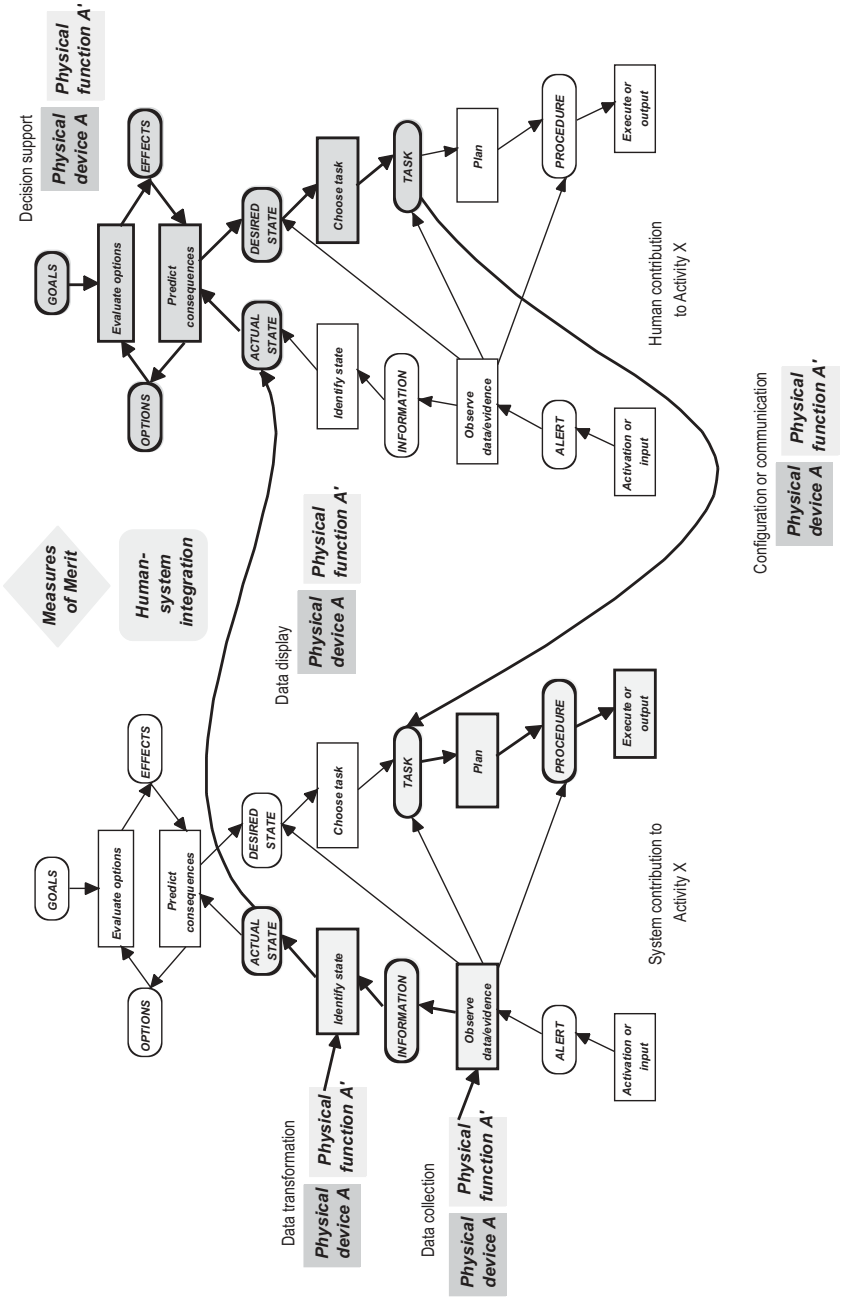


Figure 6.8: Decision ladders linked to show human-system integration.

step we had to rely on the aircraft manufacturers to provide information about their HSI solutions. If the right information was made available, we could then use Rasmussen's decision ladder formalism to illustrate and discriminate the different levels of HSI for each control task.

Figure 6.8 shows a template linking two decision ladders that we used to characterize how the processes of information gathering and analysis, hypothesis testing and generation, and planning and execution were allocated across human and computer in a particular design. We also used the template to indicate the various physical devices and physical functions (from the work domain analysis) that support the information processes shown on the decision ladders. Based on the work of Sheridan and Verplanck (1978), we could then summarize the level of HSI that a particular design offered in the following way:

- HSI Level 1: human performs the whole interpretation or decision action
- HSI Level 2: system generates the options for action or interpretation
- HSI Level 3: system generates options for interpretation or action and suggests best option for human to implement
- HSI Level 4: system generates options for interpretation or action and implements best action if human authorizes
- HSI Level 5: system generates options for interpretation or action, implements best option, and informs humans if requested.

For each control task, the HSI characterization may be at a single fixed level or adaptive over several levels.

Our experience during evaluation was that while there was a lot of information about the physical devices of AEW&C, there was not much information about the integration, cooperation, and communication mechanisms between humans and machines. This information should become available when detailed development of the system begins by Boeing, the winning manufacturer. We expect to use our activity analysis to monitor the development of the HSI solutions as the AEW&C system is developed.

We also note that we have started to use our activity analysis to evaluate alternative crewing concepts and to examine teamwork issues for AEW&C (Naikar, Drumm, Pearce, & Sanderson, 2000). This work involves some aspects of socio-organizational analysis. So far, we have used our models to generate a new team design for AEW&C that subject-matter experts think is promising. This team design will be evaluated in future research.

6.5 TRAINING-SYSTEM DESIGN

Our interest in using work domain analysis for training-system design has been stimulated by a number of concerns. One concern is that transfer-of-training

Functional Structure	Training Needs	Functional Requirements
Functional Purposes: why a work domain exists or the reasons for its design	*Training Objectives:* purpose for training workers is to fulfill the function purposes of a work domain	*Design Objectives:* training system must be designed to satisfy the training objectives of the work domain
Priorities and Values: criteria for ensuring that purpose-related function satisfy systen objectives	*Measures of Performance:* criteria for evaluating trainee performance or the effectiveness of training programs	*Data Collection:* training system must be capable of collecting data related to measures of performance
Purpose-related Functions: functions that must be executed and coordinated	*Basic Training Functions:* functions that workers must be competent in executing and coordinating	*Scenario Generation:* training system must be capable of generating scenarios for practising basic training functions
Physical Functions: functionality afforded by physical devices in the work domain and significant environmental conditions	*Physical Functionality:* workers must be trained to exploit the functionality of physical devices and operate under various environmental conditions	*Physical Functionality:* training systems must simulate the functionality of physical devices and significant environmental conditions
Physical Form: physical devices of the work domain and significant environmental features	*Physical Context:* workers must be trained to recognise functionally-relevant properties of physical devices and significant environmental features	*Physical Attributes:* training system must recreate functionally-relevant properties of physical devices and significant features of the environment

Figure 6.9: Connection between the functional structure of a work domain, training needs, and the functional requirements of training systems.

research has generally failed to show strong transfer from training devices to operational systems. This led us to wonder if the normal way of identifying training requirements is flawed. In addition, there is a recurring concern in training-system design with issues of fidelity. Designers of training systems have become well aware that something other than physical fidelity must be used to guide design decisions. The concept of functional fidelity has been promoted. However, there has been no principled method of distinguishing functional from non-functional fidelity. As work domain analysis is explicitly designed to identify and map the functional constraints of a work system, it seemed that this framework might be used to identify those characteristics of a training system that encompass functional fidelity.

Figure 6.9 shows how we have used work domain analysis for transforming the functional structure of a work domain into functional properties to recreate in a training system. By the use of this framework, each layer of the abstraction hierarchy is translated into particular kinds of training needs, which is then translated into particular kinds of functional requirements for training systems (Naikar & Sanderson, 1999).

Other uses of this framework include tracing the impact of leaving out parts of the functional structure of a work domain from a training device (Lintern & Naikar, 2000). For example, if the attributes of physical form B are

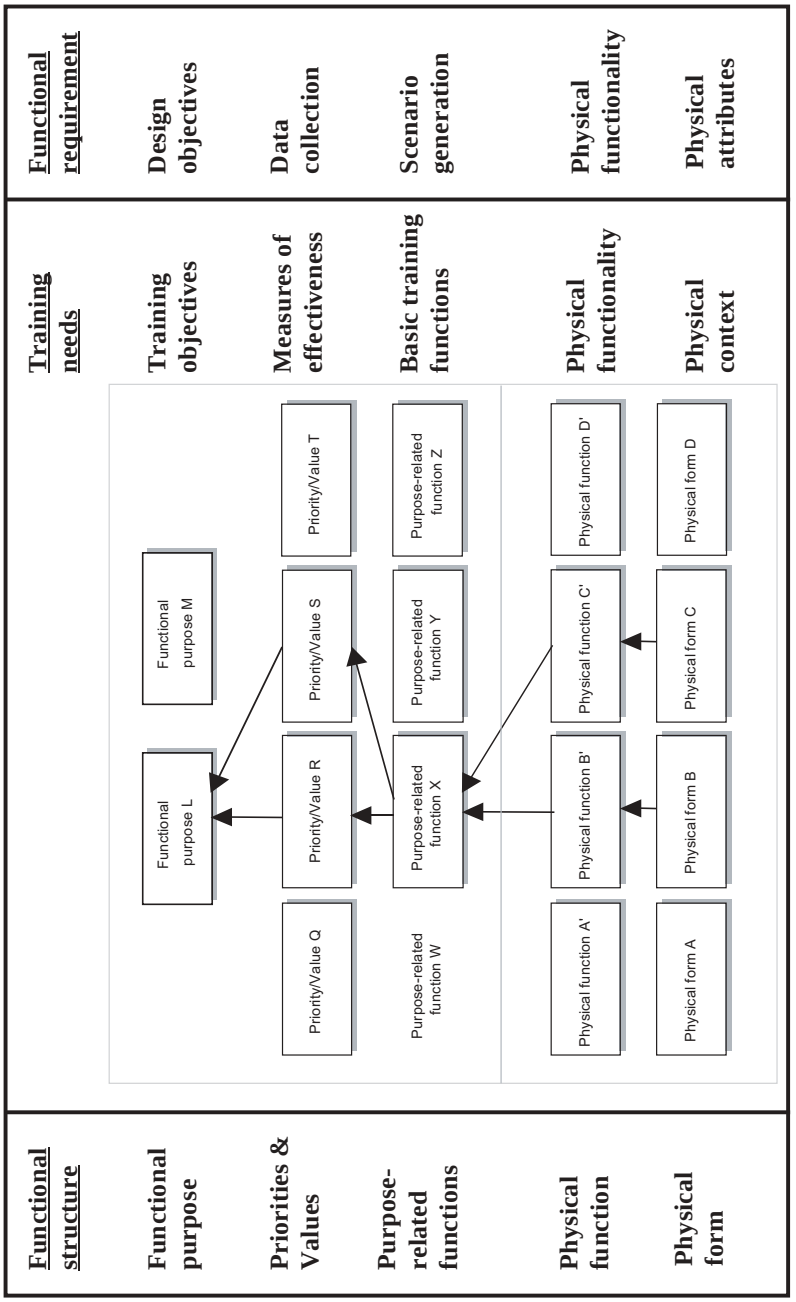


Figure 6.10: The abstraction hierarchy offers a means for tracing the impact of various design decisions on training.

not recreated in a training-system, we can use the links in the abstraction hierarchy to determine the impact this would have on training workers to perform higher-level functions (Figure 6.10). Moreover, work domain analysis can also be used to identify the requirements for part-task training. For example, in designing a part-task trainer for purpose-related function X, we could use work domain analysis to identify the functional relationships that must be present in a training device to support training of that function.

At DSTO we have used work domain analysis to help the Australian Defence Force in purchasing a training system for F/A–18 pilots. As the F/A–18 aircraft is currently undergoing a major system upgrade, our job was to identify the training needs of pilots and the functional requirements for a training system for the up-graded aircraft. To conduct the F/A–18 work domain analysis we used various tactical, training, and flight manuals, and input from subject matter experts. Figure 6.11 shows (1) a global view of the F/A–18 abstraction hierarchy, and (2) a sample of functions from each layer of the abstraction hierarchy. A detailed description of how this framework was used to identify the training needs of F/A–18 pilots, and the functional requirements for a suitable training system can be found in Naikar and Sanderson (1999).

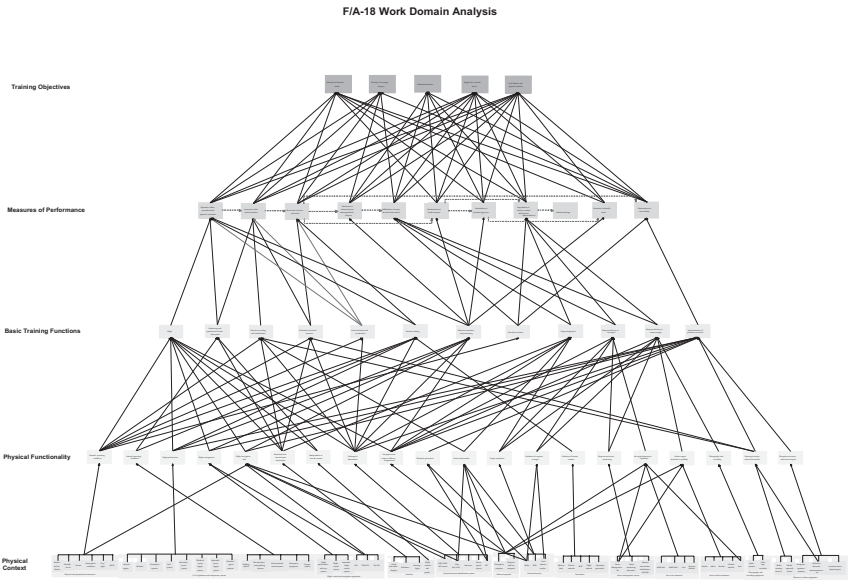
6.5.1 Training Objectives and Design Objectives

The functional purposes layer of the F/A–18 abstraction hierarchy lays out the training objectives of the F/A–18 work domain and the design objectives for a suitable training system. Figure 6.11(b) illustrates that the ultimate goal of training F/A–18 pilots is to ensure the security of sovereign airspace and to maintain the initiative for offensive action. In turn, in designing a training system for this work domain, the goal is to develop a device that supports the training objectives of the F/A–18 work domain.

6.5.2 Measures of Effectiveness and Data Collection

The priorities and values of the F/A–18 work domain describe measures of effectiveness for evaluating trainee performance, and the data collection requirements for an F/A–18 training system. For example, on strike missions, we need to evaluate whether F/A–18 pilots degraded the combat effectiveness of the enemy, whether they caused unnecessary or excessive damage, and whether the pilot operated within procedural constraints (e.g., rules of engagement) and physical constraints (e.g., time, distance, fuel). Thus, the data collection capabilities of a training system must be suitable for capturing this type of information.

We can also measure trainee performance at levels lower than the priorities and values layer of the abstraction hierarchy (Lintern & Naikar, 2000). For



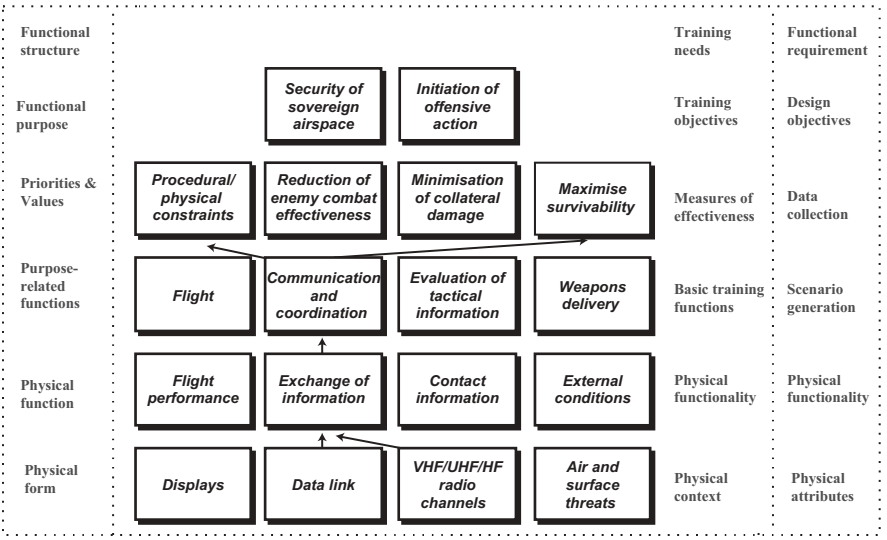


Figure 6.11: (a) A global view of the F/A–18 abstraction hierarchy.
(b) A sample of functions from each layer of the F/A–18 abstraction hierarchy.

example, we may want to measure whether F/A–18 pilots control various aircraft subsystems appropriately (physical form layer), or we may want to measure the accuracy of weapons delivery in terms of distance from target (purpose-related functions layer). Although these measures do not relate directly to the essential priorities and values of the F/A–18 work domain, they do provide diagnostic indices of pilot performance.

6.5.3 Basic Training Functions and Scenario Generation

The purpose-related functions of the F/A–18 work domain inform the basic training functions of F/A–18 pilots and the scenario-generation requirements for a training system. Thus, Figure 6.11 shows that in training F/A–18 pilots we must be concerned not only with flight and weapons delivery, as is typical of many fighter-pilot training programs, but also with communication and coordination, and evaluation of tactical information. Thus, from this layer of the abstraction hierarchy, we can derive the capabilities for scenario generation that a training device must have to support the F/A–18 training program.

Although we have promoted the purpose-related functions layer as the defining layer for the basic training functions of a work domain, the goal of training should be to teach students to exploit all possible means for realizing a target function (Lintern & Naikar, 2000). So, for example, F/A–18 pilots should be trained to reach the target function of communication and coordination via voice channels, data link, and by signaling with the airframe (e.g., tilting the wings of the aircraft). In addition, F/A–18 pilots should also be trained in the effects that their actions at one level can have on higher-level functions (Lintern & Naikar, 2000). For instance, by communicating new rules of engagement to a wingman via radio channels, a pilot may be compromising his survivability as the radio transmissions from his aircraft may be noticed by an enemy pilot. We can use the means-ends relations of an abstraction hierarchy to develop a comprehensive statement of the scenario generation requirements of a training system.

6.5.4 Physical Functions

The physical functions layer of the F/A–18 abstraction hierarchy reflects that pilots must be proficient at manipulating the functionality of aircraft subsystems, and also at operating under different external conditions. For example, Figure 6.11 illustrates that F/A–18 pilots must be proficient at manipulating the flight performance characteristics of the aircraft, and operating under different weather conditions and levels of hostility. Thus, the fourth layer of the

abstraction hierarchy conveys the physical functions to recreate in a training system to support training of higher-level functions.

6.5.5 Physical Context and Physical Attributes

The physical context for training F/A–18 pilots and the physical attributes of training systems are reflected in the final layer of the abstraction hierarchy. For example, F/A–18 pilots must be competent at operating different kind of physical devices, such as visual displays and data link. In addition, they should be capable of recognizing different features of the external environment, such as terrain and types of air and surface threats. Thus, a training system for F/A–18 pilots must recreate these physical properties.

Recently, the Australian Defence Force released parts of the F/A–18 work domain analysis to potential manufacturers for the F/A–18 training system. In time, the F/A–18 work domain analysis will form the basis for a detailed design specification of the F/A–18 training device. However, preliminary work has shown that to fully complete the specification for this system, we will need to go beyond work domain analysis. Lintern and Naikar (2000) describe the following areas as requiring alternative forms of analysis: (1) critical skills to emphasize in training, (2) the form in which to implement particular functional requirements, (3) the levels and types of fidelity of various physical components, and (4) special features for supporting instruction. These additional requirements are not limited to work domain analysis but are necessary even if one adopts more conventional approaches to training-system design.

Our experience in using work domain analysis for training-system design is that this framework offers several advantages over more conventional techniques. One advantage is that by using work domain analysis we can derive the functional requirements for training systems directly from the functional structure of the work domain itself. In contrast, Instructional Systems Design (ISD) focuses on identifying typical tasks or procedures that will be trained with the new device. Once this analysis is complete, an additional step is required to go from the description of tasks to a description of functional requirements. This step is most often done informally and task by task.

Another advantage of work domain analysis is its suitability for specifying training devices that are not yet operational. For brand new combat systems, or for combat systems that are being upgraded, it is difficult to fully anticipate training requirements. Workers will find new ways of using the platform as they develop expertise with it, and ways of using the platform will also be influenced by the capability of future enemy systems. Conventional techniques cannot inform the design of training systems for these types of events, as it is difficult to develop descriptions of tasks and procedures for ways of using a plat-

Purpose of Analysis	Properties of the Information-Action Workspace	Tools	
		Knowledge Acquisition	**Knowledge Representation**
Work Domain	Physical & Purposive Constraints	Document Analysis Reviews by Subject Matter Experts	Abstraction-Decomposition Matrix
Activities	What needs to be done in the Work Domain (Work Functions & Control Tasks)	Cognitive Walk-Through, Study of Work Practices	Decision Ladder
Strategies	Strategies for Management & Control (Planning, Adapting)	Critical Decision Methods, Interaction Analysis, Verbal Protocol Analysis	Information Flow Map
Social-Organisational	Collaborating Actors & Organisational Structure	Communications Analysis, Interaction Analysis	Integrate information from tools above
Concepts & Competencies	Human Capabilities & Limitations (SRK, Conceptual Distinctions)	Repertory Grid Analysis & Review of Decision Ladder	Concept Map & Skills-Rules-Knowledge Frame

Figure 6.12: A set of tools used for knowledge acquisition and knowledge representation in each phase of a cognitive work analysis for identifying the design-relevant properties of an information-action workspace. The tools shown here illustrate how it is possible to proceed with a cognitive work analysis. However, the diversity of tools that is available is too great for this figure. Different tools could be substituted with good effect, depending on the demands of the project and the expertise of the analysts (Seamster, Redding, & Kaempf, 1997; Lintern, in press).

form that are not yet known. Work domain analysis, however, avoids this problem by focusing on event-independent properties of a work domain.

6.6 INFORMATION WORK SPACES

It is said that we live in the information age. Although information has always been critical to human action, it sometimes seems that our modern society has overburdened us with information. We do, in fact, function quite well with a relatively small amount of information if that information is relevant, timely, and organized to suit our natural capabilities of perception, interpretation, and action. Where we are overburdened with information or are unable to make good decisions, it is primarily because information is poorly organized, fragmented, or pitched at a level of abstraction that does not link directly into the

Level of Abstraction	**Representation Requirements**	**Formats**
Functional Purposes		
Priorities and Values	Flow Mass Value Balance Accumulation Dispersion	Configural Displays of Balances & Relationships Between Functions & States Limit Envelopes
General Functions	Relations Intended States Trajectories Offensive Capability Defensive Capability	Configural Displays Constraint Boundaries Threat & Lethality Shadows Guides Predictor Elements & Envelopes Symbolic Diagrams Target Lists Priority Indicators
Physical Functions	Status of process variables with reference to target states and to limits of acceptable operation	
Physical Form	Topography of the work system	Object Representations Icons, Symbols, Signs Mimic Diagrams Pictorial Representations Flow Maps Ingress & Egress Routes Locations Fields of Action

Figure 6.13: A typology of display formats for different levels of abstraction. (Adapted primarily from Rasmussen, 1998; but also with reference to Dinadis & Vicente, 1999; and to Pejtersen, 1992).

functional interpretation-action sequence. What is needed is an appropriately configured functional interface (Lintern, Waite, & Talleur, 1999).

Following the lead of Pejtersen (1992) in her design of an interface for a children's library, we have proposed to use CWA as the analytic method for accomplishing much of the early conceptualization for the design of an information system (Figure 6.12). Work domain analysis identifies the functional

requirements that must be made visible at the interface, and activity analysis identifies what needs to be done in the work domain. strategies analysis identifies how the operator can interact with this system. For an information workspace, the primary product of the strategies analysis is a map that shows what information is needed and how it flows through the work system.

Our area of concern is military command and control, which is quite a different form of information system from a children's library. Nevertheless, the conceptual challenges to the designers of the information interfaces are similar, as are the representation requirements and the display formats that might be useful. Thus the tools for knowledge acquisition and knowledge representation should also be similar (Figure 6.13). At this stage, these ideas remain conceptual (Lintern & Naikar, in preparation) but we propose to test them in the development of a command and control information space. The aim is to integrate forms for perception and action into a virtual workspace in a manner that will support access to essential information and that will provide means for testing and implementing decisions.

6.7 CONCLUSIONS

In our work in air defense contexts over the last two years, we have used CWA for many purposes other than interface design, such as specification of training programs, specification of simulator needs, design of research programs, intelligent agent modeling, and evaluation of design solutions, and we have found that CWA provides significant insight (see Sanderson et al., 1999 for applications not discussed in this paper). We have also found that the products of CWA have been reusable across many different purposes. Consequently, CWA has become an intellectual framework for certain groups within DSTO, where human factors practitioners, training specialists, simulator constructors, cognitive scientists, and operations researchers can communicate and dovetail their activities.

There is considerable effort involved in performing CWA, and some of the conceptual material the analyst needs to understand to do it properly has subtle but critical differences from more familiar approaches to human engineering. However, we suggest that CWA provides an approach to human engineering that is no more complex than what has been suggested in more traditional human engineering programs and may actually be simpler. Not only are the five phases of CWA tightly linked, but also the products of the analyses can be reused in the ways described above. We therefore look forward to seeing how the cells in Figure 6.3 are populated as cognitive engineers gain experience with CWA over the next few years.

REFERENCES

Benda, P. J., & Sanderson, P. M. (1999a). New technology and work practice: Modelling change with cognitive work analysis. In M.A.S. C. Jonson (Eds.), *Human-computer interaction* (INTERACT99) (pp. 566–573). Amsterdam: IOS Press

Benda, P. J, & Sanderson, P. M. (1999b). Representation of changes to anesthesia work domain and activity structures. *Proceedings of the 43rd Annual Meeting of the Human Factors and Ergonomics Society* (pp. 563–567). Santa Monica, CA: HFES.

Department of Defence (1995). *The capital equipment procurement manual* (CEP-MAN). Canberra, Australia: Defence Publishing Service.

Department of Defence (1999). *The defence procurement policy manual* (DPPM). Canberra, Australia: Defence Publishing Service.

Dinadis, N., & Vicente, K. J. (1999). Designing functional visualizations for aircraft system status displays. *International Journal of Aviation Psychology, 9*, 241–269.

Leveson, N. G. (2000). Intent specifications: An approach to building human-centered specifications. *IEEE Transactions on Software Engineering*, *26* (1), 15–35.

Lintern, G. (in press). A review of applied cognitive task analysis in aviation by Thomas L. Seamster, Richard E. Redding, & George L. Kaempf. *International Journal of Aviation Psychology.*

Lintern, G., & Naikar, N. (1998). Cognitive work analysis for training system design. In *Proceedings of the Australasian Computer Human Interaction Conference—OZCHI98* (pp 252–259). Los Alamitos, CA: IEEE Computer Society.

Lintern, G., & Naikar, N. (2000). The use of work domain analysis for the design of training systems. *Proceedings of the 44th Annual Meeting of the Human Factors and Ergonomics Society Vol. 1* (pp. 1–201) Santa Monica, CA: HFES.

Lintern, G., & Naikar, N. (in press). *A virtual information-action workspace for command & control*. Melbourne, Australia: Aeronautical & Maritime Research Laboratories, Defense Science & Technology Organization.

Lintern, G., Waite, T., & Talleur, D. A. (1999). Functional interface design for the modern aircraft cockpit. *International Journal of Aviation Psychology, 9*, 225–240.

Naikar, N., Drumm, D., Pearce, B., & Sanderson, P. M. (2000). Designing new teams with cognitive work analysis. *Paper presented at The Fifth Australian Aviation Psychology Symposium,* Manly, Australia.

Naikar, N., & Sanderson, P. M. (1999). Work domain analysis for training-system definition and acquisition. *International Journal of Aviation Psychology, 9* (3), 271–290.

Naikar, N., & Sanderson, P. M. (2000a). Evaluating design solutions with work domain analysis. *Proceedings of the 44th Annual Meeting of the Human Factors and Ergonomics Society. Vol. 1* (pp. 1-202–1-205) Santa Monica, CA: HFES.

Naikar, N., & Sanderson, P. M. (2000b). *Evaluating designs proposals for complex systems with work domain analysis.* Manuscript submitted for publication.

Naikar, N., Sanderson, P. M., & Lintern, G. (2000). *F/A–18 work domain analysis: identification of training needs for simulator acquisition* (DSTO Client Report, DSTO–CR–0117). Melbourne, Australia: Aeronautical and Maritime Research Laboratory.

References

Pejtersen, A. M. (1992). The BookHouse: An icon based data base system for fiction retrieval in public libraries. In B. Cronin (Ed.), *The marketing of library and information services 2* (pp. 572–591). London: ASLIB.

Rasmussen, J., Pejtersen, A., & Goodstein, L. P. (1994). *Cognitive systems engineering.* New York: Wiley.

Sanderson, P., Naikar, N., Lintern, G., & Goss, S, (1999). Use of cognitive work analysis across the system life cycle: Requirements to decommissioning. In *Proceedings of the 43rd Annual Meeting of the Human Factors and Ergonomics Society Vol 1* (pp. 1-206–1-209). Santa Monica, CA: HFES.

Sanderson, P. M., & Naikar, N. (2000). Temporal coordination control task analysis for analyzing human-system integration. *Proceedings of the 44th Annual Meeting of the Human Factors and Ergonomics Society*. Santa Monica, CA. HFES.

Seamster, T.L., Redding, R.E., & Kaempf, G.L. (1997). *Applied cognitive task analysis in aviation*. Hants, England: Ashgate.

Sheridan, T., & Verplanck, W. (1978). *Human and computer control of undersea teleoperators* (Technical Report). Cambridge, MA: Massachusetts Institute of Technology, Man-Machine Systems Laboratory.

Vicente, K. J. (1999). *Cognitive work analysis: Towards safe, productive, and healthy computer-based work*. Mahweh, NJ: Erlbaum.

Vicente, K. J., & Rasmussen, J. (1992). Ecological interface design: Theoretical Foundations. *IEEE Transactions on Systems, Man, and Cybernetics, 22*(4), 589–606.

7 Analyzing the Cognitive System From a Perceptual Control Theory Point Of View

KEITH C. HENDY
Defence and Civil Institute of Environmental Medicine

DAVID BEEVIS
Defence and Civil Institute of Environmental Medicine

FREDERICK LICHACZ
Defence and Civil Institute of Environmental Medicine

JACK L. EDWARDS
AIM

ABSTRACT

The starting point for the design of any complex system should be analysis. For systems where human functions are predominantly "cognitive," the method of analysis should capture this essentially human activity. Traditionally human engineering analyses have been based on a hierarchical decomposition of system missions, functions and tasks. Perceptual Control Theory (PCT) provides a theoretical framework for guiding this process. PCT reorients the approach from a serial function analysis, function allocation, task analysis process, to a hierarchical goal analysis. The hierarchical goal analysis combines the previously separate processes into one. With PCT it is inescapable that goals at all levels are candidates for assignment to an agent (human or machine).

Two new analyses emerge from the PCT framework. The first, a stability analysis, looks to see if certain external variables can be simultaneously under multiple control. If conflicting goals or incompatible internal perceptual, cognitive, or machine functions, could cause these multiple control situations to be unstable, then the designer has to find a way to separate control or otherwise ensure stability. The second analysis looks at the upward flow of information in the system. Each goal is examined to see how information existing at the

subgoal level flows up to the level above. Both analyses potentially identify new goals that must be accommodated by interface design.

7.1 INTRODUCTION

A closed, negative loop gain, feedback system is an error-correcting system. The inverse of this proposition is that all error-correcting systems can be reduced to a closed-loop, negative-gain, feedback system. If these propositions are true, and the human is seen as exhibiting error-correcting behavior, then William T. Powers' claim that all human behavior occurs as a result of a perceptually driven, goal-referenced, feedback system (Powers, 1973b), should come as no surprise. This is the tenet of Perceptual Control Theory or PCT. At best, PCT provides a veridical explanation of how humans form and emit behaviors; at the very worst, PCT is a normative model of human behavior.

The PCT model suggests a multilayered system, with multiple goals providing the reference points for a hierarchical organization of control loops. These loops provide control at many levels—from the lowest levels of sensory processing, upward to the satisfaction of abstract goals such as the need for self-esteem and actualization. In PCT terms, an emitted action or behavior is in response to the presence of an error, or difference, signal. The emitted action is transmitted purposefully, with the intention of changing the state of the world so that the operator's perception can be made to match a desired state or goal, which reduces the error signal to zero. It is a fundamental thesis of PCT that it is the perception that is controlled, not the behavior.

The starting point for the design of any complex system should be analysis. For systems where human functions are predominantly "cognitive," the method of analysis should capture this essentially human activity. Traditionally human engineering analyses have been based on a hierarchical decomposition of system missions, functions, and tasks analysis (MIL–HDBK–46855A, 1999). A method is proposed in this paper, based on PCT, for conducting this type of HFE analysis. PCT provides a theoretical framework for guiding this process, and reorientates the approach from a serial process of function analysis, function allocation, task analysis, to a unified process of hierarchical goal analysis. The hierarchical goal analysis combines the previously separate processes into one. With PCT the fact that goals at all levels are candidates for assignment to an animate or inanimate agent, is inescapable. Two emergent properties have been identified with PCT:

- The need to consider stability in multiagent systems, and
- The need to consider both the upward and downward flow of information in the system.

It is important to understand the context within which the proposed method is intended to function. A PCT-based analysis is intended to serve front-end Human Factors Engineering (HFE) analysis requirements, as outlined in documents such as MIL–HDBK–46855A (1999) or in IEEE–1220 (1994). The output of this process feeds the specification of human systems design requirements that must encompass each of the domains specified by the U.S. DoD for human systems integration in their Department of Defense (DoD, 1991). These domains are as follows:

- **Manpower:** The number of military and civilian personnel required and potentially available to operate, maintain, sustain, and provide training for systems.
- **Personnel:** The cognitive and physical capabilities required to be able to train for, operate, maintain, and sustain materiel and information systems.
- **Training**: The instruction or education, and on-the-job or unit training required to provide personnel their essential job skills, knowledge, and attitudes.
- **Human Factors Engineering:** The integration of human characteristics into system definition, design, development, and evaluation to optimize human-machine performance under operational conditions.
- **System Safety:** The design features and operating characteristics of a system that serve to minimize the potential for human or machine errors or failure that cause injurious accidents.
- **Health Hazards:** The design features and operating characteristics of a system that create significant risks of bodily injury or death; prominent sources of health hazards include acoustics energy, chemical substances, biological substances, temperature extremes, radiation energy, oxygen deficiency, shock (not electrical), trauma, and vibration.

In this sense, the proposed method sits above the level of many of the specific tools or techniques that might contribute to the mechanics of a top-level analysis. For example, in the cognitive task analysis literature some methods (e.g., cognitive work analysis, see Vicente, 1999) are intended to be comprehensive human systems analysis processes, while others are more limited in scope and are focused on specific aspects of the process, such as expert knowledge elicitation.

The structure of this paper is first to elaborate on the link between structured front-end HFE methods and systems engineering and then to present some basic ideas from the perceptual control theory paradigm that leads to an alternative form of analysis. The PCT-based systems analysis described is a method for conducting front-end human factors engineering analysis. In that it deals with the goals and knowledge of the humans in the system, it has an association with cognitive systems engineering, but it is equally concerned with the sensory, perceptual, and psychomotor requirements of the human machine

interface. Finally, in discussion, the proposed PCT-based analysis method is compared to more traditional human factors engineering methods and to Vicente's (1999) Cognitive Work Analysis (CWA).

At first exposure to these ideas, it is not always obvious that PCT evokes all the power and knowledge of closed-loop control systems theory from engineering. Therefore, proponents of PCT do not argue from a purely descriptive position, but can apply what is known about closed-loop control behavior to human cognition. While many of the ideas contained in PCT are represented within other paradigms, it is their synthesis under PCT that is perhaps the main contribution of this method. It is argued that adopting the PCT approach to systems analysis does not usurp mainstream psychological theory or established tools like cognitive task analysis. PCT provides a framework within which these ideas may be embedded.

7.2 HUMAN SYSTEMS ANALYSIS

Engineering provides a technology base that bridges the gap between a body of scientific knowledge and the application of that knowledge to design. The engineering technology base consists of the methods, procedures, and tools for applying this pool of scientific knowledge to design. Engineering is divided into disciplines that draw on different science bases. The discipline known as Human Factors Engineering (HFE) fits naturally within this framework. HFE draws on a knowledge base of engineering principles and methods that is shared with other engineering disciplines, together with a specialist knowledge base that describes what we know about human capabilities and limitations. It is arguably the least mature of the engineering disciplines although possibly not the newest. Systems engineering appears to have largely emerged from the RAND Corporation during the 1950s and 1960s (Checkland, 1981, p. 136); HFE can trace its origins back at least to the 1940s (Green, Self, & Ellifritt, 1995).

Systems engineering has typically dealt with the constraints imposed by the technological side of the design equation, particularly with cost-benefit trade-offs of alternative design solutions, while human engineering has considered the constraints imposed by human capabilities and limitations. These disciplines meet at the human interface with the potential for some crossover where the transfer of information and human performance become necessary considerations in systems integration activities.

The goal of systems engineering analysis is the transformation of an operational need into a system configuration (Diamond, 1989). A key to systems engineering activities is the functional decomposition of a system to the level where candidate solutions can be identified (DoD, 1990).

The belief that large-scale systems development can benefit from a structured top-down approach to design, characterizes *hard* systems thinking in

Peter Checkland's terms (see Checkland, 1981, Chap. 5). While there may be disagreement about the best tactics for implementing this approach, the strategic issues are quite clear. A formal method of systems analysis typically involves (the following is reported in Checkland, 1981, p. 136, and is adapted from a 1955 report by Hitch from the RAND corporation):

- A statement of objective(s)
- Alternative design solutions
- The costs involved in the implementation of each solution
- Mathematical representations of the system, and
- Criteria relating objectives, costs, and resources required in implementing a preferred or optimum alternative.

7.2.1 Structured Analysis Techniques

As a set of methods and procedures, the classical systems engineering process as currently practiced (e.g., see IEEE–1220, 1994) specifically considers the constraints imposed by both the human and the environment on design. Early texts on systems engineering included human engineering as a topic with an emphasis on the "…man machine link" (Goode & Machol, 1957); however, structured techniques for analyzing human systems have been part of basic HFE practice for many years (e.g., Chaillet, 1967; McCormick, 1976; Meister, 1985; Van Cott & Kinkade, 1972; Woodson, 1981). Typical of these is the set of procedures described as Mission, Function, Task Analysis (MFTA).

MFTA represents a comprehensive top-down analysis that mirrors the process used by systems engineers (IEEE–1220, 1994) as shown in Figure 7.1. MFTA is appropriately used during the early stages of systems development, that is, during the concept development and feasibility phases. MIL–HDBK–46855A (1999) describes this process and provides guidance for how one might conduct this type of analysis. Fundamental is the concept that analysis and design are tightly linked, and that validation and verification are carried out at each step. MIL–HDBK–46855 specifies a structure, which includes, but is not restricted to:

- Scenario development and mission analysis
- Function analysis and allocation
- Equipment selection
- Task analysis (including critical task analysis), and
- Workload/performance analysis.

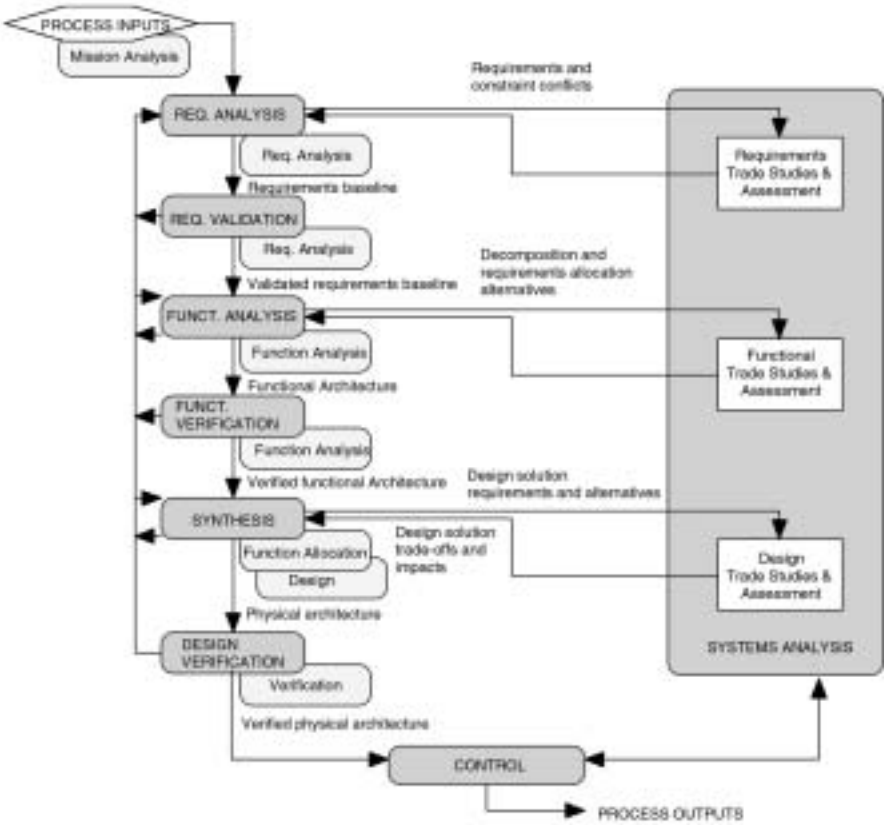


Figure 7.1: Relationship between the systems engineering (in dark gray) and human engineering processes (in light gray). (From IEEE–1220, 1994).

7.2.2 Domains of Analysis

Typically systems analysis will include (Van Cott & Kinkade, 1972, p. 4):

- The explication of system requirements and constraints
- The description of system functions
- Detailed descriptions of operational event sequences (including environmental conditions), and
- Detailed descriptions of component processes.

This suggests a two-dimensional, high-level, taxonomy for analysis that has, at least, both time and functional area aspects. For example:

7.2.2.1 Time-Based (While these categories have obvious relevance to vehicular systems, all systems demonstrate equivalent time-based phases from startup to shutdown, or from commissioning to decommissioning and disposal.)

- Prepare for mission
- Prepare for departure
- Departure
- Transition
- Mission
- Return
- Arrival
- Shutdown and secure

Each of these time-based components should then be broken out by functional area, as follows—

7.2.2.2 Functional Area-Based

- Primary mission
- Training (the need to design for embedded or on-the-job training)
- Abnormals
- Maintenance
- Sustain or replenish

To this list could be added team functions (i.e., behaviors and behavior modifications that emerge when individuals interact as team or group members, for example, see Annett & Cunningham, 2000). At a recent meeting between Canada, the Netherlands, and United Kingdom, the following tentative list of team behaviors was identified using the PCT structure as a framework (Belyavin, Cain, Lessens, & Hendy, 2000):

7.2.2.3 Team Coordination (optimize throughput, balance workload, minimize workload)

- Coordinate tasks
- Monitor task demand
- Establish priorities
- Allocate responsibilities (reactive)
- Take responsibility (proactive)
- Coordinate resources (manage the collective time-knowledge-attention equation)

- Monitor the load of team members
- Initiate load correction
- Monitor the knowledge of team members
- Initiate calibration of knowledge states
- Monitor the attentional locus
- Direct attention

7.2.2.4 ***Team Error Correction*** (support successful goal achievement)

- Monitor team goal achievement
- Provide feedback

7.2.2.5 ***Team Maintenance*** (maintain team health)

- Establish personal authority (establishing trust)
- Establish legitimate authority
- Motivate the team
- Monitor the affective state of team members
- Establish and maintain communication channels in human-human interactions
- Etc.

Without a formal structure such as this, it is possible that a designed system might serve the primary mission but not accommodate the requirements for departure and return, or on-the-job training and maintenance. The time-based dimension of this taxonomy imposes sequence at the top level of analysis but does not necessarily impose sequence at lower levels of the analysis, a criticism sometimes made of what are called traditional task analysis methods (e.g., p. 76 of Vicente, 1999).

The sequence of mission, function, and task analysis outlined above does more than support human factors engineering efforts. These analyses also provide information for the other four Human Systems Integration (HSI) domains. Thus, as with systems engineering, it can be argued that classic MFTA is ecological, in that the analyses describe the context and environment in which operators and maintainers perform their tasks.

7.2.3 Reliability of Human Systems Analyses

Currently the approach outlined in IEEE–1220 (1994) and MIL–HDBK–46855A (1999) has the advantage of many years of practice

behind it and the availability of a plethora of tools for its implementation (e.g., http://dticam.dtic.mil/hsi/index.html; McMillan, Beevis, Salas, et al., 1989; Beevis, Bost, Döring, et al., 1999). Yet even though these methods and tools exist, design continues to have its failures, even for the most simple of everyday things (Darnell, 2000).

Design can break down from many factors. It may be that the customer was insensitive to human factors and did not emphasize their importance to the developer (Hendrick, 1990). It may be that the developer's personnel were untrained in human factors, or that their effort was focused on the final detailed design stage of development and was unable to influence the system concept (Beevis, 1987). It may be the failure of systems developers to actually use a comprehensive process of systems analysis (Beevis, et al., 1999, p. xxi–xxiii) that is at the heart of many design errors rather than the actual form of the analysis used. Or it may be that the tools used to implement these procedures are not sufficiently mature or that the methods and procedures are wrong or do not capture important emergent properties (Checkland, 1981, pp. 72–84). Then there are those who argue that traditional methods of analysis are doomed to failure for the analysis of complex open systems and, by implication, one might trace design failures to these flawed methods (Vicente, 1999, Chap. 3).

The Achilles heel of MFTA, as traditionally practiced, may not be in the overall structure of the method but rather in its implementation. In MFTA, interface design is generally done at the task or lowest level of analysis. As will be seen from the PCT hierarchical goal analysis described later in this paper, this limitation presents a number of problems. For example, the typical MFTA approach identifies operator tasks related to mission-specific functions. The functional decomposition does not identify crew functions because it is performed before functions are allocated to operators, maintainers, or machines. As a result, the analysis may not identify any functions or multi-operator tasks associated with crew coordination, consultation, resolution of ambiguity, etc., which can have an important bearing on crew compartment design (Beevis, Lessens, & Scuffle, 1996). Some other problems are identified by Vicente, albeit within the limited context of task analysis procedures in isolation.

7.3 PERCEPTUAL CONTROL THEORY

The basic perceptual control theory model for goal-directed human behavior, at a single level of abstraction, is shown in Figure 7.2. PCT, combined across all possible levels of abstraction, describes a hierarchical model of many processing layers (Hendy, East, & Farrell, 2000; Powers, 1973b, 1989, 1992a, 1992b).

True to its control heritage, a PCT loop consists of a set point (the perceptual goal), an input transformation process that maps incoming sensory

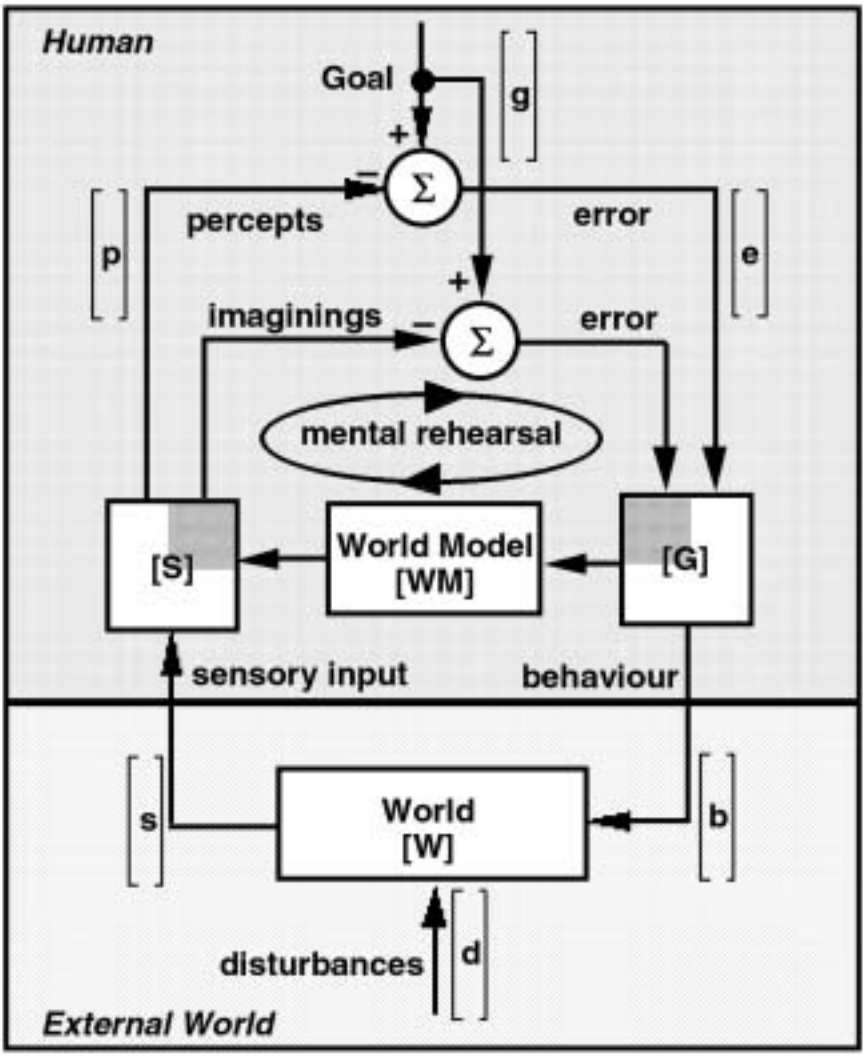


Figure 7.2: The Perceptual Control Theory model.

data into perceptual information, and an output transformation process that maps a difference or error signal into an output behavior. The loop is closed through its influence over a set of variables in the environment. PCT includes an inner loop, operating on or influencing internal memory structures. When this internal loop is exercised, no observable behaviors need be seen at the interface between the human and the world. The hierarchical nature of the PCT model is captured in a representational form by the use of a vector and

matrix notation for the various system states and the transformation functions at the input and output sides of the loop.

Humans and machines interact and communicate with each other (human-human and machine-human) via their influence on various variables in the external world. Just as humans have goals that determine the set point of various control loops, machines have various references or programmed objectives that represent the goals of the system designers. Hence, the general structure of the PCT model applies to machines as well as humans. Teams and groups interact through their mutual influence on these external world variables. In Figure 7.3 a two-person team is shown operating on a common world environment represented by a complex environment function **W**. The concept shown in Figure 7.3 can be generalized to teams or groups of any number. Team members can be composed of any mixture of humans and machines.

From what we know of multivariate controllers (e.g., see Van de Vegte, 1986) we can predict how such systems might achieve stable control. In Figure 7.3 both operator *i* and operator *j* are initiating actions that operate on the shared world environment **W**. As both operators' actions affect the shared environment, their control loops are potentially cross-coupled with the potential for divergent or unstable behavior. As Powers (1992a) points out, the potential for conflict (i.e., the degree of coupling between *i* and *j*) depends both on the extent that *i* and *j* attempt to act on the same environmental variables, and to the extent that *i* and *j*'s perceptions are formed by linearly dependent transformations on these variables.

In situations where there is strong cross coupling, it is critical for stability that the goals and the perceptual (input) and behavioral (output) transfer functions are very closely matched. When controllers are coupled and the loop gains are quite different, the controller with the highest loop gain (authority gradient) will generally override the other.

Decoupling the loops implies that nonoverlapping roles and tasks have been defined for each of the agents. The need for an executive function (command, leadership) to oversee the allocation of roles and tasks, set common goals and establish common mental models is specific to the multicrew case. Information flow or communication between crewmembers, to execute this function, becomes an essential emergent property in this situation (Hendy, 1995, 1998). Note that the potential for multiagent instability comes only from mutual influence on external variables. No agent can directly influence the internal variables of another agent.

The constraints on human information processing, within the modules of the PCT loop, are described by the Hendy et al. Information Processing (IP) model (Hendy et al., 2000; Hendy, Liao, & Milgram, 1997). Together the IP/PCT models provide a strong integrating framework for analyzing and predicting human information processing behaviors (Hendy & Farrell, 1997).

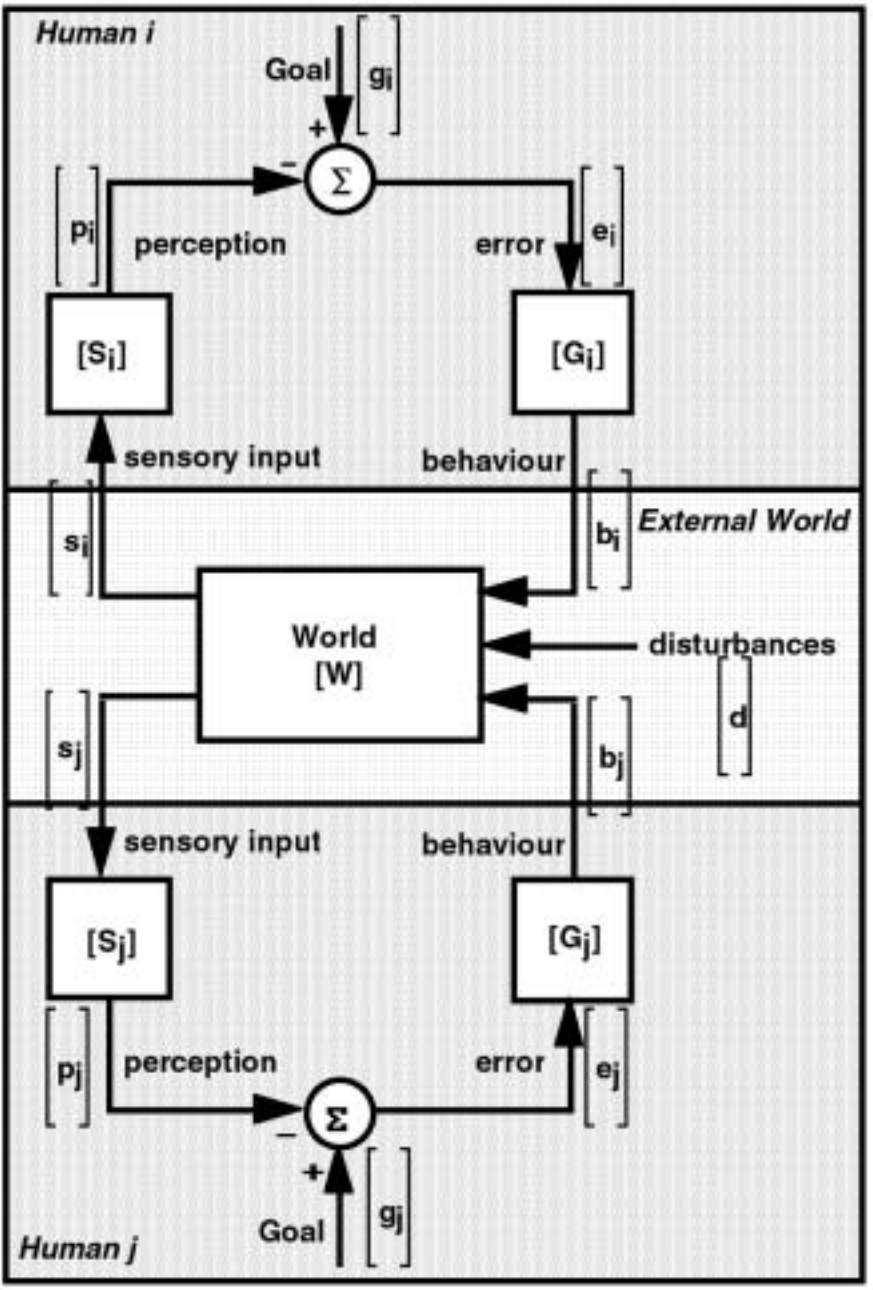


Figure 7.3: Multiple agents interacting through their influence on shared environmental variables.

7.3.1 Information Processing Models and PCT

Generally, a model is a representation that mirrors, duplicates, imitates, or in some way illustrates a pattern of relationships observed in data or in nature. In this way a model becomes a kind of mini-theory, a characterization of a process and, as such, its value and usefulness derive from the predictions one can make from it and its role in guiding and developing theory and research (Eysenk & Keane, 1990). Indeed, one of the purposes of building models is to make observations more comprehensible (Solso, 1991).

Cognitive psychologists are interested in the study of how we gain information about the external world. As Lachman, Lachman, and Butterfield (1979) assert (see also Neisser, 1967) cognitive psychology is about how people take in information, how they recode and remember it, how they make decisions, how they transform their internal knowledge states, and how they translate these states into behavioral outputs. Cognitive psychologists try to achieve this end by focusing on the internal psychological structures and operations that are involved in the transformation of information from stimulus to response (Roediger, Rushton, Capaldi, & Paris, 1984; Solso, 1991). Since the 1950s, this endeavor has been guided primarily by the information-processing paradigm (note that the Hendy et al., 1997, IP model is a specific example of this general class of model).

Rooted firmly in this framework, cognitive psychologists have viewed complex human behavior as the result of how a person transforms information between stimulus and response (Hintzman, 1978; Lachman et al., 1979; Reed, 1982; Simon, 1980). Guided by the IP paradigm, cognitive psychologists believe that information is transformed (i.e., processed) and analyzable across a series of processing stages during which specific operations are performed on incoming information. The response is assumed to be a result of the outcome of this series of stages and operations (Garner, Hake, & Eriksen, 1956; Hintzman, 1978; Knobel & Shaughnessy, 1999; Logan, Coles, & Kramer, 1996; Marr, 1982; Previc, Yauch, DeVilbiss, et al., 1995; Solso, 1991; Sternberg, 1969). The success of the IP paradigm in enhancing our understanding of complex human behavior has been demonstrated in such fields of research as lexical processing (Becker & Killion, 1977; Besner & Chapnik Smith, 1992; Herdman, Chernecki, & Norris, 1999), the effects of narcosis on divers (Fowler, Mitchell, Bhatia, & Porlier, 1989), the effects of closed-head injuries on information processing (Schmitter-Edgecombe, Marks, Fahy, & Long, 1992), and visual pattern recognition (Hughes, Layton, Baird, & Lester, 1984).

The prototypic cognitive model broadly divides the cognitive processes into three components or structures:

- Detection of stimuli
- The storage and transformation of stimuli, and

- The production of responses (Solso, 1991).

Cognitive psychologists then attempt to determine how information is modulated both within and between these structures. This "…cognitive architecture provides the missing theoretical integration, and is thus far broader in its scope than most theories…cognitive architectures are designed to capture the basic principles of operation built into the cognitive system" (Eysenk & Keane, 1990, p. 31). In addition to this basic architecture, Marr (1982) has stated that a model of information processing must have certain properties before a complex behavior can be fully understood. First, a model must be able to detail how different kinds of information (e.g., perceptual, higher order) map onto one another. Second, the model must detail how this information is represented. For example, is information coded as patterns of activation or as localized representations? Third, the model must provide a structure from which the above properties can operate. This third level involves the derivation of an algorithm from which the transformation of information is to take place. Consequently, the algorithm is dependent upon the manner in which information is represented (e.g., serial, parallel, connectionist, localist, etc). The development of these algorithms is further dictated by situation and task-specific attributes as well. As should be evident, model building, certainly within cognitive psychology, involves a progression from the conceptual to the computational. The discovery of these structures and what happens to information within and between these structures is particularly important because when a person has difficulty performing a task, the psychologist can then attempt to identify which stage is the primary source of this difficulty and then attempt to remedy it (Reed, 1982).

PCT both embodies and extends the traditional IP approach to understanding the underlying cognitive components to behavior. Structurally, PCT embodies the IP approach insofar as it broadly divides the cognitive system into three similar areas: perception (i.e., detection of stimuli), a goal stage which is integral in the storage and transformation of stimuli, and a response stage that initiates an overt behavior. Operationally, PCT outlines the transfer functions required to explain the manner in which stimuli are recoded and manifested into a response. Importantly, PCT together with the IP model emphasizes the importance of the time-constrained nature of the cognitive system within any behavioral context (Hendy & Farrell, 1997; Hendy, et al., 1997; Hendy & Lichacz, 1999). The cognitive system is assumed to be a limited-capacity processor having both structural and resource limitations (Eysenk & Keane, 1990). The IP model claims that all factors that have traditionally been seen as contributing to operator workload can be reduced to their effect on the amount of information to be processed or to their effect on the time available for processing. The IP model, through assumed limits on the rate at which information can be processed, determines the dynamic behavior of the percep-

tual control loop. When the rate of information processing demand exceeds the capability of the cognitive structure to respond, information remains unprocessed; in other words, it is shed. Error is attributed directly to the information shed. But the IP model is an adaptive model and asserts that the human information processor will adapt to excessive demands by a strategy shift, to either extend the time available before a decision has to be made, or to adopt a less information-intensive strategy that will reduce processing time. In an absolute sense, less information-intensive processing strategies will be less precise (this is the speed vs. accuracy trade-off), leading to longer goal achievement times unless the goal is relaxed to be consistent with the loss of precision. Errors due to the time-constrained behavior of the IP model will not be corrected by the perceptual control loop. Errors due to information shedding will only be reversed when the time pressure can be returned to acceptable values (say around 70–80%, see Hendy & Farrell, 1997).

PCT extends the traditional IP account of behavior by directly addressing the contributions that an individual's past experiences, expectations, and goals, as well as feedback make to the overall performance environment. Traditionally, the IP paradigm has been criticized for its assumption that stimuli impinge on an inactive and unprepared organism (Eysenk & Keane, 1990). Mental models and goals represent an important component of the PCT environment. PCT explicitly addresses the role(s) and manner in which mental models and goal states affect information processing and ultimately behavior. In PCT terms, transient errors are the inevitable result of imperfect mental models. Imperfect mental models produce transformations that resolve some, but not all, of the uncertainty in the error signal, and while an imperfect mental model may reduce the error signal (hence, it might be termed an appropriate mental model), it may not be optimum in the sense of producing fastest goal achievement (as determined by the settling time of the loop). Errors in the sensation-perception and action formation stages can be due to either a lack of the requisite knowledge in the first place, or to retrieval errors and biases from memory structures (including decision biases). In all cases, goal achievement should eventually be possible as long as the feedback loop remains intact (this translates to the management of attention and knowledge) and the system is error tolerant (a missile fired in error is unlikely to be recoverable). Feedback is essential to the adaptation of the loop and the gaining of new knowledge about a variable under perceptual control. Hence, to gain situation awareness about a particular world state, one must attend to, or control, that state.

In sum, PCT and the IP paradigm are not so dissimilar in that both share several basic characteristics: both view people as autonomous, intentional beings who interact with the external world; stimuli are acted upon by similar processes which manipulate and transform this information into symbols which relate to things in the external world and direct behavior; both specify the symbolic processes and representations which underlie performance on

cognitive tasks; and both view the mind as a limited-capacity processor having both structural and resource limitations. More importantly, some might argue, as with the IP model, PCT provides researchers with a structure within which specific hypotheses can be tested, and enables them to predict events on the basis of the model.

7.4 PCT SYSTEMS ANALYSIS

The following procedure for systems analysis is firmly rooted in PCT, and therefore is based on the notion that humans and machines can be described in terms of a hierarchical control model. At all levels of abstraction, human activities will be directed to satisfying a hierarchical set of goals. Throughout this discussion it will be assumed that *goals* describe human set points, while the neutral term *objective* will be used to describe machine set points. Similarly *behavior* will be associated with human activity while machines will have *output*.

According to PCT the hierarchical structure of goals and objectives, from the highest level of abstraction to the lowest, represents the hierarchy of control loops that potentially will be active during the life of the system. Any goal/objective not served by a control loop has no influence over a variable in the external world, and will cause no behavior/output to be emitted. Alternatively, all system variables that are to be influenced must be associated with a goal or objective. It follows then that all goals and objectives must have the potential to be assigned to either a human in the system or to a machine. This will become a major point of departure between the PCT approach and more traditional function-task analysis (MIL–HDBK–46855A, 1999).

The described approach is shaped by and directly supports the implementation of the IP/PCT model (Hendy, 1997; Hendy & Farrell, 1997) within the Integrated Performance Modeling Environment (IPME http://199.170.148.19/toolsite/Tools/Shrtdesc/sindipme.htm) software. IPME is a tool for conducting front-end human engineering and human performance modeling for validation and verification analysis (in the past IPME has been used for traditional MFTA analyses but is entirely compatible with the shift to the PCT approach). This places PCT systems analysis within an integrated and comprehensive framework for HFE front-end analysis. Yet this linkage between PCT systems analysis and IPME could be broken while still retaining the integrity of the approach. If one was not wedded to IPME as an analysis tool, different data capture strategies could lead to the specification of the interface requirements.

7.4.1 Decomposing the Goal Structure

Some ground rules for performing a Hierarchical Goal Analysis (HGA) under this framework are as follows:

- Until assigned to human or machine all set points are generalized goals/objectives. When assigned to humans they become *perceptual* goals—they are what you want to *perceive*. (GOAL: *object of effort or ambition*: Sykes, 1982). Hence all PCT goal statements are of the form "I want to perceive [*goal statement in the form of a noun or noun clause*]." Goals are what drive *human* activity. It must be possible to assign all active goal/objective statements in an analysis, from the highest to the lowest levels, to a human or a machine in the system. Any goal/objective not assigned is not actively controlling—it is unlikely that a goal/objective will be achieved if it is not assigned to an agent within the system (it is possible that external disturbances could serendipitously drive a variable to a desired state).
- All control loops involve a variable that is influenced (coq: *controlled*) by the loop action, for example, the status of a mission, the temperature of a room, the altitude of the aircraft, the rotational speed of a propeller, etc. If there is a goal/objective, there must be an influenced variable. For humans these variables can be either internal (not directly observable by a third party) or external and therefore observable (see Figure 7.2). While the same can be said for the machine, human controllers can only know about those variables that are observable in the environment.
- Subgoals/subobjectives occur at the next level down in the hierarchy. They will be decomposed, in general, into even lower level goals/objectives. Decomposition into lower level goals follows a means-end hierarchy.

Table 7.1 demonstrates a hypothetical hierarchical decomposition of system objectives for a notional airborne platform. The decomposition is shown arbitrarily to be four layers deep; however, an analysis to any depth can be carried out by this method. It is also incomplete, as only one branch of the tree is shown. Starting with the two leftmost columns of Table 7.1, the analysis first identifies the highest-level objective and the associated variable that is to be influenced. The starting point for analysis is not arbitrary but can be reasonably set at the highest level variable that is to be controlled by the system under consideration. Information may flow from this level to and from higher-level external systems (often organizational or political) but no higher-level variables are *under the control of this system*. Each objective is then decomposed according to a means-end hierarchy until the lowest level of control is reached. Typically this will occur at the fourth or fifth level (c.f. Fig 4.2 of Beevis et al.,

Table 7.1: Hierarchical Goal Analysis Four Levels Deep

Goal/Objective (Level N)	Influenced ("Controlled") Variable	Assignment (Pick List)	Subgoals/Subobjectives (Level N-1)
Level 1			
1 ...the conduct of the mission	Mission occurring	Mission Commander	1.1 ...aircraft being preflighted 1.2 ...aircraft taking-off and departing 1.3 ...aircraft ingressing to the mission area 1.4 ...aircraft conducting mission segment *1.5 ...abnormal handled* *1.6 ...restasking completed* 1.7 ...etc.
Level 2			
1.2 ...aircraft taking off and departing	Flight status	Pilot Flying	1.2.1 ...aircraft is prepared for TO 1.2.2 ...the aircraft rolling on the runway 1.2.3 ...the aircraft intersecting the outbound track and climbing to cleared altitude
Level 3			
1.2.1 ...aircraft is prepared for take off	Preparedness for take off	Aircraft Commander	1.2.1.1 ...the TO clearance has been received and understood 1.2.1.2 ...the pre-TO checklist has been completed 1.2.1.3 ...the departure has beeen planned 1.2.1.4 ...the TO brief has been prepared 1.2.1.5 ...the TO brief has been planned 1.2.1.6 ...the TO brief has been delivered and confirmed 1.2.1.7 ...etc.
Level 4			
1.2.1.1 ...the TO clearance has been received and understood	Clearance status	Aircraft Commander	No lower levels in the analysis
1.2.1.2 ...the pre-TO checklist has been completed	Checklist status	Aircraft Commander	No lower levels in the analysis

1999; and the Abstraction Hierarchy of Vicente, 1999, p. 157). For simplicity, the clause "I want to perceive," that might be associated with each goal/objective, is replaced by an ellipsis (…) in Table 7.1. The goal structure must reflect the environmental constraints identified in parallel system engineering studies. The effects of some constraints will be obvious while others will not be evident until more detailed interface specifications are developed.

Initially no assignment is made to a specific operator or machine. The assignment of objectives is a major engineering decision that fundamentally shapes the designed system. Note that an objective could be assigned to one or more human operators or machines. While not an exact science, there are tools for guiding this process (e.g., see Beevis et al., 1999, pp. 79–102). The assignment of objectives continues until the process is complete or until a decision has been made as to which variables are going to be left uncontrolled (i.e., without error correction), with their values determined entirely from the status of their lower

level objectives. For example, if no one is actually responsible for, or is going to try to control, the conduct of the mission (the highest level objective in Table 7.1), this objective does not have to be assigned. The fact that all the sub-goals are being controlled may be sufficient. However, the requirement to address this type of issue is explicit in this process and therefore is traceable in the analysis.

Because all levels may be assigned to a human, all levels are potentially tasks in the vocabulary of MFTA. The formerly distinguishable processes of function and task analyses are now inseparable. There is now no point of demarcation between functions and tasks; hence, the hierarchical goal analysis combines what has traditionally been done under function and task analysis into a single process. This has major implications for design (as will be seen in discussing the requirements for an upward flow of information in the designed system) and clearly separates the PCT analysis from traditional MFTA. This structure supports the expansion of the analysis in either direction (up to higher levels or down to lower levels) at any time. There is no issue with representing all activities at the same level of abstraction as the decomposition proceeds (in traditional MFTA one might arrive at what was formerly described as the task level after three stages of decomposition in some branches, and at five stages in other branches). This creates difficulties with some analysis tools.

The top-level goals/objectives will generally represent the system at what might be seen as its functional purpose (conduct a particular mission), whereas the lower levels objectives generally serve an interaction at the level of the physical interface (tune a radio, release a weapon—c.f., the Abstraction Hierarchy of Vicente, 1999, p. 157). A fundamental difference between this approach and traditional MFTA is that the designer must make a decision, at each level of abstraction, as to what loops are going to be controlled. This means that goals/objectives may be assigned at all levels from the highest to the lowest. *Any goal/objective not assigned to an agent is not controlled.* Generally higher-level loops are satisfied by some combination of lower-loop activity (e.g., a logical sum of lower-loop states). *If feedback is broken at any level, there is no closed loop control, and if a loop is not controlled there is no direct error correction.*

7.4.2 Analyzing the Cognitive and Perceptual Components

Once the hierarchies of goals and objectives are identified and assigned, a detailed analysis of the cognitive/information processing aspects of the activity can be completed. Table 7.2 is a draft template for describing goal-directed human activities, based on the structure of the perceptual control loop of Figure 7.1. It contains fields for describing the input sensory processes, the output/behavioral processes and the perceptual/cognitive processes that characterize central rather than peripheral processing. Recognizing that the transformations that map sensation into perception and error into behavior draw on

Table 7.2: A Template for Analyzing the Cognitive and Perceptual Aspects of a PCT Loop

ASSIGNED OPERATOR						WORLD	
		Required knowledge states	**Perceptual/ cognitive processes (pick list)**	**Ending conditions**	**Output/behavior (pick list)**	**Output interface**	
Goal	**Output**	Declarative: •	•	•	•	•	
	Σ	Situation: •		**Influenced Variable(s) (Internal)**	IVI:	**Influenced Variable(s) (External)**	EV1: EV2: EV3:
				Initiating conditions	**Input/sensation (pick list)**	**Input interface**	
	Input			•	•	•	

knowledge structures held in memory, there are fields for both the long-term declarative and procedural knowledge, and the transient situation specific knowledge (situation awareness), required to achieve the goal.

Influenced variables, both internal and external, are tracked. Initiating and ending conditions are those that cause the attentional mechanism to shift control from one loop to another. The contents of the output and input interface fields must reflect the sensory and behavioral mechanisms that have been assumed. Initially these fields may contain generic statements such as a *hand controller or a visual display*. As design proceeds, these descriptions can be refined in light of constraints imposed by the environment and the capabilities and limitations of the human. Note that the interface between the operator and the world must bridge the gap between human sensory and effector processes, and the variable that is being influenced. Both world and operator constraints will determine the specification of this interface.

The information gathered in Table 7.2 directly feeds tools such as IPME for operator workload and performance prediction (e.g., see Hendy & Farrell, 1997). Hence the analysis and validation/verification stages are tightly linked. For example:

- The initiating and ending conditions, along with the required knowledge states provide the logic for running and terminating the task in a task network or equivalent simulation. Some of the strategic aspects of the task are contained in these logical statements, and while most

tools currently support crisp logic only, fuzzy reasoning (McNeill & Freiberger, 1993) could be incorporated in the future

- Table 7.2 is arranged so that the *Input/Sensation, Perceptual/Cognitive* and *Output/Behavior* processes fields directly feed the IP/PCT human operator information-processing module in IPME. Therefore Table 7.2 reflects the current information requirements of IPME and could be considered a minimum data set for the purposes of discussion. A useful addition to Table 7.2 would be, at least, a field containing a narrative description of the activity
- The Internal Variables (IVs) are the specific knowledge state(s) influenced ("controlled") by the loop in question
- The declarative and procedural knowledge that one brings to the job is that knowledge required to form perceptions and appropriate outputs/behaviors for *this* loop. Accumulating this knowledge across the task inventory defines the experience/training requirements for each operator and so feeds a training analysis. One could represent levels of expertise in terms of missing declarative and procedural knowledge that can be used to control the functioning of a task network simulation
- The situation specific knowledge is that which must exist, together with the declarative and procedural knowledge, to form appropriate perceptions and outputs/behaviors for *this* loop. It will include the pre-existing status of the initiating condition. Tracking the situation specific knowledge allows a task model to modify performance in terms of the level of situation awareness. By explicitly tracking the influenced variables, it should be possible to track emerging situation knowledge. Note that the knowledge required (both situation and declarative/procedural) is that which is required to do the job, rather than that gained (internal influenced variable) as a result of doing the task. That is, the knowledge required is a precursor (must exist before the goal can be actioned). Knowledge gained while the goal is being actioned is tracked by the state of the internal influenced variable
- The exit state of the ending condition (that is the state of the variable(s) when the task is completed and attention switches to another task) would be known to the operator (and manifested in an internal influenced variable).

To ease data entry requirements between the PCT analysis and IPME, Table 7.2 indicates that pick lists would be used for several of the fields rather than relying on free text entry. Obviously these data are tied to the requirements of the IP/PCT model in IPME and might be augmented or replaced by other data if a different theoretical framework were being pursued. The IP/PCT implementation within IPME models multi-task interference in the visual, auditory, psychomotor/kinesthetic, and cognitive domains (Hendy &

Table 7.3: Pick Lists for Cognitive/Perceptual Categories Describing Goal-Directed Human Activites

INPUT/SENSATION (FROM THE ANALYST)	COGNITIVE/PERCEPTUAL PROCESS CATEGORIES FROM TABLE 4 OF DCIEM 97-R-71 (DEFAULT VALUES)
VISION	
1. Central	
1.1 Text, dial reading	Verbal encoding
1.2 Pattern, spatial relationship, tracking, graphic displays	Spatial encoding, visual pattern recognition
2. Peripheral	Automatised, highly learned perception
AUDITION	
1. Tone or simple auditory signal	
2. Speech input (incidental to the primary task)	Automatised, highly learned perception
3. Auditory pattern	Passive (pre-attentive) monitoring of auditory signals
? Auditory localization	Semantic (use verbal) decoding
4. Speech input (attended to, salient to the primary task)	Spatial decoding Verbal decoding, speech recognition
KINESTHETIC	
1. Tactile	
1.1 Simple stimulus	Automatised, highly learned perception
1.2 Complex stimulus	Spatial encoding
MEMORY	
1. Recall from memory	
1.1 Accessible, familiar	
1.2 Verbally coded	Automatised
1.3 Spatially coded	Verbal decoding
1.4 Semantically coded (potentially new category)	Spatial decoding
1.5 Complex operation	Semantic (use verbal) decoding Recall
OUTPUT/BEHAVIOR (FROM THE ANALYST)	
VOICE	
1. Voice output	Speech production
PSYCHOMOTOR	
1. Manual output	
1.1 Simple	Automatised, highly learned response
1.2 Difficult but familiar	Spatial encoding
1.3 Complex and/or unfamiliar	Memorization/recall, calculation, estimation, deduction, reasoning
MEMORY	
1. Commit to memory (LTM and STM)	Memorization

Farrell, 1997). The linkage with IPME is shown formally in Table 7.3. The left-hand column shows a simple taxonomy for the input and output processes, which include activities, that are entirely internal to the human (drawing from memory structures and putting down new memory traces) as well as those that interact with the external world. These are matched to the set of perceptual/cognitive categories implemented in the IP/PCT version of IPME.

Table 7.4 is a simplified form of the Table 7.2 template for objectives that are assigned to machine agents rather than humans. For machines one might not need to track internal variables, although one would always track the set point and external influenced variables. A machine might have internal states that would be of interest from a designer's point of view, but the humans can only know about the variables that are given form in the external world. In human systems modeling, tracking the internal variables of the humans in the system allows one to explore the concept of the team mental model. Similarly, one could track the machine internal states to see if the humans and machines diverge as

Table 7.4: Template for Analyzing Non-Intelligent (Machine) System Components

ASSIGNED SYSTEM					WORLD	
			Operations	**Output**	**Output interface**	
Objective or set point		**Output**	• • •	•	•	**Influenced Variable(s) (External)**
	Σ		**Influenced Variable(s) (Internal)**	IVI: IV2:		EV1: EV2: EV3:
			Operations	**Sensor Data**	**Input interface**	
		Input	• • •	•	•	

to a common understanding of the world. The transformation operations might be tracked too as an aid to the equipment design. These transformations will also determine the states of the internal and external influenced variables.

Tables 7.5 and 7.6 illustrate how these templates might be completed using the categorization schema shown in Table 7.3. In the first case the activity involves both internal and external variables; in the second case only internal variables are involved.

7.5 ANALYSES EMERGING FROM THE PCT APPROACH

Two forms of analysis emerge from the PCT approach to systems decomposition. These analyses do not appear to be explicit in any other form of work analysis. They are:

- An analysis to investigate the potential for instabilities caused by multiple control of common variables, and
- An analysis to investigate how information from low-level goals flows up to high-level goals.

Table 7.5: Illustration of a Completed Template for an Activity Involving Both Internal and External Variables

CO PILOT						WORLD	
		Required knowledge states	**Perceptual/ cognitive processes (pick list)**	**Ending conditions**	**Output/behavior**	**Output interface**	
Goal	**Output**	Declarative: • radio op procedure • comm procedure	1 Automatised 3 Verbal encoding 5 Memory recall and memorisation	• clearance obtained and understood (e = 0)	**Voice Output** • establish radio link (voice, keyboard etc.) • transmit clearance request **Memory** • add contents of clearance to memory	• Radio head • PTT • display controller • audio pickup and/or • keyboard	
1.2.1.1 ...the TO clearance has been received and understood	Σ	Situation: • radio ON/OFF • position in checklist • aircraft status • flight plan/ intentions		**Influenced Variable(s) (Internal)**	IVI: Clearance belief (1) IV2: Clearance status (1)	**Influenced Variable(s) (External)**	EV1:ATC clearance (1) EV2: Clearance request (1) EV3: VHF 1 Freq.
				Initiating conditions	**Input/sensation (pick list)**	**Input interface**	
	Input	• clearance status		• clearance not obtained and understood (e ≠ 0) • if e ≠ 0 obtain clearance else do nothing	**Memory** (1.2 recall verbally encoded) **Vision** (1.1 Central, text) **Audition** (4 Attended speech) **Kinaesthetic** (1.1 Simple tactile input)	• audio transducer and/or visual display • published charts/ frequencies	

7.5.1 Stability Analysis

In any system both humans and machines are exerting an influence over many variables in the environment. Depending on the division of control at any point in time (shared or segregated), the commonality of goals/references, and the compatibility of the transformation functions that shape input and output signals, the system might be either stable or unstable. The potential for instability is obvious when two agents are simultaneously trying to drive a variable in different directions according to incompatible set-points or internal transformations (this is what sometimes happens when two people, approaching head-on, attempt to pass one another in a corridor). This analysis investigates the existence of potential instabilities between human-human and human-machine control. In this analysis human and machine control are placed side by side.

Table 7.6: Illustration of a Completed Template for an Activity Involving Internal Only

AIRCRAFT COMMANDER						WORLD	
		Required knowledge states	**Perceptual/ cognitive processes (pick list)**	**Ending conditions**	**Output/behavior (pick list)**	**Output interface**	
Goal	**Output**	Declarative: • ATC procedures • aircraft performance	3 Verbal decoding 4 Spatial decoding 5 Memory recall and memorization	• plan made (e = 0)	**Memory** • add contents of plan to memory	•	
1.2.1.3 ...the departure has been planned	Σ	Situation: • ATC clearance (1) • flight plan • NOTAMs • plan status		**Influenced Variable(s) (Internal)**	IV1: Plan (1) IV2: Plan status (1)	**Influenced Variable(s) (External)**	EV1: EV2: EV3:
				Initiating conditions	**Input/sensation (pick list)**	**Input interface**	
	Input			• plan not made (e ≠ 0) • if e ≠ 0 make plan	**Memory** (1.5 Complex operation)	•	

The first three columns of Table 7.7 can be generated by parsing the database produced by the decomposition of system goals/objectives. Entries are sorted by external variable. The potential for simultaneous control can be determined by examining the goal/objective assignments by agent (which operator or machine). This information is entered in Column 4. If a network simulation or other time based tool has been run, this could be a time-based analysis. Training should prevent instabilities occurring due to systemic differences in procedural or declarative knowledge; however, there is the potential for differences in transient or situation specific knowledge states. Loss of synchronization in situation awareness is likely to vary over the mission time.

As the systems designer develops strategies to resolve these conflicts, new goals/objectives will be added to the master list (shown in the last column of Table 7.7). Note that this is a potentially recursive process. Generally one might exercise this process once or at most twice. Strategies for attaining stable multiagent control include:

- Separation of control
- Ensuring compatible set points and transformation functions, or
- Allowing one controller to dominate through high-loop gain.

Most resolution strategies will involve informing each of the players as to who or what is controlling. If control is not separated (including allowing one

Table 7.7: Stability Analysis

Influenced Variable (external)	Operator or System Assignment	Goal/Objective/ Reference	Potential for Simultaneous Control (pick list)	For stable control
	• Operator 1 • Operator 1 • Operator 2 • FMS	• ...the aircraft rotated for TO • ...the glideslope intersected • ...the altitude maintained • ...the altitude at x feet	• Operator 1 and FMS • Operator 1 and Operator 2 • Operator 1 and 2 and FMS	• push/pull disconnects AP with voice and visual warnings • hand over procedures, voice and visual warning if multiple inputs are detected • as above

loop to dominate), it must be blended smoothly by ensuring that all agents have compatible goals and internal transformation functions.

7.5.2 Support to Higher-Level Goals

This analysis should start at the lowest level and work up. This is the reverse process to the function/task decomposition. Traditional top-down decomposition by function and task traces the downward flow of information in a system. This matches the downward flow of action exerted on the system that is inherent in the means-end hierarchy. However, to exercise control (error correction) at any level, the state of the influenced variable(s) at that level must be fed back. In traditional MFTA there is no explicit mechanism for tracing the upward flow of information in the system which is required to achieve error correction at all levels. In Table 7.8, the database generated by the decomposition of system goals/objectives is parsed by goal/objective. Generally this would be started one level up from the lowest level goal (the example in Table 7.8 was arbitrarily completed at the top two levels of this analysis). A level N goal and its assignment are listed in the first two columns of Table 7.8. For each goal/objective at level N, the subgoals/ sub-objectives at level N–1 are now listed, along with their assignments, in the next two columns of Table 7.8. The data in the first four columns of Table 7.8 comes directly from the database.

This process yields two new pieces of information:

- The highest-level goal is controlled by each machine in the system. This is particularly salient in considering decision support systems. Few machine implementations share top-level goals with the human (e.g., in modes other than full auto, an aircraft Flight Management System ([FMS]) has no top level objective to complete a flight to a given location, even though it supports all the lower level goals of

Table 7.8: Analysis of the Upward Flow of Information in the System (Support to Higher-Level Goals)

Goal/objective	Assignment	Subgoals/subobjectives	Assignment	To support higher level goal/objective
		1.1 ...aircraft being pre-flighted	AC	AC to brief MC
		1.2 ...aircraft taking-off and departing	PF	PF to brief MC
		1.3 ...aircraft ingressing to the mission area	Mission Computer	TD to display current aircraft position
		1.4 ...aircraft conducting mission segment	Mission Computer	As above with mission status indicated
		1.5 ...abnormal handled	AC	AC to brief MC
		1.6 ...retasking completed	All crew	All stations to brief MC
		1.7 ...etc.		

maintaining a heading, altitude, etc.). Obviously if the machine/automation doesn't support all the lower-level goals, it doesn't completely support the level above

- Even if all lower-level goals are supported, this information has to be combined in some way to provide feedback at the level above. This will apply to both human and machine controllers. Interface design should facilitate this process, and the fourth column in Table 7.8 provides the additional goals that the interface must support to achieve this end. The analyst must provide the resolution shown in the last column of Table 7.8. This analysis will make the designer consider *how the information is passed up* to support higher-level functions and activities. One should not rely excessively on human memory in performing this upward synthesis of information. There are various mechanisms for fusing/storing/manipulating lower-level data to support higher-level goals/objectives and their assignments. This is a potential application for various levels of machine "intelligence" and decision aids.

Note that if the higher-level goal, and the supporting subgoals one level down, are all assigned to the same agent, then there may be no need to flow the information up. It could be assumed that as all goals are assigned to the same agent, that agent would have access to all information. However, the existence of the higher-level goal involves two processes:

- The upwards flow of information, and
- The synthesis of that information to form a perception that can be compared to the goal state.

Design should accommodate both requirements. The intent of data fusion systems is to roll up low-level data to support a higher-level perception ("big picture"). Decision support systems have similar requirements.

7.6 APPLICATION OF PCT SYSTEMS ANALYSIS

In this section, the application of PCT-based analyses to systems requirements specification is discussed. First, the use of a PCT analysis of a future land forces command and control system is described, specifically as it compares to the more traditional MFTA approach. It will be seen that the PCT analysis provides additional insight into the requirements for interface design. Second, the potential benefits of the PCT approach in analyzing adaptive intelligent interfaces are considered. Developing processes for capturing the requirements of these types of interfaces is perhaps the next grand challenge for human factors specialists. Because a PCT based analysis develops a hierarchical goal structure for the system, it appears to fit well with the requirements for goal-plan tracking structures that often drive such systems (Edwards, 1997) and with the overall notion of explicit user models (Edwards & Sinclair, 2000).

7.6.1 Land Forces Command System

An MFTA of the Canadian Land Forces Tactical Battlefield Command System (TBCS) was completed in preparation for writing detailed requirements specifications and future procurement action. A single segment of the TBCS analysis was selected for a proof-of-concept application of the PCT based systems analysis and comparison with the traditional approach (Dahn & Lowdon, 2000). This work was performed under contract to BAE Systems—Canada, and Micro Analysis and Design (MA&D) of Boulder, Colorado, USA. Both contractors were highly skilled and practiced in the use of traditional MFTA, but had minimum exposure to the PCT approach.

Specific objectives of the study included:

- Assessment of the applicability of the PCT protocol, relative to traditional analyses
- Qualitative assessment of PCT implementation issues, and
- Identification of lessons learned.

The following extracts from the contractor's report summarize the outcome of this exercise:

The comparison of PCT and non-PCT results suggests that the PCT approach is highly appropriate to conceptual phase-analyses, and offers benefits over the traditional MIL–HDBK–46855 approach used previously. The analysis of "support to higher-level goals" increases the likelihood that workload associated with monitoring/supervising/controlling the performance of others is captured and correctly represented in the model. This was evidenced by the 37 additional goals identified during this analysis which had been missed during the original TBCS analysis. Admittedly, some of the lower-level tasks should have been identified during the previous analysis, but there is still currently no logical process for identifying workload associated with the performance of functions above the task level. Similarly, the stability analysis provides a structured process for identifying situations when a particular variable could potentially experience multiple controls—i.e., either human-human or human-machine. This process also supports the identification of solutions to ensure the potential instabilities are effectively managed—e.g., through system design, procedures, etc. The traditional approach provides no process for ensuring that instability issues are adequately addressed during the conceptual-phase analysis.

There is another aspect of PCT that warrants consideration. Although the analysts at BAE Systems admit to limited exposure to the process, the general perception is that PCT is not as intuitive as the traditional methods, and probably requires a greater degree of training/experience to become truly proficient. This is particularly significant for those not previously exposed to control theory. There is also the consideration that the end-user/customer/ Subject Matter Experts (SMEs) will have similar difficulty in grasping/comprehending the significant aspects of the process. This concern should certainly not be a show stopper, particularly since the limited experience of the analysts renders it premature to pass judgment in this area. However, the utility of the process is clearly an issue that warrants consideration.

The success of the PCT analysis in identifying 37 additional goals, missed in the traditional MFTA, is heartening. These goals were all associated with support to higher-level intent. For the TBCS, it would seem that established command and control procedures had largely addressed the issue of stability, as the potential for simultaneous control was not a factor in the analysis of this segment of operation.

The contractors identified ease of use as a potential problem in the application of this new process. While the validity of a process is always of primary purpose, ease of use influences the acceptance of the method by a design team.

Ease of use is also a major factor in the timeliness of the deliverables. Ultimately ease of use will be a contributing factor in whether the process is used in design. While MFTA is familiar to anyone with basic project management or systems engineering skills, PCT systems analysis varies sufficiently from this model that a learning curve is to be expected (the level of specialist knowledge required of CWA has already been identified as an issue by Chin, Sanderson, and Watson ([1999]) for a similar domain of application).

PCT systems analysis was implemented using IPME for this proof-of-concept study. IPME currently makes no particular concessions to the PCT method, although it is compatible with the process. The ease of applying the PCT method is likely to be improved when an interface is developed that directly supports PCT (Dahn & Lowdon, 2000). Intelligent tutoring and help can also lead the analyst through the process (e.g., see Edwards, Hendy, & Scott, 2000).

7.6.2 Intelligent Adaptive Interfaces

The concepts that make up perceptual control theory, taken individually, are not unique. They appear in many theories, some which have been around for a very long time. What makes the contribution unique is the combination of those concepts into an approach that promises to add important insights into the analysis, design, and implementation of intelligent, adaptive systems.

It is difficult to say how long the notion of purposeful behavior has been with us but it is discussed early in the life of psychology by William James, among others, later in elaborated theories of learning, and still later in cognitive psychology and information processing theories. More recently, various subfields in artificial intelligence (e.g., the subfield of planning) deal with purposeful behavior.

Many of the concepts used in the description, explanation, and prediction of purposeful behavior overlap these disciplines, concepts like goals, plans, states, feedback, hierarchies, closed loops, conflict, cooperation, and so on. The different ways these concepts are defined, elaborated, and used can provide useful cross-disciplinary insights.

In the interest of building intelligent, adaptive systems, it is worthwhile to take some time to examine how some of those concepts are treated and what other disciplines have to say to PCT about how they might be used. In particular, this section will look at PCT from the perspective of some of the concepts used in AI for building intelligent, adaptive systems. Aspects of its purpose include clarifying differences in the use of same or similar terms, providing a basis for transforming usage between the two fields, identifying apparent discrepancies, highlighting areas where questions remain, and laying a foundation for future refinement and elaboration of ideas necessary to building those systems.

7.6.3 Two Views

PCT's basic unit of purposeful behavior is the reference signal, taken from control theory. Unlike control theory, the reference signal is not given from the outside by some observer or user of the control system but emerges from within. For humans, reference signals are inferred and cannot be observed directly.

Reference signals exist at many different levels of abstraction and are linked together in ways that can help define all complex systems. At any given level of abstraction, a reference signal is compared to a perceptual signal (at that same level) and if a difference is detected, the system produces responses intended to change the perceptual signal in ways that will reduce that difference. The results of those changes provide feedback in the form of an altered perceptual signal that moves through the same process again. All of this happens in a closed loop system not easily described in terms of traditional notions of cause and effect.

In AI, in the area of planning, for example, the notion of goal is a key concept. In general, a goal is a state and there are two states of particular importance in plan generation work: the current state of the world and a desired state held by some entity (person or machine). In a similar way to the negative feedback system in PCT, the object of planning is to reduce the difference between the current state and the desired state and, in the case of AI, this is done through the application of operators. Early work in this area conducted by Newell and Simon (1963, 1972) made use of a means-ends analysis in the application of operators to reduce that difference.

During this process, it is important to understand the preconditions, or conditions for applying an operator, and to make sure they are met before attempting to apply the operator. A second aspect of the process is understanding the effects that will be produced in the world when the operator is applied. Effects add and subtract facts from the system's current model of the world. Examining preconditions and effects leads to another aspect of AI planning systems: the detection and avoidance of conflicts among goals and subgoals. For example, satisfying one goal may produce an effect that prevents the pursuit of an important, related goal or may actually reverse the effects produced by the successful pursuit of earlier goals.

Similar to PCT, there are different levels at which goals occur, both higher and lower relative to the goal currently in focus. Subgoals support the achievement of goals above them and supergoals are the goals that motivate the goals below. In other words, subgoals are the means by which supergoals (ends) are achieved.

From these two descriptions, it is clear that there are a number of points on which PCT and AI seem to agree. They both support the notion of goals, the concept of differences between goal states and current world states, and the need to reduce those differences to bring the (perception of the) world in line with the desired goals. Further, they support the notion of an organized hier-

archy of goals. The treatment of preconditions and effects in AI is accommodated in PCT through perceptual signals that reflect the nature of the current state of the world and effects that operate on it. An inner PCT loop, referred to in earlier sections, would seem to allow for rehearsals of reference, perceptual and error signals, and this is consistent with a reasoning component in AI systems that examines optional subgoal structures that might lead to a successful plan. Further elaboration of these inner loops and their relationships within and across levels would be instructive.

7.6.4 The Notion of Plans

A corresponding and complementary concept to goals in AI is the notion of plans. Plans are sets of subgoals (possibly only one) that support the achievement of some higher-level goal. The goal is specified at some level and subgoals are identified that will help achieve that goal. The goal of course is a supergoal relative to all the subgoals in the plan but we refer to it simply as a goal and its meaning will be understood in context.

In PCT, there seems to be some reluctance to embrace the notion of plans either because it seems to imply knowing what may not be knowable (Powers, 1993) or that plans somehow reflect brittle, repeated actions that cannot accommodate changes in the real world:

> There can be no plan of action precise enough to carry out this process that the driver accomplishes every day. What we really see is not a series of repeated actions that have repeated consequences, but a series of variable actions that have repeated consequences. (Forssell, 1997, p. 9).

Subgoals seem to be implied; however, PCT's hierarchical control model points to a need for such a hierarchy based on perceptual dependencies.

> The rationale for hierarchical classes of perceptual control is based on the observation that certain types of perception depend on the existence of others. Higher level perceptions depend on (and, thus, are a function of) lower level perceptions. (Forsell, 1997, p. 4)

Since a reference signal is identified in PCT as synonymous with goal, it stands to reason that a hierarchy of such reference signals is a hierarchy of goals. Reference signals at lower levels support those higher up and are required to reduce the error signals at higher levels. Thus, those lower-level signals would seem to constitute ways (plans) for achieving the higher-level signal (goal).

Passing over PCT objections to the use of the word "plan," what seems problematic is how the various reference signals operate in concert with each other to produce intelligent, adaptive behavior. That is, how is a set of low-level reference signals chosen from many possible alternate sets in a way that permits the successful achievement of a higher-level reference signal?

Assuming all these relationships to be hardwired flies in the face of a concept like adaptation and even the claims by PCT theorists themselves. Appealing again to some internal operations (inner loops) provides a way of linking PCT concepts with adaptation to changing circumstances in the world.

In many circumstances, goal achievement requires only the application of previously learned behavior in the presence of familiar circumstances (some world state), monitoring the environment for changes, and using other learned methods to adjust behavior so as to accommodate those changes (as in PCT's familiar example of driving a car to some desired destination).

In contrast, planning an action or activity that is new or that may require substantial variation from past behavior demands some kind of (inner loop) rehearsal for the behavior to be effective. Some form of rehearsal and even practice may be necessary to establish the new behavior required to achieve the desired goal. Participating low-level behaviors, or at least aspects of those behaviors, may not need to be learned.

To help bridge the gap between PCT and AI approaches on the issue of plans, PCT might be more explicit about the relationships among reference signals in its hierarchically organized systems. Specific examples of adaptive systems created using the PCT approach would be helpful, along with the more general principles and methods used to define relationships within a PCT hierarchy.

7.6.5 The Nature of Hierarchies

A key difficulty in constructing hierarchies is establishing clear criteria for defining their levels. Lower layers seem more easily identifiable and the criteria for differentiating them easier to specify. Likely, this is due to the fact that "behaviors" at those levels are far more limited, simpler and easier to identify, detect, and measure than those at higher levels. The PCT hierarchy illustrates these points nicely.

Powers (1973a, 1990) proposed a hierarchy of eleven levels with the lowest level systems controlling perceptions that represent intensity from the environment. The next level is sensations, which includes things like sounds and colors and which are functions of intensities at the lowest level. A third level supports the control of configurations, which are combinations of sensations and, so it goes, on up the hierarchy.

Descriptions of these low-level control systems seem quite reasonable; it seems clear how they support one another and how a system might be con-

structed based on such a hierarchy. As you ascend the hierarchy, however, things become increasingly tentative and questionable. Incidentally, this point can be made about many non-PCT hierarchies as well.

At the upper end of Powers' hierarchy are levels called categories and programs. Categories capture the notion of class membership and programs specify "if-then" contingencies. At these levels, Powers has moved into areas that involve symbolic systems and it is not at all clear why that shift occurs and how criteria used to establish those levels relate to criteria used to differentiate lower levels of the hierarchy. Powers' highest levels of principles and system concepts refer to very broad notions, namely, generalizable rules and disciplines such as science, mathematics, and art.

Two of these higher levels are worth noting relative to the concepts of goals and plans in AI. The first is the next level up from categories, called sequences. Sequences represent unique orderings of lower-order perceptions and provide a first hint at how PCT might be seen to accommodate the notion of plans. The next level up from that one is programs, as defined in the previous paragraph. The notion of a program captures much of what might be considered a goal and the plan for achieving it, including the specification of contingencies that contribute to the adaptiveness of the plan or to the use of a different plan altogether.

Returning to the nature of hierarchies, a little reflection on the levels in this proposed PCT hierarchy will show that it is not at all clear how the higher levels are related in any coherently logical, smoothly transitioning way to the lower levels, nor is it clear whether the criteria used to differentiate any two levels bear any resemblance to those used to distinguish any other pairs. More theoretical work is needed of course, along with implementations and demonstrations of systems that attempt to use those higher levels.

7.6.6 Hierarchies and Plans

To further elaborate the notion of hierarchies and plans, the following example from Hendy et al. (2000) is instructive.

> Human behavior is commonly described with action verbs, rather than perceptual states. For instance, going from the first floor of a building to the top floor, one might walk to the elevator, press the door button, enter the elevator, press the numbered button, wait for the doors to close then open, and exit the elevator. This sequence of events might lead the reader to believe that the human is directly controlling their actions, perhaps in response to some stimulus. However, implicit in this elevator scenario is the goal of reaching the top floor, the initial

> perception of being on the first floor, and a series of actions that move the current perception closer to the goal state (p. 7).

Although the example is meant to show that a characterization using action verbs can be represented as changes in perceptual states, it also provides a convenient way to illustrate how AI might represent those perceptual states in a hierarchy of goals and subgoals (plans). One of many useful rule sets for understanding such examples, developed by Abelson some years ago (see Schank & Abelson, 1977), is formulated as follows:

Actions Cause States; States Enable Actions

Applying this rule set to the above example, along with the concepts of goal and subgoal (states) and actions (operators), we have:

State 1	(Subgoal5):	Being on first floor
Action 1	(Opt1):	Walk to elevator
State 2	(Subgoal4):	Being at elevator
Action 2	(Opt2):	Press button for elevator
State 3	(Subgoal3):	Elevator at first floor (and) doors open
Action 3	(Opt1):	Walk into (Enter) elevator
State 4	(Subgoal2):	Being in elevator
Action 4	(Opt2):	Press button for desired floor
State 5	(Subgoal1):	Elevator at desired floor (and) doors open
Action 5	(Opt1):	Walk out of (Exit) elevator
State 6	(Goal):	Being on the top floor

If the states above adequately reflect conditions in the real world, then the actions that follow are consistent not only with output functions invoked by the error signals of PCT but also with cognitive theories which require the evaluation of testable conditions to trigger consequent behavior and even behavioral notions of chained responses to (antecedent controlling) discriminative stimuli.

In AI, the entire sequence can be thought of as a plan for achieving the final goal (State 6: *Being on the top floor*), but actions at any level can be understood to achieve the goal represented in the following state. Each state is a subgoal in the service of the final goal (state) and the actions and subgoals together constitute a plan for reaching that final goal, at whatever level that may be defined.

Comparing this to PCT, subgoals are equivalent to controlled real-world variables and thus to perceptual states that occur as consequences of an entity's acting on the world, and from the actions (and properties [states]) of other entities (e.g., the actions and states of the elevator).

As actions achieve each of the subgoals, the final goal (reference signal): *being on the top floor* is compared against the current world state and an error signal gen-

erated which shows that the difference is less than before. That process is repeated as each of the subgoals is achieved until the error signal is effectively zero.

If the plan is a clear one and the currently satisfied condition is a recognizable condition within the plan, the next action to reduce that difference with the final goal should be clear also. If the next "logical" condition is not satisfied, however, replanning may need to take place.

What happens for example if the elevator is out of order? Cognitive theories permit replanning by allowing the person to use alternative routes such as taking the stairs or asking an attendant for help in getting the elevator working. Conditions in this scenario that could influence whether and when such replanning takes place might include a light above the elevator door that provides information about where the elevator is and whether it appears to be working.

How might PCT handle the problem of replanning? Will the person determine that the stairs are a best alternative under the circumstances or are they left standing at the elevator repeatedly pushing the button? That is, does the behavior continue to repeat since the error signal remains the same or does the control system have a way to effectively replan? An intriguing question, and one that the reader, in finding an answer, should also uncover other areas of correspondence between AI and PCT.

7.6.7 Hierarchies Revisited

As indicated, there are fuzzy aspects to hierarchies, deciding what criteria to use in differentiating the different levels, how many there are, and so on. In the example above, the action-state sequence can be thought of as a kind of hierarchy leading to a final goal.

That particular concept of hierarchy is based primarily on identifying an appropriate sequence of actions and states rather than identifying different levels of abstraction or granularity in those actions and states. In a sense, the whole sequence, as described, could be thought of as existing at one level in an abstraction hierarchy. Within the PCT hierarchy, that might be the program's level.

To illustrate, contrast the action-sequence hierarchy with that involved when performing any given action in that sequence, for example, pressing the elevator button. To accomplish that, one has to move the hand, extend a finger and perhaps tilt the body slightly forward as the finger comes into contact with the elevator button. Those acts involve bringing muscles into play, altering muscle cells and, at an even more fundamental level, firing neural impulses. The goal of pushing the button then is achieved by a set of lower-level actions in a different kind of hierarchy than the action-sequence one above, by body movement and position, the flexing of muscles, changes in muscle cells, and activation of neural impulses.

Table 7.9: Classification of Goal Relationships

	Negative Interactions	**Positive Interactions**
Internal	Conflict: Mutually opposing goals held by a single entity (person or machine)	Overlap: Goals achieved more easily together than apart
External	Competition: Mutually opposing goals held by different entities (people or machines)	Concord: Mutually beneficial goal possessed by several entities

Powers admits that he doesn't know how many levels there are and that he doesn't know what determines the reference signal for the highest level. He points out that there are perhaps thousands or even more control systems, for example, one or more for each of the 800 muscles in the body. He also recognizes that some of those systems may operate independently and simultaneously, or even dependently.

Much of our understanding about hierarchical representations for the design and implementation of adaptive systems is yet to come. Other forms of representation are also possible and should be considered for what they have to say to hierarchical systems such as those in the PCT and AI fields.

7.6.8 Final Thoughts on Shared Concepts for Intelligent Adaptive Interfaces

This section has explored only a few of the concepts from PCT and AI that continue to be important in the analysis, design, and implementation of intelligent adaptive interfaces. It is not meant to be a thorough review and comparison of the two areas, only a primer on some of the relations among their concepts.

Many other aspects of PCT and AI could have been discussed such as how each might handle conflict and cooperation, within and among sets of control systems (people and machines), or how they might account for the modeling of one set of control structures by another in the service of understanding what goals (reference signals) are appropriate in interpersonal situations (see for example Edwards & Sinclair, 2000).

Since some time has been spent on stability analysis in this paper, classification of goal relationships from AI might motivate an exploration of how PCT can accommodate these in its approach to building adaptive control sys-

tems. The following is a modification from Wilensky (1983), and shows one way to classify goal relationships.

Competition, cooperation, and coordination of these different relationships among goals, within and across entities, needs to be accommodated within PCT and it will be instructive to explore just how PCT deals with each.

Finally, the questions raised here and elsewhere in this section are not meant to argue for PCT or AI in the analysis, design, and implementation of intelligent adaptive interfaces, but rather to encourage the elaboration of ideas within both so that correspondences and differences can be understood more clearly. Achieving that goal will provide real opportunities for mutually beneficial discussion and effective refinement and extension in adaptive system building.

7.7 DISCUSSION

This paper proposes a hierarchical goal analysis, based on the theoretical underpinnings of PCT, as an alternative to the traditional MFTA approach to human systems analysis. It is argued that a PCT-based HGA addresses many of the deficiencies associated with traditional MFTA. The overall structure of PCT-based HGA is similar to hierarchical task analysis (Annett & Cunningham, 2000) which also assumes that humans behave as goal-driven, closed-loop controllers. PCT, however, brings a more formal application of control theory into play and along with this comes the need for error correction at all levels within the hierarchy. The formal representation of closed-loop control theory within PCT spawned the two emerging analyses related to the detection of instabilities in the system and support for higher-level goals.

Vicente (1999) describes a comprehensive design and analysis process he calls *Cognitive Work Analysis* (CWA), based on Ramussen's approach (Rasmussen, Pejtersen, & Goodstein, 1994) to cognitive systems engineering. He argues (p. 48) that "…work analysis should begin with, and give primary importance to, the constraints that the environment imposes on worker's actions." The desired objective of this approach is said to be to "…ensure that workers will acquire a veridical mental model of the environment, so their understanding corresponds as closely as possible, to the actual behavior of the context with which they interact." CWA also attempts to avoid the pitfalls of traditional MFTA, but does so by taking a more revolutionary approach. Vicente (1999, Chap 3) has been a most vocal critic of what he calls the traditional methods of task analysis and points to what he believes are fatal flaws, particularly for the design of complex open systems, with any method that is mission or scenario based. Such approaches are seen to be normative and instructional as missions, and scenarios are thought to result in designs that impose a prescribed sequence of actions upon workers. By implication this would include PCT-based HGA because of its association with MFTA.

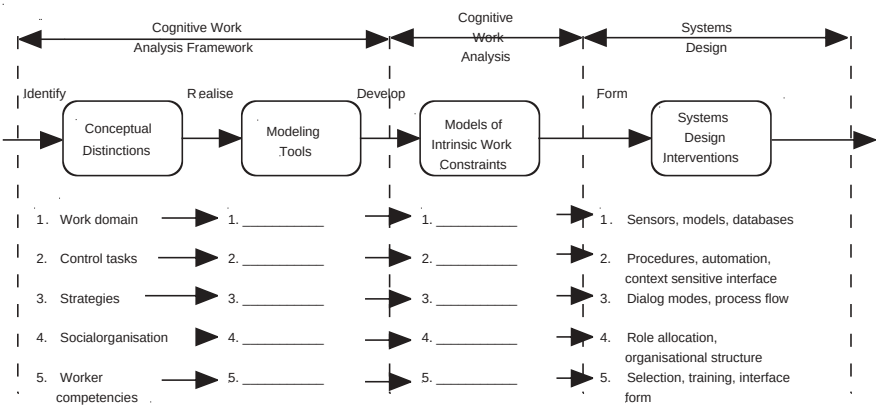


Figure 7.4: A framework for cognitive work analysis. (From Vicente, 1999, p. 136.).

Because of this uncompromising position, it is worthwhile making some comparisons between the methods.

CWA consists of a process and an evolving series of tools for conducting this process. The framework of CWA is shown in Figure 7.4 (the modeling tools and models that bridge the gap between the five conceptual distinctions and the corresponding five design interventions have been omitted from the diagram). Figure 7.4 outlines a structured process that has much in common with procedures that have evolved for human systems analysis as described earlier in this paper. CWA differs from the approach to human systems analysis described previously, not so much by the overall form of the process but by the specific methods used to implement it. While the necessary link between CWA and the rest of the engineering process has not been formally elaborated, there is sufficient commonality with other methods that one could imagine CWA partially mapping onto the structure of Figure 7.1. The one area that doesn't appear to be covered in CWA is a mechanism for verifying the design decisions, the final step in the iterative design process shown in Figure 7.1.

On the other hand, the overall framework of the PCT systems analysis method largely parallels that of traditional MFTA, and therefore fits within the systems engineering process, described in Figure 7.1. The differences between MFTA and PCT stem from the combination of the function and task analysis phases in PCT into a single process of hierarchical goal analysis, and in the two emerging analyses (stability analysis and support to higher level functions) that come from this method. At the heart of these differences is the realization that all goals, from the highest to the lowest, are candidates for assignment to a system agent. While the data requirements for the two methods are a little different, there is not a huge conceptual gap between the HGA and MFTA. The basic flaw in traditional MFTA is a result of an analysis

method that explicitly traces the downward flow of information in a system but only captures upward flow fortuitously.

Fundamental to PCT is the notion that goal-directed human activity is driven by a process of closed-loop negative feedback control. Vicente (1999) makes repeated references to a closed-loop negative feedback model of the human information processor throughout his book, and uses this model to support several of his assertions. Yet he doubts the appropriateness of this form of representation for human action in complex systems (p. 73). His position appears to stem from the difficulty in giving explicit form to the transformation functions that map sensation into perception and error into action according to Powers' model. This difficulty is clearly acknowledged, but it doesn't in any way invalidate the model. The are many complex physical systems operating strictly according to a closed-loop feedback process that would possibly defeat any analyst to derive transformation functions from first principle calculations. That is one of the reasons that systems identification is used, to build empirical models of complex processes from which stable controllers can be built (DiStefano, Stubberud, & Williams, 1967; Phillips & Harbor, 1991, p. 57, p. 355). Fuzzy controllers have extended this capability from purely deterministic control to rule-based controllers. Just as is the case with a human controller, it is difficult to predict the output of a fuzzy controller under all conditions. However, this doesn't change the fact that the system is under set-point-driven, closed-loop, negative feedback control.

The extensive preoccupation with timelines and sequence that some authors see reflected in traditional methods of analysis appears to be taken out of context in many cases. For example, Vicente (1999) is most concerned with the inability to predict sequence or timing of activities in the presence of disturbances. He uses the example of vehicle guidance in the presence of wind gusts to underscore his point. He argues (p. 76) that because one cannot necessarily predict the pattern of wind disturbances, one cannot determine the timeline of human actions that are intended to counter the effects of the disturbance (note that Bourbon, 1996, uses precisely the same example to argue that it is actually the presence of closed-loop control that allows an operator to cope with these unpredictable disturbances). The message seems to be that you can't design an interface between the operator and the vehicle in the light of this uncertainty. Yet one doesn't need to know the timeline of steering actions to propose that a manual control in the form of a steering wheel be provided for vehicular control. The requirement for such an interface would come from a detailed MFTA, which is relatively neutral with respect to timeline or sequence. Further, knowing the statistical properties of the wind disturbance (mean, variance, max), the dynamic response of the vehicle and the transfer function of the operator, one could determine all design parameters for the interface (size of steering wheel, steering ratio, power steering requirements, augmented visual displays, the need for control quickening, etc.).

Time or sequence is not a parameter in traditional function or task analysis. Top-down decomposition of functions into lower level functions and tasks can be made entirely without reference to time or sequence. Decomposition proceeds in terms of an action means-end hierarchy (c.f., Vicente, 1999, p. 162). Even Functional Flow Diagrams (e.g., Beevis et al., 1999, p. 53) are only weakly time related. They capture sequence when sequence is important to system function (you can't fight the mission until you get airborne, etc.) but accommodate both serial and parallel branching. Sequence becomes an issue only when one needs to chart inter-dependencies between communicating entities (humans and machines), capture a logical progression with time, or establish safety interlocks between systems. Task sequence information might be used in tools for timeline analysis and performance/workload prediction, but many modern tools are not restricted to cases with prescribed task sequences (Hendy & Farrell, 1997). It is also important to understand the role of timeline and performance prediction in the total framework of front-end human engineering analyses. These methods are used to verify that the emerging design is likely to meet specified performance criteria, at least for normal operation. It is not necessary to predict system performance under all possible sequences of operation to answer this question (Hendy & Farrell, 1997).

While recognizing the different tools involved (an action means-end hierarchical function decomposition versus a structural means-end abstraction hierarchy) it would seem that work domain analysis in CWA equates to some extent to function analysis in MFTA (Vicente, 1999, p. 212). Similarly there are equivalencies between control task analysis in CWA and task analysis in MFTA. However, while function and task analysis are inexorably linked through a hierarchical decomposition in MFTA, or through the HGA in a PCT-based analysis, the link between work domain analysis and control task analysis is less well defined. Some loose equivalencies between MFTA, CWA, and PCT are shown in Table 7.10 (it is realized that these equivalencies are open to interpretation, as there are extensive differences in the overall implementation and order of the processes that go beyond similarities in individual steps). It is assumed that MFTA and PCT analyses would be performed in a systems engineering context as shown in Figure 7.3 and therefore would include a detailed analysis of technological and environmental constraints.

It is stated that (Vicente, 1999, p. 182) "…control task analysis should identify what needs to be done, independently of how or by whom…." The notion that control tasks act on the work domain (Vicente, 1999, Fig. 8.1) should be very familiar to PCT proponents as it is a form of analysis that represents human and machine activities by a common format and describes their interaction through a vector of variables in the external world (p. 190). Yet Vicente (1999) is less clear about how one generates the database of what has to be done in the system, without driving this process bottom up from an assumed display concept or physical description of the plant, or top down

Table 7.10: Some Loose Equivalencies Between Procedures Used in MFTA, CWA, and PCT Systems Analysis

MATA	CWA	PCT
Mission Analysis	Not specified	Mission Analysis
Scenario Generation	Not specified	Scenario Generation
Function Analysis	Work Domain Analysis	
Function Allocation	Social Organization and Cooperation Analysis	Hierarchical Goal Analysis
Task Analysis	Control Task Analysis	
Critical Task Analysis	Strategies	
Training Analysis	Worker Competencies	Tracking declarative and procedural knowledge by operator
Performance Prediction	Not specified	Integrated Performance Modeling Environment (IPME)

from some form of function, task, or hierarchical goal analysis (c.f., Vicente, 1999, p. 201). His example for the DURESS task (pp. 181–214) implies a top-down hierarchical decomposition of the functional purpose of the system (by goal and subgoal), with the decision ladder providing structure to this process (Fig. 8.9). The HGA of PCT allows for the decomposition of goals from the highest to the lowest to be independent of *who* but has an implicit *how* associated with the way goals are broken down into subgoals in accordance with a means-end hierarchy. It could be argued that control task analysis in CWA also implies the *how* by the pathways traced in the decision ladder.

What is apparent is that CWA doesn't easily accommodate manpower and personnel trade-off studies. Manning is really an exercise in accountability and workload management. Both aspects are of concern for CWA, but there isn't a clear method within the process for the implementation of trade-off studies to examine these issues. The introduction of automation into a system combines accountability and workload aspects with direct system performance factors such as the latencies involved in meeting goals/objectives and the precision with which these goals can be achieved. There are also human-in-the-loop concerns

related to the development and maintenance of situation awareness that one should address during the design phase. PCT combined with the IPME tool can quantitatively address these issues in both a static and dynamic representation. With CWA the assessment is, at best, subjective.

In CWA it appears that the intention of the Abstraction Hierarchy (AH) is to define all the objects (variables) in the work domain that might be acted on, and store information about them in a data base (Vicente, 1999, Fig. 8.6) for display at the operator interface. PCT identifies both a vector of environmental variables that must be influenced (*controlled*) if the hierarchy of goals identified in the HGA is to be achieved, and the knowledge that needs to exist to service these goals. HGA therefore identifies the variables or objects in the work domain that are to be acted on, including those variables associated with high-level goals. Arguably PCT identifies, at the very least, a subset of the variables described by the AH. This subset is bounded at one end by the highest-level functional purpose of the system and at the other by the lowest level at which control is to be exercised and therefore spans much of the AH space. It would seem that the AH and the HGA capture similar information and, if this is the case, the main distinguishing features of CWA and PCT are the tighter coupling between PCT's HGA and the concept of control, through the perceptual control loop, than occurs in CWA's AH and Control Task Analysis (CTA). Further, PCT is clear about the upper bound of the analysis, unlike CWA.

In Tables 7.5 and 7.6, examples are given of knowledge requirements and perceptual and cognitive processes involved in satisfying specific goals within a HGA. As they are described here they are minimal representations of these attributes, largely at the descriptive level, driven by the specific requirements of the IPME modeling environment. This does not exclude the use of more detailed knowledge elicitation tools in combination with the PCT-based HGA. Indeed something like Rasmussen's Decision Ladder (e.g., see Vicente, 1999, Fig. 8.4) or Klein's taxonomy (Klein, 2000) could be used to further decompose the hierarchy. A tool for conducting a PCT-based HGA would include, at the very least, fields for detailed narrative descriptions of the plans, knowledge, decision processes, and subgoals at each level as appropriate.

The amount of work required to perform a CWA seems to expand exponentially as the complexity of the system grows (a PCT-based HGA would be expected to increase more linearly). One would expect that there would be AHs for each of the functional areas identified in an earlier section of this paper, namely: (1) primary mission, (2) training (the need to design for embedded or on-the-job-training), (3) abnormals, (4) maintenance, and (5) sustain or replenish. Similarly there may be different AHs for the time-based behavior of the system, from startup to shutdown. Then what of multiple actors with different levels of responsibilities in a large system (say a ship's command center that is a system of systems)? The work domain of a sensor operator is quite different from the work domain of the commander in a ship's operations room. The

notion of nested AHs is already presented in Chin et al. (1999). One would have to iterate this process through every possible manning concept. Yet the starting point of the CWA analysis, which is the generation of an abstraction hierarchy, is supposed to precede the allocation of responsibilities in the system.

MFTA is a relatively mature process that has much in common with accepted systems engineering and project management methods. This is not a case for whether it is a good method or not, just an argument that it is a reasonably well-understood paradigm. CWA and PCT have not had the same opportunity to demonstrate their successes and failures. CWA is a complex process that is still evolving. While Vicente's (1999) book is an excellent starting place for those with an interest in the area, it is hardly a primer on implementing CWA. There are still many unknowns related to the link between the AH and CTA and in the number of AHs that should be developed in complex multicrew and multifunction systems. Tools for the application of CWA are still in the development stage.

PCT based systems analysis is even less well known, but due to the parallels between MFTA and PCT systems analysis, many existing tools can be adapted with relatively little effort, and the overall process can be recognized by those skilled in the traditional method. The similarities and differences between CWA and PCT systems analysis are not yet fully understood and will become more apparent as both methods are applied to systems development and lessons are learned. What can be said of PCT in general, and of the PCT based systems analysis in particular, is that:

- Goal decomposition in the PCT's HGA is based on a means-end hierarchy—because of the 1:1 coupling between goal (internal state) and influenced variable (objects in the work domain) this is basically the same as CWA's structural means-end decomposition
- PCT systems analysis is based on a theoretical framework that describes goal-directed human behavior, machine activity, and communication between actors in the system. PCT brings all the rigor of control theory to the problem
- PCT explicitly generates behaviors that are shaped by both the actor and the environment—learned mental models reflect environmental dynamics for stability…in essence learning is a system identification issue
- All perception and action are shaped by internal knowledge structures whether they be right or wrong, good or bad…feedback and adaptation make this work in many cases…feedback allows less than perfect mental models to null the error (better mental models result in more rapid goal achievement)…an adaptive loop learns new and better mental models as actors interact with the system
- Perceptions and actions will be subject to constraints due to the capability and limitations of the actor (this is the realm of human engi-

neering)…the physical interface will constrain the possibilities for action…the external variables will be constrained by the environment (both their range and cross coupling—this is the realm of the systems engineer)…constraints at the interface may make goals unachievable
- PCT clearly identifies the top and bottom levels of analysis
- PCT is scalable and therefore can be extended upward or downward, as circumstances require, at any time
- PCT links display with action through goals
- Information is not displayed at the interface unless it contributes to the knowledge required to support the achievement of some goal/objective that has been designed into the system.

MIL–HDBK–46855 provides a process model that can be used to establish an HFE process compatible with an HSI program. Any new developments, such as PCT or CWA have to be compatible with the overall HSI process. We are confident PCT is, although it is clear that it will require training or experience to use. From that perspective it is no different from other HFE techniques, because university courses in HF typically do not teach HFE techniques except at the most rudimentary level.

7.8 SUMMARY

Structured analysis methods have been shown to contribute to the systems design process. Such methods have characterized front-end human engineering analysis for several decades. Typical of these structured approaches is Mission, Function, Task, Analysis (MFTA). Yet as systems have become more cognitive, some argue that the traditional methods fail to capture important aspects of the system specification, particularly with respect to the characteristics of the human-machine interface. This paper presents an alternative approach, PCT systems analysis, which claims to overcome many of the problems associated with traditional MFTA. Some comparisons are made with another method for human systems analysis, cognitive work analysis.

This paper presents a method of front-end human engineering analysis, based on the perceptual control theory model for goal-directed human behavior, with contributions from a time-based information processing model of the human operator. While this method might trace its origins to traditional MFTA, there are fundamental points of departure that distinguish the new method from the old. Central to these differences is:

- The replacement of separate function and task analyses with a unified hierarchical goal analysis
- The requirement to consider all goals, from the highest level to the

lowest, as candidates for assignment to an agent(s) (either human, machine) in the system

- The analyses that emerge from this process that trace both the stability of a system with multiple sources of control, and the upward flow of information from each level to support the level above.

PCT analysis and systems analysis can proceed in lockstep with design, shaped both by the constraints of the work domain and the constraints of human actors that are to populate the system as operators and maintainers. It supports the specification of goal and plan tracking databases for intelligent adaptive interface design, and has been used in one proof-of-concept application with encouraging results and tangible contributions to analysis. PCT systems analysis can make use of existing tools although the development of dedicated interfaces, to support this form of analysis, is needed. This will contribute to usability.

In contrast, CWA takes a more revolutionary path by arguing that design must start at the work domain and proceed through several processes before human capabilities and limitations start to shape the evolving system. Arguably it is a more complex process to implement, and the toolset is still in development. It has yet to be formally integrated into the engineering design process and is weak on methods for validating and verifying the design from a human perspective. However if, as some would claim, mission and scenario based analyses will inevitably fail in guiding successful designs of complex open systems, and PCT-based HGA cannot shake itself free of these criticisms, then the method proposed in this paper has a limited future. It is the view of the current authors that this issue is not yet settled.

REFERENCES

Annett, J., & Cunningham, D. (2000). Analysing command team skills. In J. M. Schraagen, S. F. Chipman, & V. L. Shalin (Eds.), *Cognitive task analysis* (pp. 410–415). Mahwah, NJ: Erlbaum.

Department of Defense (1990). *Systems engineering management guide.* Washington, DC: U.S. Defense Systems Management College, U.S. Government Printing Office.

Department of Defense (1991). *Department of Defense Directive 5000.1. Defense acquisition, Part 7, Section B: Human systems integration.* Washington, DC: U.S. Department of Defense.

Becker, C. A., & Killion, T. H. (1977). Interaction of visual and cognitive effects in word recognition. *Journal of Experimental Psychology: Human Perception and Performance, 3,* 389–401.

References

Beevis, D. (1987). Experience in the integration of human engineering effort with avionics systems development. In *Proceedings of the AGARD Symposium on the Design, Development and Testing of Complex Avionics Systems* (AGARD CP–417, pp. 27–1—27–9). Neuilly-sur-Seine, France: Advisory Group for Aerospace Research and Development.

Beevis, D., Bost, R., Döring, B., Nordo, E., Oberman, F., Papin, J. P., Scuffle, H., & Streets, D. (1999). *Analysis techniques for human-machine systems design*. Wright-Patterson AFB, OH: Crew Systems Ergonomics/Human Systems Technology Information Analysis Center.

Beevis, D., Lessens, P., & Schuffle, H. (1996). *Improving function allocation for integrated systems design.* Wright-Patterson AFB, OH: Crew Systems Ergonomics Information Analysis Center.

Belyavin, A., Cain, B., Lessens, P., & Hendy, K. (2000). Identifying team behaviours. *Meeting on Team Modelling.* Soesterberg, The Netherlands: TNO.

Besner, D., & Chapnik Smith, M. (1992). Models of visual word recognition: When obscuring the stimulus yields a clearer view. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 18,* 468–482.

Bourbon, W. T. (1996). On the accuracy and reliability of predictions by Perceptual Control Theory: Five years later. *The Psychological Record, 46* (1), 39–47.

Chaillet, R. F. (1967). Human factors requirements for the development of U.S. Army materiel. *Ergonomics, 10* (2), 278–286.

Checkland, P. (1981). *Systems thinking, systems practice.* New York: Wiley.

Chin, M., Sanderson, P., & Watson, M. (1999). Cognitive work analysis of the command and control work domain. In *Proceedings of the 1999 Command and Control Research and Technology Symposium.* Newport, RI: Naval War College.

Dahn, D., & Lowdon, D. (2000). *A perceptual control theory approach to human systems analysis: Summary Report* (PW&GS Contract No. W7711–9–7552/001 TOR for DCIEM, Toronto, Canada). Kanata, Ontario, Canada: BAE Systems Canada.

Darnell, M. J. (2000). *Bad human factors designs.* http://www.baddesigns.com/index.shtml.

Diamond, F. I. (1989). Avionics system engineering—An introduction. In *Systems Engineering* (AGARD–LS–164). Neuilly-sur-Seine, France: Advisory Group for Aerospace Research and Development.

DiStefano, J. J., Stubberud, A. R., & Williams, I. J. (1967). *Feedback and control systems*. New York: McGraw-Hill.

Edwards, J. L. (1997). *Design and development of intelligent aiding for LOCATE workspace layout tool* (PW&GS Contract No. W7711–6–7321 for DCIEM, North York, Canada). Toronto, Ontario, Canada: Artificial Intelligence Management and Development Corporation.

Edwards, J. L., Hendy, K. C., & Scott, G. (In press). A testbed for adaptive aiding using the LOCATE workspace layout tool. *Journal of User Modeling and User Adapted Interaction.*

Edwards, J. L., & Sinclair, D. (2000). Designing intelligence: A case of explicit models and layered protocols. In M. M. Taylor, F. Néel, & D. G. Bouwhuis (Eds.), *The structure of Multimodal Dialogue II.* Philadelphia: John Benjamins.

Eysenk, M. W., & Keane, M. T. (1990). *Cognitive psychology: A student's handbook.* London: Erlbaum.

Forssell, D. (1997). *PCT introduction and resource guide*. CSG group materials online. http://www.ed.uiuc.edu/csg/.

Fowler, B., Mitchell, I., Bhatia, M., & Porlier, G. (1989). Narcosis has additive effects rather than interactive effects on discrimination reaction time. *Human Factors, 31,* 571–578.

Garner, W. R., Hake, H. W., & Eriksen, C. W. (1956). Operationism and the concept of perception. *Psychological Review, 63*, 149–159.

Goode, H. H., & Machol, R. E. (1957). *System engineering: An introduction to the design of large-scale systems.* New York: McGraw-Hill.

Hendrick, H. W. (1990). Factors affecting the adequacy of ergonomic efforts on large-scale-system development programs. *Ergonomics, 5,* 639–642.

Hendy, K. C. (1995). Situation awareness and workload: Birds of a feather? In *Proceedings (CP–575) of the AGARD AMP Symposium on "Situation Awareness: Limitations and Enhancement in the Aviation Environment"* (pp. 21–1—21–7). Neuilly-sur-Seine, France: Advisory Group for Aerospace Research and Development.

Hendy, K. C. (1997). An information processing approach to workload and situation awareness. In R. S. Jensen & L. A. Rakovan (Eds.), *Proceedings of the 9th International Symposium on Aviation Psychology* (pp. 1423–1428). Columbus, OH: The Ohio State University, The Aviation Psychology Laboratory.

Hendy, K. C. (1998). CRM: More than just talk, talk, talk, talk… *RAAF Flight Safety Spotlight, 98* (1), 12–15.

Hendy, K. C., East, K. P., & Farrell, P. S. E. (2000). An information processing model of operator stress and performance. In P. A. Hancock & P. A. Desmond (Eds.), *Stress, workload and fatigue: Theory, research, and practice* (pp. 34–80). Hillsdale, N.J: Erlbaum.

Hendy, K. C., & Farrell, P. S. (1997). *Implementing a model of human information processing in a task network simulation environment* (DCIEM 97–R–71). North York, Ontario, Canada: Defence and Civil Institute of Environmental Medicine.

Hendy, K. C., Liao, J., & Milgram, P. (1997). Combining time and intensity effects in assessing operator information processing load. *Human Factors, 39* (1), 30–47.

Hendy, K. C., & Lichacz, F. (1999). Controlling error in the cockpit. In R. S. Jensen (Ed.) *Proceedings of the 10th International Symposium on Aviation Psychology* (pp. 658–663). Columbus, OH: The Ohio State University, The Aviation Psychology Laboratory.

Herdman, C. M., Chernecki, D., & Norris, C. (1999). Naming cAsE aLtErNaTeD words. *Memory & Cognition, 27*, 254–266.

Hintzman, D. L. (1978). *The psychology of learning and memory.* San Fransisco: Freeman.

Hughes, H. C., Layton, W. M., Baird, J. C., & Lester, L. S. (1984). Global precedence in visual pattern recognition. *Perception and Psychophysics, 35*, 361–371.

IEEE–1220 (1994). *Application and management of the systems engineering process.* Piscataway, NJ: Institution of Electrical and Electronic Engineers.

Klein, G. (2000). Analysing command team skills. In J. M. Schraagen, S. F. Chipman & V. L. Shalin (Eds.), *Cognitive task analysis* (pp. 417–429). Mahwah, NJ: Erlbaum.

References

Knobel, R. C., & Shaughnessy, M. (1999). Human cognition and information processing: Potential problems for a field dependent sequential information processor. In D. Harris (Ed.), *Engineering psychology and cognitive ergonomics*. Aldershot, United Kingdom: Ashgate.

Lachman, R., Lachman, J. L., & Butterfield, E. C. (1979). *Cognitive psychology and information processing*. Hillsdale, NJ: Erlbaum.

Logan, G. D., Coles, M. G. H., & Kramer, A. F. (1996). Introduction. In A. F. Kramer, M. G. H. Coles, & G. D. Logan (Eds.), *Converging operations in the study of visual selective attention* (pp.#xv–xxv) Washington, DC: American Psychological Association.

Marr, D. (1982). *Vision.* New York: Freeman.

McCormick, E. J. (1976). *Human factors in engineering and design* (4th Ed). New York: McGraw-Hill.

McMillan, G. R., Beevis, D., Salas, E., Strub, M. H., Sutton, R., & van Breda, L. (Eds.). (1989). *Applications of Human performance models to system design*. New York: Plenum Press.

McNeill, D., & Freiberger, P. (1993). *Fuzzy logic*. New York: Simon & Schuster.

Meister, D. (1985). *Behavioral analysis and measurement methods.* New York: Wiley.

MIL–HDBK–46855A (1999). *Human engineering program process and procedures.* Washington, DC: Department of Defense.

Neisser, U. (1967). *Cognitive psychology*. New York: Appleton-Century-Crofts.

Newell, A., & Simon, H. A. (1963). GPS: A program that simulates human thought. In E. A. Feigenbaum & J. A. Feldman (Eds.), *Computers and thought*. New York: McGraw-Hill.

Newell, A., & Simon, H. A. (1972). *Human problem solving.* Englewood Cliffs, NJ: Prentice-Hall.

Phillips, C. L., & Harbor, R. D. (1991). *Feedback control systems* (2nd Ed.). Toronto: Prentice-Hall.

Powers, W. T. (1973a). *Behavior: The control of perception.* New York: Aldine De Gruyter.

Powers, W. T. (1973b). Feedback: Beyond behaviorism. *Science, 179* (4071), 351–356.

Powers, W. T. (1989). An outline of control theory. In R. S. Marken (Ed.) *Living control systems—Selected papers of William T. Powers* (pp. 253–293). Gravel Switch, KY: Control Systems Group.

Powers, W. T. (1990). A hierarchy of control. In R. J. Robertson & W. T. Powers (Eds.), *Introduction to modern psychology: The Control-Theory view.* Gravel Switch, KY: Control Systems Group.

Powers, W. T. (1992a). CT psychology and social organizations. In G. Williams (Ed.), *Living control systems II—Selected papers of William T. Powers* (pp. 91–127). Gravel Switch, KY: Control Systems Group.

Powers, W. T. (1992b). The fallacy of behaviorism. In G. Williams (Ed.), *Living control systems II—Selected papers of William T. Powers* (pp. 233–240). Gravel Switch, KY: Control Systems Group.

Powers, W. T. (1993). *The dispute over control theory*. Paper presented at the 1993 CSG Meeting. Durango, CO: Control Systems Group.

Previc, F. H., Yauch, D. W., DeVilbiss, C. A., Ercoline, M. S., & Sipes, W., E. (1995). In defense of traditional views of spatial disorientation and loss of situation awareness: A reply to Navathe & Singh's "An operational definition of spatial disorientation." *Aviation, Space, and Environmental Medicine, 66*, 1103–1106.

Rasmussen, J., Pejtersen, A. M., & Goodstein, L. P. (1994). *Cognitive systems engineering.* New York: Wiley.

Reed, S. K. (1982). *Cognition: Theory and applications*. Monterey, CA: Brooks/Cole.

Roediger, H. L., Rushton, J. P., Capaldi, E. D., & Paris, S. G. (1984). *Psychology*. Boston: Little, Brown & Company.

Schank, R. C., & Abelson, R. P. (1977). *Scripts, plans, goals and understanding.* Hillsdale, NJ: Erlbaum.

Schmitter-Edgecombe, M. E., Marks, W., Fahy, J. F., & Long, C. (1992). Effects of severe closed-head injury on three stages of information processing. *Journal of Clinical and Experimental Psychology, 14*, 717–737.

Self, H. C., Green, R. J., & Ellifritt, T. S. (1995). *50 years of human engineering: history and cumulative bibliography of the Fitts Human Engineering Division.* Wright-Patterson Air Force Base, OH: Armstrong Laboratory, Crew Systems Directorate.

Simon, H. A. (1980). The social and behavioral sciences. *Science, 209*, 72–78.

Solso, R. L. (1991). *Cognitive psychology*. Boston: Allyn & Bacon.

Sternberg, S. (1969). The discovery of processing stages: Extensions of Donders' methods. In W. G. Koster (Ed.), *Attention and Performance II.* Amsterdam, The Netherlands: North-Holland.

Sykes, J. B. (Ed.). (1982). *The concise oxford dictionary* (7th Ed.). Oxford, United Kingdom: Clarendon Press.

Van Cott, H. P., & Kinkade, R. G. (Eds.). (1972). *Human engineering guide to equipment design*. New York: Wiley.

Van de Vegte, J. (1986). *Feedback control systems.* Englewood Cliffs, NJ: Prentice-Hall.

Vicente, K. J. (1999). *Cognitive work analysis: Towards safe, productive, and healthy computer-based work.* Mahwah, NJ: Erlbaum.

Wilensky, R. (1983). *Planning and understanding.* Don Mills, Ontario, Canada: Addison-Wesley.

Woodson, W. E. (1981). *Human factors design handbook.* New York: McGraw-Hill.

8 Cognitive Cockpit Engineering: Coupling Functional State Assessment, Task Knowledge Management, and Decision Support for Context-Sensitive Aiding

ROBERT M. TAYLOR
DERA Center for Human Sciences

MICHAEL C. BONNER
DERA Center for Human Sciences

BLAIR DICKSON
DERA Center for Human Sciences

HOWARD HOWELLS
DERA Air Systems

CHRISTOPHER A. MILLER
Honeywell Technology Center

NICHOLAS MILTON
Epistemics Ltd.

KIT PLEYDELL-PEARCE
University of Bristol

NIGEL SHADBOLT
University of Southampton

JENI TENNISON
Epistemics Ltd.

SHARON WHITECROSS
University of Bristol

ABSTRACT

The UK Ministry of Defence (MOD) in conjunction with the Defence Evaluation Research Agency (DERA) have established a program of applied research concerned with the development of cockpit adaptive automation and decision aiding for military fast-jet pilots. The operational requirement for this *cognitive cockpit* project arises from the possibility of a highly automated future offensive air system, involving a mix of manned and uninhabited air vehicles. In complex, rapidly changing military environments, increased dependencies on automation present significant challenges to maintaining effective human cognitive involvement in systems functioning. A human-centered approach to system design is needed that is based on human cognitive requirements for the control of system functional purpose, decision-making usability, and effectiveness in context of use. Technology is needed to assist rather than replace the future aircrew in cognitive work with systems involving high levels of task automation. Support will be needed that is adaptive and context-sensitive, to be responsive to changing mission requirements, in particular for in-flight situation assessment and mission replanning, in other words, decision support to provide the right information, in the right way, and at the right time. Technology needs to consider the aircrew's physiological and behavioral state, adaptively responding to an individual's indications of overload, distraction, and incapacitation. This chapter describes a program of research in cognitive systems engineering that seeks to couple pilot functional state assessment, knowledge-based systems for situation assessment and decision support, with concepts and technologies for adaptive automation and cockpit adaptive interfaces. The intention is to provide a scientific quantitative assessment of a broad range of options for intelligent pilot-aiding. This is to be based on sound cognitive systems engineering principles for system cognitive control, which keeps the pilot in control of the system, rather than the system controlling the pilot.

8.1 INTRODUCTION

8.1.1 Cognitive Design Requirements

Cognitive systems engineering seeks to bring together consideration of the environment, artifacts, and agents (human and machine) in a system of systems approach to design (Hollnagel & Woods, 1983; Rasmussen, 1986; Norman 1986). It tries to make sense of the mutual interactions between people and their environments under a variety of changing conditions (McNeese, 1995). This

supports a much needed human-centered, rather than technology-centered, approach to systems design, with a strong understanding of the role of artifacts—machines, tools, computers (i.e., things that make us smart or dumb)—and the requirements of the context of use and of system functional purpose. The need for this approach has arisen generally from human problems of working with automation and computers, and from considerations of human error and safety, in addition to efficiency and productivity. This has led to a focus on analysis of cognitive work and environmental constraints, and ideas such as context of use, cognitive control, situated cognition, and other ecological issues (Hollnagel 1993; Rasmussen, Pejtersen, & Goodstein, 1994; Vicente, 1999). These ideas and approaches are particularly relevant to the implementation of intelligent aiding systems in complex environments such as military aviation.

In aviation, computer-based cockpit automation has been designed generally to replace rather than to support human functions. Implementation of conventional automation, particularly in civil aviation, has sought to reduce or simplify crew tasks, so as to enable cost savings from reductions in crew complements, human error, and training. However, in the military aviation environment, human involvement is needed in systems control to govern the system's functional purpose, and particularly to provide the strategic guidance and tactical flexibility needed in rapidly changing, complex military operations. In the environment of use, the complexity of military aviation task domains is such that without considerable computerized assistance the aircrew would not be able to cope with the very large number of potentially relevant features and a vast number of possible responses. Perceiving and interpreting all of the relevant features and choosing an appropriate response within the tight temporal constraints of the domain will challenge any intelligent agent—whether human or machine.

One method of reducing the task and cognitive load on aircrew, enabling the pilot to concentrate unique cognitive skills on critical tasks, is the provision of intelligent knowledge-based aiding systems with the context sensitivity needed to provide the right information, in the right way, at the right time (Eggleston, 1993). Providing an aircrew with usability aiding makes the crew station easier to use and—determines when and how to deliver proposals and notifications. The introduction of intelligent aiding systems requires cockpit systems engineering to consider the cognitive requirements in the specification and design of cockpit processes, in addition to the basic system physical design (Eggleston, 1993; Taylor, MacLeod, & Haugh, 1995). Eggleston redefines cognitive design requirements as "all the system factors that are essential for it to behave at a conceptual (symbolic and abstract) level of understanding and engage in a knowledge level discourse with the user." He notes that conventional cockpits, aimed at providing information delivery and a control system, have cognitive requirements imbedded in their basic design, captured through mission, task, information, and workload analyzes. In contrast, intelligent

cockpits aimed at mission task and usability-aiding, through interagent, knowledge-based, conceptual, mixed initiative transactions, have the additional cognitive design requirements of the design of the knowledge base and reasoning processes that need to be embedded in the system process architecture.

Validated psychological methods and techniques are needed to capture cognitive requirements of the essential high-level internal processes of user's mental models. The methods available for cognitive systems engineering are becoming increasingly diverse and mature and are available for use as a systematic practice, such as the Work Domain Analysis (WDA) Workbench (Sanderson, Eggleston, Skilton, & Cameron, 1999). They include cognitive modeling, Cognitive Work Analysis (CWA), functional decomposition, Cognitive Task Analysis (CTA), control task analysis, concept mapping, Knowledge Acquisition (KA), knowledge modeling, Ecological Interface Design (EID), and prototype story boarding. Some of the methods available for understanding user mental models are illustrated in Figure 8.1 (adapted from McNeese, 1995). For the purposes of providing intelligent knowledge-based aiding, a key development has been the evolution of the CommonKADS knowledge engineering methodology (Schreiber et al., 1999). This provides a structured approach to knowledge modeling, with a comprehensive approach to the aiding context distinguishing the requirements of the organization, task, and agent.

Significant progress has been made towards the provision of intelligent knowledge-based aiding systems in a variety of application environments, sufficient to warrant their serious consideration for the next generation of military

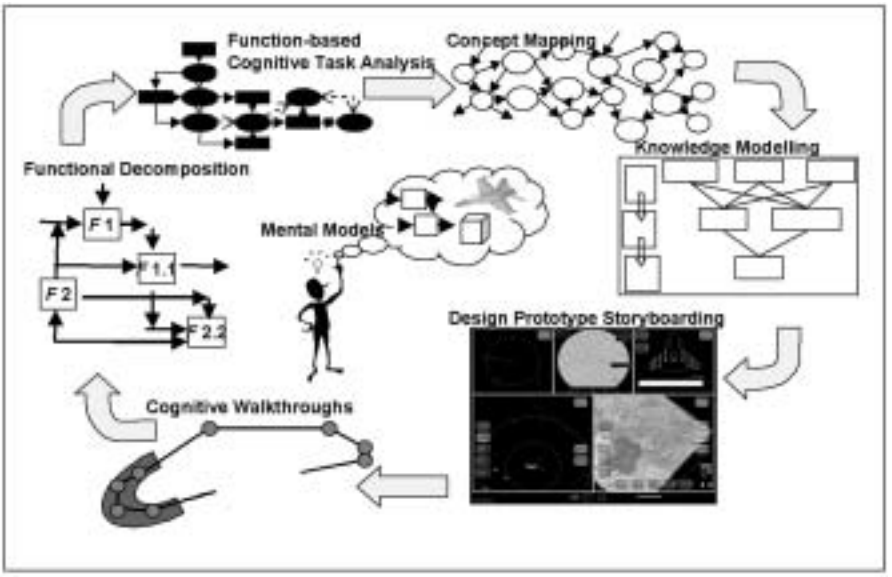


Figure 8.1: Cognitive engineering methods. (Adapted from McNeese, 1995)

aircraft, both manned and uninhabited, and controlled from the ground. Recently, DERA has been tasked with investigating cognitive technologies and cognitive design requirements and with providing proof-of-concept technical demonstration of options and benefits for future envisioned air systems MOD procurements (currently Future Offensive Air System [FOAS], Future Carrier-Borne Aircraft [FCBA], and Eurofighter Upgrade). The resultant DERA Cognitive Cockpit (COGPIT) program provides research on intelligent aiding systems in which the relationship between the pilot and the system is flexible and context-dependent to support adaptiveness (Taylor, Howells, & Watson, 2000). This flexibility is derived from a functional architecture that couples on-line monitoring of the pilot's functional state, and on-line task knowledge management and decision support for context-sensitive aiding, deriving information to mediate the timing, saliency, and autonomy of the aiding. The potential system benefits include the following:

- Real-time pilot functional state assessment for cockpit task adaptation
- Real-time support for situation assessment, task prioritization and decision making
- Real-time idiosyncratic and bespoke cockpit ergonomics, and
- Real-time safety net, with potential to recover to base an incapacitated pilot.

8.1.2 Background—The Development of Intelligent Pilot-Aiding Systems

Historically, the aircraft pilot and cockpit systems have had a master-slave relationship, with full pilot authority for aircraft control functions. This relationship changed with the introduction of computer control technology, with the pilot acquiring systems monitoring and supervisory roles. In the late 1970s, ideas arose for more intelligent cockpit systems, with an interactive and synergistic pilot-system relationship, providing cooperative rather than conflicting advice and control (Reising, 1979; Rouse, 1976, 1988). The crew-adaptive cockpit proposed sensors for monitoring the pilot's state, Artificial Intelligence (AI) software enabling the computer to learn, and pictorial displays allowing efficient presentation of cockpit information. This developed into a form of "R2D2" intelligent agent cooperating with the pilot as a Human-Electronic Crewmember (HEC) team, raising issues of human-computer teamwork, trust, technology capability maturity, cognitive requirements, and architectures (Emerson, Reinecke, Reising, & Taylor, 1988; Reising, Taylor, & Onken, 1999; Taylor & Reising, 1994).

Developments in advanced computer technology now make intelligent pilot-aiding through an HEC team realizable, including real-time data acquisition, fusion and processing, and computer modeling and AI-inferencing tech-

niques, such as expert systems, Knowledge-Based Systems (KBS), and neural nets (for a recent review, see Taylor & Reising, 1998). Beginning with the U.S. Air Force/Defense Advanced Research Projects Agency (DARPA) Pilot's Associate (PA) program (1985–1992), expert systems showed the potential of AI to support the pilot's problem analysis and solution generation. PA research identified human factors issues of adaptive automation, dynamic function allocation, levels of system autonomy and trust, and introduced goal-plan tracking for inferencing pilot intent. The U.S. Air Force Small Business Innovation/Research (SBIR) Hazard Monitor provided a real-time KBS for supporting system malfunction management in transport aircraft. Now, the U.S. Army's Rotorcraft PA (RPA) provides a *cognitive decision aiding system* and *cockpit information manager* (Miller, Guerlain, & Hannen, 1999).

In Europe in the 1990s, AI efforts on pilot-aiding have centered on the French Co-pilote Electronique (CE), and on the German civil and military Cockpit Assistant Systems. In contrast to PA, the CE program focused on AI support for problem recognition and situation assessment. The German CASSY project provided flight test of flight management KBS for rerouting of civil aircraft. Situation assessment modules provided perception, diagnosis, decisions and communication management, with pilot intent and error recognition functions. CAMMA has extended this application of KBS to military missions (for detailed technical information on CE, and CASSY/CAMMA, see Reising, Taylor, & Onken, 1999).

In the UK in the late 1980s, the joint Industry/MOD Mission Management Aid (MMA) project applied conventional computer techniques to sensor fusion, situation assessment, and dynamic planning. Using deterministic, rule-based, event-driven logic, MMA found positioning (rerouting) and EW functions more difficult to assist and automate reliably than fuel and time-management tasks. Lessons-learned have been applied to Eurofighter to reduce pilot workload. MMA identified the need for context-sensitive prioritization of interrupts in high workload phases. Subsequently, industry research has used AI model-based reasoning with multiple-goals to provide context-sensitive prioritization for intelligent warning systems for civil cockpit applications. Applying KBS to safety critical functions poses certification problems. At Farnborough in the 1990s, MOD Navy sponsored AI research by DERA Aircraft Sector focusing on KBS for aiding aircrew mission decision-making in helicopter antisurface warfare and airborne early warning (Zanconato & Davies, 1999). This has led to the development of real-time multi-agent KBS software, and new methodologies for knowledge acquisition and management (Martin & Howells, 1995).

Other MOD RAF-sponsored human factors research at DERA/Center for Human Sciences (CHS) focused on the cognitive issues in HEC teamwork, in particular situation awareness, trust, and cognitive compatibility (Taylor & Selcon, 1993; Taylor, Shadrake, Haugh, & Bunting, 1996). It was followed by

work on cognitive engineering issues, associated interfaces, and the operation of adaptive automation and decision support (Taylor, Finnie, & Hoy, 1997; Taylor, Shadrake, & Haugh, 1995). The results highlighted the risks of poor awareness of functioning with dynamically changing automation and the problems of cognitive bias associated with acceptance of automation advice. This work generally raised concern with the problems of maintaining effective human control of critical decisions and complex system functions with high levels of automation. It identified the need for further cognitive-engineering work on cognitive control issues and on supporting adaptiveness.

8.1.3 Cognitive Systems Engineering Challenges—Supporting Adaptiveness

To assist future pilots perform their critical cognitive tasks, technology is needed for automation and decision support that is context-sensitive and adaptive, in other words, responsive to changes in the operating environment, mission requirement, and operator capability. Adaptiveness can be considered as the ability of the human-machine system to perform in an appropriate, context-sensitive manner in different situations (Miller & Goldman, 1999). Adaptiveness is needed to support in-flight situation assessment, retasking, and replanning, and increasingly to avoid casualties, fratricide, and collateral damage. Adaptiveness and context sensitivity in use have traditionally been provided by the pilot knowing when and how to change the plan. To support and enhance adaptiveness, technology needs to respond to context divisions with sensitivity that is both precise and accurate (i.e., supports handling of critical events, in the appropriate manner and at the appropriate time.) This increased adaptiveness needs to be achieved without increasing crew workload and without the unpredictability often associated with the action of conventional automation (Miller & Goldman, 1999; Miller, Pelican, & Goldman, 1999).

Decision aids in particular need adaptiveness, ideally to both individual user characteristics and to changing task situations, to be useful in complex, dynamic problem-solving environments (Rouse & Rouse, 1983). Currently, intelligent knowledge-based aiding systems are available that are capable of responding to changes in the aircraft and the environment. Technology under development seeks to monitor and respond adaptively to changes in the mission plan, and to provide inferencing of pilot intent. Cognitive technology is also needed that is influenced by the aircrew's physiological and behavioral state, adaptively responding to an individual's indications of overload, distraction, and incapacitation. Integration and implementation of these cognitive technologies will need to be based on sound cognitive engineering principles. Cognitive technologies and architectures are needed that support the required

levels of human control over critical system functions, and that keep the crew in control of the system, rather than the system in control of the crew.

Early research on adaptive decision-aiding (Rouse & Rouse, 1983) considered the form of adaptation (user or task), the locus or mode of adaptation (designer, user, or aid), the method of adaptation (task allocation, partitioning, or transformation), and the communication of information (implicit or explicit). Recently, the focus has shifted towards the nature of the knowledge underlying the task adaptation. Intelligent aiding systems previously attempted, or currently under development, can be distinguished in terms of the tasks and roles that they perform, and the knowledge that they manipulate, indicating levels of capability maturity (Geddes, 1997):

- Assistant—Performs specific tasks when instructed by the operator, using basic task and situation knowledge. For example, such a system could provide the pilot with an assessment of a threatening aircraft when asked
- Associate—Automatically recognizes that the operator requires assistance (using complex task and situation knowledge, and basic user knowledge), and provides some level of support. For example, such a system could recognize a threatening situation and automatically provide the pilot with all threat information
- Coach—Using complex task, situation, and user knowledge, these systems are capable of recognizing the need for automation to achieve a mission objective, and providing instructions to the operator on how to achieve it. For example, the pilot is presented with the most threatening aircraft first, in accordance with the higher-level goal of maximizing own-ship survivability.

In this analysis, the drivers for aiding capability maturity are the complexity of task, situation, and user knowledge managed by the system. Technological advances in both artificial intelligence and the physiological monitoring of human performance have the potential to provide complex task, situation, and user knowledge, sufficient to allow these higher levels of intelligent aiding to be realized. Thus, in the future, intelligent aiding systems will be considered more as fully integrated, intelligent cockpits that take on agent-like properties, rather than as traditional cockpits with a discrete automation control center. Intelligent aiding systems have the potential to provide the following capabilities (Eggleston, 1997):

- Respond intelligently to operator commands, and provide pertinent information to operator requests
- Provide knowledge-based state assessments
- Provide execution assistance when authorized
- Engage in dialogue with the operator, either explicitly or implicitly, at

a conceptual level of communication and understanding, and
- Provide the operator with a more usable and nonintrusive interface by managing the presentation of information in a manner appropriate to the content of the mission.

The requirement to provide support in the appropriate internal and external "context" can be realized through a functional architecture with the following attributes (Taylor & Reising, 1998):

- A model of human decision-making and control abilities
- The ability to monitor pilot performance and workload through behavioral and physiological indices, and
- The ability to predict pilot expectations and intentions with reference to embedded knowledge of mission plans and goals.

Experience with conventional automation has highlighted the need for the state of highly complex avionics systems to be readily *comprehensible* to operators. Cockpit displays and controls for interacting with intelligent aiding systems must be particularly easy to understand and to operate, since their benefits arise only in use, and not automatically. Principles of cognitive compatibility and efficiency, such as simplicity and consistency, need to be design drivers so as to reduce rather than to increase workload from using cognitive aids. However, this requirement may be at odds with the ability of the automated system to remain *flexible.* Billings (1997) argues that there is an inverse relationship between system comprehensibility and system flexibility, especially when such systems exploit adaptive-aiding technologies. This corollary is unfortunate given that system comprehensibility and system flexibility are both important design drivers for adaptive aiding. Miller and Goldman (1999) and Miller, Pelican, and Goldman (1999) argue that the implication of this corollary is that, for any increase in system flexibility or adaptiveness, there must be an accompanying increase in either operator workload (i.e., the amount of cognitive effort required to operate the system), or in unpredictability for the operator (i.e., the inability of the human to know what the automation will do at any given time) (see Figure 8.2). Allowing operators to choose various levels of interaction for the tasks they are required to conduct can mitigate this problem.

The use of intelligent aiding within the cockpit requires aiding and interaction in real time. In real-time operations the correctness of the system is dependent not only on the correctness of its result, but also on meeting stringent timing requirements. The deadlines for tasks that a real-time system must perform can be characterized as *hard, firm, or soft.* Failure to meet a firm deadline means that the results of the computation have no utility. This is in contrast to soft deadlines where the results of the computation are still useful after the deadline has elapsed, but have decreasing utility as a function of the time

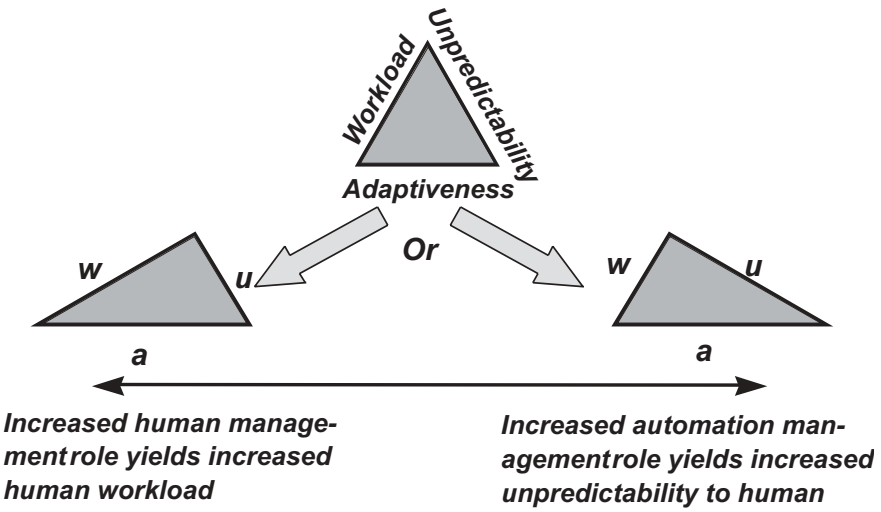


Figure 8.2: Relationship between system adaptiveness, human workload, and predictability. (From C. A. Miller, M. Pelican, & R. Goldman, Tasking interfaces for flexible interaction with automation: Keeping the operator in control, *Proceedings of the International Conference on Intelligent User Interfaces*, Association for Computing Machinery [ACM], 1999. Reprinted with permission.)

elapsed (Hayes-Roth 1991). For knowledge-based tasks, systems may be useful that can produce aiding at *anytime,* such as through progressive reasoning, providing the best advice immediately available when called upon to provide support. Some of the requirements for the real-time operation of intelligent aiding systems, identified by Hayes-Roth (1991) include the following:

Cognitive Versatility

- Multifaceted expertise—The system should be able to perform different types of reasoning tasks in an attempt to solve problems in a variety of domains utilizing a number of problem-solving techniques
- Concurrent reasoning activities—The system must be capable of simultaneous reasoning about a number of concurrent activities
- Incremental reasoning—The system must be able to integrate information over time to produce an accurate assessment of the current situation
- Explanation—The system should be able to explain all aspects of its behavior in the time available.

Management of Integration

- Functional asynchrony and parallelism—The system must be able not only to investigate anomalies, but also perform routine actions within specified time limits
- Continuous operation—The system must be capable of functioning over extended time periods
- Functional Integration—The system should be able to perform accurate reasoning even where certain conditions affect normal output of the reasoning process (e.g., recommendations may differ as a function of weapons fit).

Management of Complexity

- Selective attention—The system may encounter situations in which it cannot process all the data in real time. Therefore, the system must be able to make choices about which data are the most important and disregard extraneous data. However, it is imperative that the system still be alert to new data that might be critical to the current task
- Automatic performance—The system must be able to deal with complex anomalies or situations while performing important routine activities in a timely manner
- Focused reasoning—The system must be able to control its reasoning such that it can achieve specific goals. The system will face more "problems" than it can solve in real time, and so it is important that the system must be able to choose the most urgent problem(s) to solve first.

Real-Time Performance

- Guaranteed interoperation latencies—The system must be able to guarantee that it can achieve certain goals within the prescribed timeframes
- Time-stress responsivity—The system should be able to respond to increased pressure on time resources by decreasing its response latency
- Graceful degradation—The system must be able to reduce response latency as a function of time stress by compromising precision and confidence in a graduated manner
- Speed-knowledge independence—The system must be able to produce consistent response latencies despite the inclusion of new knowledge.

8.2 THE COGNITIVE COCKPIT

8.2.1 Assisting Cognitive Work

Arising from DERA CHS human factors work on cognitive issues in HEC teamwork, adaptive automation, and decision support, the need was identified for crew-centered, cognitive engineering research on cognitive control issues in supporting adaptiveness. It was envisioned that there was a need to provide a *cognitive cockpit.* This would be a form of electronic crew-member or cockpit cognitive assistant, constructed using cognitive engineering principles and methods. It would provide a principled implementation of cognitive technologies that use knowledge about how people perceive, think, and act adaptively, such as knowledge-based systems (Taylor, 1997; Taylor & Finnie, 1999). Figure 8.3 illustrates some of the initial concepts for a prototype DERA cognitive cockpit (from Taylor & Finnie, 1999).

In this early COGnitive cockPIT (COGPIT) conceptualization, potential threat locations are shown as cued adaptively (visual and 3D audio) off boresight, in the helmet-mounted display, with the form guided by head/eye location monitoring. Pilot health monitoring is operating in the background, showing no unexpected indications (customized functional abstraction/decomposition available on pilot request). The pilot has a "panic" button to provide inputs on subjective status and load. Systems health shows normal status using an adaptively reduced, iconic information "stamp." A rerouting proposal (best-calculated computer plan) is presented for pilot critiquing, in response to an identified air threat, using an adaptively enlarged Situation Awareness (SA) panel. This SA panel is provided with a background attitude indicator for spatial orientation SA, with accept, reject, and explanation key options for computer proposals. The intended automatic maneuver (with pilot interrupt, if required) is shown using a head-up, pathway-in-the-sky and pictorial aircraft velocity cue. A summary indication (functional abstraction/decomposition available on pilot request) of the mission effectiveness, based on mission plan and goal tracking, is shown in the form of a mission "goal ball." The internal diameter indicates the level of effectiveness, and the degree of risk. The operating status of three aiding agents—skill (coach, with flight path control), rule (three R2D2s, two currently active) and knowledge-based (Odin, advising tactical reroute)—is shown by pictorial adaptive icons with a summary of the allocated control authority or "balance."

Coupled with the confidence-building DERA KBS work, and encouraging DERA CHS Corporate Research Programme work on monitoring cognitive load, these ideas led in 1998 to an enhanced MOD Royal Air Force (RAF) Applied Research Programme project on a technical demonstrator for auto-

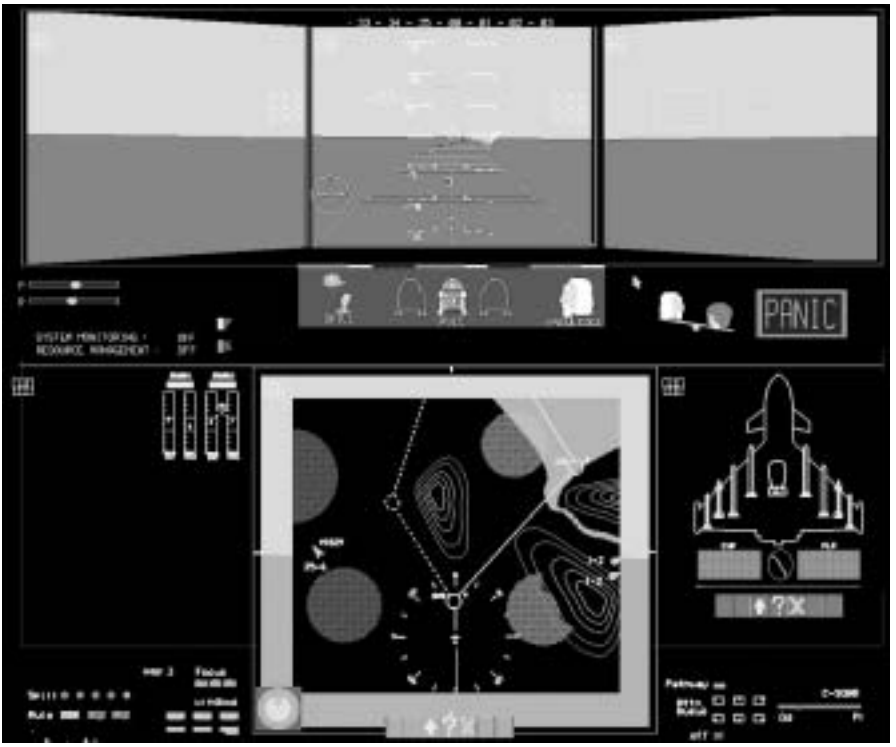


Figure 8.3: Initial conceptual prototype for the DERA cognitive cockpit.

mated decision support, known as the DERA Cognitive Cockpit (Taylor, Howells, & Watson 2000). This program seeks to provide a human-centered contribution to a system-of-systems approach, with an overall focus on system functional purpose, usability and affordability, and cost-effectiveness in use. The operational aim is to allow the pilot, in control of the aircraft or an Unmanned Air Vehicle (UAV) "to concentrate his skills towards the relevant critical mission event, at the appropriate time, to the appropriate level." The project is originally scoped as a three-year program, led by CHS, with multi-disciplinary internal DERA (CHS and Aircraft Sector) and significant external contractor involvement (Epistemics Ltd., University of Southampton, University of Bristol, Honeywell Technology Center). The DERA COGPIT work is conducted with international collaboration through The Technical Cooperation Program (TTCP) HUM Technical Panel 7, Human Factors in Aircraft Environments, and under bilateral agreements. This includes the U.S. Air Force Research Laboratory (USAFRL), Human Effectiveness Directorate, the Defence and Civil Institute of Environmental Medicine (DCIEM), Canada, and the Defence Science and Technology Organisation (DSTO), Air Operations Division (AOD), Australia. European bilateral collaboration is

with Sweden FOA, under Project Longboat (Linde & Berggrund, 1999), and recently with The Netherlands, National Aerospace Laboratory (NLR), through Project Nightwatch. Collaboration with the USAFRL began with work on virtually augmented and adaptive cockpit interfaces (Hettinger, Cross, Brickman, & Haas, 1996; Haas et al. 1997), and continues with the USAFRL current real-time human engineering program. Banbury, Bonner, Dickson, Howells, and Taylor, (1999) provide a detailed technical review of the COGPIT program, covering the first two years of work.

The DERA Cognitive Cockpit program, while broadly aimed at supporting future MOD air systems procurements needing intelligent aiding, is focused initially on the single-seat, fast jet role, and in particular the Future Offensive Air System (FOAS) pilot. The complexity, time-pressure, and rapidly changing uncertainties of the single-seat, fast jet environment provide a major cognitive engineering challenge for implementing intelligent knowledge-based pilot aiding systems. Figure 8.4 illustrates the major functions of the offensive air role using a WDA abstraction-decomposition breakdown (Vicente, 1999).

The military aviation domain is characterised by being uncertain and having shifting goals, dynamic evolution, time stress, action feedback loops, high stakes, and multiple players. While operators may wish to remain in charge, and it is critical that they do so, today's complex systems no longer permit them to be fully in charge of all system operations at all times as in earlier cockpits and workstations (Miller, Guerlain, & Hanner, 1999; Miller, Pelican, & Goldman, 1999). Cockpit automation has been, and will continue to grow more intelligent and more sensitive to context and mission objectives. But no one seriously believes that cockpit automation and decision aids can or should replace pilot control. Instead, they must free up pilot resources to concentrate on the most important tasks and must create in the pilot a situation awareness that allows him/her to make decisions correctly and very quickly.

To handle the kinds of unpredictable events experienced in the military environment, the functions in Figure 8.4 have to be accomplished by a combination of planning and reacting, with various degrees of human involvement. Many of these functions can be aided by computer-based systems and automated to a greater or lesser degree, depending on their simplicity and complexity and the maturity of the relevant technology, and on the predictability or uncertainty of factors governing their performance, and hence the need for involving human judgment. In human systems generally, the operator sets the goals and functional purpose of the behavior, and through the use of aids, automation, and other artifacts (things that make the user smart or dumb) provides the coordination and adaptation directed at coping with the complexity and uncertainty in the world. The military pilot needs to be involved in critical decisions affecting the safety of the aircraft, and the effectiveness of the mission. The pilot needs to be able to concentrate his cognitive skills "on the critical mission event, at the appropriate time, to the appropriate level." In addi-

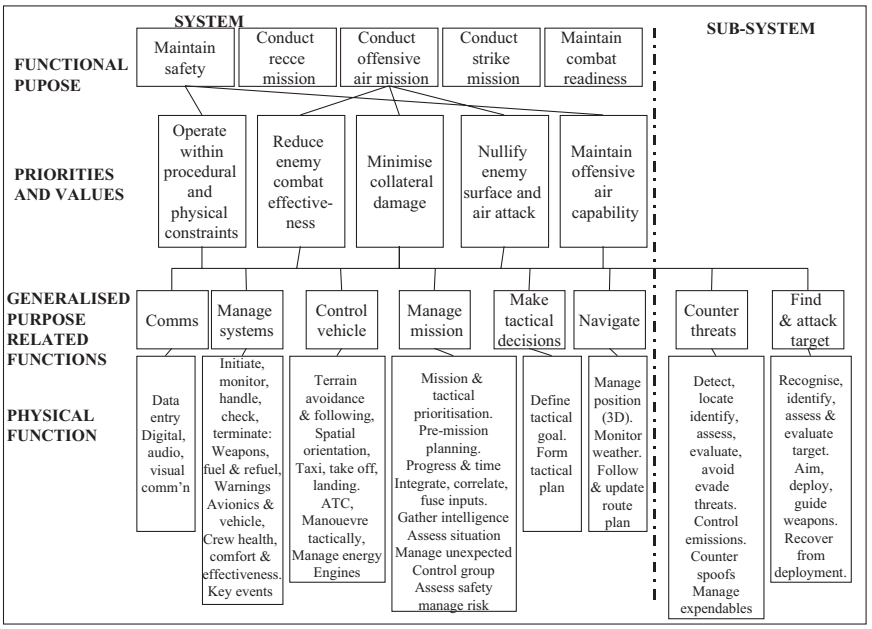


Figure 8.4: Offensive air functions.

tion to the goal setting and high-level control functions, the pilot's expert knowledge and adaptive, pattern recognition and decision-making cognitive skills in uncertain situations are difficult to emulate, and remain key capabilities for computer-aided support.

8.2.3 Functional Integration and Cognitive Control

The emerging situation with automation capability begs questions about the appropriate roles for pilot and smart automation in future military aircraft. It is becoming increasingly apparent that the dedicated and limited roles of today's automation systems (e.g., one system devoted to collision avoidance, another completely separate system devoted to route planning, etc.) cannot sustain the requirements for advanced missions. This is because such systems leave the most complicated task, that of integration and holistic decision making, to the human. By contrast, functional integration is an important characteristic of advanced intelligent aiding systems, in that the required behavior can be shared across many functional components, including the user (Geddes, 1997). That is, several functional components can collectively perform many of the same behaviors as the pilot—because they are aware of each other, capable of sharing information, aware of overall mission goals, and capable of inte-

grating their behaviors in the same way the pilot would. Functional integration of cockpit duties provides for a more robust and flexible integrated system when compared to systems based upon more strict function allocation to individual and unique components. It is expected that future intelligent aiding systems will be considered more as fully integrated, intelligent cockpits that take on agent-like properties, rather than as traditional cockpits with a discrete automation control center.

Adaptive automation occurs when the control decisions concerning the onset, the offset, and the degrees of automation are shared between the operator and machine. Within such a system the human operator remains "in-the-loop" and the automation intervenes only when the operator requires support to meet operational requirements—but it does not require explicit human instructions to do so. At the highest level of capability maturity, adaptive automation systems seek to augment and enhance human judgment and responsibility "intelligently," while mitigating against their limitations, by adapting to the changing requirements of both the operator and the external situation. These systems can be considered as "intelligent" insofar as they exhibit behaviors that are consistent with human-like characteristics (Taylor & Reising, 1998).

The requirements for command, control, and communication with conventional "intelligent" and hybrid computer systems are of particular interest. The use of intelligent and adaptive automation technology allows both the human and the machine components to influence and jointly support cognitive functioning, providing joint cognitive control (Hollnagel, 1996, 1997). Hollnagel illustrates how in a joint cognitive system, the decisions required can be considered a task net, through which some particular path can be taken, and where the tasks can be flexibly assigned to either human or computer performance. The specific elements can be performed by the human, by the automation, or alternatively by both, say if insufficient information may or may not be available for the automation to work (Figure 8.5).

Such joint cognitive systems require new levels of human-computer interaction, such as cooperative teamwork between human and electronic crewmembers, with dynamic allocation of functions responding to changing aircrew and mission requirements. This will require an adaptive interface that is aware of, and continually responds, to such changes.

To provide a principled development of intelligent aiding with the required levels of pilot control, we have established a systematic approach to cognitive control to guide the COGPIT program (Taylor 1997; Taylor & Finnie, 1999). This framework is based on the concepts and implications of Perceptual Control Theory (PCT) (Farrell & Chery, 1998; Powers, 1973; Taylor, 1992) and on the theory of cognitive control of complex systems (Brehmer, 1992; Hollnagel, 1997; Rasmussen, 1986). The work is influenced by a recent theory of Information Processing (IP) load under time pressure (Hendy, Liao, & Milgram, 1997) and with DERA/University of Cardiff work on a theory of

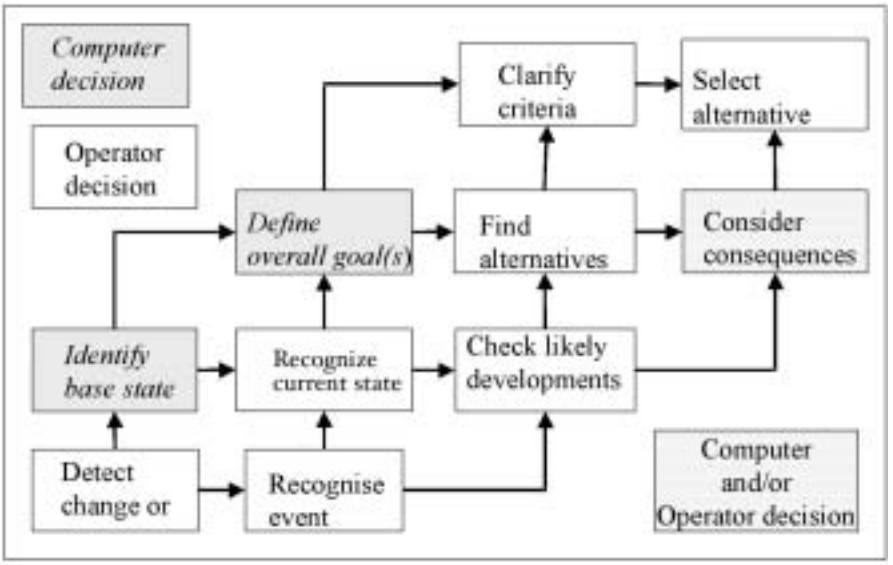


Figure 8.5: Task net of decision making. (Adapted from E. Hollnagel, Decision support and task nets, in S. A. Robertson [Ed.], *Contemporary ergonomics 1996*, Taylor & Francis, 1996. Reprinted with permission.)

cognitive streaming, emphasizing the importance of conflict of process, rather than content (Banbury, 1999; Tremblay & Emery, 2000). The general approach highlights the importance of functional integration, rather than partitioning and allocation, and of joint cognitive control, between the pilot and the intelligent aiding systems. In this way, a more direct and systematic consideration of the cognitive engineering and control issues can be achieved. For example:

- The incorporation into intelligent aiding systems of the ability to track the operator's goals and plans (e.g., the difference between current and desired states) and to infer the intent of the operator
- The use of abstraction hierarchies and system aggregation methods during task decomposition to determine important interactions and emergent properties within the knowledge base
- The importance of information utility in the design process (e.g., a focus on the information used, rather than the resultant action)
- The importance of error diagnosis and rectification
- The enhancement of system stability through the balance of feedback (i.e., reactive) and feed-forward (i.e., proactive) control information (Brehmer, 1992)
- The recognition of differences in cognitive control strategies between Skill, Rule, and Knowledge-based (SRK) levels of performance

(Rasmussen, 1986)

- The incorporation of planning horizons (e.g., scrambled, opportunistic, tactical and strategic) into cognitive control strategies (Hollnagel, 1997), and
- The use of intelligent aiding to critique operator performance and prevent cognitive bias and other forms of human error.

Using Rasmussen's SRK taxonomy (Rasmussen, 1986), the broad aim is to provide support for the pilot's knowledge-based behavior, in handling complex, unpredictable situations, where new procedures need to be formulated, reducing the pilot's cognitive decision-making load through the provision of KBS advice and assistance with knowledge management. Furthermore, the intention is to provide automation of rule—and skill-based behavior, if feasible and as required, for application in simple, predictable situations, where successful procedures are known to work, enabling the pilot to concentrate on critical strategic and tactical decisions. Broadly, for simple problems, the aim is to provide computer-based solutions where successful procedures are known (rule-based processing), and when procedures need to be formulated (knowledge-based processing). For complex problems, the intention is to provide computer-based support where known procedures are probably applicable (rule-based processing), and when new procedures need to be developed (knowledge-based processing). Support decision-making where the outcome is unpredictable;

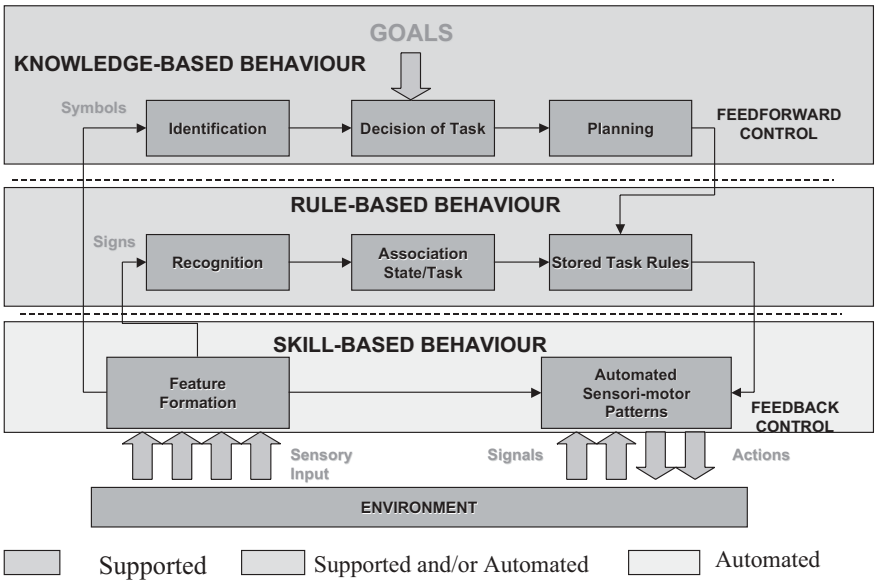


Figure 8.6: Strategy for aiding skill-, rule-, and knowledge-based behavior.

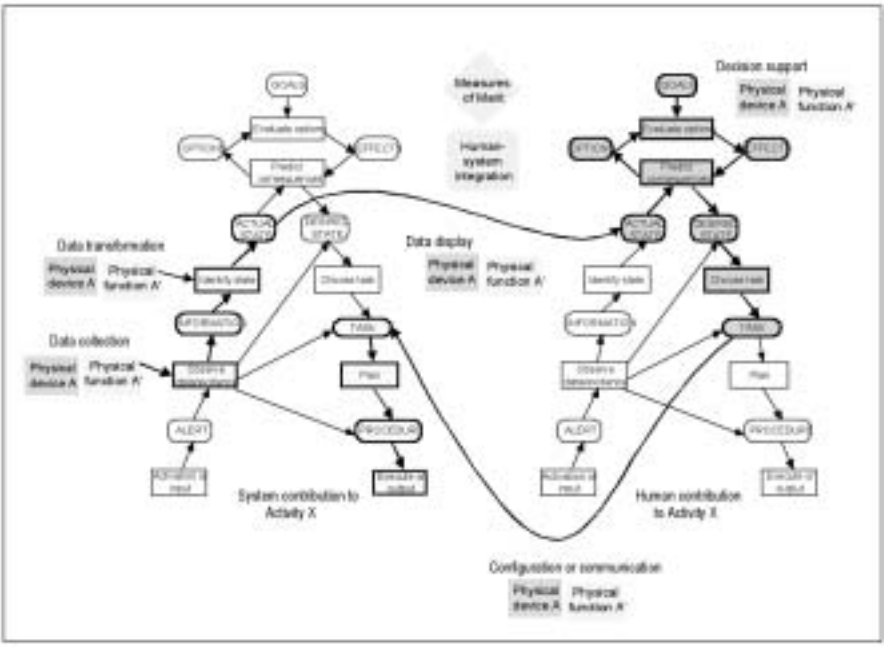


Figure 8.7: Decision ladders for control task analysis.
(From Sanderson, 1999, personal communication)

automate decision-making where the outcome is predictable. Figure 8.6 illustrates the strategy for aiding through automation and/or support in relation to skill, rule, and knowledge-based levels of behavior.

A typical aiding configuration is illustrated in Figure 8.7, using the SRK decision ladder modeling tool (Rasmussen, 1976; Vicente, 1999). Any given activity can comprise a combination of skill-, rule-, and knowledge-based process, and the information processing steps can be represented as a decision ladder. Combinations of decision ladders can be used to model collaborative work between agents. Decision ladders have been used in CWA for CIA to identify levels of human-system integration in Royal Austrian Air Force (RAAF) airborne early warning and control (Sanderson, Naikar, Lintern, & Goss, 1999). In Figure 8.7, provided by Sanderson (personal communication), two decision ladders are linked together to show different human and computer contributions in computer-assisted decision-making. This might apply to a complex problem, with automation of the relatively simple aspects of situation assessment and plan execution, and with KBS support for the more complex options evaluation. The computer is responsible for data collection and transformation and for communicating the actual state (i.e., computer situation assessment). The human is then responsible for interpreting the consequences

for system goals, for identifying the required state, and the task to be performed to achieve the desired state. These decisions require formulation of new procedures, which could be assisted by KBS decision support, but they are too complex for the computer alone to provide a successful solution. The computer is then responsible for analyzing the identified task, formulating the plan of procedures, and executing the plan. Other combinations are possible, with different computer and human decision roles and responsibilities. This form of cognitive activity analysis is useful since it allows the analyst to understand the human operator's goals in performing activity, it helps identify possible sources for competition of operator's attention, and it describes the forms of collaborative work and aiding that will be essential for mission effectiveness.

In summary, the significant technical challenges for the COGPIT program are as follows:

- Improving adaptiveness without increases in workload or unpredictable automation
- Providing real-time, context-sensitive aiding with accuracy and precision to be useful and trustworthy, with tracking of operator's goals and plans to infer intent
- Building an integrated KBS for prioritizing pilot tasks and aiding decisions
- Supporting adaptiveness in skill-, rule-, and knowledge-based levels of performance, critiquing performance, preventing cognitive bias, and aiding error rectification
- Providing useful functional state information for task adaptations and interruptions
- Providing quantitative, scientific assessment of a broad set of aiding options using measures of effectiveness based on mission task performance.
- Providing a blend of automation levels and pilot cognitive control strategies with:
    - Pilot executive authority for controlling the system
    - Stability through feedback (reactive) and feed-forward (proactive) control
    - Strategically planned pilot control at the knowledge-based level (feed-forward), and
    - Automation of reactive skill and rule-based responses (feed-back).
- Focusing system design on functional purpose, control allocation and information utility using cognitive work analysis methodologies.

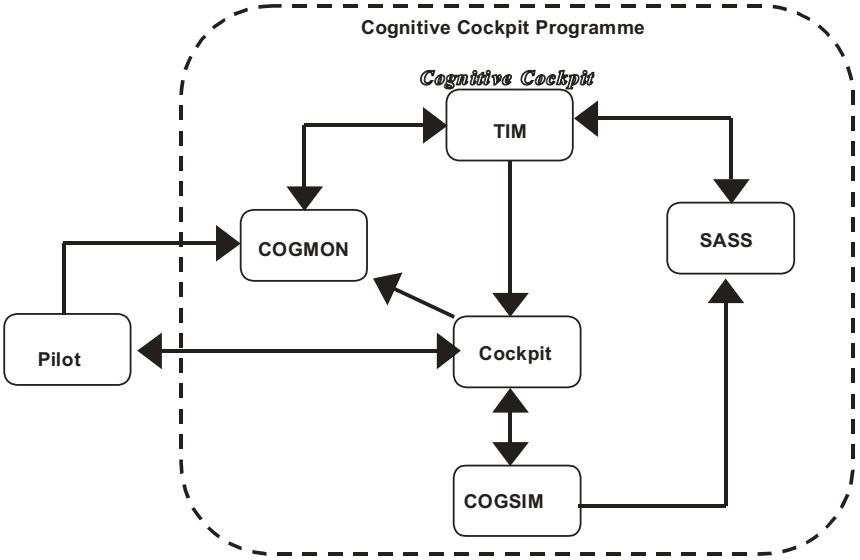


Figure 8.8: COGPIT agents architecture.

8.2.4 Functional Architecture—Agents, Communication, and Tasks

The COGPIT systems under development involve the interacting agents and communications shown in Figure 8.8. A simplified activity diagram, representing the processes performed by these agents, in support of updating the mission plan, is shown in Figure 8.9. Ultimately, the aim is to increase system adaptiveness by enabling changes in the mission plan in response to changes in the situation. To achieve this, the COGPIT will monitor three aspects of the situation: *the pilot* to take account of his physiological and cognitive state; *the environment,* both external to the aircraft and the aircraft systems; and *the mission plan* to indicate current and future pilot actions (Tennison, 1999). Four agents with different tasks can be distinguished as comprising the COGPIT system:

Cognition Monitor (COGMON)—a module that monitors the pilot's physiology and behaviour to provide an estimation of pilot state. This module is concerned with on-line analysis of the psychological, physiological, and behavioral state of the pilot. Primary system functions include continuous monitoring of workload, and inferences about current attentional focus, ongoing cognition and intentions. It also seeks to detect dangerously high and low levels of arousal. Overall, this system provides information about the objective and subjective state of the pilot within a mission context. This information is used to optimize pilot performance and safety and provides a basis for the implementation of pilot-aiding (Pleydell-Pearce & Dickson, 2000).

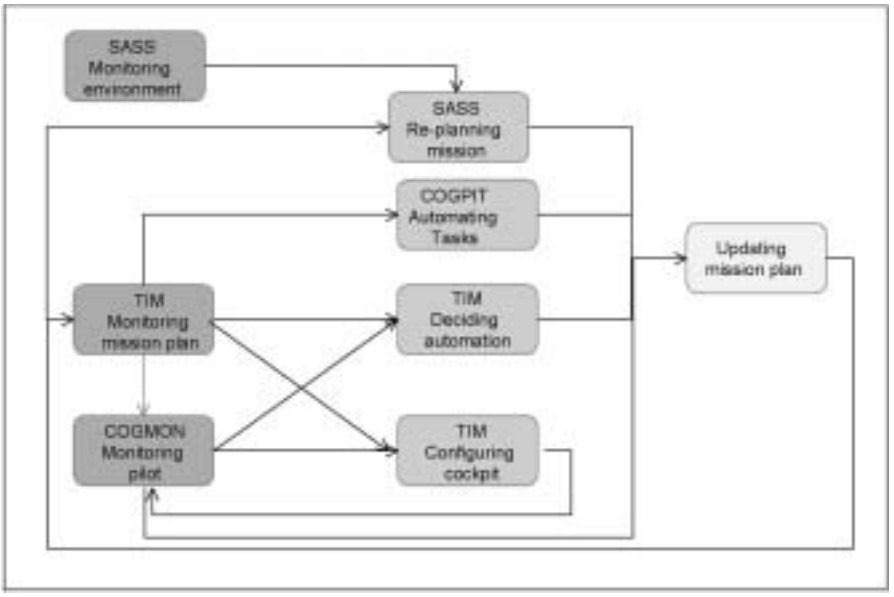


Figure 8.9: COGPIT agents, processes, and tasks.

Situation Assessment Support System (SASS)—a module that monitors the status of the aircraft situation and the outside environment and recommends actions. This module is concerned with on-line mission analysis, aiding, and support provided by real-time, multi-agent KBS software. This system is privy to the current mission, aircraft (e.g., heading, altitude, and threat) and environmental status, and is also invested with extensive a priori tactical, operational, and situational knowledge. Overall, this system provides information about the objective state of the aircraft within a mission context and uses extensive KBS to aid and support pilot decisions (Shadbolt, Tennison, Milton, & Howells, 2000).

Tasking Interface Manager (TIM)—a module that monitors the mission plan and manages the interface presented to the pilot. This module is concerned with on-line analysis of higher-order outputs from COGMON and SASS and other aircraft systems. A central function for this system is maximization of the goodness of fit between aircraft status, pilot state, and tactical assessments provided by the SASS. These integrative functions enable this system to influence the prioritization of tasks and, at a logical level, to determine the means by which pilot information is communicated. Overall, this system allows pilots to manage their interaction with the cockpit automation by context-sensitive control over the allocation of tasks to the automated systems (Bonner, Taylor, & Miller, 2000).

COGPIT Simulation Test Environment—COGSIM is concerned with the specification and provision of a proof-of-concept, Technical Demonstrator, simulation test environment for pilot-aiding. This includes the form and function of a cockpit that interprets and initiates display and automation modifications upon request, and in which the COGPIT modules will be implemented, tested, and validated. The cockpit will use adaptive interface technologies for multimodal communication. It will use aiding taxonomies and existing HF analysis methods and human-computer interaction guidelines. Computer application tools are used for prototyping, simulation, and scenario management (VAPS, VEGA, Stage).

8.3 MONITORING THE PILOT—COGNITION MONITOR

The COGMON is a COGPIT system designed to provide real-time information about the cognitive-affective state of a pilot. It derives data from four principal sources: physiology, behavior, context, and subjective states. Data from these sources are combined to update a real-time model of pilot state. This model can then be used as a basis for optimizing pilot performance, enhancing safety, and for the implementation of various on-board cockpit-aiding systems. The structure of COGMON is shown in Figure 8.10. This section provides an overview of the architecture of COGMON, its underlying theoretical basis, and ends with a discussion of the nature and uses of its outputs.

8.3.1 COGMON Functions—Pilot State Assessment

One of the basic principles underlying COGMON is the view that the term "workload" is too limited and should be replaced by the more embracing concept of "operator state." With regard to aircraft environments we view "pilot state" as a multidimensional concept. It includes, for example, levels of stress and alertness, current physical and mental demand, current locus of attention, nature of cognitive activity, current context as well as higher-order concepts such as pilot intent and situation awareness. At present there is no single measure that even remotely provides information about these various aspects of pilot state. For this reason, COGMON continuously samples a range of variables to provide a real-time model of pilot state. The data sources upon which COGMON relies can be divided into four general classes and these are now discussed in turn.

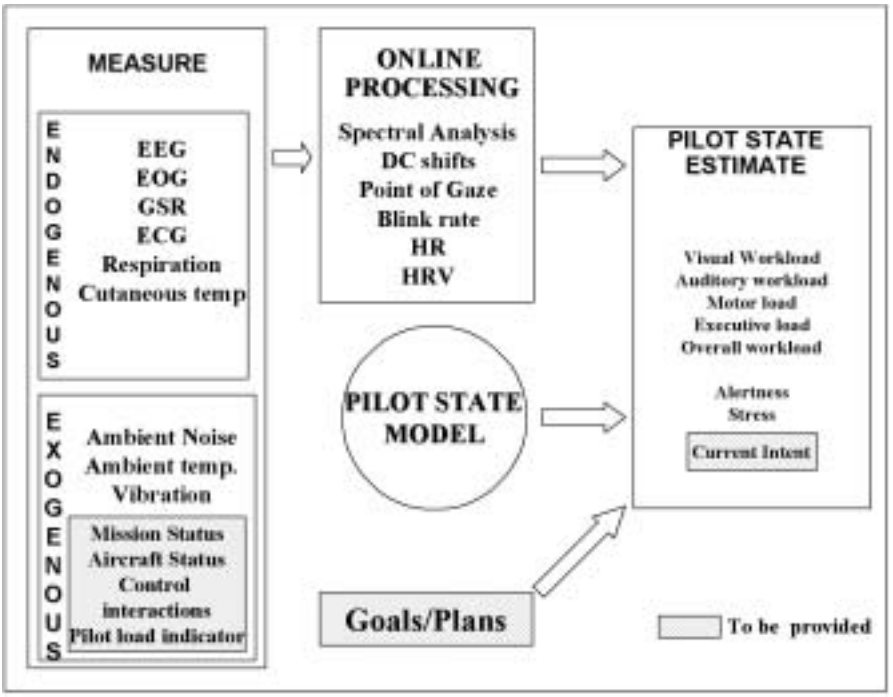


Figure 8.10: COGMON structure.

8.3.2 COGMON Measures

8.3.2.1 Physiological Measures. A full review of COGMON physiological recording and analytical facilities is beyond the present scope. However, the system includes measurement of heart rate, respiration rate, electromyogram, electrodermal activity, skin temperature, electro-oculogram, and electroencephalographic (EEG) activity. These measures provide information concerning levels of autonomic reactivity (e.g., stress) as well as information about current levels of alertness. Measurements of eye-movement activity and blink rate provide an index of visual workload, and recent improvements in biosensor technology and signal processing have allowed a dramatic improvement in locus-of-gaze detection (when head position is known). However, an optical solution to gaze location is presently seen as most promising. Recordings of brain electrical activity from the scalp also provide information about workload. For example, COGMON is capable of recording slow cortical potentials within the EEG which have been shown to be sensitive to fluctuations in cognitive demand (e.g., Pleydell-Pearce, McCallume, & Curry, 1995) and capable

of differentiating load imposed upon distinct cognitive systems (e.g., Pleydell-Pearce, 1994). COGMON also employs spectral decomposition and coherence analysis of EEG to differentiate levels of cognitive load.

It is worth noting that many physiological measures are correlated. For example, heart rate and electrodermal activity are both influenced by respiration rate (e.g., Bernston et al., 1997) and many biosensors are sensitive to thermal and vibratory artifact. For this reason, COGMON uses various mathematical tools aimed at uncoupling correlations between its incoming physiological variables. Finally, physiological sensors can be time-consuming to apply. So, the development of COGMON includes the design of fast-fit biosensors including helmet-mounted nonpolarising EEG electrodes.

8.3.2.2 Behavioral Measures. While physiological measures provide a wide range of useful information, they are presently poor at providing fine-grained information about specific forms (i.e., contents) of cognitive activity. For this reason, behavioral data, and in particular, interactions with cockpit controls, provide a rich database, which can be used to make inferences about cognitive state. Interactions with controls are monitored by COGMON for two general purposes. First, such measures permit strong inferences about the nature of ongoing cognitive activity. For example, manual interaction with a visually guided cockpit control that uses an on-screen cursor typically indicates visuospatial workload and permits the inference that visual, somatosensory, and motoric attention are invested in that task. A second major aspect of COGMON is based on the view that a great deal of pilot behavior can be decomposed into separate largely encapsulated procedures or algorithms. A crucial aspect of COGMON function is therefore the facility to recognize when these specific procedures/algorithms are being performed. Such inferences rely heavily upon interpretation of interactions with aircraft controls, although other measures taken by COGMON can supply additional information.

COGMON refers to a database to detect the onset and track the progress of specific procedures. When a particular procedure is detected, COGMON uses a stored functional taxonomy to provide information about affective and cognitive states such as stress and workload that are likely to accompany the procedure. This kind of information derives in part from a priori subjective measures (see ahead). It also depends upon a deconstruction of procedures into components based upon logical analyzes. The database can also indicate many other factors such as whether the procedure is one which when started must be taken rapidly towards completion or can be left to "idle" in the background. The database also contains information about which distinct procedures can be combined without mutual interference on both logical and empirical grounds. Novel or unusual procedures adopted by pilots may not be correctly recognized. Under such circumstances, COGMON can still gain some information based

upon lower-level monitoring of interactions with controls. For example, COGMON monitors all vocalizations from, as well as auditory inputs to, pilots. While this information may not be analyzed to the level of meaning, it does provide useful information about ongoing cognitive processes.

Specific combinations of particular procedures indicate more global goals and permit inferences about pilot intent. At this more macroscopic level, COGMON attempts to infer pilot intent using a pre-existing database in which the probable significance of particular procedural combinations is stored. Analysis at this level may also be guided by pertinent contextual information (see ahead). However, at this level, novel or unusual combinations of particular procedures may be enacted in the pursuit of complex unknown goals. Finally, the interpretation of some interactions with controls can be ambiguous. However, such sources of ambiguity can be minimized in carefully designed cockpits.

8.3.2.3 Subjective Measures. Subjective measures of pilot state are those provided by the pilot. In conventional settings these are often paper and pencil tasks (e.g., the NASA Task Load Index, Hart & Staveland, 1988). COGMON makes use of two kinds of subjective measure. "Prospective" measures can be signaled by the pilot to COGMON at any time and include communications such as "I am—drowsy, bored, stressed, or experiencing high levels of workload." We call this system the *Pilot Load Indicator* (PLI). Communication is currently made via pushbuttons. The direct communication of subjective states to COGMON provides useful additional information although the use of this system is currently seen as an issue of pilot preference. Under conditions of high stress and high workload the PLI could constitute an extra source of load although it does have a single prominent "emergency" button to signal such states. Furthermore, incorporating such measures within COGMON gives the pilot a direct link with on-board flight systems and does not therefore treat the individual as passive and "out-of-the-loop." For similar reasons we are considering the possibility of providing direct though simplified pilot feedback concerning current levels of pilot state inferred by COGMON.

A priori subjective measures are those that have been collected on the basis of interviews with pilots. Identifiable algorithms and procedures (defined above) are rated in terms of factors such as probable degrees of accompanying workload and stress. Thus when any actual task is detected, COGMON can make use of this existing knowledge. Furthermore, pilots can supply a priori information about the ease with which various separate tasks can be combined and the kinds of load that are associated with tasks and their components (e.g., visual/auditory/somatosensory, spatial/verbal or estimates of task time pressure). Finally, information concerning the stress and workload consequences of failures of various cockpit systems as well as influences of contextual measures (next) is contained within the database.

8.3.2.4 Contextual Measures. Context provides a powerful basis for interpreting pilot state data. COGMON has access to contextual information, which includes factors such as altitude, speed, levels of threat, and whether aircraft controls are functioning normally. This provides COGMON with a context for interpreting incoming data. COGMON also collects low-level contextual information as well. Examples of this include ambient noise, luminance, vibration and temperature, which are all factors known to influence pilot performance, and outputs from biosensors.

8.3.3 COGMON Implementation

8.3.3.1 Customized Systems. A characteristic feature of human performance is that there are widespread differences in behavioral and physiological responses to similar situations. This means that conclusions based upon average findings from a group of individuals may only correlate weakly with the behavior of a particular individual. However, scientific approaches to problems such as mental workload are usually based upon data averaged across subjects. In contrast, less research has attempted to identify unique but reproducible changes within single individuals. A major feature of COGMON is that it is designed to learn about the behavior of individuals and to look for predictable regularities in their particular responses to changing patterns of workload. This means that COGMON holds a database for each pilot that is activated when that pilot is identified. This is seen as a supplement to other aspects of COGMON, because in the absence of such a database, it would rely upon its noncustomized systems.

8.3.3.2 Convergent Processing. The previous sections indicate that COGMON processes a large amount of data. Although the various forms of data can be treated as separate variables, the relationships between different data sources will contain valuable information. For example, the absence of an arousal reaction to a mild threat, such as a low-altitude warning, may indicate that the pilot is confident and in control. However, it might instead indicate a loss of situation awareness caused by dangerously low levels of arousal. In recognition of the importance of convergent processing, COGMON is capable of performing complex on- and off-line multivariate analysis to improve inferences about pilot state. These routines include the facility to look for redundancy within measures. In other words, if two COGMON measures provide near identical information, then it makes sense to select the measure that is easiest to collect and process. A further benefit of convergent processing is that hidden predictive trends can often be discovered in the relations between data sets that cannot be obtained from either dataset alone. COGMON research has also employed artificial neural networks to search for "hidden" patterns within data.

8.3.3.3 A Model of Pilot State. Broadly speaking, COGMON provides an estimate of sleep-wakefulness, relaxation-stress, cognitive load (including an assessment of load imposed upon distinct modalities), an index of currently active procedures (algorithms), and an assessment of current intents and some specification of longer-term goals. COGMON outputs may also permit some estimates of situation awareness. For example, failure to have performed any (or recent) actions that might signal awareness of a particular threat would constitute grounds for inferring a deficit in situation awareness. Similarly, sustained focus of attention on a single task serves to warn that situation awareness may have decreased. Taken together, these various goals of COGMON processing constitute our multidimensional model of pilot state.

8.3.3.4 The Nature and Uses of COGMON Outputs. At present, COGMON is one component of a program aimed at the production of a cockpit that can monitor pilot state and implement automization and various forms of aiding as and when appropriate. In this system various aspects of aircraft control can be taken over by the SASS, for example, when the pilot is heavily overloaded. Decisions about which tasks will be automated are taken by the TIM, which is supplied with a constantly updated model of pilot state by COGMON. The TIM system uses this information to maintain pilot performance at optimal levels. For example, which task(s) might benefit from automation or how and where warning should be displayed? Similarly, COGMON can warn TIM if the pilot is dysfunctionally stressed, overloaded, or even underloaded and drowsy.

Another function of COGMON is its capacity to store data for later off-line analysis. This allows it to examine patterns of performance in detail, improve prediction on future flights, and update the precision of its bespoke analyzes. This facility also provides a useful tool for flight training, debriefing, and a basis for improving various aspects of flight management. More generally, COGMON architecture employs computational principles that mean its individual components can function in isolation from the whole. This is even true of the systems that interpret interventions with cockpit controls, which will work in conjunction with any suitably specified functional taxonomy. For this reason the system can be easily adapted to other platforms (in part or in entirety) and also constitutes a stand-alone research tool.

8.4 MONITORING THE ENVIRONMENT—SITUATION ASSESSOR SUPPORT SYSTEM

The Situation Assessment Support System seeks to demonstrate a COGPIT knowledge-based subsystem that will provide a dynamic assessment of the operational context and generate recommendations to support tactical deci-

sion-making. Knowledge-based decision support systems are becoming a recognized technology in the defense industry, with situation assessment and awareness recognized as a key capability in military decision-support systems. This section describes the current state of development for the knowledge-based system component of the COGPIT program, including how the SASS fits into the COGPIT and the structured methodology used to develop it. First, some background is given by describing work on other knowledge-based decision support systems involving situation assessment.

8.4.1 The Development of Knowledge-Based Systems

Previous collaborations between DERA and Epistemics Ltd have included two projects which developed real-time knowledge-based systems with a major emphasis on situation assessment. These projects were Helicopter Aircrew Decision Support (HADS) and Future Organic Airborne Early Warning (FOAEW).

8.4.1.1. Helicopter Aircrew Decision Support. In collaboration with Cambridge Consultants Ltd., this project developed a helicopter-based decision-support system for antisurface warfare. The system provides automated support for the key decisions in the principal mission tasks. It interprets available sensor data to determine the identity of each surface vessel, then plans optimum routes for helicopters to move closer to vessels to confirm their identity and analyze any threat that they may pose. Route planning takes into account the speed and direction of vessels, while prioritizing according to their possible threat.

A knowledge-based approach allowed the informal reasoning involved in the task to be described and used in a flexible manner. In such tasks, no conclusions can be certain, and they depend upon other information that is similarly uncertain. The known features of a particular contact are matched with typical descriptions of certain types of vessel: for example, a contact with a high speed is likely to be a warship or merchant ship, rather than a fishing vessel. In this application, a knowledge-based system provides an extra level of support and supervision to increase operational efficiency. The underpinning real-time, multi-agent software required for the HADS system is described by Martin and Howells (1995).

8.4.1.2 Future Organic Airborne Early Warning. This project successfully demonstrated the feasibility of a knowledge-based decision support system to aid a helicopter-based Airborne Early Warning (AEW) crew in detecting and eliminating enemy aircraft. The system performs such key tasks as placement of the helicopter barrier, identification of hostile aircraft, management of Combat Air

Patrol (CAP) aircraft, and fuel/position management. Without such a system, it is expected that future AEW operator workload will increase to levels likely to have a detrimental effect on the performance of AEW operations.

Epistemics Ltd. performed all knowledge acquisition for the system using the PC PACK software toolkit (Schreiber et al., 1999, Chapter 8), and facilitated the implementation carried out by Cambridge Consultants Ltd. The structure of the knowledge models constructed in PC PACK was replicated in the system architecture to aid in the validation, upgrading, and maintenance of future systems (Zanconato & Davies, 1999). During KA sessions, extensive use was made of generic, reusable models of problem solving, which are supported within the GDM tool in PC PACK. A full description of the GDM tool and the use of this method are given in O'Hara, Shadbolt, and Van Heijst (1998).

As Zanconato and Davies (1999) point out, the system developed was not intended as an autonomous system with which the FOAEW operator has minimal interaction. Instead, it was required to be a cooperative system in which the system and operator are able to utilize the skills most appropriate to their capabilities. As such, the design of the Man-Machine Interface (MMI) was crucial to successful operation. Hence, the system was designed to interface with the Royal Navy's latest AEW MMI. Using this system configuration in a concept demonstrator, operator's confidence in the accuracy and reliability of the advice provided increased significantly. The dynamic filtering of information coupled with the MMI displays implemented was felt to provide temporal and consistency gains in achieving overall situation assessment (Davies, 1999).

8.4.2 SASS Functions—Situation Assessment and Tactical-Decision Making

Part of the COGPIT Technical Demonstrator will be a knowledge-based decision support system, termed the SASS. The COGPIT Technical Demonstrator is intended to showcase the role of future cognitive technologies within the cockpit, with an initial focus on the FOAS (and FCBA) role. A summary of the functions of FOAS under consideration, differentiated at levels of abstraction and system decomposition, is shown in Figure 8.4. SASS is currently scoped to support offensive air mission functions, in particular monitoring the situation and recommending actions to support tactical decision-making.

As with previous approaches to situation assessment, the SASS handles situation assessment on a task-by-task basis with no separate module or agent performing situation assessment. It is believed that this integrated approach is best suited to such applications, since the knowledge used by human operators when performing situation assessment is best acquired and modeled within the context of the task being performed. In other words, expert human operators conceptualize situation assessment in a task-specific way and not as a separate activity (Klein, 1995). This approach still allows specific information on situa-

tion assessment to be requested from the knowledge-based decision support system, for example for explanation to the human operator or use in another automated module, without the need for a specific situation-assessment module.

8.4.3 SASS Methodology

By exploiting software and toolkits developed under MOD Corporate Research Programme funding, the work seeks to define, design, and construct a decision support subsystem prototype to operate in scenarios associated with the target aircraft (FOAS and FCBA), using real-time multi-agent software. The development of the SASS follows the CommonKADS model for the development of knowledge-based systems (Schreiber et al., 1999). CommonKADS is a development methodology that is the result of a number of research and applied projects on knowledge engineering over the past 16 years and has been used in a wide variety of business contexts.

CommonKADS describes a number of knowledge-level models that should be developed prior to the implementation of a KBS. These models are:

- Organizational model—organizational analysis to identify the opportunities for knowledge-intensive systems within it
- Task model—identification of the major tasks involved within the organization
- Agent model—modeling of the agents (humans, information systems, and other entities) that carry out tasks within the organization
- Knowledge model—an implementation-independent description of the knowledge components involved in carrying out a task
- Communication model—a description of the interactions between the various agents involved in a task
- Design model—a technical system specification that indicates how the knowledge model and communication model will be implemented within a specific environment.

Figure 8.11 shows how the CommonKADS models are combined: the organizational, task, and agent models provide information for the knowledge and communication models, which themselves provide information for the design model. The resulting models are then implemented according to structure preserving design principles: the implemented code should retain the organization and structure of the antecedent models (knowledge model, communication model, etc.).

The development of the organizational model used a structured approach to examine the organization and assess the feasibility of knowledge-based solutions for the problems that are identified. The assessment of feasibility of

1. For more information, see http://www.epistemics.co.uk/products/

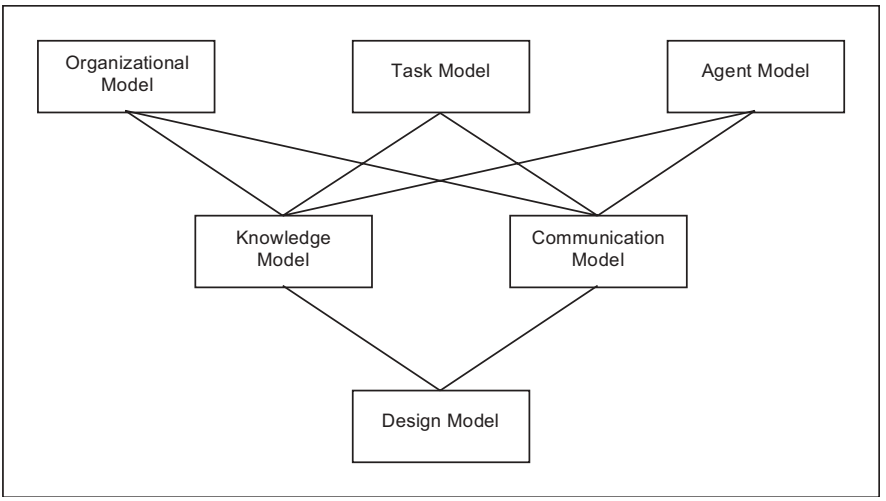


Figure 8.11: The CommonKADS models.

potential for knowledge-based solutions is summarized in Table 8.1. This scoping procedure uncovered four main areas that could benefit from knowledge-based decision support: plan assessment, system health checks, the attack phase of the mission, and the Defensive Aids Suite (DAS)/reroute task.

Extensive KA and validation have been undertaken with appropriate RAF and RN aircrew subject-matter experts (SMEs) over four vignettes on each of these task areas, leading to the production of the knowledge-base document. The individual task decompositions and detailed knowledge captured during this phase provide the basis for future architectural and software-design processes. This encapsulates all relevant expertise, for integration and aiding pilot tactical decision-making in the proposed COGPIT simulation test environment. For the purpose of the COGPIT Technical Demonstrator, current work focuses on the DAS/reroute task, which involves the use of the DAS and rerouting to counter problems caused by threats and weather.

The development of the knowledge model was substantially aided through the reuse of models, structure, and content used in the development of decision-support systems for HADS and FOAEW as described earlier. While those systems were used within helicopters, and with different tasks, a number of concepts could be reused due to the fact they were all systems to be deployed in a military airborne context.

The KA involved in the development of the CommonKADS models for the SASS has utilised a number of KA techniques, including structured interviews, laddering, repertory grid analysis, card sorts, and 20 questions. The KA was conducted in parallel with knowledge modeling, in consultation with the experts, which improves the validity of the models. The PC PACK and MetaPACK

Table 8.1: Prioritized Areas of SASS Support and Assessed Case of Supply

	Easy for Supplier	**Medium for Supplier**	**Difficult for Supplier**
Higher Priority for User	Advise on need for DAS Advise on need to route	Aid in target detection, target ID & designation Aid pilot in confidence to make weapon release	Reprioritize SA tasks Assess impact of pop-up threats & possible actions Advise on possible DAS actions Advise on optimal rerouting Advise on actions if events not occur in time
Medium Priority for User	Advise on possible actions if marshaling problems Aid in preparastion of sensor & weapon Highlight weapon/ platform/sensor system problems & possible actions Check crucial mission events occur on time	Generate and update prioritized "to do" list Aid classification of pop-up threat Use alternative means to monitor platforms if no data-link	Filter incoming information
Lower Priority for User	Highlight COMMS/ navigation system problems & possible actions Advise on EMCON	Control of automation level Anticipate/plan response to loss of INT updates Advise on sensor management	Control interaction with pilot Assess impact based on DAS information

toolsets, developed by Epistemics Ltd, have been essential in supporting the acquisition and modeling processes. The results of the KA are (1) a number of scenarios in which the SASS, and the COGPIT as a whole, can be demonstrated and evaluated, and (2) knowledge documents giving implementation-independent

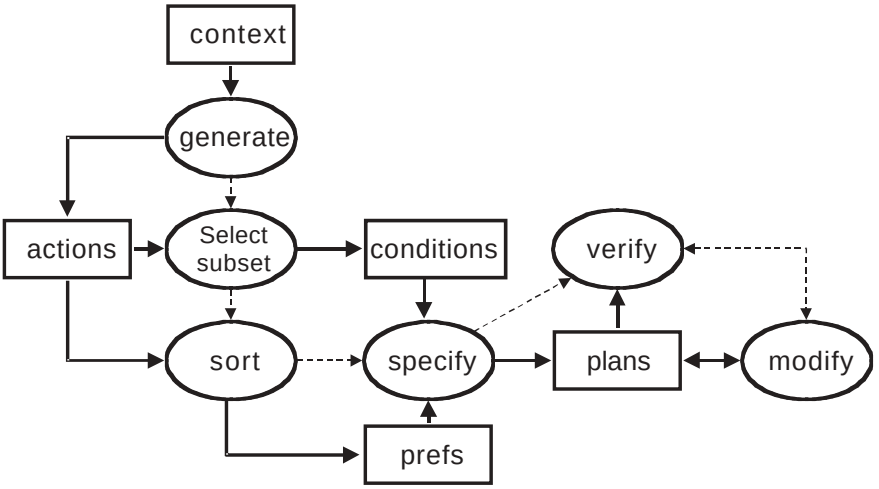


Figure 8.12: SASS replanning task process.

models of the knowledge involved in the relevant tasks. As an example, Figure 8.12 illustrates the SASS replanning task process based on CommonKADS "planning" and "rescheduling" templates that is adaptable to other tasks (e.g., plan assessment, target attack phase), and other mission support systems.

8.4.4 SASS Implementation

The implementation of the SASS will involve three stages. The first stage is a conceptual implementation, using the CLIPS expert-system shell, in which SASS will give advice on the best course of action given static situations. The second stage involves the integration of the SASS with the other modules of the COGPIT, involving the dynamic exchange of information between them. The final stage will involve the implementation of decision support for the other tasks.

The approach seeks to establish the power and utility of an incremental and structured knowledge-oriented development methodology. This improves the efficiency of KA, a classic bottleneck in system development. Moreover, this leads to substantial reuse of knowledge that has been elicited at great cost in previous projects. Finally, it is possible to demonstrate the enhanced maintainability of systems developed in this way. Together, these developments should decrease the risk associated with knowledge-intensive system development.

8.5 MONITORING THE MISSION PLAN AND CONFIGURING THE COCKPIT—TASKING INTERFACE MANAGER

The Tasking Interface Manager seeks to demonstrate real-time adaptive automation and real-time task, interface, and timeline management to support pilot operations in the COGPIT. The intended TIM application is to enable the pilot to concentrate his/her cognitive capabilities on the tactical aspects of the mission and off-load the routine activities to automation. Ideally, this would allow the pilot to remain in a planned feed-forward activity, while most, if not all reactive feedback requirements are met by decision aiding and automation. More specifically, the function of the TIM is to track goals and plans and to manage the pilot/vehicle interface and system automation. The TIM utilizes output from the SASS and the COGMON to adaptively present information and adaptively automate tasks according to the situation context and the pilot's internal state. The main features of a tasking interface are a shared mental model, the ability to track goals, plans and tasks, and the ability to communicate intent about the mission plan. This section describes the current state of development of a tasking interface component of the COGPIT program, that allows the aircrew to retain executive control of aircraft and mission parameters, while benefiting from such computerized assistance.

8.5.1 TIM Functions—Task, Timeline, Interface, and Automation Management

Among other things, as the integrated automation systems in an adaptive cockpit become more aware and capable of augmenting or even replacing pilot activities in some cases, new forms of interaction between human and automation become both possible and necessary. Our goal is the creation of an adaptive or "tasking" interface that allows an aircrew to pose a task for automation in the same way that they would task another skilled crewmember. It affords aircrew the ability to retain executive control of tasks while delegating their execution to the automation. A tasking interface will necessitate the development of a cockpit control/display interface that allows the pilot to change the level of automation in accordance with mission situation, pilot requirements, and/or pilot capabilities. It is necessary that both the pilot and the system operate from a shared task model, affording the communication of tasking instructions in the form of desired goals, tasks, partial plans, or constraints that accord with the task structures defined in the shared task model.

The function of TIM is to track goals and plans and to manage the pilot/vehicle interface and system automation. The central feature of the COGPIT Technical Demonstrator is to afford the pilot the capability to concentrate his skills towards the relevant critical mission event, at the appropriate time and to the appropriate level. This does not necessarily imply the exclusion of all other

data from the pilot; rather mission-critical information will be of primary focus and other temporally noncritical but mission important data will be presented at a lower level of salience.

The work exploits the lessons learned from the U.S. Army's Rotorcraft Pilot's Associate program (Miller & Goldman, 1999; Miller, Guerlain, & Hannen, 1999; Miller, Pelican, & Goldman, 1999) through consultancy with the U.S. developer of the tasking approach. Operators allowed to choose various levels of interaction for the tasks they are required to conduct can mitigate the problem of unpredictability of automation. This notion can be described in terms of a "tasking interface" that allows delegating a task to automation in the same way one might task an intelligent, knowledgeable subordinate. The aim is to produce a tasking interface solution tailored to the requirements of the COGPIT project that is compatible with the outputs of the SASS and CM work. TIM will utilize the monitoring and analysis of the mission tasks provided by the SASS combined with the pilot state monitoring of the COGMON to afford adaptive automation, adaptive information presentation, and task and timeline management.

Honeywell Technology Center is developing the functional requirements for the TIM. The overall architecture of an adaptive cockpit we are working with involves 12 functions, with a natural flow of information and control across the functions as loosely illustrated by Figure 8.13 and described below.

Assess pilot state information and actions. Tracking the pilot's physiological and/or cognitive state should serve as one of two broad inputs for allocating automation and configuring information presentations; the other is the needed information and task performance as dictated by the mission. This function is performed by the COGMON in our architecture.

Assess aircraft and world states and events. Tracking the state of the aircraft and the world serves as one of two broad inputs for allocating automation and configuring information presentations; the other is the pilot state. It is important to maintain this information separate from pilot actions to be able to discriminate intended states from ones that happen serendipitously. This is a function of the Situation Assessor Support System in our architecture.

Store goals and plans. This GP repository is a database of the goals, plans, and tasks that relate to the mission plan. This includes the tasks that the aircrew and aircraft system are capable of doing (possible tasks), those that relate to the specific mission plan (planned tasks) and those that are currently active (current tasks). TIM will maintain this repository in our architecture.

Interpretation of pilot and world state into intended or actual goals and plans. This function takes the outputs of the pilot state and action information described above, along with auxiliary information as needed, and includes a set of pilot-intended and actual goals and plans. These are then written into the "active" task layer of the goals and plans repository. The interpretation of pilot state and actions into pilot intended plans is a function performed by TIM in our architecture.

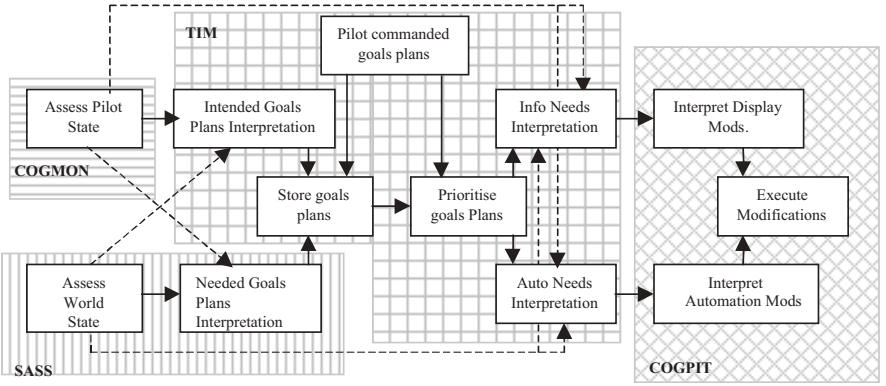


Figure 8.13: Flow of information across functions
(——→ primary - - - - → secondary).

Interpretation of pilot and world state into needed goals and plans. This function is intended to provide a parallel view of what tasks need to be performed, not necessarily those that the pilot wants to be or is performing. These are then written into the "active" task layer of the GP repository, though they should be flagged as having a different status than tasks the pilot is actually working on. This task is performed by the SASS in our architecture.

Capture of pilot statements/commands about goals and plans. This function is intended to represent explicit pilot inputs (as opposed to implicit or inferred ones) about his/her goals and plans. These are also written to the "active" or "planned" layers of the GP repository, though they should be flagged as having a different status that the other tasks in those layers. The capture, storage, and integration of these commands are a function of the TIM.

Prioritization of goals and plans. This function is responsible for asserting some degree of importance or priority on the tasks that exist in the active layer of the GP repository. This will implement some prioritization policy defined by designers and modified by pilots. The prioritization of goals is a TIM function.

Interpretation of pilot state, world state, and goals and plans into information needs. This function is responsible for determining an aggregate set of information needs from what can be determined about the pilot and world state. The function may reduce to determining the set of information needs associated with those active tasks the pilot is doing or going to do. The interpretation of pilot state, world state, and goals and plans into information needs is a TIM function.

Interpretation of pilot state, world state, and goals and plans into automation needs. This function is responsible for determining an aggregate set of tasks that are and need to be performed at the current time, then, based on reasoning about pilot capabilities, preferences and automation authorizations, for developing a set of tasks for automation to perform. The interpretation of pilot state, world state, and goals and plans into automation needs is a TIM function.

Interpretation of information needs to display modifications. Given the information needs determined above, the role of this function is to determine how to best meet those information needs in the current cockpit and the mission context. This function will take the current information display, information on the pilot's attentional state, and cognitive and perceptual resources and the environmental context to determine display modifications. This is a function performed by the cockpit automation and control/display systems, in accordance with the TIM interpretation of need.

Interpretation of automation needs to control modifications. The role of this function is to take the automation needs already determined and determine how best to meet them in the current context. This function will take the currently invoked automation, the pilot's attentional state, cognitive and physical resources, and the mission context to determine control modifications. This is a function performed by the cockpit automation and control/display systems, in accordance with the TIM interpretation of need.

Execution of display and automation modifications. Once the display and automation modification requirements are known, this function is responsible for actually implementing them in the aircraft by placing a demand to the cockpit for their activation. This is a function performed by the cockpit automation and control/display systems, in accordance with the TIM interpretation of need.

8.5.2 TIM Implementation

8.5.2.1 Shared Task Model. To develop a tasking interface, it is essential to be able to code, track, and dynamically modify user's goals and plans. The use of a "task model" format shared by both the operator and the knowledge-based planning system affords a high level of coordination between the human and the supporting system (Miller, Guerlain, & Hannen, 1999; Miller, Pelican, & Goldman, 1999). Figure 8.14 shows the general architecture for tasking interfaces. This includes a Graphic User Interface (GUI) in the form of a "Playbook" and a mission analysis component, which are based on, and communicate with each other, through a shared mental model.

The development of the shared task model for the TIM will be based upon the analytical studies that have been conducted to develop a mission description and concomitant information requirement for an offensive air mission. The knowledge elicitation utilized to develop these documents included a number of KA techniques, structured interviews, laddering, and verbal protocols based on the Goals, Means Task Analysis methodology of Roth and Mumaw (1995). The mission description and information requirement documents were

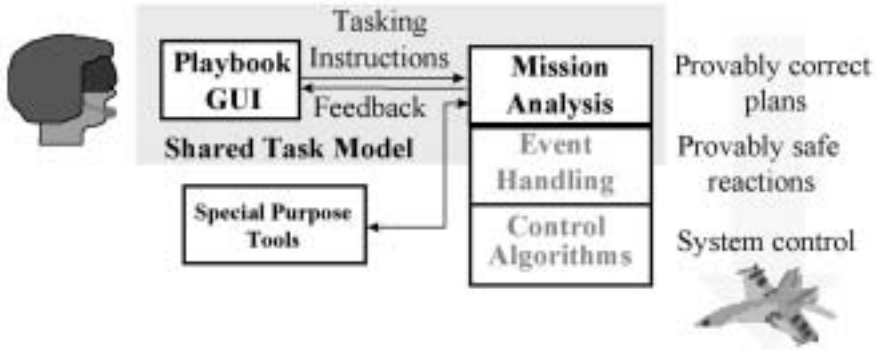


Figure 8.14: General architecture for tasking interfaces. (From C. A. Miller, M. Pelican, & R. Goldman, Tasking interfaces for flexible interaction with automation: Keeping the operator in control, *Proceedings of the International Conference on Intelligent User Interfaces*, Association for Computing Machinery [ACM], 1999. Reprinted with permission.)

developed using Jaguar SMEs and validated across Tornado ground attack and air intercept variants.

To support a tasking interface, a task model must be organized via functional decomposition, wherein there are alternative methods to achieve each task or goal. These tasks must be representative of the way pilots think of their domain and use operator-based labeling conventions (Miller, Guerlain, & Hannen, 1999; Miller, Pelican, & Goldman, 1999). The task model used for the COGPIT uses three task categories: generic tasks that are constant for a particular task for any mission, mission specific tasks that are constant for a particular task within a particular mission, and specific tasks that differ for each instance of a particular task.

8.5.2.2 TIM's Task-Tracking Capabilities. The Goal Plan Tracking (GPT) system is intended to take the form of a three-pass assessment. The first pass takes cockpit manipulation and interface information to infer a goal, a plan/objective, and a task (for example pilot stick inputs might imply SAM avoidance or acceptance of a new target or need to abort the mission). The second pass would use contextual information provided by the SASS to disambiguate the first pass (e.g., a SAM site in search mode has been located 20° on the right at approximately 20km). The final pass, which is pilot direct input, would only be used if the assessment were incorrect (for example in this situation the pilot would agree with the assessment and the TIM would then act upon this assessment to request interface modifications and automation requirements from the cockpit). The initial TIM build will be to provide a Mission Plan Tracking (MPT) capability, with later expansion to a full GPT system.

8.5.2.3 Communication About Intent. One of the goals of TIM is to allow the pilot to interact with advanced automation *flexibly* at a variety of levels. This allows the pilot to smoothly vary the "amount" of automation used depending on such variables as time available, workload, criticality of the decision, degree of trust, etc.—variables known to influence human willingness and accuracy in automation use (Riley, 1996). It further allows the human to flexibly act within the limitations imposed by the capabilities and constraints of the equipment and the world—a strategy shown to produce superior aviation plans and superior human understanding of plan considerations (Layton, Smith, & McCoy, 1994).

There are three primary challenges involved in the construction of a tasking interface:

- A shared vocabulary must be developed, through which the operator can flexibly pose tasks to the automation and the automation can report how it intends to perform those tasks. This challenge was discussed above
- Sufficient knowledge must be built into the interface to enable making intelligent choices within the tasking constraints imposed by the user. This is the role of the information and automation needs interpreters
- One or more interfaces must be developed which will permit inspection and manipulation of the tasking vocabulary to pose tasks and review task elaborations in a rapid and easy fashion.

This final challenge is one that will have to be undertaken for the FOAS fighter domain. The goal is to allow the human operator to communicate tasking instructions in the form of desired goals, tasks, partial plans, or constraints in accordance with the task structures defined in the shared task model. These are, in fact, the methods used to communicate commander's intent in current training approaches for U.S. battalion-level commanders (Shattuck, 1995). One of the authors (Miller, Guerlain, & Hannen, 1999; Miller, Pelican, & Goldman, 1999) has developed prototype tasking interfaces based on a playbook metaphor wherein the set of available plans can be described and visualized in a comparatively limited vocabulary of previously defined "plays" that can then be adapted rapidly to the current context. Figure 8.15 is an example of a prototype ground-based GUI for a tasking interface used to control unmanned combat air vehicles.

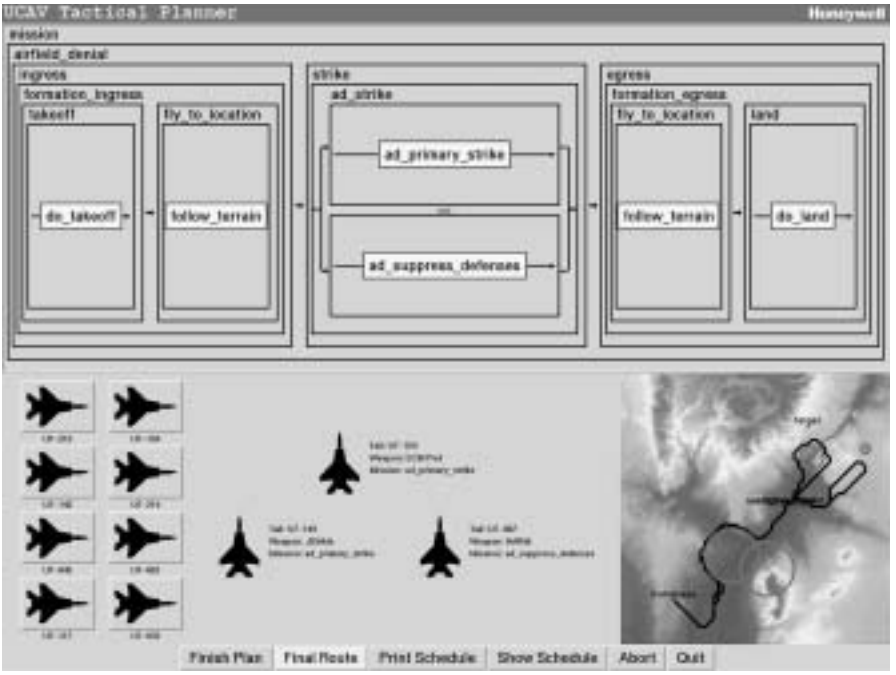


Figure 8.15: Prototype tasking interface GUI. (From C. A. Miller, M. Pelican, & R. Goldman, Tasking interfaces for flexible interaction with automation: Keeping the operator in control, *Proceedings of the International Conference on Intelligent User Interfaces*, Association for Computing Machinery [ACM], 1999. Reprinted with permission.)

8.6 PROTOTYPING, SIMULATION, AND TESTING INTELLIGENT AIDING

The COGPIT simulation and test environment is intended to provide a proof-of-concept cockpit technical demonstration of intelligent pilot-aiding (including COGMON, SASS, and TIM functions), enabling comparison of a broad set of options and providing quantifiable assessment of aiding benefit. COGSIM provides the form and function of a cockpit that interprets and initiates display and automation modifications upon request, and in which the COGPIT Technical Demonstrator modules will be implemented, tested, and validated. It will use aiding taxonomies and existing cognitive engineering and human factors analysis methods and human-computer interaction guidelines (Banbury et al., 1999).

8.6.1 COGSIM Functions—Specification, Analysis, Development, and Test

The development of the COGPIT Technical Demonstrator environment is guided by a taxonomic approach. The tenet of this approach is to scope what mission-related cockpit tasks are appropriate for machine assistance, the degree of such assistance, and the cockpit interfaces through which this interaction is likely to occur. The organization of this work is shown in Figure 8.16.

The construction of the COGPIT demonstration environment reflects an iterative approach in that the initial specification and development are followed by experimental trials, the results of which are then used to modify the environment's displays and formats. Thus, the work undertaken will include:

- Prototyping and demonstration of the cockpit and cognitive implications of intelligent aiding concepts, with appropriate human factors studies and analyzes
- Development of human-centered cockpit design principles and human-system interface guidance for automated decision support
- Provision of practical findings for cockpit design.

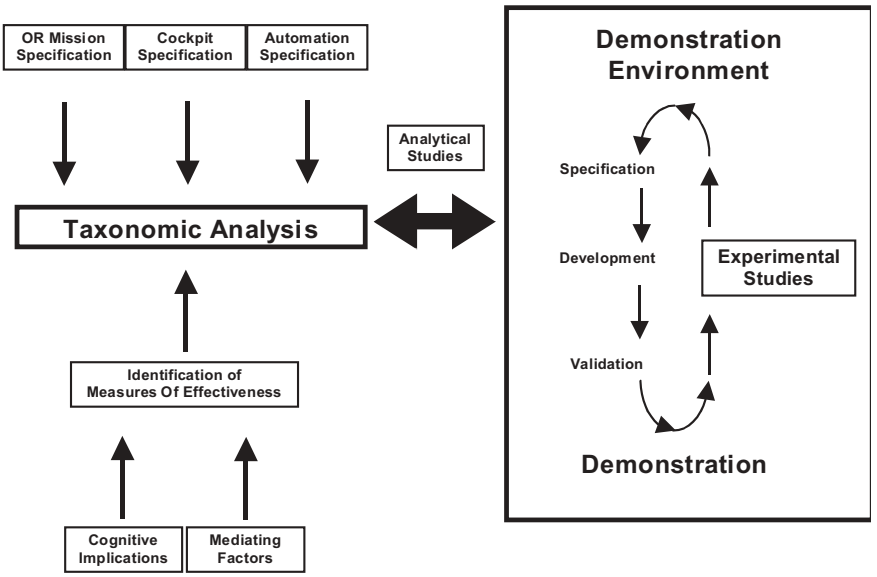


Figure 8.16: Structure of the COGPIT analytical and test work.

8.6.2 COGSIM Methodology

8.6.2.1 Measures of Effectiveness. For evaluation of COGPIT options and benefit, Measures Of Effectiveness (MOE) will be identified to provide task and mission performance metrics in the subsequent empirical evaluation. Each MOE must be mindful of the cognitive implications (i.e., capabilities and limitations of human information processing) and mediating factors (i.e., situation and environmental constraints) that exist. Mission-based MOEs provide monitoring of specific performance parameters in relation to phases of specific missions. Early work under the RPA program has shown that identification of MOEs for measuring the effects of decision aiding is a complex problem. Traditional measures of mission effectiveness, such as mission completion, arrival accuracy, and threat exposure, may not be sensitive to the effects of aiding manipulations (Casper, 1997). Linkage of MOEs to systems functions is needed to assess and analyze the benefits of specific forms of aiding. It is intended to use system functional abstraction/decomposition as a framework for developing a top-down, Function-based MOE (FMOE) system for the assessment of COGPIT intelligent aiding at the functional level. An FMOE system is also a potential source of high-level pilot feedback information, such as the "goal balls" idea for supporting the pilot's mission situation awareness, as illustrated in Figure 8.3 (Taylor & Finnie, 1999). A program of empirical validation testing is planned, with collaboration on assessment methodologies from the USAF Adaptive Interfaces program, in particular the Global Implicit Measures (GIM) approach to assessing aiding of situation awareness (Vidulich & McMillan, 2000).

8.6.2.2 Human Performance Modeling. In addition to human-in-the-loop testing, it is intended to develop a pilot cognitive model to support testing. The pilot cognitive model will be developed using the Cognitive Network of Tasks (COGNET) cognitive task analysis tool, and the iGEN software modeling environment. COGNET is particularly suited to modeling real-time operations with multitask demands on attention. The intention is to develop an executable entity model for insertion into the COGSIM scenario management tool (STAGE), for the purposes of capability and scenario development. Furthermore, the aim is to develop a COGNET model of human performance, from the mission description and task analysis data, for incorporation into COGSIM for assessment of decision aiding and automation tasks.

8.6.2.3 Aiding Taxonomy. An initial Decision Aiding Taxonomy (DAT) has been produced to provide a development framework for the suite of intelligent

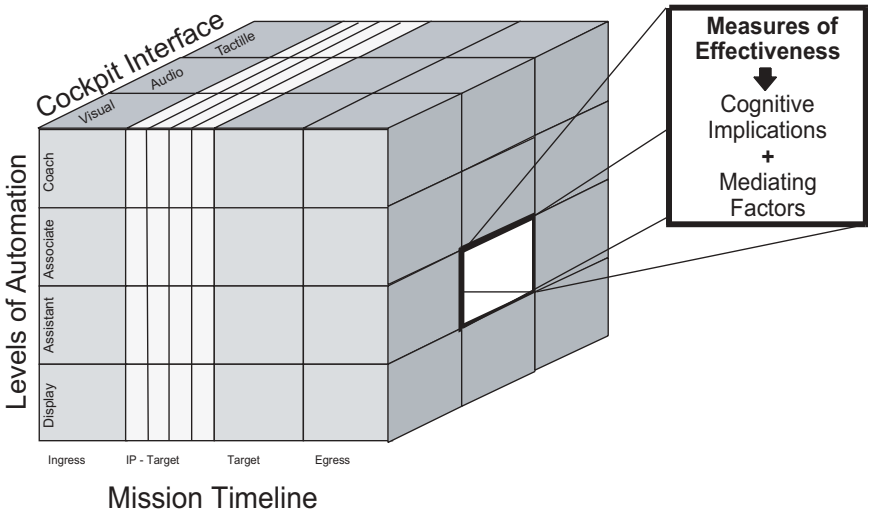


Figure 8.17: Generic format of initial decision-aiding taxonomy.

decision aids that will comprise the Cognitive Cockpit and to allow assessment of the progress of the research (see Figure 8.17). To achieve these ends, the DAT scoped a number of areas:

- The role of the human
- The role of the decision aid
- The level of automation possible
- The number of behavioral and cognitive functions possible
- The operational requirements of the scenario in which both the human and decision-aid were expected to operate, and
- The cockpit interface technologies through which this interaction can occur.

In doing so, the approach allows responsibilities to be allocated between the human and automated system, for a given mission segment, and through a specific cockpit interface (Banbury et al., 1999).

8.6.3 COGSIM Implementation

The COGSIM's main function is to provide a medium-low fidelity simulation of the out-the-window view, head-down displays, cockpit controls and displays, while also being responsible for modeling and controling the scenario

and all the entities in the simulation (ground, air, and sea). The following commercially available software applications were selected for use:

- Virtual Application Prototyping Software (VAPS) which provides a rapid development environment specifically for avionics interfaces and systems
- Scenario Toolkit and Generation Engine (STAGE) which drives the simulator scenario and handles all entities and allows some degree of entity scripting, map importing, and distributed interactive simulation broadcasting
- Flight Simulation (FLSim) which provides a medium fidelity aerodynamic aircraft model which will be used for "ownship"
- MultiGen which is a modeling and simulation suite of software tools and modules
- VEGA which provides an out-the-window scene with the capability of handling large terrain databases.

The current state of development of the COGPIT program is summarized in Table 8.2, using the WDA abstraction-decomposition framework. The architecture of the COGPIT is being developed to include an XML-server approach to communications with other modules in the assignment. XML was chosen as an ideal means of rapid and reliable distribution of data and integration of modules developed on different machines, operating systems and networks, because the HTTP protocol and internet-based technology has been well proven. Due to the performance-cost ratio and recent rapid advancement in PC technology, PCs where chosen over other platforms such as SGIs for this assignment.

8.6.3.1 Baseline Cockpit. The interface technologies selected for inclusion in the cockpit simulation were chosen from an analysis of the maturity of candidate technologies for implementation in FOAS time-scales (Table 8.3). The intention is to provide comparison of the baseline cockpit (Eurofighter/F22 interface standard) with candidate cockpit configurations with TIM adaptive interfaces supported by COGMON and SASS. Consideration will be given to investigating the support for adaptiveness afforded by flexible, large area head-down displays, and multimodal display techniques for head-up, out-of-the-cockpit operations, in particular, the use of a Helmet-Mounted Visual Display (HMD) coupled with voice and 3D audio cueing.

8.6.3.2 Information Requirements Analysis. Information requirement analysis performed with the project SME pilot on the mission scenario description and storyboard, has identified the preferred modality, saliency, and method of dis-

Table 8.2: Summarized COGPIT Functional Decomposition

Abstraction Level	**Functional Decomposition**
Functional purpose and constraints	Help demonstrate pilot aiding benefit; Support development of plot aiding functionality; Support growth of DERA cognitive engineering capability
Abstract functions, priorities and values	Human-system performance assessment; Systen emulation, simulation; Pilot T&E; System design; Development; Research tool; Requirements analysis
Generalized purpose related functions	STAGE Entity management; VEGA/VAPS Responding to pilot interaction; VAPS/FLSim Providing cockpit system options; SASS/TIM Adaptation and pilot aiding; COGSIM Monitoring performance parameters; MultiGen/VEGA Providing realism; COGMON Assessing pilot functional state
Physical functions	COGMON Monitoring pilot parameters; SASS Monitoring environment, Replanning mission; TIM Configuring cockpit, Deciding automation, Monitoring & updating mission plan; COGSIM/Cockpit Automating tasks, Entity representation, Gathering mission, contact & pilot interaction data, Exchange information, Display warnings, interrupts, status & tasks
Physical forms and systems	DVI/DVO, 3-D sound, Radio voice links, Data links, HOTAS, Touch screen, Visually coupled HMD; Geo data base, Map; RWR, MAWS, Radar, DASS, ECM Chaff/flares; Warning system, Automatic FCS, Mission planning & re-planning systems; SASS Tactical KBS; TIM GPT system; COGMON biosensors, amplifiers; HTTP server, blackboard

play information for the baseline cockpit, and indicated the need for prioritization of the saliency of TIM advice (Taylor, Abdi, Dru-Drury, & Bonner, 2000). Mission analysis indicates that the primary display requirement is to

Table 8.3: Baseline Cockpit Interface Technologies

Display Technologies	Control Technologies
• Helmet Mounted Display	• Direct Voice Input
• Large Head-Down Display	• Active Throttle and Stick Control
• Direct Voice Output	• Hands-On Throttle And Stick (HOTAS)
• 3D Audio	• Helmet Tracking
	• Eye Tracking (pilot-state monitoring)
	• Hard Keys/X-Y Controller
	• Touch Screen

support head-up, eyes-out of the cockpit operation. The work has examined requirements for information at five levels of saliency, namely:

- Background—information
- Hinting—messages
- Influencing—suggestions
- Directing—warnings
- Compelling—alerts

SME analysis has indicated the need for TIM information to be managed, organized, and easily digested, so as not to unduly add to pilot workload. Direct voice input/output (DVI/DVO) is the preferred primary modality for TIM dialogues. Initial analysis has identified the need to provide a distinction between feed-forward primary information, and feedback secondary information, with appropriate levels of saliency in display, for example, center HMD location for feed-forward, and peripheral HMD location for feedback.

8.6.3.3 Control Requirements Analysis. In addition, consideration will be given to examining options for supporting adaptation using alternative control technologies, in particular coupling Hands-On-Throttle-And-Stick (HOTAS) operations with DVI and head-tracking (Hudgins et al., 1998). Eye-tracking is to be considered initially only for pilot-state monitoring, rather than for controlling systems, because of limitations on aiming accuracy with current technology. Allocation of control functions to HOTAS is guided by the requirements for speed of learning, ease of use and simplicity of operation. HOTAS is a primary mode of control for many critical functions, but it is intended to provide only a back-up to DVI and soft keys for TIM input. Particular attention is being given in the design of DVI protocols to control task information requirements and the

design of feedback (Dru-Drury, Farrell, & Taylor, 2001). This is to provide an implementation of DVI consistent with principles of PCT (Powers, 1973; Taylor, 1992). Specifically, this is to allow the operator to flexibly pose tasks to the automation and for the automation to report on its intended action.

8.6.3.4 Control of Tasks. Recent analysis of the operator requirement for pilot authorizing and control of levels of automation, with the envisioned TIM support, has led to the development of the COGPIT PACT system (Bonner, Taylor, Fletcher, & Miller, 2000). The PACT system uses military terminology (Under Command, At Call, Advisory, In Support, Direct Support, Automatic) to distinguish realistic operational relationships for five aiding levels, with progressive pilot authority and computer autonomy supporting situation assessment, decision making, and action (Table 8.4). These are a reduced, practical set of levels, with clear engineering and interface consequences, derived from the ten levels of automation for human-computer decision-making proposed by Sheridan and VerPlanck (1978). The PACT terminology and selection of levels are based on operational considerations that are consistent with theory to afford usability and compatibility with military user cognitive schemas and models. It is envisaged that mission functions and tasks, at different levels of abstraction, will be allocated to these levels. The operator could control this allocation in a number of ways:

- Preset operator preferred defaults
- Operator selection during pre-flight planning
- Changed by the operator during in-flight replanning, and
- Automatically changed according to operator agreed, context-senstive adaptive rules.

8.6.4 COGSIM Demonstration and Test

8.6.4.1 Test Functions. The intended COGPIT application is to enable the pilot to concentrate his/her cognitive capabilities on the tactical aspects of the mission (knowledge-based) and off-load the routine (rule-based and skill-based) activities to automation. In effect, this will allow the pilot to remain in a feed-forward loop while, most, if not all, feedback requirements are met through decision aiding and automation. The principal functions that will need to be tested are as follows:

- That the SASS provides useful rule-based decision-aiding informa-

Table 8.4: PACT System for Pilot Authorization of Control of Tasks

Levels	Operational Relationship	Computer Autonomy	Pilot Authority	Adaptation	Information on Performance
5	**Automatic**	Full	Interrupt	Computer monitored by pilot	On/off Failure warnings Performance only if required
4	**Direct Support**	Action unless revoked	Revoking action	Computer backed up by pilot	Feedback on action. Alerts and warnings on failure of action
3	**In Support**	Advice, and if authorised, action	Acceptance of advice and authorizing action	Pilot backed up by the computer	Feed-forward advice and feedback on action; Alerts and warnings on failure of authorised action
2	**Advisory**	Advice	Acceptance of advice	Pilot assisted by computer	Feed-forward advice
1	**At Call**	Advice only if requested	Full	Pilot, assisted by computer only when requested	Feed-forward advice, only on request
0	**Under Command**	None	Full	Pilot	None, performance is transparent

tion, according to the situational context. For example, progressively providing avoid, evade, and defeat action requirements against ground and air threats as the scenario develops

- That the COGMON provides useful pilot-state information (cognitive capability) according to the pilot's physiological condition. For example, providing the TIM with the information that the pilot is high on visual and cognitive workload coupled with a high alertness and high arousal but low activity
- That the TIM affords the ability to adaptively provide information

according to the situational context and either selectively (pilot controlled) or adaptively (TIM-controlled) offload tasks to automation in accordance with the mission plan. For example, the TIM could adaptively increase the automation level on aspects of the DAS and aircraft defensive maneuvre to allow the pilot to concentrate on the ramifications of the threat avoidance to mission completion.

The basic experimental test design is illustrated in Table 8.5. In this design, the independent variables are the classes of cockpit and types of aiding. The classes of cockpit comprise conventional (i.e., baseline) and candidate (i.e., with adaptive aiding and adaptive automation). The types of aiding comprise, none (control condition), COGMON, SASS, and COGMON working together with SASS. TIM provides the integration of COGMON and SASS in the candidate cockpit, with adaptive aiding and adaptive automation. Function-based MOEs will provide a large choice of dependent variables to measure performance. The selection of the dependent variables is related to two factors: (1) the external validity of the measure in terms of its relevance to the real world; and (2) the internal validity of the measure in terms of its suitability to measure a particular aspect of performance.

Table 8.5: Basic Experimental Test Design

Between Subjects	**Conventional**				**Candidate**			
Within Subjects	None	COG-MON	SASS	COG-MON + SASS	None	COG-MON	SASS	COG-MON + SASS

8.6.4.2 Usage Scenario. The design of the COGPIT and its Technical Demonstration are based upon MOD customer-agreed scenarios and missions. Based on a NATO Studies Advisory Group European scenario and mission, three individual scenarios have been derived for technical demonstration of DAS/Rerouting (weather, threat update, SA–8 pop-up threat). These have different mission plans, timelines, and priorities and provide variations or tweaks to exercise specific functionality (known threat, loss of data link and RAP, loss of GPS, hostile AA, chaff failure, fuel leak, two missiles, ambiguous RWR tracks). The scenarios were originally developed as aids to the SASS knowledge acquisition to give contexts for the processes involved in the FOAS mis-

sion. In addition, the scenarios were chosen to show key situations where the COGPIT should aid the pilot. They are therefore selected to be challenging (e.g., flying over unfamiliar terrain, for extended duration and therefore fatiguing, and able to defeat the baseline cockpit), to demonstrate added value of aiding, and to show the COGPIT concept of operation.

8.6.4.3 Mission Story-Boards. Mission descriptions have been captured as mission timelines and analyzed by pilot SME to provide decomposition into goals, functions and tasks, systems operations, control inputs, and information display requirements. The methodology is derived from role-playing narrative procedures used to identify technologies and cueing for air-to-ground fighter integration (Boucek et al., 1996; Montecalvo, Redden, Rolek, Orr, & Barbato, 1994). The mission descriptions cover all potential tasks within a mission. They will be used to develop MOEs and to analyze automation and adaptive aiding requirements, in accordance with the DAT and PACT frameworks. Mission story-boards providing detailed, scripted tactical vignettes have been created in accordance with strategic factors in the high-level scenario and used successfully for the purposes of technical demonstration. The method of demonstration used to date has been a coupling of real-time simulated flight involving some automated actions (DAS, not auto-pilot) together with cognitive walk-through. The approach used is similar to the decision-centered approach to story-boarding used under the U.S. Navy TADMUS program by Miller, Wolf, Thorndsen, and Klein (1992) to provide focus on the situational dynamics. The mission story-boards are intended to drive the detailed COGSIM development to demonstrate COGPIT functionality. The priorities for COGPIT story-board development were the need to follow the mission description development, to allow each COGPIT component to demonstrate its functions, and that they should be based on KA with pilot SMEs, and not developed arbitrarily.

During demonstration, the simulation paused at selected decision points to provide explanation of SASS provided plans and to describe TIM automation of aircraft systems and presentation of information to pilots. COGMON operation has been demonstrated stand-alone with a simulated cockpit (F22) computer-game flying task. This showed significant sensitivity to types of physical activity, input and output modalities, and to changing levels of concentration, arousal, and cognitive load. Future work will seek to provide a real-time demonstration of the integrated functioning of COGMON and SASS working through TIM in the COGSIM cockpit.

8.7 CONCLUSIONS

The work has exercised a wide range of cognitive systems engineering methods in bringing together cognitive technologies for intelligent knowledge-based pilot-aiding. These cognitive technologies include pilot functional state monitoring—in its infancy in providing on-line measurement and interpretation for task adaptation—and task knowledge management and decision support for context sensitive aiding—applying relatively mature knowledge engineering techniques to support adaptiveness in real time. Considered in terms of the capability maturity levels suggested by Geddes (1997), the coupling of cognitive technologies proposed in the COGPIT project provides a capability at the level of "coach," using complex task, situation, and user knowledge. As proposed, the system will be capable of recognizing the need for automation to achieve a mission objective and of providing instructions to the operator on how to achieve it, and/or implement the required automation where necessary. A summary of the methods, tools, and techniques used on the COGPIT project in the phases of development of the COGPIT systems, including cognitive systems engineering, is shown in Table 8.6.

Functional analysis of cognitive work provides essentially the foundations for the successful development and implementation of cognitive technologies for pilot-aiding. Recent developments on cognitive work analysis seem particularly promising in providing a broad set of models and tools for human systems analysis, based on a high-level functional analysis (abstraction/decomposition framework), but they are not designed for ease of transfer into computer code. The CommonKADS methodology and PC PACK software toolkit for knowledge engineering seem particularly useful for implementing knowledge-based systems. However, there is sufficient commonality in the CWA and CommonKADS approaches, to afford validation of either.

Work to date has provided mission-based functional decomposition, cognitive task analyzes, knowledge acquisition and modeling, interface prototyping, initial proof-of-concept simulation, and cognitive story-board evaluation. The analysis is based on assumptions concerning future capabilities and technical developments, which require SMEs to extrapolate from their knowledge base. This poses considerable problems in validation. A particularly difficult area is the analysis of cognitive requirements of future automation capabilities.

A baseline conventional EF22 cockpit has been built, with initial scenario scripting for a partial prototype proof-of-concept demonstration. The idea that the scenario should be sufficiently difficult, so as to defeat the baseline cockpit, is an important scenario and COGPIT design driver. The validity of the scenario and the missions needs to be checked and maintained to ensure the validity of the findings for the intended, platform-specific application. However, the basic aiding concepts and technologies are likely to be generalizable to other applications and domains.

Table 8.6: Summary of COGPIT Engineering Methods, Tools, and Techniques

Phase of Work	COGMON	SASS	TIM	COGSIM
Requirements	KA Custom profile Empirical analysis Structuralist inter-ference model of performance limitation	KA PC Pack Meta Pack Knowledge documents	KA GMTA COGNET	KA GMTA WDA Control task analysis PCT Cognitive Streaming SRK taxonomy DAT taxonomy EID Aiding HCI Style Guide
Specification and Design	Modular redundant architecture Near real-time Interleaved processing Polynomial decomposition Spectral analysis Regression analysis Coherence analysis Slow wave detection Loss line detection Artifact rejection and correction Rule-based artificial intelligence Recursive artificial neural networks	CommonKADS Organisational model Task model Agent model Knowledge model Communication model Design model	CommonKADS based	Interface Design Document HTML Corba Sockets
Implementation	Ethernet protocol Risk PC C code ARM code Time critical assembly language Basic 5	CLIPS expert system shell Ethernet protocol	Ethernet protocol Visual C++ Blackboard-based mission plan tracker	VAPS VEGA MultiGen FLSim STAGE LADBM Ethernet protocol
Simulation	Acorn RISC PC RISC-OS	Microsoft NT	Microsoft NT	Microsoft NT iGEN
Test and Evaluation	Iterative empirical validation F22 Computer game FMOE	FMOE Prototype story board Cognitive walk-through	FMOE Prototype story board Cognitive walk-through	Aiding Test plan PC-based and COGSIM-based experiments Iterative empirical validation FMOE

There has been some initial development of the COGPIT modules. Work so far indicates that on-line pilot functional state assessment is feasible with current computing power, and looks like providing useful information for cockpit and task adaptation. In particular, the increased power of individual profiles for developing custom adaptations seems a highly promising development.

Knowledge engineering methodology can provide useful on-line knowledge-based systems to support for pilot replanning tasks, and this has the potential for wider application. The traditional KA bottleneck has been significantly reduced by the provision of a structured methodology and tool set (CommonKADS, PC Pack and Meta Pack). Demonstration has highlighted the criticality of the timing of KBS advice in context.

Useful assistance in the management of cockpit interfaces, tasks, and automation can be provided by a tasking interface system based on a shared task model. The development of an effective TIM, with which pilots can interact easily, will be critical for the successful integration and acceptance of the outputs of the COGMON and SASS subsystems. The technical specification of a tasking interface for this type of system is a major task, particularly as the functional components require iterative development, precluding early definition of inputs and outputs. While it is relatively easy to track tasks instantiated in a mission plan, it becomes very difficult to track and support tasks that deviate from the intended plan. Tracking deviations requires the system to infer likely pilot intent, which is inherently problematic.

Further work is needed to identify the precise methods for cockpit adaptation and their benefits and to determine the optimization of control/display interfaces, in particular for DVI/O dialogue and HMD ramifications. A function-based system for MOEs could provide useful information on mission confidence for on-line pilot feedback, and for analysis of the benefits of aiding options. Future work seems likely to extend the system functionality and scenarios, to provide integration of subsystems for evaluating candidate cockpit options, and to consider wider applications, such as supporting the control of multiple UAVs, and export to other work domains.

REFERENCES

Banbury, S. (1999). *Experimental assessment of current models of interference between concurrent activities* (DERA/CHS/MID/CR990091/1.0). Farnborough, England: DERA Center for Human Sciences.

Banbury, S., Bonner, M., Dickson B., Howells H., & Taylor, R. M. (1999). *Application of adaptive automation in FOAS (manned option)* (DERA/CHS/MID/CR990196/1.0). Farnborough, England: DERA Center for Human Sciences.

References

Berntson, G. G., Bigger, J. T., Eckberg, D. L., Grossman, P., Kaufman, P. G., Malik, M., Nagaraja, H. N., Porges, S. W., Saul, J. P., Stone, P. H., & Van Der Molen, M. W. (1997). Heart rate variability: Origins, methods and interpretive caveats. *Psychophysiology, 34*, 623–648.

Billings, C. E. (1997). *Aviation automation: The search for a human-centered approach*. Hillsboro, NJ: Erlbaum.

Bonner M. C., Taylor R. M., Fletcher K., & Miller C. (2000). Adaptive automation and decision aiding in the military fast jet domain. In *Proceedings of the Conference on Human Performance, Situation Awareness and Automation: User-Centered Design for the New Millennium*.

Bonner, M., Taylor R. M., & Miller C. (2000). Tasking interface manager: Affording pilot control of adaptive automation and aiding. In P. T. McCabe, M. A. Hanson, & S. A. Robertson (Eds.), *Contemporary ergonomics 2000* (pp. 70–74). London: Taylor & Francis.

Boucek, G. S., Orr, H. A., Williams, R. D., Montecalvo A. J., Redden, M. C., Rolek, E. P., Cone, S. M., & Barbato, G. J. (1996). *Integrated mission/precision attack cockpit technology (IMPACT). Phase II: Cueing benefits of large tactical situation displays, helmet mounted displays, and directional audio* (Tech. Rep. WL–TR–96–3076). Wright-Patterson AFB, OH: Wright Laboratory

Brehmer, B. (1992). Dynamic decision making: Human control of complex systems. *Acta Psychologica, 81*, 211–241.

Casper, P. A. (1997). A full mission simulation success story: RPA simulation at CSRDF yields promising results. In *Proceedings of the Human Factors and Ergonomics Society 41st Annual Meeting* (Volume 1, pp. 95–99). Santa Monica, CA: HFES.

Davies, A. J. (1999). *Assessment of FOAEW decision support concept demonstrator and advice to DOR(Sea) concerning route to staff target definition* (Tech. Rep. DERA/AS/SID/CR990117/1.0). Farnborough, England: DERA Center for Human Sciences.

Dru-Drury, R., Farrell, P. S. E., & Taylor, R. M. (2001). Cognitive cockpit systems: Voice for cognitive control of tasking automation. In D. Harris (Ed.), *Engineering psychology and cognitive ergonomics – Volume five: Aerospace and transportation systems* (pp. 99–107). Aldershot, England: Ashgate.

Eggleston R. G. (1993). Cognitive interface considerations for intelligent cockpits. In *AGARD Conference Proceedings No 520, Combat automation for Airborne Weapons Systems: Man/Machine Interface Trends and Technologies* (pp. 21-1–21-16). Neuilly-sur-Seine, France: NATO Advisory Group for Aerospace Research and Development.

Eggleston, R. (1997). Adaptive interfaces as an approach to human-machine co-operation. In M. J. Smith, G. Salvendy, & R. J. Koubek, (Eds.), *Design of human-computer systems: Social and ergonomic considerations* (pp. 495–500). Amsterdam: Elsevier.

Emerson, T., Reinecke, M., Reising, J., & Taylor, R. M. (1988). *The human-electronic crew: Can they work together?* (Tech. Rep.WRDC–TR–TR–7008). Wright-Patterson AFB, OH: Wright Laboratory,

Emerson, T., Reinecke, M., Reising, J., & Taylor, R. M. (1992). *The human-electronic crew: Is the team maturing?* (Tech Rep. WL–TR–TR–92–3078). Wright-Patterson AFB, OH: Wright Laboratory.

Farrell, P., & Chery, C. (1998). PTA: Perceptual control theory based task analysis. In *Proceedings of the Human Factors and Ergonomics Society 42nd Annual Meeting* (Volume 2, pp. 1314–1318). Santa Monica, CA: HFES

Geddes, N. D. (1997). Associate systems: A framework for human-computer co-operation. In M. J. Smith, G. Salvendy, & R. J. Koubek, (Eds.), *Design of human-computer systems: Social and ergonomic considerations* (pp. 237–242) Amsterdam: Elsevier.

Haas, M. W., Beyer, S. L., Dennis, L. B., Brickman, B. J., Hettinger, L. J., Roe, M. M., Nelson, W. T., Snyder, D. B., Dixon, A. L., & Shaw, R. L. (1997). *An evaluation of advanced multi-sensory display concepts for use in future tactical aircraft* (Tech. Rep. AL/CF–TR–19997–0049). Wright-Patterson AFB, OH: Armstrong Laboratory.

Hart, S. G., & Staveland, L., (1988). Development of NASA–TLX (Task Load Index): Results of empirical and theoretical research. In P. A. Hancock & N. Meshkati (Eds.), *Human mental workload* (pp. 139–183). Amsterdam: Elsevier.

Hayes-Roth, B. (1991). Making intelligent systems adaptive. In K. Van Lehn (Ed.), *Architectures for intelligence*. Hillsdale, NJ: Erlbaum.

Hendy, K., Liao, J., & Milgram, P. (1997). Combining time and intensity effects in assessing operator information processing load. *Human Factors, 39*, 30–37.

Hettinger, L. J., Cress, J. D., Brickman, B. J., & Haas, M. W. (1996). Adaptive interfaces for advanced airborne crew stations. In *Proceedings of the 3rd Annual Symposium on Interaction with Complex Systems* (pp. 188–192). Los Alamitos, CA: IEEE Computer Society Press.

Hollnagel, E., (1993). *Human reliability analysis: context and control*. London: Academic.

Hollnagel, E. (1996). Decision support and task nets. In S. A. Robertson (Ed.), *Contemporary Ergonomics 1996* (pp. 31–36). London: Taylor & Francis.

Hollnagel, E. (1997). Control versus dependence: Striking the balance in function allocation. In M. J. Smith, G. Salvendy, & R. J. Koubek (Eds.), *Design of human-computer systems: Social and ergonomic considerations* (pp. 243–246). Amsterdam: Elsevier.

Hollnagel E. & Woods, D. D., (1983). Cognitive systems engineering: New wine in new bottles. *International Journal of Man-Machine Studies, 18*, 583–600.

Hudgins, B., Leger, A., Dauchy, P., Pastor, D., Pongratz, H., Rood, G., South, A., Carr, K., Jarret, D., McMillan, G., Anderson, T., & Borah, J., (1998). *Alternative control technologies* (Tech. Rep. 7. AC/323[HFM]TP/3). Neuilly-sur-Seine, France: NATO Research and Technology Organisation.

Klein, G. (1995). *Naturalistic decision making*. Wright-Patterson AFB, OH: Crew System Ergonomics Information Analysis Center.

Layton, C., Smith, P., & McCoy, E., (1994). Design of a cooperative problem solving system for enroute flight planning: An empirical evaluation. *Human Factors, 36*, 94–119.

Linde L., & Berggrund U., (1999). *Interface design and decision support in a cognitve cockpit* (Tech. Rep. FOA–R–99–01097–706–SE). Linkoping, Sweden: FOA Defence Research Establishment.

Martin, S., & Howells, H., (1995). Real time software for knowledge based systems. In *IEEE Colloquium on Real Time Systems*. London: IEEE.

McNeese, M., (1995). Cognitive engineering: A different approach to human-machine systems. *CSERIAC Gateway, 6*(5), 1–4.

Miller C. A. & Goldman, R. (1999). Tasking interfaces: Associates that know who's the boss. In J. Reising, R. M. Taylor, & R. Onken, (Eds.), *The human electronic crew: The right stuff? Proceedings of the 4th joint GAF/RAF/USAF workshop on human-computer teamwork, Kreuth, Germany* (Tech. Rep. AFRL–HE–WP–TR–1999–0235, pp. 97–102). Wright-Patterson AFB, OH: Air Force Research Laboratory.

Miller C. A., Guerlain S., & Hannen, M. (1999). The rotorcraft pilot's associate cockpit information manager: Acceptable behaviour from a new crew member. In *Proceedings of the American Helicopter Society, 55th Annual Forum*, Montreal, Quebec.

Miller, C. A., Pelican, M., & Goldman, R. (1999). Tasking interfaces for flexible interaction with automation: Keeping the operator in control. In *Proceedings of the International Conference on Intelligent User Interfaces*, Redondo Beach, CA.

Miller T. E., Wolf, S. P., Thorndsen, M. L., & Klein, G. K. (1992). *A decision-centered approach to story-boarding anti-air interfaces* (Final Tech. Rep. Task 3). Fairborn, OH: Klein Associates.

Montecalvo A. J., Redden, M. C., Rolek, E. P., Orr, H. A., & Barbato, G. J. (1994). *Integrated mission/precision attack cockpit technology (IMPACT). Phase I: Identifying technologies for air-to-ground fighter integration* (Tech. Rep. WL–TR–94–3143). Wright-Patterson AFB, OH: Wright Laboratory.

Norman, D. A. (1986). Cognitive engineering. In D. A. Norman & S. W. Draper (Eds.), *User-centered system design* (pp. 32–61). Hillsdale, NJ: Erlbaum.

O'Hara, K. & Shadbolt, N. R. & Van Heijst, G. (1998). Generalised directive models: Integrating model development and knowledge acquisition. *International Journal of Human-Computer Studies, 49*, 497–522.

Pleydell-Pearce, C.W. (1994). DC potential correlates of attention and cognitive load. *Cognitive Neuropsychology, 11*, 149–166.

Pleydell-Pearce, K., & Dickson, B. (2000), Cognition monitor: A system for real-time functional state assessment. In P. T. McCabe, M. A. Hanson, & S. A. Robertson (Eds.), *Contemporary ergonomics 2000* (pp. 65–69). London: Taylor & Francis.

Pleydell-Pearce, C. W., McCallum, W. C. & Curry, S. H., (1995). DC shifts and cognitive load. In G. Karmos, M. Molnar, V. Csepe, I. Czigler, & J. E. Desmedt (Eds.), *Perspectives of event related potentials research. Supplement 44 to Electroencephalography and Clinical Neurophysiology* (pp. 302–311). Amsterdam: Elsevier.

Powers, W. T. (1973). *Behavior: The control of perception*. Chicago: Aldine.

Rasmussen, J. (1976). Outlines of a hybrid model of the process plant operator. In T. B. Sheridan & G. Johannsen, (Eds.), *Monitoring behaviour and supervisory control* (pp. 371–383). New York: Plenum.

Rasmussen, J. (1986). *Information processing and human machine interaction: An approach to cognitive engineering*. New York: Holland.

Rasmussen, J., Pejtersen, A. M., & Goodstein, L. P. (1994). *Cognitive engineering: Concepts and applications*. New York: Wiley.

Reising, J. (1979). The crew adaptive cockpit: Firefox here we come. In *Proceedings of the 3rd Digital Avionics Conference*, Fort Worth, TX.

Reising, J, Taylor, R. M., & Onken, R. (1999). *The human-electronic crew: The right stuff?* (Tech. Rep. AFRL–HE–WP–TR–1999–0235). Wright-Patterson AFB, OH: Air Force Research Laboratory.

Riley, V. (1996). Operator reliance on automation: Theory and data. In R. Parasuraman & M. Mouloua (Eds.), *Automation and human performance: Current theory and applications* (pp. 19–36). Hillsdale, NJ: Erlbaum.

Roth, E. A., & Mumaw, R. J. (1995). Using cognitive task analysis to define human interface requirements for first-of-a-kind systems. In *Proceedings of the Human Factors Society 32nd Annual Meeting* (Volume 1, pp. 520–524). Santa Monica, CA: HFS.

Rouse , W. B., (1976). Adaptive allocation of decision-making responsibility between supervisor and computer. In, T. B. Sheridan & G. Johannsen (Eds.), *Monitoring behaviour and supervisory control* (pp. 295–306). New York: Plenum.

Rouse, W. B. (1988). Adaptive aiding for human/computer control. *Human Factors, 30*, 431–443.

Rouse W. B. & Rouse S. H. (1983). *A framework for research on adaptive decision aids* (Tech. Rep. AFAMRL–TR–83–082). Wright-Patterson AFB, OH: Air Force Aerospace Medical Research Laboratory.

Sanderson, P., Eggleston, R., Skilton, W., & Cameron, S. (1999). Work domain analysis workbench: Supporting cognitive work analysis as a systematic practice. In *Proceedings of the Human Factors and Ergonomics Society 43rd Annual Meeting* (Volume 1, pp. 323–327). Santa Monica, CA: HFES.

Sanderson, P., Naikar, N., Lintern, G., & Goss, S. (1999). Use of cognitive work analysis across the system life cycle: From requirements to decommissioning. In *Proceedings of the Human Factors and Ergonomics Society 43rd Annual Meeting* (Volume 1, pp. 318–322). Santa Monica, CA: HFES.

Schreiber, A. Th., Akkermans, J., Anjewierden, A., De Hoog, R., Shadbolt. N., Van De Velde, W., & Wielinga, B. (1999). *Knowledge engineering and management: The CommonKADS methodology*. Cambridge, MA: MIT Press.

Shadbolt, N. R., Tennison, J., Milton, N., & Howells, H. (2000). Situation assessor support system: A knowledge-based systems approach to pilot aiding. In P. T. McCabe, M. A. Hanson, & S. A. Robertson (Eds.), *Contemporary Ergonomics 2000* (pp. 60–64). London: Taylor & Francis.

Shattuck, L. (1995). *Communication of intent in distributed supervisory control systems*. Unpublished doctoral dissertation. The Ohio State University, Columbus, OH.

Sheridan T. B., & VerPlanck, W. L. (1978). *Human and computer control of undersea tele-operators* (Tech. Rep.). Cambridge, MA: MIT Man Machine Laboratory.

Taylor, R. M. (1992). Principles for integrating voice I/O in a complex interface. In *Proceedings of AGARD–CP–521* (pp. 21-1–21-21). Neuilly sur Seine, France: NATO Advisory Group for Aerospace Research and Development.

Taylor, R. M. (1997), Human electronic crew teamwork: Cognitive requirements for compatibility and control with dynamic function allocation. In M. J. Smith, G. Salvendy, & R. Koubek (Eds.), *Design of human-computer systems: Social and ergonomic considerations* (pp. 247–250). Amsterdam: Elsevier.

Taylor, R. M., Abdi, S., Dru-Drury, R., & Bonner, M. C. (2001). Cognitive cockpit systems: Information requirements analysis for pilot control of automation. In D. Harris (Ed.), *Engineering psychology and cognitive ergonomics—Volume five: Aerospace and transportation systems* (pp. 81–88). Aldershot, England: Ashgate

Taylor, R. M., & Finnie, S. E. (1999). The cognitive cockpit: Adaptation and control of complex systems. In J. Reising, R. M. Taylor, & R. Onken (Eds.), *The human-electronic crew: The right stuff? Proceedings of the 4th joint GAF/RAF/USAF workshop on human-computer teamwork. Kreuth, Germany* (AFRL–HE–WP–TR–1999–0235, pp. 75–88). Wright-Patterson AFB, OH: Air Force Research Laboratory.

Taylor, R. M., Finnie, S., & Hoy, C. (1997). Cognitive rigidity: The effects of mission planning and automation on cognitive control in dynamic situations. In, *Proceedings of the Ninth International Symposium on Aviation Psychology* (Volume 1, pp. 415–421). Columbus, OH: The Ohio State University.

Taylor, R. M., Howells, H., & Watson, D. (2000). The cognitive cockpit: operational requirement and technical challenge. In P. T. McCabe, M. A. Hanson, and S. A. Robertson (Eds.), *Contemporary ergonomics 2000* (pp.55–59). London: Taylor & Francis.

Taylor R. M., MacLeod I. S., & Haugh J. (1995). *Schema-based methods for cockpit information integration* (Tech. Rep. DRA/CHS/HS3/CR95035/01). Farnborough, England: DERA Center for Human Sciences.

Taylor R. M., & Reising J. (1995). *The human-electronic crew: Can we trust the team?* (Tech. Rep. WL–TR–96–3039). Wright-Patterson AFB, OH: Wright Laboratory.

Taylor, R. M., & Reising, J. R. (1998). The human-electronic crew: Human-computer collaborative team working. *RTO Meeting Proceedings 4*, (22-1–22-17). Neuilly sur Seine, France: NATO Research and Technology Organisation.

Taylor, R. M., & Selcon, S.J. (1993). Operator capability analysis: Picking the right team. In *AGARD conference proceedings No. 520, Combat automation for airborne weapons systems: Man/machine interface trends and technologies* (pp. 20-1–20-17). Neuilly-sur-Seine, France: NATO Advisory Group for Aerospace Research and Development:

Taylor, R. M., Shadrake, R., & Haugh, J. (1995). Trust and adaptation failure: An experimental study of uncooperation awareness. In R. M. Taylor & J. Reising (Eds.), *The human-electronic crew: Can we trust the team?* (Tech. Rep. WL–TR–96–3039, pp. 87–92). Wright-Patterson AFB, OH: Wright Laboratory.

Taylor, R. M., Shadrake, R., Haugh, J., & Bunting, A. (1996). Situational awareness, trust, and cognitive compatibility: Using cognitve mapping techniques to investigate the relationships between important cognitive system variables. In *AGARD conference proceedings No. 575, Situation Awareness: Limitations and Enhancement in the Aviation Environment* (pp 6-1–6-14). Neuilly-sur-Seine, France: NATO Advisory Group for Aerospace Research and Development.

Tennison, J. (1999). *Technical demonstrator of a situation assessment support system: Task, agent and communication model knowledge document* (Tech. Rep. FOAS2/WP200/Comm/KD Version 2). Nottingham, UK: Epistemics.

Tremblay, S., & Emery, E. (2000). *Features of cognitive streaming* (Tech Rep. DERA/CHS/MID/CR000138/1.0). Farnborough, England: DERA Center for Human Sciences.

Vicente, K. J. (1999). *Cognitive work analysis: Towards safe, productive, and healthy computer-based work*. Hillsdale, NJ: Erlbaum.

Vidulich M., & McMillan, G. (2000), The global implicit measure: Evaluation of metrics for cockpit adaptation. In P. T. McCabe, M. A. Hanson, & S. A. Robertson (Eds.), *Contemporary ergonomics 2000* (pp. 75–80). London: Taylor & Francis.

Zanconato, R., & Davies, A. (1999). Design for a knowledge-based decision support system to assist near littoral airborne early warning (AEW) operations. In J. Reising, R. M. Taylor, & R. Onken (Eds.), *The human-electronic crew: The right stuff? Proceedings of the 4th joint GAF/RAF/USAF workshop on human-computer teamwork* (Tech. Rep. AFRL–HE–WP–TR–1999–0235, pp. 45–56). Wright-Patterson AFB, OH: Air Force Research Laboratory.

© British Crown Copyright 2000/DERA
Published with the permission of the controller of Her Majesty's Stationery Office

9 Beyond Cognitive Systems Engineering: Assessing User Affect and Belief States to Implement Adaptive Pilot-Vehicle Interaction

EVA HUDLICKA
Psychometrix Associates

MICHAEL D. MCNEESE
The Pennsylvania State University

ABSTRACT

We describe an Affect and Belief Adaptive Interface System (ABAIS) designed to compensate for performance biases caused by users' affective states and active beliefs. The ABAIS architecture implements an adaptive methodology consisting of four steps: sensing/inferring user affective state and performance-relevant beliefs; identifying their potential impact on performance; selecting a compensatory strategy; and implementing this strategy in terms of specific Graphic User Interface (GUI) adaptations. ABAIS provides a generic adaptive framework for integrating a variety of user assessment methods (e.g., knowledge-based, self-reports, diagnostic tasks, physiological sensing), and GUI adaptation strategies (e.g., content- and format-based). The ABAIS performance bias prediction is based on empirical findings from emotion research combined with detailed knowledge of the task context. The initial ABAIS prototype was demonstrated in the context of a U.S. Air Force combat task, used a knowledge-based approach to assess the pilot's anxiety level, and adapted to the pilot's anxiety and belief states by modifying selected cockpit instrument displays in response to detected changes in those states. Preliminary results indicate feasibility of the ABAIS approach, raise a number of further research questions, and suggest specific requirements for a successful, operational affect and belief-adaptive interface (e.g., limiting the number, type, and resolution of affective and belief states; using multiple methods and individualized data for user state assessment; implementing "benign" adaptations [e.g., adaptations

should never limit access to existing information]). The focus on affect and belief represents a new area of research in joint cognitive systems. Results of this effort suggest that existing cognitive systems engineering methods, and the resulting designs, may not go far enough if they limit themselves to the exclusive consideration of cognitive and motor factors, and fail to place adequate emphasis on affect and beliefs as critical factors influencing performance.

9.1 INTRODUCTION

The mutual influence of cognitive schemata and contextual constraints is considered the accepted basis for cognitive systems engineering practices (Hollnagel & Woods, 1983; Rasmussen, Pejtersen, & Goodstein, 1994). Less generally accepted is the fact that *affective states* can also dramatically influence human performance and decision making, via effects on attention, perception, situation assessment, and ultimately action selection (LeDoux, 1992; Williams et al., 1997).

Recent research provides increasing evidence that individual differences in general, and affective states in particular, have a major impact on performance (Deckert et al., 1994; Eysenck, 1997; Isen, 1993; LeDoux, 1992; Mineka & Sutton, 1992; Williams et al., 1997). Affective states influence a variety of perceptual, cognitive,, and motor processes by influencing both low-level perceptual, cognitive and motor processes (e.g., attention, memory), and by influencing higher-level processes such as situation assessment, decision making, and judgment. Examples of these influences include:

- Altering the nature of attentional processing (e.g., change focal area, increase/reduce size of focal area, bias attention toward or away from particular stimuli, etc.)
- Helping to activate (or inhibit) particular perceptual and cognitive schemata that enhance (or limit) the perception of processing of specific stimuli
- Promoting (or inhibiting) the selection of particular actions, and influencing the accuracy and speed of selected motor responses.

Similarly, experimental studies indicate that the user's current assessment of the situation, in other words, his/her *belief state*, plays a critical role in the decision making and, ultimately, response selection.

As computer systems requiring user adaptation and associated User Interface (UI) technologies mature and proliferate into critical applications, and increasingly heterogeneous user populations, it becomes particularly important that they recognize and adapt to individual user characteristics. To accommodate these requirements, we suggest that Cognitive Systems Engineering (CSE) methods must therefore enlarge their traditional focus on

cognition and context to include the user's individual characteristics. Successful user adaptation requires the understanding of these individual characteristics and their interactions with task and situation variables, including other human and synthetic agents, to create successful designs (see McNeese, Chapter 3 this volume). This is particularly true for the user's dynamic characteristics that involve affective and belief states, since both these factors strongly influence performance. This requirement is necessary for the wide variety of decision support and user interface systems being deployed in the aviation domain (e.g., see Reising, Chapter 1 and Taylor et al., Chapter 8 this volume).

While some progress has been made in user-modeling and adaptive user interfaces (see Section 9.3 below), the majority of existing decision-support and cognitive modeling systems continue to assume normative performance, and fail to adapt to the individual characteristics of particular users, whether those that are relatively stable over time, or those that are susceptible to situation influences. This is particularly true for adaptation with respect of the user's current situation assessment (belief states) state and current emotion (affective states).[1]

This existing lack of detection, assessment, and modeling of belief and affective states on the one hand, and adaptation to these states on the other, in the majority of human-machine systems can lead to nonoptimal behavior at best, and critical errors with disastrous consequences at worst. This is increasingly evidenced by a variety of accidents and incidents attributed to the broad area of "human error" that exist in commercial and military aviation, a variety of industry processes, and, increasingly, in health care (Institute of Medicine, 1999).

To address these influences we must develop user models that take into account the effects of affect and belief on performance, and develop strategies for adapting the machine aiding and user interface to the users' individual, possibly idiosyncratic, affect and belief states. A number of issues arise in developing adaptive user interfaces capable of identifying the user's affective and belief state and compensating for the resulting biases in performance. The primary challenges are:

- Effective assessment of the user's current affective state
- Prediction of its influences on performance, in the current task context
- Identification of strategies that could compensate for the potential performance biases
- Generation of corresponding Graphic User Interfaces (GUIs) and associated computer system adaptations.

To address these challenges we developed an Affect and Belief Adaptive Interface System (ABAIS) adaptive methodology and implemented a prototype ABAIS system. The development of ABAIS and the associated adaptive methodology are analogous to cognitive system engineering practices and may be considered as one example of what McNeese (Chapter 3 this volume) con-

1. We use the term "emotion" and "affective state" interchangeably, referring to transient states, with distinct triggers and individual decay functions, roughly at the level of basic emotions or Category 2 emotions (Panskepp, 1994).

siders as a *knowledge as model as design* approach, wherein knowledge is defined as inclusive of affective and belief states. In many cases, however, this approach goes beyond the scope of narrow implementations of cognitive systems engineering by the very fact of including emotions and beliefs.

The ABAIS prototype implements this methodology in terms of an architecture shown in Figure 9.1. The architecture consists of four modules. *User State Assessment* provides a framework for integrating a variety of methods to identify the user's affective and belief state (e.g., knowledge-based, self-reports, diagnostic tasks, physiological sensing). *Impact Prediction* integrates generic empirical findings with the results of task-specific Cognitive Affective Personality Task Analysis (CAPTA) (Hudlicka, 2000a) to predict the most likely effects of user states on performance. *Strategy Selection* combines the CAPTA results with individual preferences to derive an appropriate compensation strategy. Finally, *GUI Adaptation* implements this strategy by modifying the content and/or format of the user interface (see Figure 9.2).

ABAIS implements an adaptive methodology framework capable of adapting the system interface format and content to the user's affective state and situation-specific beliefs that might influence performance (Hudlicka, 2000b; Hudlicka & Billingsley, 1999, 1998).

The ABAIS prototype was developed and demonstrated in the context of an Air Force combat task simulation. ABAIS assessed the pilot's anxiety and belief states via a knowledge-based approach, using information from a variety of sources (e.g., task characteristics, pilot personality, etc.), predicted the effects of user state on performance, and suggested and implemented specific GUI adaptation strategies based on the pilot's information presentation preferences (e.g., modified icon/ display to capture attention, etc.).

This chapter is organized as follows. First, we provide a review of existing empirical research on the effects of affect and belief states on performance, and a generic summary of affect assessment methods (Section 9.2), and briefly discuss specific existing work in affective assessment and adaptation (Section 9.3). Next, we describe the ABAIS adaptive methodology and the system architecture that implements this methodology (Section 9.4). We then briefly describe the task context: Air Force fighter pilot sweep task, to provide the necessary background for the concrete examples of system functionality described in subsequent sections (Section 9.5). We then outline an enhanced form of cognitive task analysis, which explicitly includes the possible effects of affect and personality traits on performance, and is therefore termed *cognitive affective personality task analysis* (CAPTA) (Section 9.6). Next we describe the process of user affective and belief state assessment and behavior prediction (Section 9.7), and the strategy selection and specific GUI adaptation strategies (Section 9.8). We then illustrate the overall ABAIS prototype functionality by a brief description of system performance in the context of the demonstration tasks: Air Force fighter pilot sweep task (Section 9.9). The paper concludes with a

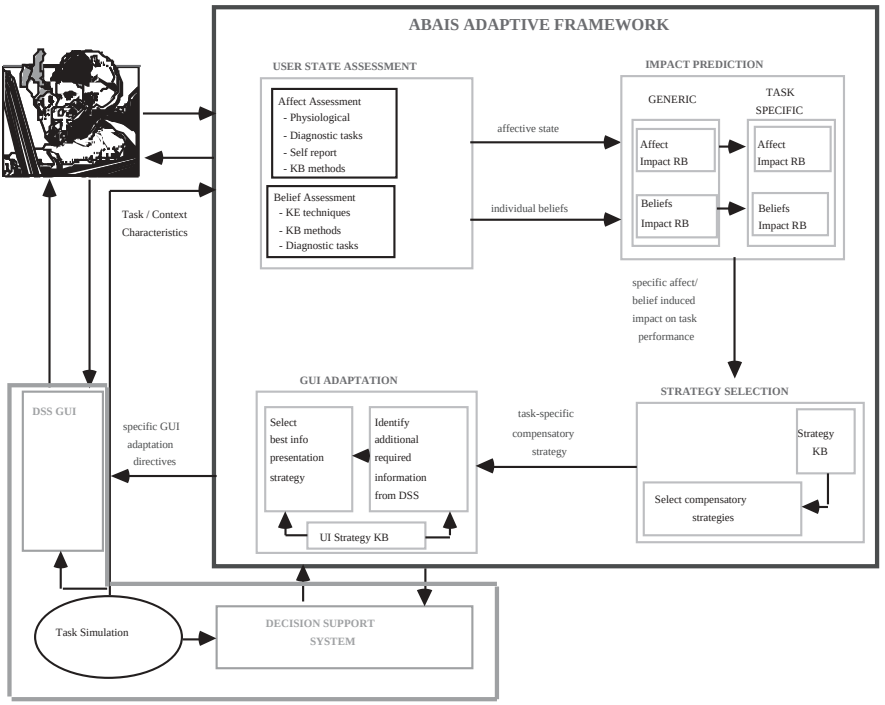


Figure 9.1: ABAIS system architecture.

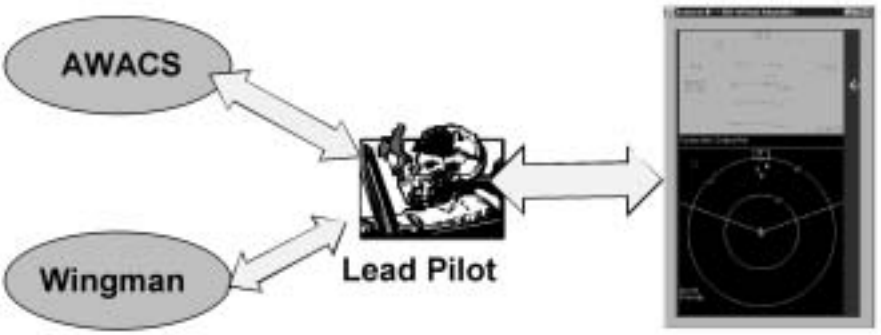


Figure 9.2: High-level diagram of key demonstration task components.

summary, conclusions, and brief outline of future work (e.g., system performance evaluation) and generalizability of the ABAIS methodology to other domains (Section 10.0).

9.2 AFFECTIVE AND BELIEF STATES: EFFECTS ON PERFORMANCE AND ASSESSMENT

This section provides background on existing research most relevant to our effort to develop an affect and belief adaptive system. Section 9.2.1 summarizes the effects of emotion on cognition and performance. Section 9.2.2 reviews generic methods for assessing affective states. Section 9.2.3 summarizes research on situation awareness and its relevance to assessing, and adapting to, the user's belief state.

9.2.1 Effects of Affective States on Performance

Although central to human development and functioning, emotions have, until recently, had a somewhat marginal status in cognitive science, neuroscience, and human factors. Over the past ten years, however, important discoveries in neuroscience and experimental psychology have contributed to an interest in the scientific study of emotion. A growing body of evidence from neuroscience research points to the existence of circuitry processing emotionally relevant stimuli (i.e., stimuli that threaten or benefit the survival of the organism or its species) (LeDoux, 1989). LeDoux and colleagues have studied fear conditioning in rats and identified a number of key results: (1) existence of dedicated circuitry processing stimuli that threaten or benefit organism or species survival; (2) evidence that emotional circuitry performs fast, less differentiated processing and behavior selection (e.g., freezing behavior in rats); and (3) evidence that this processing is mediated by connections linking sensory organs directly to emotional circuitry in the brain, specifically, the amygdala (LeDoux, 1992). Cognitive psychologists have described a variety of appraisal processes involved in inducing a particular emotional state in response to a situation (Lazarus, 1991) and several models have been proposed (Ortony et al., 1988), some of which have been implemented in computational models (Bates et al., 1992; Elliot, 1992; Scherer, 1993). Damasio and colleagues (1994) have studied humans with brain lesions and identified the role of emotion in human information processing and decision-making, suggesting that emotions "prune" the search spaces generated through cognitive processing. Recent research thus provides evidence for the impact of emotion on cognitive processing and the central role of emotion in the control of behavior. The emerging findings also begin to blur the distinction between what has traditionally been thought of as the separate realms of cognition and emotion.

Of relevance to the ABAIS system are the consistent findings by cognitive and clinical psychologists regarding the differential impact of various emotional states on cognition and a number of affective states and personality traits have been studied extensively (e.g., anxiety, positive and negative affect,

Table 9.1: Effect of Emotion and Personality Traits on Cognition: Examples of Empirical Findings

Anxiety and Attention (Williams et al., 1997; Mineka and Sutton, 1992).
Narrowing of attentional focus
Predisposing towards detection of threatening stimuli
Affective State and Memory (Bower, 1981; Blaney, 1986)
Mood-congruent memory phenomenon—positive or negative affective state induces recall of similary valenced material
Obsessiveness and Performance (Persons and Foa, 1984; Sher et al., 1989)
Delayed decision-making
Reduced ability to recall recent activities
Reduced confidence in ability to distinguish among actual and imagined actions and events
Affect Judgement and Perception (Isen, 1983; Williams et al., 1997)
Depression lowers estimates of degree of control
Anxiety predisposes towards interpretation of ambiguous stimuli as threatening

obsessiveness, extraversion, etc.). These factors influence perceptual and cognitive processes, including attention, perceptual categorization, memory, and general inferencing and judgment. Examples of findings are shown in Table 9.1. These findings provide an empirical basis for predicting the generic effects of emotional states and personality traits on performance. These generic effects can be used in the absence of task-specific information, and also serve as guiding principles for the affective/cognitive task analysis required to generate specific performance effects, in the context of particular situations.

9.2.2 Assessment of Affective States

Existing methods for affect assessment include psychological self-report instruments, physiological sensing, facial expression recognition (Ekman & Davidson, 1994; Kaiser et al., 1998; Picard, 1997), speech analysis, diagnostic tasks, and expert observer evaluation (knowledge-based assessment). However, none of these methods alone provide a definitive solution to this difficult task. Assessing affective states is an inherently difficult problem, in large part due to the variation in the expression of these states, both across and within individuals (Picard, 1997). Effective assessment of emotional states therefore requires a combined use of a number of methods. Below we provide general background information about several methods and outline their relevance for ABAIS. Section 9.3 then discusses specific affective modeling and adaptation systems using some of these methods.

9.2.2.1 Psychological Instruments/Self-Reports. Self-reports using standardized psychological instruments (i.e., "pencil and paper" or computer-administered questionnaires) represent an established means of affect and personality trait assessment, in both clinical and experimental settings. A variety of validated instruments exist for a broad range of affective states and personality traits, both "normal" and "pathological" (e.g., specific affective state and trait measures, specific personality trait instruments, social performance assessment instruments, workload and stress measures) such as the Minnesota Multiphasic Personality Inventory [MMPI] [Hathaway & McKinley, 1989]; State-Trait Anxiety Scale; Anxiety Sensitivity Index [Peterson & Reiss, 1987]; Positive and Negative Affect Scales—PANAS [Watson, Clark, & Tellegen, 1988); Beck Depression Inventory, etc.]. In addition, domain-specific instruments also exist (e.g., aviation-oriented Armstrong Laboratory Aviation Personality Survey [ALAPS] [Retzlaff et al., 1997].

While it is often the case that the best way to find what people are feeling is to ask them, this method has several associated difficulties. Specifically, (1) use of self-reports can be impractical in real-time environments except to provide background information; (2) self-reports can be inaccurate due to the user's inability to recognize certain affective states which may, nevertheless, influence performance; and (3) self-reports may be inaccurate because the user may not wish to reveal the desired information.

In spite of these drawbacks, self-reports can be a valuable resource for affect assessment, provided the following criteria are met:

- The task environment provides opportunities for brief, simple self-assessment of specific affective states (e.g., a simple dialog box with a "Are you now feeling anxious?" and a "YES/NO" option)
- Users can be trained to accurately differentiate between the affective states of interest (e.g., high-and low-anxious state)
- Cooperation of users in providing accurate information can be ensured.

Implications for ABAIS. Existing instruments can be used in two ways as part of the self-assessment procedure in ABAIS. First, during an off-line, initial assessment, to provide background information and suggest generic effects on performance. Second, during brief on-line assessment, where one or two items would be presented to the users *during* task performance, to provide a specific, targeted assessment of their current affective state. This latter application of self-reports would need to be compatible with the task context and would not be appropriate in all situations.

9.2.2.2 Physiological Sensing. A large number of physiological assessment methods exist, varying in intrusiveness, reliability of the obtained data, and diagnos-

ticity. Affect assessment via a combination of physiological sensing methods is currently an active research area, in part due to the technological advances in wearable computing, and in part due to the increasing interest in emotion research (Picard, 1997). While physiological sensing methods provide objective assessments in terms of measurable, physical signals, a number of issues exist regarding their usefulness in reliably identifying distinct affective states. These are both theoretical and practical. From the *theoretical perspective*, these methods rely on the assumption that different affective states have unique, detectable physiological signatures. Much research has been devoted to the identification of "basic" emotions (Ekman & Davidson, 1994) and their differentiation using physiological measures (Cacioppo et al., 1993). However, the results of these efforts are often difficult to replicate. (Note, however, recent work by Picard and colleagues [Vyzas & Picard, 1998], who report increasing success in differentiating among several of the basic emotions using physiological data.) From the *practical perspective,* physiological sensing has until recently been both cost-prohibitive and impractical, due to the amount and type of equipment necessary for accurate assessment (e.g., large, bulky equipment for heart-rate measures; uncomfortable, interfering electrodes for EEG and EMG, etc.).

Fortunately, however, there have been both theoretical and practical advances in recent years, to make selected physiological assessment feasible. *On the theoretical side*, one approach is to limit the assessment to a differentiation between high and low arousal, and positive and negative valence, rather than a large set of affective states. Arousal is reflected in a number of physiological measures, including heart rate, pupil size, and skin conductance measures. Valence is best assessed by monitoring selected facial muscles via EMG sensors (i.e., corrugator muscle activity increases during negative affect and decreases during positive affect, and activity of the zygomatic muscle increases during positive affect). *On the practical side*, the emerging technology of wearable computers makes fast, unobtrusive measurement of a variety of physiological signals feasible.

Implications for ABAIS. While a number of experimental methods are theoretically available, results of the literature search reveal that most reliably assessed affective measures are arousal and valence. The best practical signal for arousal detection is heart rate (Cacioppo et al., 1993; Hugdahl, 1995; Orr, 1998). Other measures of arousal, such as galvanic skin response, pupil size, blood volume pressure, etc., either do not provide additional data and/or are not as readily assessed. While skin conductance measures represent a better direct measure of anxiety, as opposed to arousal, the requirement of finger or palm sensors makes these impractical in computerized, automated environments.[2] The best means of assessing valence appears to be facial EMG, focusing on the corrugator and zygomatic muscle groups. The selection of specific methods and measures is thus highly task- and interface-dependent. Given the task considerations of the ABAIS prototype, the most appropriate physiological signal appears to be heart rate as a measure of anxiety level.

2. Note, however, rapidly emerging nonintrusive and wearable devices that may make complex physiological assessment practical in the near future.

9.2.2.3 Diagnostic Tasks. Diagnostic tasks represent an additional means of assessing affective state in real time, using a variety of behavioral observations and performance metrics (e.g., accuracy, reaction time, type and timing of specific errors, etc.). Diagnostic tasks require either passive user monitoring (e.g., speed of responding to alerts), or, in certain conditions, the injection of specific diagnostic probes to collect measures of interest (e.g., sequence of artificial stimuli identification tasks, etc.). A key advantage of diagnostic tasks is their ability to provide an individualized means of assessing actual performance biases in real time, rather than relying solely on indirect measures of anxiety/arousal. As with physiological sensing, while it would be unrealistic to use this method for differentiation among subtle affective states, it appears feasible to indirectly infer levels of anxiety or key personality traits (e.g., obsessiveness, aggressiveness, etc.). Thus, for example, failure of the user to repeatedly respond to critical new data can be interpreted as task neglect due to anxiety, and repeated interrogation of data sources for the same items can be interpreted as obsessiveness.

Implications for ABAIS. This approach to affect assessment appears feasible, provided the following criteria are met:

- Domain-specific diagnostic tasks can be identified and customized to individual users and task context
- Individual baseline performance data can be collected for comparison during real-time assessment
- Implementation of a nonintrusive means of collecting the necessary data in real time is feasible.

9.2.2.4 Expert Observer Evaluation—Knowledge-Based Assessment. This method uses an expert, knowledge-based system approach, where a number of knowledge bases are first constructed from knowledge elicited from experts or technical literature, or derived from cognitive task analysis. This knowledge is encoded in a variety of representational formalisms (e.g., rules, belief nets, etc.). Both static user data (e.g., individual history, personality traits, task characteristics) and dynamic user data (e.g., current affective state, workload, physiological signals, etc.) then provide data used by the inferencing mechanisms to derive likely user affective state from factors such as current task context, personality traits, and individual history. This approach in effect emulates an expert observer, familiar with the task and the user, who combines the relevant data and determines the most likely user state. A key advantage of this method is that it allows the *simulated* implementation of a combination of multiple methods by assuming the existence of a variety of static and dynamic data and integrating these during assessment (e.g., pilot's individual history, real-time task events, simulated heart-rate measures, etc.). In some sense, this approach is analogous to the cognitive appraisal theory of affect generation, which posits

that affective states result from cognitive evaluations of the individual's goals and expectations, current situation, interpersonal environment, etc. A key distinction between the knowledge-based and cognitive appraisal process, other than the obvious difference between who is doing the assessment, is that the former draws on a broader variety of data (e.g., physiological data, diagnostic tasks, etc.).

Implications for ABAIS. The ability to combine a wide variety of data makes the knowledge-based approach an ideal candidate for an initial prototype implementation, designed to test the feasibility of integrating multiple assessment methods within a single inferencing formalism and architecture. In addition, there are the obvious benefits of increasing assessment reliability by integrating multiple data sources.

9.2.3 Effect of Belief States on Performance

Much recent research in decision making and skilled human performance, particularly in dynamic, real-time settings, has focused on the concept of situation assessment and situation awareness (Endsley, 1995). Briefly, situation awareness refers to the individual's ability to rapidly identify salient cues in the incoming data and map those cues onto a small set of relevant situations, which then guide further action selection. Situation assessment and situation awareness may take various forms and are highly dependent upon the mutual constraints among individuals, teamwork, and environmental factors. A series of extensive studies of situation assessment in the military and other real-time settings have been conducted by Klein and colleagues (1989) as well as others (Elliot et al., 2001; Serfaty et al., 1997; Wellens, 1993). The Klein (1997) approach has labeled this process Recognition-Primed Decision-Making (RPD), and identified RPD as a key element in tactical planning.

To the extent that affective state and personality traits influence attention, perception, and cognition, they play a major role in influencing all aspects of situation assessment and belief formation, from cue identification and extraction, to situation classification, and finally decision-selection. A critical relevant area of study is the research in cognitive biases, resulting from the application of cognitive heuristics, many of which may be unconscious (Tversky & Kahneman, 1981). Specific biases identified include confirmation bias, primacy and recency effects, over-generalization, etc.

Situation assessment, situation awareness, cognitive biases, and perceptual cues represent key elements of the cognitive-perceptual processes that help users make sense of the world. Together, these processes ultimately define the users' personal belief systems, which then guide and direct perception, decision-making, and performance. As experiences accumulate with certain degrees of regularity, a set of residual beliefs and knowledge schemas are grad-

ually constructed which eventually result in the development of the users' *mental models.* Mental models serve as templates through which impinging sensory stimuli are organized into meaningful wholes and allow a rapid generation of expectations about the likely evolution of the world state. The use of mental models as a cognitive metaphor has been posited by a number of researchers. A common definition is that of Johnson-Laird, who defines mental models as "structural analogues of the world which may represent spatial relations between entities, causal-temporal relations among events, and contain an imaginary world model used to compute projective relations for an image" (Johnson-Laird, 1983, p. 410). He also suggests that mental models vary in form and content according to their purpose which is typically to predict, explain, or control. Mental models represent a set of mental shortcuts, which meld beliefs and schemas together for use in real-world situations.

One can thus see that beliefs are very much cognate with situation assessments (e.g., Young & McNeese, 1995) and demonstrate how "pure" cognitive processing, as advocated in human information-processing models (e.g., Lindsay & Norman, 1977), is shaped by context. In turn, contemporary approaches, for example, situated cognition (Resnick et al., 1991), distributed cognition (Woods et al. 1994), sensemaking (Weick, 1979), naturalistic decision making (Zsambok & Klein, 1997), and cognition in the wild (Hutchins, 1995) share common ground with CSE in that they suggest the mutually dependent roles among cognition, beliefs, and contextual variation. However, most all of these perspectives, including CSE, while lending credence to the value of belief formation (in some fashion), typically fail to address the role of emotions and affect.

Implications for ABAIS. The ABAIS belief assessment component primarily corresponds to the situation assessment discussed above, in that the currently active beliefs and knowledge schemas influence all stages of the situation assessment process. The assessment of a user's belief state thus amounts to the identification of his/her active mental model (sets of beliefs and knowledge schemas) that guide situation assessment. The situation assessment literature helps identify both the distinct stages of situation assessment, and the role that specific knowledge plays in this process (Klein, 1997; Lipshitz & Ben Shaul, 1997). The cognitive bias literature then helps identify the set of specific performance errors that can result from cognitive biases. Both sources provide a systematic basis for identifying possible belief states, for analyzing individual history information and applying it to dynamic belief assessment, and for identifying the relationships between specific affective states and cognitive biases.

9.3 RELATED WORK IN AFFECT AND BELIEF ASSESSMENT AND ADAPTATION

Relevant related research is conducted by the *user modeling* community. The term "user modeling," however, is used in a different sense in this community, from that typically used by the CSE researchers. Within the CSE community, this term typically implies a detailed characterization of the user's mental models in terms of some representational structure (e.g., concept maps [Zaff, McNeese, & Snyder, 1993], cognitive task-analytic structures such as goal and task hierarchies, causal models, etc.). Such models are typically nonexecutable, although in some cases may be part of an executable human performance model. In contrast, the user modeling community focuses less on the internal mental representations and more on static descriptions of behavioral and perceptual preferences (e.g., user preferences for information retrieval, alternative strategies during tutoring sessions, etc.). Nevertheless, there are important areas of overlap between the CSE and user modeling community, namely, the issues of user assessment and adaptation.

Research in this area has evolved rapidly over the past ten years and the number of systems, methods, and applications that attempt to solve these problems is too large to allow an exhaustive discussion of each potentially relevant system. The traditional nonaffective application domains include information filtering, document retrieval, web navigation (Pohl & Nick, 1999), tutoring, and personal assistants (Maes, 1994; Mitchell et al., 1994) in a variety of domains. Of particular relevance for this chapter are early concepts and prototypes for adaptive intelligent interfaces for the fighter-pilot domain (e.g, Fraser et al., 1989; McNeese, 1986). The early work in the pilot associate program is an example that focused on multi-level adaptation (Rouse, 1988). These early programs provided a much-needed experimental context for today's successful Rotorcraft Pilot's Associate and more advanced envisioned worlds (see Taylor, Chapter 8, this volume). Examples of recent user assessment and adaptation efforts include a fighter-pilot adaptive interface that assesses the pilot's workload and implements content, format, and modality adaptations in the cockpit displays (Mulgund et al., 2001) and analogous work in the context of air-traffic management (Harper et al., 2000).

The user states assessed typically consist of a variety of cognitive aspects of user knowledge, preferences, and performance (e.g., capabilities and limitations, interaction style preferences, goals and information needs, domain models, specific knowledge "bugs," generic problem-solving knowledge, etc.) (Dietrich et al., 1993). The methods for obtaining the user knowledge include implicit inferencing (e.g., probabilistic information filtering and classification approaches, knowledge-based inferencing and statistical approaches to derive user model information, machine learning techniques to automatically construct a user model from identified patterns in collected behavioral data

[McNeese, 1989; Pohl & Nick, 1999]), and explicit queries directly to the user, with varying degrees of user involvement in model construction and maintenance (Fleming & Cohen, 1999). The adaptations include modifying information retrieval criteria, changing tutoring strategies or material, and modifying user interfaces (e.g., Fijakiewicz & DeJong, 1998; McNeese & Katsuyama, 1987; McNeese, Rentsch, & Perusich, 2000).

Although the focus on affective assessment and adaptation in the broad area of user modeling is relatively recent, much interesting and relevant work exists. Systems are being developed and evaluated in applied settings, and basic research is being conducted on specific methods.

Below we provide brief descriptions of several representative systems and approaches, both those focusing on developing *tools* in particular application settings (e.g., tutoring [Elliot et al., 1997] and call monitoring [Petrushin, 2000]), and those focusing on exploratory development and assessment of particular *methods* (e.g., facial expression analysis [Kaiser et al., 1998], and physiological signal analysis [Healy & Picard, 2000]).

Elliot and colleagues have developed a pedagogical agent, based on the Affective Reasoner system (Elliot, 1992), which attempts to enhance its effectiveness through the assessment of the user's affective state (Elliot, Lester, & Rickel, 1997). The agent uses a type of cognitive appraisal process (Lazarus, 1991) to infer the affective state, implementing an enhancement of a computational cognitive appraisal model outlined by Ortony, Clore, and Collins (1988), and considers a variety of user characteristics (e.g., goals, principles that guide behavior, preferences, etc.). The assessment focus is on emotions such as hope and fear, and on the user's goals and goal characteristics (e.g., most critical goals, expectations regarding the achievement of this goal), in an attempt to capture the user's motivation. The data used include self-reports and behavior observations, and techniques include various AI-inferencing approaches (e.g., case-based abductive reasoning based on particular behavioral manifestations, recent individual history of the user, such as the "just failed on task" or "just succeeded on task"). More recent efforts attempt to infer the user's affective state based on the agent's affective state, thus implementing a form of software empathy.

In an effort to enhance and enrich human-computer interaction in general, Breese and Ball (1998) have developed an affective adaptive architecture that assesses the user's affective state in terms of the fundamental dimensions of valence and arousal, and personality in terms of the dimensions of dominance and friendliness. The system uses dynamic Bayesian belief networks to combine a variety of observable data (e.g., speech speed, facial expressions, word choice). (This work focuses more on the inferencing processes required to derive a particular state than on the data extraction itself.) Once a state and personality trait are identified, the interaction agent generates a response whose affective and personality tone match those identified in the user.

With a similar objective in mind, Klein developed and evaluated an experimental system that assesses a user's level of frustration using behavioral observations and self-reports and that implements an adaptation strategy that emulates human activities aimed at reducing negative affect (e.g., active listening, providing opportunity to vent, etc.) (Klein, 1999; Klein et al., 1999).

A number of recent efforts have attempted to assess user affective state via physiological sensing and a variety of prototype systems have been developed, primarily at the MIT Media Laboratory (http://www-white.media.mit.edu/vismod/demos/affect/AC_research/recognizing.html). These approaches assume distinct sets of physiological correlates characterizing particular states, and use a variety of sensors and wearable computer devices for data collection, coupled with complex pattern-recognition algorithms to identify unique patterns characterizing a particular state. Examples of these projects include Healy's work in assessing the user's level of stress (Healy & Picard, 2000), which uses four physiological signals (EKG, EMG, respiration sensor, and GSR), and the BlueEyes project at IBM (http://www.almaden.ibm.com/cs/blueeyes/) developing touch-sensitive input devices (e.g., Emotion mouse [http://www.almaden.ibm.com/cs/blueeyes/mouse.html]], by sensing and analyzing the user's pulse, temperature, and galvanic skin response to determine user's anxiety, stress and happiness, and using this information to determine success of computer-generated behaviors in a variety of intelligent "appliance" systems (e.g., intelligent TV-channel selector, etc.).

The BlueEyes project also includes methods using visual data based on observable behaviors, for example, facial expression (focusing on eyebrows and corners of mouth), gaze tracking, and gesture observation to assess the user's cognitive, affective, and physical states relevant to human-machine interaction (e.g., anxiety, happiness, dissatisfaction, etc.).

Extensive work in affective state assessment has been done by the Geneva Emotion Group (http://www.unige.ch/fapse/emotion/), as part of an extensive emotion research program. Particularly relevant to affect assessment is the extensive in-depth work of Kaiser, Wehrle, and colleagues, focusing on facial expression analysis (Kaiser, Wehrle, & Schmidt, 1998).

Speech is another effective source of data for affect assessment, using a variety of speech attributes (e.g., pitch, vocal energy, rate, and pauses, etc.) to differentiate among five affective states (neutral, happy, sad, angry, and fearful), and using several mathematical modeling approaches to derive the final affective state. These include K-nearest neighbors and a variety of back-propagation neural network approaches. These methods have been applied to analyze telephone conversations in support call centers and both prioritize messages and assign an appropriate human to handle particular calls (Petrushin, 2000).

In terms of pilot-vehicle adaptive interfaces, several systems attempt to assess the pilot stress level, more or less directly. The work of Mulgund and col-

leagues mentioned above (Mulgund et al., 2001) uses workload, which can be used as an indirect measure of stress, and performs cockpit display adaptations in response to high workload values.

9.4 ABAIS ADAPTIVE METHODOLOGY AND SYSTEM ARCHITECTURE

This section provides an overview of the ABAIS architecture and its constituent modules. Detailed descriptions of the key components—assessment and adaptation–are provided in Sections 9.7 and 9.8, after the necessary domain background information is provided in Sections 9.5 and 9.6.

9.4.1 ABAIS Adaptive Methodology

The ABAIS prototype implements a four-step adaptive methodology consisting of (1) sensing/inferring the individual's affective state and performance-relevant beliefs (e.g., high level of anxiety; aircraft is under attack), (2) identifying their potential impact on performance (e.g., focus on threatening stimuli, biasing perception towards identification of ambiguous stimuli as threats), (3) selecting a compensatory strategy (e.g., redirecting focus to other salient cues, presentation of additional information to reduce ambiguity), and (4) implementing this strategy in terms of specific UI adaptations (e.g., highlighting relevant cues or displays), that is, presenting additional information, or presenting existing information in a format that facilitates recognition and assimilation, thereby enhancing situation awareness (Endsley, 1995).

9.4.2 ABAIS System Architecture

The ABAIS system architecture (see Figure 9.1) implements the adaptive methodology described above and consists of four modules, each module implementing the corresponding step of the adaptive methodology:

- *User State Assessment*, which identifies the user's affective state and task-relevant beliefs
- *Impact Prediction*, which identifies the effect of user state on performance
- *Strategy Selection*, which selects a compensatory strategy, and
- *GUI /DSS Adaptation*, which modifies the user interface content and format to enhance detection, recognition, and assimilation of incoming data, that is, to enhance situation awareness.

Each of the modules is briefly described below, focusing on its input-output behavior. A more detailed description of the assessment and impact prediction modules is provided in Section 9.7, and of the strategy selection and adaptation modules in Section 9.8.

The *User State Assessment Module* receives a variety of data about the user and the task context, and from these data identifies the user's predominant affective state (e.g., high level of anxiety) and situation-relevant beliefs (e.g., interpretation of ambiguous radar return as threat). This key component of the ABAIS system is discussed in detail in Section 9.7.

The *Impact Prediction Module* receives as input the identified affective states and associated task-relevant beliefs and determines their most likely influence on task performance. The goal of the impact prediction module is to predict the influence of a particular affective state (e.g., high anxiety) or belief state (e.g., "aircraft under attack," "hostile aircraft approaching," etc.) on task performance. Impact prediction process uses Rule-Based Reasoning (RBR) and takes place in two stages. First, the *generic effects* of the identified affective state are identified, using a knowledge-base that encodes empirical evidence about the influence of specific affective states on cognition and performance. Next, these generic effects are instantiated in the context of the current task to identify *task-specific effects,* in terms of relevant domain entities and procedures (e.g., task prioritization, threat assessment). The knowledge encoded in these rules is derived from a detailed affective/cognitive task analysis, which predicts the effects of different affective states on performance in the current task context. The separation of the generic and specific knowledge enhances modularity and simplifies knowledge-based adjustments.

The *Strategy Selection Module* receives as input the predicted specific effects of the affective and belief states, and selects a compensatory strategy to counteract resulting performance biases. Strategy selection is accomplished by rule-based reasoning, where the rules map specific performance biases identified by the impact prediction module (e.g., task neglect, threat-estimation bias, failure-estimation bias, etc.) onto the associated compensatory strategies (e.g., present reminders of neglected tasks, present broader evidence to counteract threat-estimation bias, present contrary evidence to counteract failure-driven confirmation bias, etc.). As was the case with impact prediction, the strategy selection module relies on a detailed analysis of the task context that identifies specific strategies available to counteract the possible biases. This analysis then allows the construction of the strategy selection knowledge bases.

The *GUI Adaptation Module* performs the final step of the adaptive methodology, by implementing the selected compensatory strategy in terms of specific GUI modifications. A rule-based approach is used to encode the knowledge required to map the specific compensatory strategies onto the necessary GUI/DSS (decision support system) adaptations. The specific GUI modifications take into consideration information about the individual pilot

preferences for information presentation, encoded in customized user preference profiles; for example, highlighting preferences might include blinking vs. color change vs. size change of the relevant display or icon.

9.4.2.1 Ancillary Modules. Several additional modules exist in the ABAIS prototype, enabling the simulation of the demonstration task, output of the simulation results on a simulated cockpit GUI (both adapted and nonadapted versions), and a series of displays and windows supporting the analyst-system interaction (e.g., entering user and task data, and monitoring system performance). These are briefly described below.

ABAIS Simulation Module. The core ABAIS framework is integrated within a dynamic flight simulation environment and supports two modes of system operation: (1) pilot-as-user mode, where the user actually flies the aircraft and interacts with a simulated environment consisting of other friendly aircraft, enemy aircraft, radars, and weapons; and (2) analyst-as-user, where the analyst watches a simulation of a scripted task and monitors the (scripted) pilot's performance, and the system run-time performance (i.e., results of the rule-based inferencing). Both modes include a pilot GUI consisting of key cockpit displays (see Figure 9.5).

Pilot's GUI. The pilot's GUI (see Figure 9.3) consists of four displays, corresponding to the Heads-Up-Display (HUD), which combines a variety of navigation and sensor information within a single display (i.e., heading, airspeed, altitude, MACH speed, etc.) (upper portion of the display), a window showing current incoming communication as text strings (middle window), an alert notification window (middle window), and the radar and sensor display (bottom portion of display), which combines information from a variety of aircraft instruments (e.g., active radar, IFF, NCTR, and RWR), as well as datalink from other friendly aircraft. The radar and sensor display symbology follows existing cockpit standards.

Analyst's GUI. The analyst's GUI serves three functions:

- It allows specification of all ABAIS system run-time parameters, including task script editing and selection, adaptation thresholds, and execution monitoring windows.
- It allows specification of all necessary background pilot information.
- It allows display and monitoring of ABAIS simulation and run-time data.

Prior to a run, the analyst specifies the necessary background information about the pilot. In the initial ABAIS prototype demonstration, these values were entered by the analyst. In a full-scope system, some of these parameters would be entered by the user (pilot) (e.g., self-reports and individual history information), gathered during training tasks (e.g., baseline physiological or

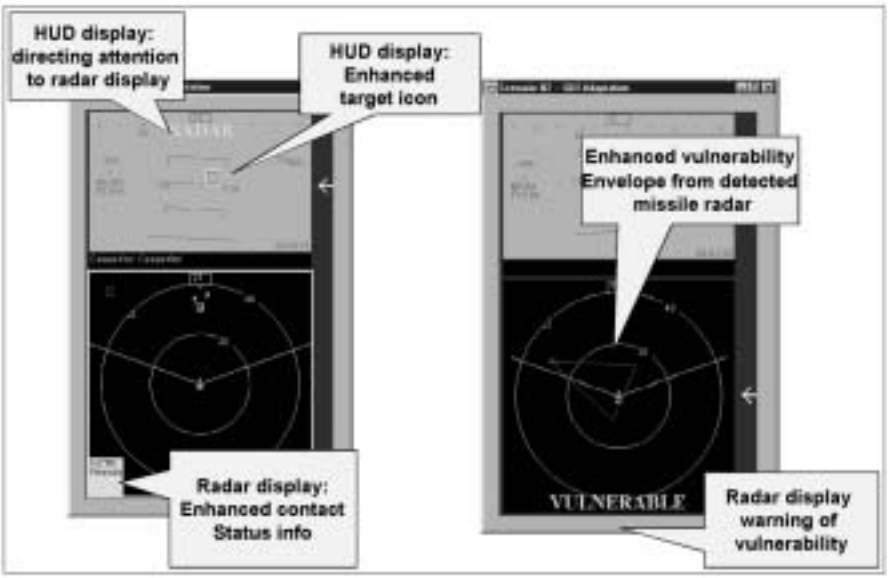


Figure 9.3: Summary of GUI/DSS adaptation strategies.

diagnostic task data), or collected automatically during an actual system run (e.g., actual physiological signals or diagnostic task results). Different categories of information are specified, including personality, skill, individual history, and adaptation preferences.

9.5 ABAIS DEMONSTRATION TASK: FIGHTER PILOT "SWEEP" TASK

Below we describe the essential details of the ABAIS demonstration task to provide the necessary background for understanding the details of the ABAIS rulebases and inferencing described in Sections 9.7 and 9.8. While the initial prototype was implemented in the context of a fighter pilot task, the overall ABAIS methodolog, and implemented framework provide a generic approach to affective adaptation and modeling (see discussion in Section 9.10.3).

9.5.1 Task Context

The demonstration task simulates a U.S. Air Force fighter pilot sweep mission, where a group of friendly pilots attempt to clear the airspace of any enemy aircraft. The aircraft is assumed to be an F–15-like fighter aircraft cockpit. During the specific mission, two friendly aircraft ("lead" and "wingman") are conducting a sweep mission over enemy territor. They are assisted by an

AWACS aircraft providing additional information, and command and control. Several unknown, presumably hostile aircraft, are approaching the friendly aircraft and the corresponding data (radar returns) are beginning to appear on the aircraft cockpit radar displays.

9.5.2 Human Context

The demonstration task focus is on the lead pilot as the "user." The lead pilot communicates with the wingman and with the AWACS aircraft operator, both via voice and electronic datalinks, which display data directly on the lead's cockpit instruments (e.g., radar display, HUD).

The lead pilot's background information (user profile) is entered by the analyst, prior to a run, by entering the user-specific data (e.g., personality traits, skill, individual history, physiological responsiveness, adaptation preferences).

9.5.3 Summary of Demonstration Task Events

Two pilots (lead & wingman) are conducting a sweep mission over enemy territory (refer to Figure 9.2). Pilots are expecting strong enemy opposition. Friendly aircraft are likely to be in the area, making fratricide a possibility. A high-anxious lead pilot misinterprets incoming unknown contact as hostile and prepares to fire by "centering the dot" on his HUD display (upper portion of the cockpit GUI display). Table 9.2 summarizes the possible effects of affective and belief states on the lead pilot's performance. At the last minute, the wingman radios that a friendly identification has been obtained on this contact. In other words, the approaching aircraft is in fact a friendly coalition aircraft. However, the lead pilot is busy targeting this aircraft and misses this transmission. Without adaptation, the lead would fire a missile and hit a friendly aircraft. With adaptation, the incoming data (friendly ID) are enhanced and the pilot's attention is directed to this information in an attempt to prevent the fratricide.

9.6 COGNITIVE, AFFECTIVE, PERSONALITY TASK ANALYSIS (CAPTA)

Given the fact that a general model of human information processing is beyond the state of the art, any user modeling effort must focus itself on particular aspects of performance during a specific task of interest or an underlying mechanism of interest. A systematic cognitive task analysis process is the foundation for the development of any such cognitive model. Since the focus

Table 9.2: Summary of Possible Affective and Belief State Influences on Pilot Behavior

Anxiety-Induced Narrowing of Attention
Lead focusing on "centering the dot"
Misses "friendly ID" voice communication from wingman
Misses results of NCTR (friendly ID) on radar display
Anxiety-Induced Perceptual Bias
Lead misinterprets ambiguous radar returns as threats
Obsessiveness-Induced Checking Behavior
Lead engages in obsessive checking behavior, delaying firing at enemy

of this effort is the influence of affect and belief states on performance, the cognitive task analysis focused not only on the space of nominal behaviors and cognitive performance, but also the larger space of possible behavior variations due to the effects of these additional factors. We term this enhanced cognitive task analysis process *Cognitive Affective Personality Task Analysis*, or CAPTA for short. CAPTA formed the basis for developing the ABAIS system knowledge bases and adaptation strategies. In this section we briefly outline the CAPTA process and distinguish it from existing cognitive task analysis techniques (9.6.1), and indicate how it supports the ABAIS adaptive methodology and relates to the ABAIS architecture (9.6.2). A full description can be found in Hudlicka (2000a).

9.6.1 Description of the CAPTA Process

The objective of Cognitive Task Analysis (CTA) is to define the user's activities during a task at a sufficient level of detail to allow computational modeling, knowledge-base definition, user-interface requirements specification, or work analysis to support the variety of tasks that comprise the definition, implementation, operation, and applications of user models.

A number of CTA techniques exist (Cooke, 1995; Essens, McCann, Cannon-Bowers, & Dorfel, 1995) using a variety of methods and representational formalisms (e.g., rules, concept maps, frames, and schemas, etc.). Their unifying objective is to define critical domain and user entities, their states, and their behaviors, including the user's actions.

CAPTA differs from the traditional CTA techniques by explicitly focusing on behavior variations due to the user's affective and belief states. This supports affect and belief adaptation by (1) making explicit the number and type of affective and belief states, and user behaviors associated with each; and (2)

producing a more complete and comprehensive description of the user's possible states and behaviors. Specifically, CAPTA addresses the following:

- What is the possible set of user's affective and belief states?
- What are the likely triggers of particular affective and belief states?
- What are the likely cognitive/perceptual schemata (beliefs), and goals, expectations, and behaviors associated with each affective state?
- What are the most likely goals, expectations, and behaviors associated with each belief state?

For the purposes of the initial ABAIS architecture prototype development, we focused the CAPTA process on the identification of affective and belief states, their triggers, and likely behaviors associated with each. Table 9.3 illustrates how these categories of information relate to the ABAIS architecture.

The CAPTA application in the current context assumes a naturalistic model of decision making (Klein, 1997) where the primary skilled processing takes place at the perceptual-situation assessment stage. Once the incoming cues are assembled into meaningful schemata and categorized into situations, there is a simpler mapping process between the situation and the selected action. This theoretical model underlies the task analysis approach implemented by CAPTA. The complete CAPTA process consists of the following steps:

- Constrain the possible user behaviors (i.e., decisions); goals, expectations, and situations (i.e., high-level cognitive and perceptual schemata); and stimuli (i.e., incoming sensory cues), within the context of the scenario and task domain. Focus on both task and self-related goals, expectations, situations, and behaviors
- Define the mappings between the cues and the situations, and between the situations and the decisions
- Identify the user's dominant personality traits and their most likely effects on affective state triggering, belief formation, and behavior
- Define the user's set of possible affective-states and the most likely transitions among these states, including specific triggers
- Identify most likely sets of beliefs, goals, expectations, and behaviors associated with each affective state
- Determine how goals, expectations, and affective states influence situation assessment and decision selection.

The last four steps, along with the focus on self-related processing (i.e., goals, expectations, situations, etc.), distinguish CAPTA from more traditional CTA techniques. During these steps, the cognitive systems engineer works closely with both the users and the subject-matter experts (SME), and draws on the available empirical evidence about the effects of particular affect and

Table 9.3: Relationship of CAPTA to ABAIS Architecture Components

CAPTA Information Category	Relevance for ABAIS Architecture	Pilot Domain Example
• Possible set of user's affective states • Possible set of user's belief states	• Basis for defining specific affective/belief state assessment strategies • Basis for predicting behavioral effects of specific states • Basis for identifying compensatory strategies associated with each state	• Pilot A can be calm or anxious • Pilot B can be calm or aggressive • Pilot can believe s/he is out of danger, about to be attacked, under attack • If a particular affective state (e.g., anxiety) reduces attentional capacity, modify display to reduce attentional demands
• Likely triggers of particular affective and belief states	• Defines knowledge-base within "User Assessment" module, supporting the assessment of user's affective and belief state	• Pilot becomes anxious when large number of unknown radar contacts appear on the screen Pilot assumes any unknown radar contacts are hostile
• User's likely response to a particular situation • User's likely behavior as a function of affective or belief state	• Defines knowledge-base within "Impact Prediction" module, supporting the prediction of specific effects of user affective and belief states • Helps define knowledge-base within "GUI Adaptation" module and specific GUI adaptation strategies	• Pilot tends to delay weapon engagement • Pilot engages in excessive information gathering • Pilot detects alarms more rapidly when presented in auditory mode

belief states on perception, goal and expectation formation, decision selection, and behavior. Thus, the generic effects of the possible affective states identified from the empirical literature are instantiated within the specific task context. By combining the available empirical evidence with the practical knowledge of the domain expert, the cognitive systems engineer constructs a space of possible behaviors which takes into account not just the nominal path through the problem-solving space, but also the variations resulting from different affective and belief states. For example, the empirical knowledge that anxiety biases attention and perception towards the detection of threatening stimuli, and the interpretation of ambiguous stimuli and situations as threatening, is combined with the expert's knowledge of the task at hand (e.g., air combat), to derive the alternative possible behaviors due to a heightened state of anxiety (e.g., a pilot may interpret ambiguous radar returns as threats and fire prematurely at friendly or neutral aircraft which happens to be in the area).

9.6.2 Use of CAPTA to Support the ABAIS Architecture Development

CAPTA thus produces a comprehensive description of the possible behaviors and behavior variations associated with particular user affective states, personality traits, and beliefs, thereby generating a more complete specification of the

user-task problem space. Such problem space definition then becomes the basis for defining critical knowledge-bases of the ABAIS architecture modules (refer to Table 9.5). For the purposes of ABAIS prototype development, only a subset of the complete CAPTA process was used: definition of user behaviors, task situations, and incoming cues; definition of situation-action and cues-situation mappings; influences of affect and belief states on situation assessment and behavior; and definition of user's affective state transition diagrams.

Below is a brief illustration of how these steps of CAPTA process supported the definition of the ABAIS architecture components, that is, the knowledge bases contained in the distinct modules (e.g., Impact Prediction).

9.6.2.1 Defining User Behaviors, Situations, and Incoming Cues. The CAPTA process must be grounded in a fixed set of possible actions, possible situations triggering those actions, and possible cues leading to the derivation of the situations. The first step in the CAPTA process is therefore to define the limiting conditions—that is, the possible outputs (behaviors), key intermediate states leading to these actions (perceived situations), and incoming data or stimuli leading to the perception of the situations (cues). While the end-points of this process are fixed by the situation context and user capabilities (available displays and sensory channels on the input side, and possible behavioral outputs on the output side), the definition of the situation set is an art rather than a science. This process relies on the knowledge and cooperation of the subject-matter expert, who must work closely with the cognitive systems engineer/model developer to select the most appropriate situations, at the correct level of abstraction, that provide coverage of the current task. Table 9.4 shows examples of cues, situations, and decisions in the current ABAIS fighter-pilot task context.

9.6.2.2 Defining Situation-Action and Cues-Situation Mappings. Once the sets of possible behaviors, situations, and cues are defined, the CAPTA defines the mappings between the cues and the situations, and the situations and the actions. In other words, based on task knowledge elicited from the SME, the model developer specifies which cues trigger which situations, and which situations in turn trigger which behaviors. This knowledge then forms the basis for defining the rules in the ABAIS Impact Prediction Module knowledge-base.

To identify the *situation-action mappings*, we began with the possible sets of behaviors and identified the situations that triggered each behavior. For example, a pilot will fire if he/she sees a hostile target within range and is cleared to fire. By working backward from the possible behaviors, we thus identified a series of triggering situations for each behavior. These situation-actions mappings were then expanded to cover the different user profiles, and catego-

Table 9.4: Examples of Pilot Cues, Situations, and Decisions

CUES (Incoming Stimuli)	Situations (Beliefs)	Decision (Behaviors)
Datalink—friendly | unknown | hostile	Hostile aircraft closing | closing	Fire weapon
IF—friendly | unknown	Presumed hostile aircraft opening | closing	Initiate intercept
NCTR—<aircraft type>	Unknown aircraft opening | closing	Initiate evasive action
RWR—no radar contact | hostile radar contact | friendly radar contact | SAM radar contact | unknown radar contact	Cleared to fire w/positive EID	Communicate w/wingman
Active radar—friendly | unknown | hostile	Firing range for <weapon> maximun | beyond maximum	Communicate w/wingman
Targets—closing | opening	Attacking hostile aircraft	Communicate w/AWACS
	Under attack from hostile aircraft	Focus on particular instrument {HUD |

Table 9.5: Example Rules Predicting Impact of Anxiety on Performance

Anxiety-Induced Narrowing of Attention
IF (multiple targets on radar) THEN (focused on signals representing unknowns or threats)
IF (multiple unknown/threats targets on radar) THEN (focused on nearest/fastest-approaching targets)
IF (data arriving on multiple cockpit instrument) THEN (attention focused on radar or HUD)
IF (data arriving on HUD and radar) THEN (attention focused on HUD)

Anxiety-Induced Misinterpretation of Cues
IF (unknown or ambiguous targets on radar | NCTR exist) THEN (assume targets are hostile)
IF (unknown air-to-air radar lock on RWR) THEN (assume under attack)
IF (no reply from IFF) THEN (assume targets are hostile)

rized into specific groups according to the affect and belief factors. Once the complete analysis was performed, the situation-action mappings were translated into the rule set in the ABAIS affect and belief impact prediction module.

To identify the *cues-situation mappings*, we began with the set of possible situations and identified the incoming cues that would indicate each situation (i.e., belief). For example, a hostile aircraft is closing if a radar-return exists representing a hostile aircraft and the radar-return indicates an approaching aircraft over time. Given the fact that CAPTA defines not only correct decision-making but a variety of possible distortions (e.g., misinterpretation of cues; incorrect weighting of individual cues, etc.), as well as a number of variations, this process involved the construction of extensive cues-situations mappings allowing for such distortions and variations. For example, while the presence of an unknown contact on the radar means just that and does not necessarily imply that the pilot is under attack, it is certainly a possibility that this contact may represent a hostile aircraft. In the case of an anxious or aggressive pilot, it may be a likely possibility. Thus the construction of these mappings involved the consideration of a large number of possible cues-situation mappings that might be used by various pilot types to conduct situation assessment

and arrive at the consequent beliefs. These cues-situations mappings are the primary source for deriving the belief assessment rules within the ABAIS User Assessment module.

9.6.2.3 Influences of Affect and Belief States. Once these basic sets of mappings were defined, they were expanded to account for the influences of the pilot's affective and belief states. This process combined existing empirical evidence, knowledge of individual user behaviors, and task-specific expertise. A systematic analysis of the possible pilot behavior and decision-making during the course of the demonstration scenario yielded a set of possible perceptual and cognitive distortions, biases, and general variations resulting from the affect and belief states. The paragraph below describes the application of this process to the analysis of the effects of anxiety on behavior and provides examples of specific rules that result from this process.

The *generic effects of anxiety* on attention include narrowing of attentional focus, difficulty focusing attention (i.e., inability to select an action and consequent delayed reaction time), and increased attention to threatening stimuli. This narrowing of attention may also result in task neglect for other critical tasks, and a failure to detect other relevant cues.

Given this generic knowledge, the CAPTA process is then used to predict *task specific* situations where these biases may influence performance, in other words, to identify situations where ambiguous cues exist which can be misinterpreted as threatening, and to identify task segments where parallel signals may occur (e.g., two signals on radar from two different sources, radar and engine instruments, etc.) and identify points where parallel tasks take place. These then allow predictions as to which of these tasks is likely to be neglected during a state of increased anxiety (e.g., pilot is more likely to pay attention to radar signals than engine instruments or radio). Examples of *specific effects* of the generic anxiety-induced biases include:

- Focusing on target or radar display and failing to notice incoming communication from other sources (e.g., radio voice communication from wingman, AWACS, etc.) or other cockpit instruments (e.g., warnings of aircraft system malfunctions)
- Focusing on target information on HUD and failing to notice new information on radar
- Interpreting ambiguous radar-returns as threats.

Examples of specific rules constructed from this knowledge and used in the Impact Prediction module are shown in Table 9.5.

9.6.2.4 Defining User's Set of Affective States and Transitions Among Them. A key aspect of the CAPTA process is the definition of the possible user affective states. This is based on the assumption that the user's affective states represent key determinants of behavior, via their influence on both the nature of perceptual, cognitive, and motor processing, and the choice of goals, formation of expectations, and ultimately formation of situation assessment and selection of specific decisions and associated behaviors (Isen, 1993; LeDoux, 1992; Williams et al., 1997).

To implement this step of the CAPTA process we have used a process we term cognitive-dynamic behavior analysis (Hudlicka, 2000a). This approach combines concepts from psychodynamic (depth psychology) and cognitive analysis of personality and behavior (Horowitz, 1991), and focuses on both intrapsychic and interpersonal patterns of thought, affect, and interaction (core role relationship models). The objective of the cognitive-dynamic behavior analysis is to identify an individual's predominant affective states (e.g., calm, anxious, withdrawn). The assumption underlying this approach to behavior analysis is that the primary determinants of behavior are the individual's cognitive schemas and that the predominant schemata controlling behavior is triggered in part by a particular affective state. We have expanded this concept to also include the individual's goals and expectation. This expanded concept then states that behavior is determined by:

- Currently active goals and expectations, which are activated based on the
- Previous and current situation assessments and current affective state, which trigger the activation of
- Specific cognitive schemata that guide perceptual and cognitive activities and decision-making.

Figure 9.4 illustrates an example of an affective state transition diagram resulting from cognitive-dynamic behavior analysis.

The process of affect state transition diagrams necessarily involves elements of affect and belief assessment. As such, it minimally consists of combining existing empirical evidence, knowledge of individual user behaviors, and task-specific expertise. A variety of issues arise here regarding the feasibility and validity of identifying these transition diagrams, both for specific individual users, and for user populations. Detailed discussion of these issues, as well as detailed procedures for the CAPTA process, can be found in Hudlicka (2000a).

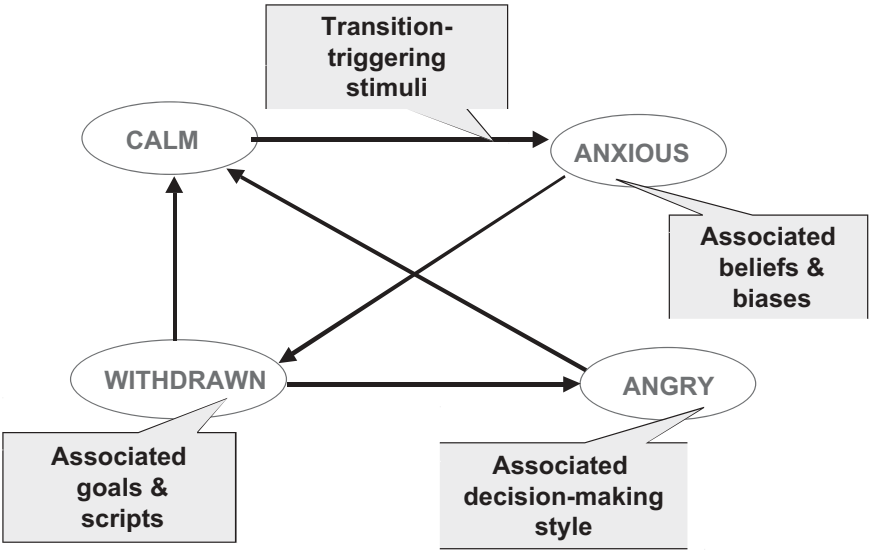


Figure 9.4: Example of an affect state transition diagram resulting from cognitive-dynamic behavior analysis.

9.7 USER STATE ASSESSMENT AND BEHAVIOR PREDICTION

Having provided the necessary background information about the ABAIS adaptive framework and architecture, and the task domain, we now turn to the key aspect of affective adaptation: the assessment of the user's affective and belief states (Section 9.7.1) and the prediction of their likely effects on performance (Section 9.7.2).

9.7.1 User Affect and Belief State Assessment

9.7.1.1 Affect Assessment. Since no single reliable method currently exists for affective assessment, the user assessment module of the ABAIS architecture provides facilities for the flexible combination of multiple methods. These include physiological assessment (e.g., heart rate); diagnostic tasks; self-reports; and use of knowledge-based methods to derive likely affective state based on factors from current task context, personality, and individual history. Each of these methods has its associated advantages and disadvantages, and none alone is currently sufficient for reliable affective state identification.

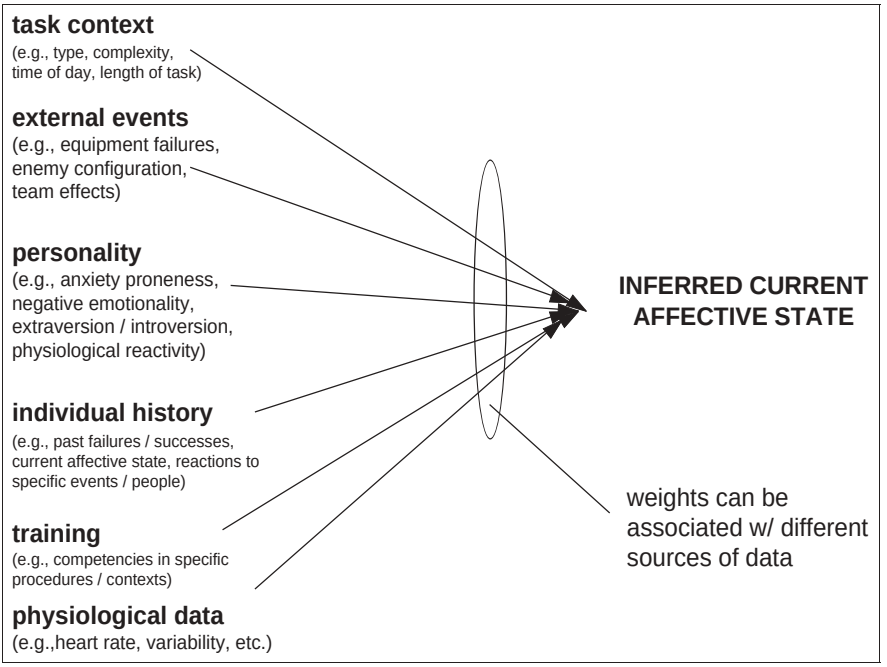


Figure 9.5: Sources of information for deriving pilot's affective state.

The initial prototype described here implements the knowledge-based approach, and assesses the user's anxiety level. The knowledge-based assessment essentially emulates the judgment of an expert observer, familiar with the key user and situation factors contributing to the user's affective state. This approach was selected for the prototype because it combines multiple sources of data (e.g., individual history, personality, task context, physiological signals), reflecting the multiple factors that influence the user's affective state (see Figure 9.5), and thus essentially provides a means of simulating the use of multiple methods. Anxiety was selected both because it is the most prevalent affect during crisis situations, and because its influence on cognition has been extensively studied and empirical data exist to support specific impact prediction and adaptation strategies.

The factors contributing to the pilot's affective state fall into two broad categories: *static* and *dynamic,* which are briefly described below. The assessment process implements a fuzzy rule-based approach which combines the influences of both static and dynamic factors to arrive at the final user affective state assessment.

Static Factors. The static factors represent influences that remain constant throughout the task (e.g., overall task difficulty, user's training and proficiency, and user's personality traits and individual history). The values of these factors are specified prior to a particular simulation via a combination of knowledge elicitation interviews with experts and users (e.g., task difficulty, level of training, personality) and off-line user assessment instruments (e.g., personality, skill level), and extensive interviews with users (e.g., individual history, personality traits). Examples of the task context and individual history static factors, and their possible values in the context of the demonstration task, are shown in Table 9.6.

Specific events from the user's history that may influence the current affective state are considered to be *individual history* factors. Given the importance of an individualized approach to affective and belief state assessment, these factors are among the most critical, particularly so since personality research indicates that the most reliable predictor of future behavior is past behavior. The individual history factors must obviously be tailored to the specific task context. Accordingly, the factors used in the initial ABAIS prototype represent the user pilot's previous experience in combat tasks.

Another critical set of static factors is the set of user's key *personality traits*, which represents a relatively stable set of characteristics that contributes to the user's affective state (e.g., emotional stability correlates with anxiety tolerance) and may influence behavior in general (e.g., obsessiveness, aggressiveness). The selection of the specific personality factors below was guided by the following criteria:

- Empirical evidence for existence of candidate factor as a distinct personality characteristic
- Empirical evidence or knowledge elicitation data indicating specific effects of the personality factor on performance, particularly in the context of the specific task (e.g., aviation and air combat tasks)
- Likelihood of the personality factor playing a role in the selected demonstration task.

Specific personality factor selection is thus informed both by domain- independent personality theory and empirical data, and, to the extent possible, by studies performed in the specific domain of interest. In the case of the ABAIS prototype, data from studies assess the pilot population in terms of a variety of standard psychological assessment instruments (e.g., NEO–PI–R, MMPI, etc. [Callister et al., 1997]), as well as instruments specialized for the fighter pilot population (e.g., ALAPS [Retzlaff et al., 1997]). Unfortunately, little systematic empirical work has been done in the general area of linking personality factors to specific performance influences and biases, at a level of analysis that would provide the type of detail necessary for real-time adaptation. We there-

Table 9.6: Examples of Static Factors Used During User State Assessment

Task Context	Individual History	Personality Traits
Task type 1 :==\{offensive | defensive\}	Successful situations in past :==\{ egress | enemy-radar-lock—hit |	Emotional stability :==\{1–10\} Impulsiveness :==\{1–10\}
Task type 2 :==\{air-to-air | air-to-grnd\}	enemy-radar-lock—evasion | unknown-radar-lock—hit...\}	Risk tolerance :==\{1–10\}
Phase :==\{planning | detection | ...\}	Failure situations in past :==\{fratricide | repeat IFF interrogations—fratricide...\}	Aggressiveness :==\{1–10\}
Complexity :==\{1...10\}	Affective reactions to individual/events :==\{	Conscientiousness :==\{1–10\}

fore also relied on general personality research (e.g., "Big 5," "Giant 3"), and on knowledge elicited from domain experts (e.g., pilots or USAF psychologists and scientists). The objective was to capture personality traits that (1) are likely to exist in the subject population (e.g., fighter pilots), and (2) exert a pronounced influence on behavior during the performance of the demonstration task. Emotional stability was the primary factor of interest under the prototype development effort, since this factor correlates with anxiety tolerance.

Dynamic Factors. In contrast to the static, the values of the dynamic factors change during the course of the task, reflecting changes in both the external environment (e.g., incoming data from sensors such as radar contacts) as well as the changing pilot state (e.g., changes in physiological signals such as heart rate). These values are provided to the affective assessment rules throughout the course of the simulation, allowing dynamic computation of their contributions to the pilot's affective state.

An important set of dynamic factors are *external events*, which contribute to the task difficulty and, as such, influence the user's affective state. These include a variety of factors relating to the state of the environment, the equipment relevant to the performance of the task, and dynamic task characteristics (e.g., state of the aircraft, equipment failures, task-specific factors such as the geometry of the intercept, any data appearing on the radar systems, team effects).

Another set of dynamic factors are *physiological data* representing specific physiological measures collected from the user during the course of the task. Due to the high degree of individual variations in physiological signals, as well as within-individual variations over time and habituation, these measures must be normalized based on the user's baseline responsiveness measures, and baseline measures for the task and the current day. In the initial ABAIS prototype we focused on assessment of the user's anxiety level and thus considered physiological data that reflect anxiety. While a variety of measures are theoretically possible, (see discussion on physiological sensing in Section 2.2.2), the most reliable measures of state anxiety appear to be those related to arousal, that is,

heart rate and skin conductance measures.[3] Although these measures reflect general arousal, rather than anxiety per se, it is assumed that during crisis situations in general, and during the demonstration scenario sweep task in particular, such arousal would be an indication of anxiety. During the initial prototype we therefore focused on heart rate as the most reliable and practical measure of arousal, using estimates derived from existing empirical literature and interviews with fighter pilots.

9.7.1.2 Belief Assessment. For the discussion below, we assume a working definition of "belief state" as representing the currently active or preferred set of knowledge constructs, schemata, or procedures guiding perception, influencing decision making, and determining the final behavioral outcome. In other words, the current belief state represents the currently active situation schemata and thus reflects the pilot's *situation assessment* and *situation awareness*. Which specific schemata are instantiated at any given time is a function of a number of factors, including user's training, individual history, personality and cognitive style differences, and affective state. Belief assessment in this context thus corresponds to what is generally referred to as situation assessment in the literature, that is, the identification of the most likely current interpretive schemata guiding situation interpretation, decision making, and subsequent action selection.

For example, in the aviation domain, a combination of pilot's training, recent events, and affective state might predispose him towards a particular interpretation of existing ambiguous data (e.g., unknown radar is hostile, approaching unknown aircraft are friendly, etc.), a pilot's training might predispose him/her toward a particular cockpit instrument-scanning pattern, and individual experience might predispose him to a specific set of expectations regarding the outcome of a particular engagement.

Given this definition of beliefs, the following problems must be addressed to identify a belief state and its potential effects on performance. *First*, the possible set of beliefs relevant for a particular task context must be identified; in other words, the situation taxonomy for the task domain must be defined. *Second*, the factors contributing to the instantiation of a particular set of beliefs during situation assessment must be identified; these can then be used to dynamically assess the pilot's belief state. *Finally,* a dynamic assessment must be performed during the task execution to determine the individual's most likely set of active schemata, that is, the dominant belief state and corresponding situation assessment. These three problems, and the corresponding solutions implemented in the ABAIS prototype, are discussed below.

Identifying the Task Situation Taxonomy. The first problem requires a detailed ontological analysis of the task domain, identifying critical cues, a taxonomy of

3. GSR may not be a practical measure in the fighter pilot setting, due to intrusive monitoring devices (i.e., finger sensors) and because the fighter pilot task environment may interfere with the data collection (i.e., finger sensors may obstruct other activities).

possible situations, and space of possible actions. This problem was addressed through the CAPTA process described in Section 9.6 above and resulted in the set of cues, situations (beliefs), and possible actions described in Table 9.6.

Factors Contributing to a Particular Belief State. A variety of factors determine the final belief state, each contributing some piece of knowledge or evidence to establish, confirm, or refute a particular belief about the current situation (e.g., unknown target is friendly or hostile). The sum total of these influences then determines the pilot's overall assessment of the current situation. There is significant overlap in the knowledge and rules used to assess the affective state and those used to infer the belief state. A critical aspect of the belief state assessment is the inclusion of the pilot's presumed affective state. This in effect allows the implicit modeling of the influence of specific affective states on cognition and distinguishes the current approach to belief/situation assessment from existing situation assessment methods (e.g., SD_PVI of Zacharias et al., 1996; Pew & Mavor, 1998).

Dynamic On-Line Belief Assessment. The process of dynamic belief assessment is the final step of user belief state assessment. During this phase the current knowledge factors contributing to the activation of particular beliefs are instantiated to derive the most likely set of activated schemata, that is the user's beliefs reflecting the current situation. This process essentially simulates, at the input-output level, the pilot's own situation assessment processes. In the context of the current demonstration scenario, the pilot's belief state is reflected in the pilot's assessment of the current situation from the available salient cues. While in theory an infinite number of situations are possible, in practice the set of situations for a particular task is generally limited (e.g., attacking versus being attacked, approaching aircraft are friendly or hostile, etc.). In fact, one of the conditions constraining the application of effective user modeling and adaptation is precisely the possibility of constraining the number of possible situations.

In the context of the ABAIS demonstration scenario we therefore limit the possible situations to those identified through initial knowledge elicitation and cognitive task analysis (see Table 9.6).

For the initial ABAIS prototype, we selected a knowledge-based approach for the dynamic belief assessment. As with affect assessment described above, this approach in effect emulates an expert observer, familiar with both the task and the specific individual. As was the case with affective state assessment discussed above, the knowledge used to dynamically derive the pilot's belief state was encoded in terms of production rules. Again, as with the affective state assessment, it is important to keep in mind that the factors, their values, and the corresponding rules, are specified in the context of the current individual-

task and can be changed as that context changes. In fact, such individualized tailoring of the ABAIS knowledge-bases is the key to its successful adaptation for a particular individual, or group of individuals in a team setting. The actual belief state was then derived from a combination of static, a priori information about the pilot and the task, and from dynamic data reflecting the changing task environment and pilot state, including the pilot's affective state. The most critical categories of factors used were (1) external events (see discussion above), (2) individual history, and (3) current affective state.

External events are described above. Here we briefly outline issues associated with the use of individual history and affective state for belief assessment.

Individual history combines the training and skill factors with specific experiences that influence the pilot's situation assessment and decision-making. In other words, specific successful or unsuccessful experiences tend to predispose the pilot towards or against certain situations and maneuvers. For example, in the current demonstration scenario, occurrence of specific recent situations may bias the interpretation of current data; in other words, if the pilot has recently experienced a situation where a number of unsuccessful IFF interrogations were followed by a final identification of that aircraft as hostile, he/she may be predisposed to conclude that if an aircraft does not respond to IFF interrogations it is in fact hostile.

The pilot's affective state plays a critical role in his/her situation assessment. By taking into account the current affective state, the ABAIS user assessment module in effect implicitly models the potential biasing influences of the different affective states and provides a structure which allows the explicit representation of the positive feedback between cognition and affect that is often seen in crisis situations. In other words, increased anxiety contributes to a particular situation assessment (e.g., aircraft is being attacked by hostile aircraft), which then limits the processing of data that could give rise to alternative interpretations and further increase the anxiety level.

9.7.2 Affect and Belief State Impact Prediction

The goal of the impact prediction module is to predict the influence of a particular affective state (e.g., high anxiety) or belief state (e.g., "aircraft under attack," "hostile aircraft approaching," etc.) on task performance. Impact prediction thus represents an essential component of the overall adaptation strategy. The impact prediction process implemented in the ABAIS prototype uses rule-based reasoning and takes place in two stages. First, the *generic effects* of the identified affective state are identified, using a knowledge-base that encodes empirical evidence about the influence of specific affective states on cognition and performance. Next, these generic effects are instantiated in the context of the current task to identify *task-specific effects*, in terms of relevant domain

entities and procedures (e.g., task prioritization, threat assessment). The knowledge encoded in these rules is derived from the CAPTA analysis process described in detail in Section 9.6, that predicts the effects of different affective states on performance in the current task context. This process is an essential component of building the impact prediction knowledge base, since the state-of-the-art of theoretical understanding and empirical research in personality and emotion do not allow accurate prediction of these influences in a generic, domain-independent manner. The separation of the generic and specific knowledge enhances modularity and simplifies knowledge-based adjustments.

9.8 ADAPTATION

Once the user affective and belief states are identified, and their likely impact is predicted, ABAIS identifies a compensatory strategy and selects a means of implementing this strategy (Section 9.8.1) in terms of specific user interface modifications (Section 9.8.2).

9.8.1 Strategy Selection

As was the case with impact prediction, the strategy selection module relies on a detailed CAPTA analysis to identify specific compensatory strategies to counteract the identified performance biases. This analysis serves as the basis for constructing the strategy selection knowledge bases, which map a specific behavioral bias (e.g., task neglect, threat-estimation bias, etc.) onto a particular strategy (e.g., present reminders of neglected tasks, present broader evidence to counteract threat-estimation bias, etc.).

Again, this process consists of two stages. First, generic strategy rules map the *generic* performance bias (e.g., task neglect) into a *generic* compensatory strategy (e.g., present reminders of neglected tasks). Next, the generic rules are instantiated in the task context, to determine the actual task-specific strategies. Examples of both generic and task-specific rules are shown in Table 9.7.

9.8.2 GUI Adaptation

Once the compensatory strategy has been identified, ABAIS moves on to the final step of implementing this strategy in terms of specific modifications to the user's interface. The GUI/DSS adaptation strategies are expressed in abstract terms, and are instantiated within the particular user-task context, taking into consideration the user preference profiles. In this final step of the user adaptation process, the ABAIS adaptation module performs three sequential functions:

Table 9.7: Examples of Rules for Compensatory Strategies Selection

Generic Rules	Task Domain Specific Rules
Anxiety	

IF (<task> importance = high) AND (pilot's assessment of <task> importance = low)
THEN
(present reminders for <task>) AND (direct attention to neglected instruments/data)
 	
IF (threat estimation bias = high)
THEN
(collect all evidence regarding radar signal identity)
 	
IF (confirmation bias = high)
THEN
(collect any contradictory evidence) AND (enhance display of evidence)
	IF (recent change in radar target status)
	THEN
	(emphasize change in status of return)
	
	IF (attention focus = HUD) AND (incoming radar data)
	THEN
	(redirect focus to radar)
	
	IF (target = unknown) AND (target belief = hostile)
	THEN
	(emphasize unknown status) AND (collect more data)

Obsessiveness	
IF (obsessiveness = high) THEN (remind of consequences of delayed decisions) (remind that no data available to provide additional information) (remind of most recent tasksd accomplished—present explicit checklists) (display task timeline and current position within timeline)	IF (likelihood of delayed attack = high) THEN (display all available info about enemy a/c) AND (display likelihood of attack by enemy a/c) AND (display envelope of vulnerability around own aircraft) AND (display reminders for attack tasks)

- Identifies additional information required based on selected compensatory strategy
- Selects best information presentation format
- Applies individual information presentation preferences and capabilities (e.g., modality preference, color blindness, etc.).

Below, we discuss some general principles guiding the selection of specific GUI adaptation method and illustrate the different alternatives in the context of the ABAIS user interface.

In general, two broad categories of adaptation are possible:

- Content-based, which *provide additional information*, and
- Format-based, which *modify the format of existing information*.

*Content-base*d *adaptation* involves the collection and display of additional data or knowledge to compensate for a particular performance bias. For exam-

ple, providing additional disambiguating data about an ambiguous stimulus helps prevent an anxiety-induced bias to identify such stimuli as threats.

*Format-base*d *adaptation* involves the presentation of existing data in an alternative format, to enhance visibility, to draw attention to neglected displays, and, in general, to facilitate detection, recognition, and assimilation of data. For example, modifying some attention-capture attribute of a display such as size, color, blink rate, etc. helps draw the user's attention to the display.

These types of adaptations have been extensively evaluated in both laboratory and field settings, indicating that even small display changes can have major impact on attentional and cognitive processing (Wickens, Sandry, & Vidulich, 1983).

9.8.2.1 Level of Adaptation. Regardless of the method chosen, adaptation eventually results in a modification of specific User Interface (UI) attributes. These include changes in overall format, different choice of icons, or changes in UI elements, including color, location, size, orientation, modality, motion of stimulus, motion on periphery to redirect attention by preattentive processes. Adaptation can thus take place at any of the four levels below:

- *Icon level*. Modify individual GUI icons by modifying one of the appearance attributes (e.g., highlighting, changing its location within a display, changing color or size, etc.) or modifying the icon appearance itself
- *Display level*. Modify the display as a whole by changing the size or location of a selected display (e.g., moving a critical display to a central location of the overall UI), changing the appearance of a display (e.g., range setting on radar), or changing the contents of a display (e.g., decluttering a display)
- *Notification level.* Augment interface by inserting new, or modifying existing, alarms and alert notifications. Examples of notification level adaptations include adding a notification string regarding desired focus of attention (see for example "RADAR" on the HUD display or "VULNERABLE" string on the radar display in Figure 9.8), or adding an icon to a display to represent new information (see triangle in Figure 9.8)
- *User interface level.* Implement global changes to the user interface as a whole or insertion of display elements designed to focus attention on particular areas of the overall UI. Examples of UI level adaptations include a reconfiguration of the entire set of instruments on the UI to reflect a different system mode, increasing the redundancy of warnings (e.g., adding an auditory warning to a visual one, etc.), or the insertion of attention-capturing and attention-directing elements designed to direct the user's attention to a particular icon or display.

Each of these levels affords different alternatives and is more or less suitable for a given situation, depending on the task, task context, and the individual. A summary of the adaptations implemented within the ABAIS prototype is shown in Figure 9.3.

9.8.2.2 Individualized Adaptation. To be effective, the above strategies must be customized by taking into account the user's (e.g., the pilot's) display and modality preferences (see Table 9.8). This is critical to any adaptive approach, due to the large individual differences that exist in human information processing and decision-making. ABAIS therefore allows the specification of multiple user display preference profiles (e.g., knowing that a particular user has a high-sensitivity to auditory signals, ABAIS suggests that auditory warnings be used to capture attention).

9.9 DEMONSTRATION OF ABAIS SYSTEM PERFORMANCE

To demonstrate the ABAIS adaptive methodology and prototype performance, we defined several representative pilot profiles (high and low anxiety), varying in personality (obsessiveness, aggressiveness), training, individual history, and adaptation preferences. Their performance within scenario segments was simulated, generating varying levels of anxiety and alternative situation assessments at different points. Due to the precise timing required to demonstrate the adaptation GUI changes, the emphasis during this initial effort was on developing the analyst-as-user mode[4] and the associated script-based simulation, which allows precise control over the external task events necessary to demonstrate the real-time adaptation.

The pilot's anxiety levels were assessed, resulting in GUI/DSS adaptations. Specifically, ABAIS predicted that the heightened level of anxiety would cause narrowing of attention and interpretation bias towards threats (see box in Figures 9.7 and 9.8 for a summary of the ABAIS-derived pilot affective and belief states), possibly causing the pilot to fail to notice a recent change in status of radar contact from unknown presumed hostile to friendly. ABAIS therefore suggested a compensatory strategy aimed at preventing possible fratricide by (1) directing the pilot's attention to the display showing the recent status change and (2) enhancing the relevant signals on the radar to improve detection (see Figures 9.6 through 9.8). Specifically, the blinking, enlarged, contact icon on the HUD display indicates a change in status. The blinking "RADAR" string displayed on the HUD, the pilot's current focus, directs the pilot to look at the radar display, which shows an enhanced contact icon indicating a change in status, with details provided in the text box in lower left corner of the display.

4. This is in contrast to the alternative user-as-pilot mode where the user actually flies the simulated aircraft during the scenario.

Table 9.8: Pilot Information Preference Profile:
Categories of Information and Related GUI Modification Options

Information Category	Options
Preferred means of enhancing visibility	Color; Size; Blinking
Preferred color for alarms	Red solid; Red outline
Preferred alarm notification modality	Visual; Auditory
Preferred attention capture means	Movement at visual periphery; Shift display to foveal region; Enhance icon visibility; Display arrow pointing to desired icon; Superimpose blinking icon

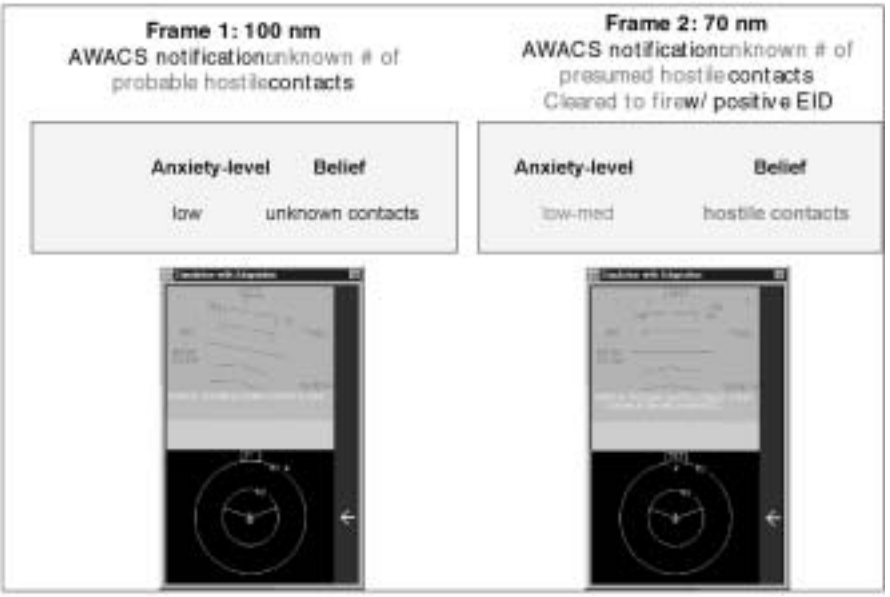


Figure 9.6: Frames 1 and 2 of the demonstration scenario: No adaptation occurs.

Informal preliminary ABAIS evaluation by an expert pilot indicated the following: (1) underscoring the importance of nondramatic, "benign adaptations," (2) general approval of the GUI adaptation strategies and modifications, (3) questions regarding the system's ability to perform *accurate* assessment in *real-time*, (4) a degree of skepticism regarding the need for affective adaptive system in an *operational*, versus training, fighter pilot cockpit, and (5) but at the same time overall enthusiasm for this type of research. A plan for extensive empirical evaluation to formally address these issues is outlined in Section 9.10.3

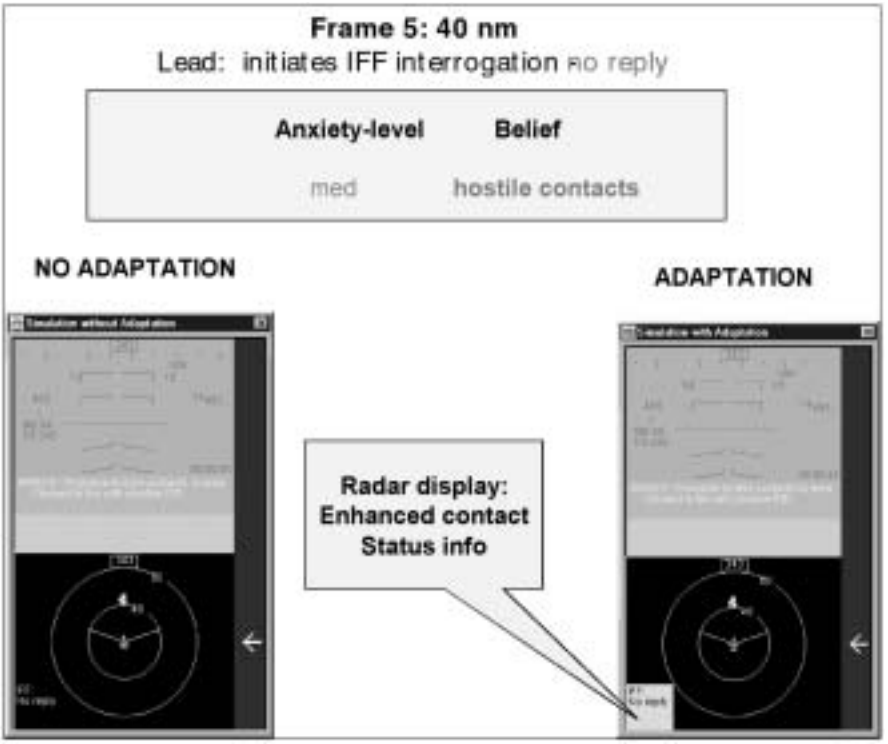


Figure 9.7: Frame 5 of the demonstration scenario: Adaptation occurs to enhance visibility and status of ambiguous radar returns.

9.10 SUMMARY, CONCLUSIONS, AND FUTURE WORK

9.10.1 Summary

The primary result of this effort was a proof-of-concept demonstration of an ABAIS, designed to provide individualized GUI and DSS adaptations based on the user's affective and belief state. ABAIS implements a four-stage adaptive methodology for the assessment of, and adaptation to, the user's affective and belief states which goes beyond traditional cognitive systems engineering practice. The ABAIS adaptive methodology was implemented within a software prototype, and demonstrated in the context of an Air Force sweep mission task. Several representative pilot profiles were defined, varying in personality, physiological responsiveness, training, individual history, and adaptation preferences. Their performance within selected scenario segments was simulated, generating varying levels of anxiety and alternatives in situation assessments at different

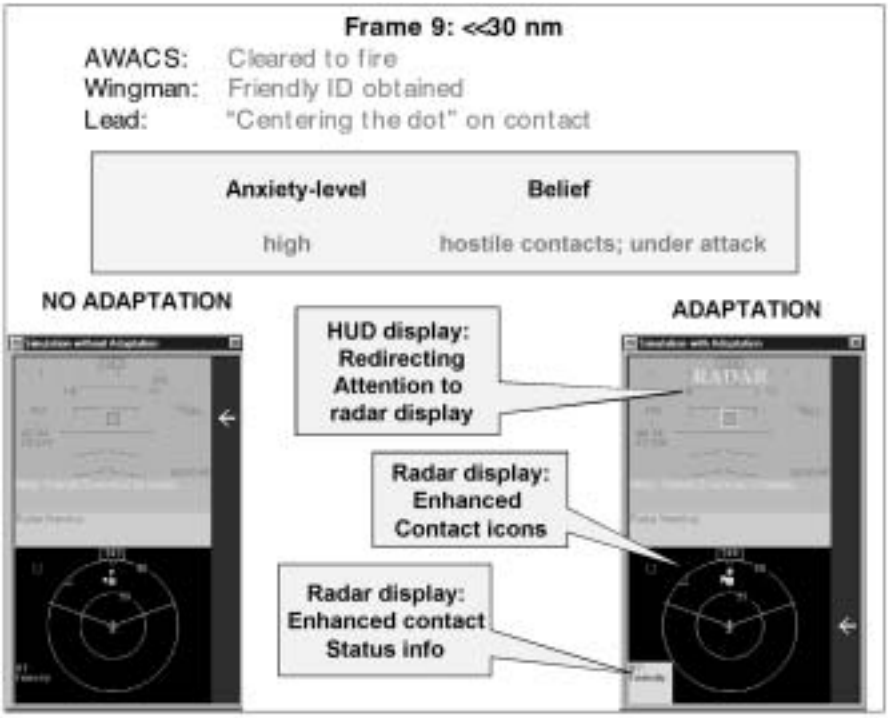


Figure 9.8: Frame 9 of the demonstration scenario: Adaptation occurs to redirect attention and enhance visibility of incoming data to prevent fratricide.

points during the scenario. The pilot's anxiety levels were assessed using the available data, resulting in GUI/DSS adaptations derived via the ABAIS adaptive methodology, using rule-bases in the four ABAIS modules. Specifically, ABAIS predicted that the heightened level of anxiety would cause narrowing of attention and interpretation bias towards threats, possibly causing the pilot to fail to notice a recent change in status of radar contact from unknown presumed hostile to friendly. ABAIS therefore suggested a compensatory strategy aimed at preventing possible fratricide by augmenting the existing cockpit GUIs to (1) direct the pilot's attention to the display showing the recent status change and (2) enhance the relevant signals on the radar to improve detection.

9.10.2 Conclusions

Development of the ABAIS proof-of-concept prototype demonstrated feasibility of the overall adaptive methodology and its implementation. ABAIS assessed the user anxiety level and belief states using a knowledge-based

approach and information from a variety of sources (e.g., static task context, dynamic external events occurring during the scenario, individual history, personality, training, and simulated physiological data), predicted the effects of these states within the constrained context of the demonstration task, and suggested and implemented specific GUI adaptation strategies, taking into account the user's individual information presentation preferences (e.g., modified an icon or display to capture attention and enhance visibility). An empirical study with actual pilots flying the ABAIS simulation and providing real-time assessment data would be required to fully assess the actual effectiveness of the assessments and adaptations in an operational context (see discussion below).

Development of a remote heart-rate monitor and its linkage with the ABAIS user-interface successfully demonstrated the feasibility of nonintrusive heart-rate monitoring, providing this information to the system, and making specific changes in the user-interface as a result of detected changes. While the monitor was not integrated into the User Assessment module due to time constraints, these preliminary results indicate that such an integration is feasible.

The implementation of the ABAIS prototype demonstrated general feasibility of the adaptive methodology, and provided information about the specific requirements for a successful, operational affective adaptive interface; namely:

- Limiting the number, type, and resolution of affective states (e.g., distinguishing between high versus low anxiety)
- Using multiple, complementary methods and multiple data sources for affective state assessment
- Providing individualized user data, including details of past performance, individual history, personality traits, and physiological data
- Constraining the overall situation in terms of situation assessment and behavioral possibilities
- Providing a wide variety of task-specific data in an electronic format
- Fine-tuning the rule-bases and inferencing to "personalize" the system to the individual user-task context, and
- Implementing "benign" adaptations, that is, GUI/DSS modifications that at best enhance and at worst maintain current level of performance (e.g., adaptations should never limit access to existing information).

9.10.3 Future Work

The objective of the initial ABAIS prototype was to implement a proof-of-concept demonstration of the ABAIS adaptive methodology, implemented within an adaptive system architecture. The existing prototype indicates that this represents a feasible approach to affective adaptation. However, much work remains to be done to demonstrate its effectiveness in an operational setting, to

implement additional enhancements, to validate and streamline the CAPTA process, and to generalize the ABAIS system to other domains. Several of these directions are briefly outlined below.

9.10.3.1 Evaluation. The most critical next step is an empirical study demonstrating improved human-system performance with adaptation and providing feedback about the ABAIS methodology, architecture, and knowledge-base enhancements, shortcomings, and operating constraints. Both qualitative (e.g., cognitive task walkthroughs, protocol analysis; heuristic evaluations) and quantitative (e.g., traditional empirical evaluations) are planned. Collectively, the results from these evaluation experiments will provide specific performance enhancements, shortcomings, and constraints, which, together, will define future improvements to the original ABAIS methodology, architecture design, and knowledge bases. Through successive approximations the ABAIS prototype would be revised (and reevaluated) to improve system effectiveness and human-system performance.

9.10.3.2 Generalizability. The next step is to demonstrate the generalizability of the ABAIS methodology and architecture across tasks and domains. ABAIS was designed to maximize generalizability. Specifically, the following elements apply across domains:

- ABAIS four-step adaptive methodology
- ABAIS architecture framework
- Assessment model integrating multiple factors
- Rules capturing generic effects of affective-state biases and generic compensatory strategies
- Integration of generic empirical data with task-specific domain data to identify possible effects of affective and belief states and possible compensatory strategies, and
- User-interface modification framework (format/content and levels of adaptation).

In addition, the CAPTA task-analytic process underlying the development of the ABAIS architecture knowledge-bases is also domain independent.

However, to implement ABAIS in a different task domain, a number of system modifications must be implemented by the developer and cognitive systems engineer, to identify the task and domain specific data and background information about users, and enter these into the existing ABAIS framework and architecture. Specifically, the following steps would be required:

- Detailed CAPTA to identify domain and task-specific effects of particular affective and belief states
- Detailed CAPTA to identify domain and task-specific compensatory strategies for their effects
- Construction of domain-specific components of the "Impact Prediction" and "Strategy Selection" KBs
- Entering user background data (individual history, personality traits, training, etc.)
- Development of an appropriate task simulation module, and
- Development of task-and domain-specific user-interfaces.

Again, an empirical evaluation will be required to determine the exact level of effort required to perform such a translation, to quantify the benefits offered by the ABAIS architecture components, and to identify any constraints to such domain transitions (e.g., applicability of generic knowledge across domains, etc.).

9.10.3.3 Enhancement and Integration of Multiple Assessment Methods. The ABAIS architecture was designed to accommodate multiple methods of affect and belief assessment, with the initial prototype focusing on the knowledge-based approach. Future work in this area therefore includes the implementation and exploration of the integrated use of multiple, complementary methods; enhancing the types of data processed by these methods (e.g., user's goals, specific knowledge types); enhancing the inferencing algorithms used during the assessment process; enlarging the set of affective and belief states identified; and addressing a variety of issues such as contradictory data and assessment resolution. Such integration will then allow the exploration and evaluation of the best "mix" of these methods for particular task types, domain types, or task-domain-user combinations. We would also like to explore a variety of second-order effects to enhance the reliability of affect assessment (i.e., explicit representation and analysis of the interaction among distinct affective states and among multiple factors influencing a particular affective state).

The field of affective computing is in its infancy. The confluence of technologies that facilitate affective assessment and adaptation on the one hand, and the increasing need and desire for such functionalities, provide a rich environment within which to investigate a number of fundamental issues about the role of emotion in human and human-computer performance and interaction. Key questions include issues such as to what extent are existing user modeling and adaptation methods applicable to affective adaptation? What emotions should and can be recognized, modeled, and adapted to in human-machine interfaces? When should an agent attempt to enhance the user's affective state? When should it adapt to the user's affective state? When should it attempt to counteract it? In

addition to this, a number of ethical issues emerge once affect enters the stage. Canamero offers an excellent summary of some of the affect-related issues that must be addressed by the user modeling community (Canamero, 1998).

The work described here represents an attempt to integrate affective computing considerations and methods into the practice of traditional cognitive systems engineering and HCI design, in other words, to move beyond the traditional cognitive and psychophysical factors in designing the human-machine interface, by explicitly integrating affective and personality considerations into the design process via CAPTA. While the initial prototype was developed within a military aviation task context, we believe that the results are applicable to a broad variety of nonaviation and nonmilitary domains.

We conclude with a quote from a virtual agent of yore: "This fills my head with ideas, only I don't exactly know what they are" (Carroll, 1941).

9.11 ACKNOWLEDGEMENTS

This work was supported in part by U.S. Air Force contract number F41624–98–C5032. The authors wish to thank the following individuals who contributed to this effort. Mr. John Billingsley of Psychometrix Associates was the software developer for the ABAIS prototype, responsible for the software system design and implementation. Mr. Bob Shaw of FCI Associates developed the demonstration scenario and served as the primary SME. Major Ray King, Major Joseph Callister, LTC Tim Sowin, and LTC Royland Marsh provided information about personality assessments within USAF. Major Daniel Callan and LTC Vincent Caterina provided useful suggestions and feedback regarding the task scenario selections. Dr. Thomas von Wiegand of MIT developed the heart monitor. Drs. Scott Orr of Harvard University and Stefan Hofmann of the Boston University Anxiety Center provided information about physiological sensing. Dr. Greg Zacharias provided helpful discussions about the Charles River Analytics SD_PVI system. Dr. Rosalind Picard of MIT Media Lab provided information about wearable computing, and background information on physiological sensing and facial recognition.

REFERENCES

Bates, J., Loyall, A.B., & Reilly, W.S. (1992). Integrating Reactivity, Goals, and Emotion in a Broad Agent. In *Proceedings of the 14th Meeting of the Cognitive Science Society*. Hillsdale, NJ: Erlbaum.

Blaney, P.H. (1986). Affect and Memory. *Psychological Bulletin, 99* (2), 229–246.

Bower, G.H. (1981). Mood and Memory. *American Psychologist, 36*, 129–148.

Breese, J,. & Ball, G. (1998). Bayesian Networks for Modeling Emotional State and Personality. In *Proceedings of Emotional and Intelligent: The Tangled Knot of Cognition.* AAAI Fall Symposium Series, (TR FS–98–03). Menlo Park, CA: AAAI Press.

Cacioppo, J.T., Klein, D.J., Bernston, G.G., & Hatfield, E. (1993). The Psychophysiology of Emotion. In J. Haviland & M. Lewis, (Eds.), *Handbook of emotions*. New York: Guilford.

Callister, J.D., King, R.E., Retzlaff, P.D., & Marsh, R.W. (1997). *Using the NEO–PI–R to Assess the Personality of U.S. Air Force Pilots* (Tech. Rep. AL/AO–TR–1997–0097). Brooks AFB, TX: Armstrong Laboratory.

Canamero, D. (1998). Issues in the Design of Emotional Agents. In *Proceedings of Emotional and Intelligent: The Tangled Knot of Cognition.* AAAI Fall Symposium, (TR FS–98–03). Menlo Park, CA: AAAI Press.

Carroll, L. (1941). *Alice in wonderland.* New York: Heritage Press.

Cooke, N. M. (1995). Varieties of Knowledge Elicitation Techniques. *International Journal of Human-Computer Studies, 41* (6), 801–849.

Damasio, A. R. (1994). *Descartes error: Emotion, reason, and the human brain.* New York: G. P. Putnam's Sons.

Deckert, J. C., Entin, E. B., Entin, E. E., MacMillan, J., & Serfaty, D. (1994). *Military Command Decisionmaking Expertise* (Report 631). Burlington, MA: Alphatech.

Dietrich, H., Malinowski, U., Kuhme, T., & Schneider-Hufschmidt, M. (1993). State of the Art in Adaptive Interfaces. In M. Schneider-Hufschmidt, T. Kuhme, & U. Malinowski, (Eds.), *Adaptive user interfaces.* New York: Elsevier.

Ekman, P., & Davidson, R.J. (1994). *The nature of emotion: Fundamental questions.* New York: Oxford.

Ekman, P., & Friesen, W. V. (1975). *Unmasking the face*. Englewood Cliffs, NJ: Prentice Hall.

Elliot, C. (1992).*The Affective Reasoner: A process model of emotions in a multi-agent system* (Tech. Rep. #32). Evanston, IL: Northwestern University Institute for the Learning Sciences.

Elliott, L. R., Schiflett, S. G., Hollenbeck, J. R., & Dalrymple, M. A. (2001). Investigation of Situational Awareness and Performance in Realistic Command and Control. In M. D. McNeese, E. Salas, & M. R. Endsley (Eds.). *New trends in cooperative activities: Understanding system dynamics in complex environments.* Santa Monica, CA: HFES.

Endsley, M. R. (1995). Measurement of Situation Awareness in Dynamic Systems. *Human Factors, 37* (1), 65-84.

Essens, P., Fallesen, J., McCann, C., Cannon-Bowers, J., & Dorfel, G. (1995). *COADE: A Framework for Cognitive Analysis, Design, and Evaluation* (Final AC/243, Panel 8, TR/17). Brussels: NATO, Defence Research Section.

Eysenck, M.W. (1997). *Anxiety and cognition: A unified theory.* Hove, East Sussex, UK: Psychology Press.

Fijalkiewicz, P., & DeJong. G. (1998). CHESHIRE: An Intelligent Adaptive User Interface. In *Proceedings of Second Annual Symposium on Advanced Displays* (pp. 15–19), College Park, MD: University of Maryland.

Fleming, M., & Cohen, R. (1999). User Modeling in the Design of Interactive Interface Agents. In J. Kay (Ed.), *User Modeling: Proceedings of the Seventh International Conference UM99.* New York: Springer.

Fraser, N. M., Hipel, K. W., Kilgore, D. M., McNeese, M. D., & Snyder, D. E. (1989). An architecture for integrating expert systems. *Decision Support Systems, 5*, 263–276.

Harper, K., Guarino, S., & Zacharias, G. (2000). Adaptive Human-Machine Interface for Advanced Air Traffic Management. In *Proceedings of 2000 AIAA, Navigation and Control Conference,* Denver, CO.

Hathaway, S. R., & McKinley, J.C. (1989). MMPI–2: Minnesota Multiphasic Personality Inventory-2. Minneapolis, MN: The University of Minnesota Press.

Healy, J., & Picard, R. (2000). SmartCar: Detecting Driver Stress. In *Proceedings of the 15th International Conference on Pattern Recognition*, Barcelona, Spain. (Also MIT Media Lab TR#525).

Hollnagel, E., & Woods, D.D. (1983). Cognitive systems engineering: New wine in new bottles. *International Journal of Man-Machine Studies, 18*, 583–600.

Horowitz, M.J. (1991). *Person schemas and maladaptive interpersonal patterns*. Chicago: University of Chicago Press.

Hudlicka, E. (2000a). *Cognitive Affective Personality Task Analysis* (Tech. Rep. # 2010). Blacksburg, VA: Psychometrix Associates.

Hudlicka, E. (2000b). Increasing SIA Architecture Realism by Modeling and Adapting to Affect and Personality. In *Proceedings of the AAAI Fall Symposium* (TR FS–00–04). Menlo Park, CA: AAAI Press.

Hudlicka, E., & Billingsley, J. (1999). ABAIS: Affect and Belief Adaptive Interface System. In *Proceedings of the 43rd Annual Meeting of the Human Factors and Ergonomics Society*. Santa Monica, CA: HFES.

Hudlicka, E., McNeese, M., J. Healy, D. Serfaty, G. Zacharias, & R. Wellens. (1999). Issues in Affective Computing—Panel Discussion Summary. In *Proceedings of the 43rd Annual Meeting of the Human Factors and Ergonomics Society*. Santa Monica, CA: HFES.

Hudlicka, E. (1998). Modeling Emotion in Symbolic Cognitive Architectures. In *Proceedings of Emotional and Intelligent: The Tangled Knot of Cognition.* AAAI Fall Symposium Series (TR FS–98–03). Menlo Park, CA: AAAI Press. (See also Pew & Mavor, 1998).

Hudlicka, E., & Billingsley, J. (1998). ABAIS: *Affect and Belief Adaptive Interface System. Report 9814.* Lincoln, MA: Psychometrix Associates. (Also AFRL–HE–WP–TR–1999–0169. Wright-Patterson AFB, OH).

Hutchins, E. (1995). *Cognition in the wild.* Cambridge, MA: MIT Press.

Institute of Medicine. (1999). *To err is human: Building a safer health system.* Washington, DC: National Academy Press.

Isen, A. M. (1993). Positive Affect and Decision Making. In J. M. Haviland & M. Lewis (Eds), *Handbook of emotions.* New York: Guilford Press.

Johnson-Laird, P. N. (1983). *Mental models.* Cambridge, MA: Harvard University Press.

Kaiser, S., Wehrle, T., & Schmidt, S. (1998). Emotional episodes, facial expressions, and reported feelings in human-computer interaction. In *Proceedings of the Xth Conference of the International Society for Research on Emotions*. Wursburg, Germany: ISRE Publications.

Klein, G. A. (1997). The recognition-primed decision (RPD) model: Looking back, looking forward. In C. Zsambok & G. Klein (Eds.), *Naturalistic decision making.* Mahwah, NJ: Erlbaum.

Klein, G. A. (1989). Recognition-Primed Decisions. *Advances in Man-Machine Systems Research, 5*, 47–92.

Klein, J., Moon, Y., & Picard, R. W. (1999). This Computer Responds to User Frustration. *CHI 99 Short Papers*. Pittsburgh, PA.

Klein, J. (1999). *Computer Responses to User Frustration* (Master's Thesis, TR 480). Cambridge, MA: MIT.

Lazarus, R. S. (1991). *Emotion and adaptation*. New York: Oxford University Press.

LeDoux, J. E. (1992). Brain Mechanisms of Emotion and Emotional Learning. *Current Opinion in Neurobiology, 2* (2), 191–197.

LeDoux, J. E. (1989). Cognitive-Emotional Interactions in the Brain. *Cognition and Emotion, 3* (4), 267–289.

Lindsay, P. L., & Norman, D.A. (1977). *Human information processing*. NY: Academic Press.

Lipshitz, R., & Ben Shaul, O. (1997). Schemata and mental models in recognition-primed decision making. In C. Zsambok & G. Klein (Eds.), *Naturalistic decision making.* Mahwah, NJ: Erlbaum.

Maes, P. (1994). Agents that reduce work and information overload. *Communications of the ACM, 37 (*7).

McNeese, M. D. (2002). Discovering how cognitive systems should be engineered for aviation domains: A developmental look at work, research, and practice. In M. D. McNeese & M. A. Vidulich (Eds.), *Cognitive systems engineering in military aviation domains: Avoiding cogminutia fragmentosa.* Wright-Patterson Air Force Base, OH: Human Systems Information Analysis Center.

McNeese, M. D., Rentsch, J. R., & Perusich, K. (2000). Modeling, measuring, and mediating teamwork: The use of fuzzy cognitive maps and team member schema similarity to enhance BMC^{3}I decision making. *IEEE International Conference on Systems, Man, and Cybernetics* (pp. 1081–1086). New York: IEEE.

McNeese, M. D. (1989). *Cerebral weevil: A machine learning model for hemispheric categorization of faces.* (Tech. Rep. AAMRL–TR–89–045). Wright-Patterson Air Force Base, OH: Armstrong Aerospace Medical Research Laboratory.

McNeese, M. D., & Katsuyama, R. (1987). Neuropsychology in the cockpit: An analysis of configurational processing, hemispheric asymmetry, and masking disturbance. *Proceedings of the 4th International Symposium on Aviation Psychology* (pp. 202–208), Columbus, OH.

McNeese, M. D. (1986). Humane intelligence: A human factors perspective for developing intelligent cockpits. *IEEE Aerospace and Electronic Systems, 1* (9), 6–12.

Mineka, S., & Sutton, S. K. (1992). Cognitive Biases and the Emotional Disorders. *Psychological Science, 3* (1), 65–69.

Mitchell, T., Caurana, R. Freitag, D., McDermott, J., & Zabowski, D. (1994). Experience with a learning personal assistant. *Communications of the ACM, 37* (7).

Mulgund, S., Rinkus, G., & Zacharias, G. (2001). Adaptive Pilot/Vehicle Interfaces for the Tactical Air Environment. In *Psychological Issues in the Design and Use of Virtual Adaptive Environments*. Mahwah, NJ: Erlbaum.

Orr, S. P. (1998). Personal communication.

Ortony, A., Clore, G. L., & Collins, A. (1988). *The cognitive structure of emotions.* New York: Cambridge University Press.

References

Panksepp, J. (1994). The Basics of Basic Emotion. In P. Ekman & R. J. Davidson (Eds.), *The nature of emotion.* New York: Oxford.

Persons, J. B., & Foa, E. B. (1984). Processing of fearful and neutral information by obsessive-compulsives. *Behavior Research and Therapy, 22*, 260–265.

Peterson, R. A., & Reis, S. (1987). *Test manual for the anxiety sensitivity index.* Orland Park, IL: International Diagnostic Systems.

Petrushin, V. A. (2000). Emotion Recognition Agents in Real World. In *Proceedings of the AAAI Fall Symposium*. (TR FS–00–04). Menlo Park, CA: AAAI Press.

Pew, R. W., & Mavor, A. S. (1998). *Representing human behavior in military simulations.* Washington, DC: National Academy Press.

Picard, R. (1997). *Affective computing*. Cambridge: MIT Press.

Pohl, W., & Nick, A. (1999). Machine learning and knowledge representation in the LabUR approach to user modeling. In J. Kay (Ed.), *User modeling: Proceedings of the Seventh International Conference UM99.* New York: Springer.

Rasmussen, J., Pejtersen, A. M., & Goodstein, L. P. (1994). *Cognitive systems engineering: Concepts and applications.* New York: Wiley.

Reising, J. M. (2002). Cognitive engineering and its relationship to future aviation systems. In M. D. McNeese & M. A. Vidulich, (Eds.), *Cognitive systems engineering in military aviation domains: Avoiding cogminutia fragmentosa!* Wright-Patterson AFB, OH: Human Systems Information Analysis Center.

Resnick, L., Levine, J., & Teasley, S. D. (1991). *Perspectives on socially shared cognition.* Washington, DC: American Psychological Association.

Retzlaff, P. D., Callister, J. D., & King, R. E. (1997). The *Armstrong Laboratory Aviation Personality Survey (ALAPS): Norming and Cross-Validation,* (Tech. Rep. AL/AO–TR–1997–0099). Brooks AFB, TX: Armstrong Laboratory.

Rouse, W. B. (1988). Adaptive aiding for human/computer control. *Human Factors, 30* (40), 431–443.

Scherer, K. (1993). Studying the emotion-antecedent appraisal process: The expert system approach. *Cognition and Emotion, 7,* 325–355.

Serfaty D., MacMillan J, Entin, E. E., & Entin E. B. (1997) The Decision-Making Expertise of Battle Commanders. In C. Zsambok & G. L. Klein (Eds.) *Naturalistic decision making.* Mahwah, NJ: Erlbeum.

Taylor, R. M., Bonner, M. C., Dickson, B., Howells, H., Miller, C., Milton, N., Pleydell-Pearce, K., Shadbolt, N., Tennison, J., & Whitecross, S. (2002). Cognitive cockpit engineering: Coupling functional state assessment, task knowledge management, and decision support for context sensitive aiding. In M. D. McNeese & M. A. Vidulich, (Eds.), *Cognitive systems engineering in military aviation domains: Avoiding cogminutia fragmentosa!* Wright-Patterson AFB, OH: Human Systems Information Analysis Center.

Tversky, B., & Kahneman, D. (1974). Judgments under uncertainty: Heuristics and biases. *Science, 185*, 1124–1131.

Vyzas, E., & Picard, R. (1998). In *Proceedings of Emotional and Intelligent: The Tangled Knot of Cognition.* AAAI Fall Symposium Series (TR FS–98–03). Menlo Park, CA: AAAI Press.

Watson, D., Clark, L. A., & Tellegen, A. (1988). Development and validation of brief measures of positive and negative affect: The PANAS Scales. *Journal of Personality and Social Psychology, 54*, 1063–1070.

Weick, K. E. (1979). *The social psychology of organizing*. Reading, MA: Addison-Wiley.

Wellens, A. R. (1993). Group situational awareness and distributed decision making: From military to civilian applications. In N. J. Castillian (Ed.), *Individual and group decision making: Current issues*. Hillsdale, NJ: Erlbaum.

Vicente, K. J. (1999). *Cognitive work analysis: Towards safe, productive, and healthy computer-based work.* Mahwah, NJ: Erlbeum.

Wickens, C. D., Sandry, D., & Vidulich, M. (1983). Compatibility and resource competition between modalities of input, output, and central processing. *Human Factors, 25*, 227–248.

Williams, J. M. G., Watts, F. N., MacLeod, C., & Mathews, A. (1997). *Cognitive psychology and emotional disorders*. New York: Wiley.

Woods, D. D., Johannesen, L. J., Cooke, R. I., & Sarter, N. B. (1994). *Behind human error: Cognitive systems, human error, and hindsight.* Wright-Patterson Air Force Base, OH: Crew Systems Ergonomics Information Analysis Center.

Woods, D. D. (1998). Designs are hypotheses about how artifacts shape cognition and collaboration. *Ergonomics, 41* (2), 168–173.

Young, M. F., & McNeese, M. D. (1995). A situated cognition approach to problem solving. In P. Hancock, J. Flach, J. Caird, & K. Vincente (Eds.), *Local applications of the ecological approach to human-machine systems.* Hillsdale, NJ: Erlbaum.

Zacharias, G. L. (1996). SAMPLE: *Situation Awareness Model for Pilot-in-the-Loop Evaluation* (Final Report R95192). Cambridge, MA: Charles River Analytics.

Zaff, B. S., McNeese, M. D., & Snyder, D. E. (1993). Capturing multiple perspectives: A user-centered approach to knowledge acquisition. *Knowledge Acquisition, 5* (1), 79–116.

Zsambok, C. E., & Klein, G. (1997). *Naturalistic decision making.* Mahwah, NJ: Erlbaum.

Acronyms

AEW&C	Airborne Early Warning and Control
ABAIS	Affect and Belief Adaptive Interface System
ACTA	Applied Cognitive Task Analysis
ADS	Abstraction Decomposition Space
AH	Abstraction Hierarchy
AI	Artificial Intelligence
AKADAM	Advanced Knowledge And Design Acquisition Methodology
AOD	Air Operations Division (Australia)
AWACS	Airborne Warning and Control Systems
C2	Command and Control
C3	Command, Control, and Communication
C3I	Command, Control, Communication, and Intelligence
CACSE	Computer-Assisted Cognitive System Engineering
CAP	Combat Air Patrol
CAPTA	Cognitive-Affective Personality Task Analysis
CAST	Constraint Analysis and Synthesis Technique
CCT	Cognitive Complexity Theory
CDM	Critical Decision Method
CDTL	Collaborative Design Technology Laboratory
CE	Copilote Electronique
CHS	Center for Human Sciences (DERA, UK)
CMU	Carnegie Mellon University (US)
COGMON	COGnition MONitor
COGNET	COGnitive NETwork of tasks
COGPIT	COGnitive cockPIT
COPE	C3 Operator Performance Engineering
COGSIM	COGpit SIMulation test environment
CPM-GOMS	Cognitive, Perceptual, and Motor GOMS
CPM-GOMS	Critical Path Method GOMS
CS	Control Strategy
CSCW	Computer-Supported Cooperative Work
CSE	Cognitive Systems Engineering
CT	Control Task
CTA	Cognitive Task Analysis
CTA	Control Task Analysis
CWA	Cognitive Work Analysis
DARPA	Defense Advanced Research Projects Agency (US)
DAS	Defensive Aids Suite
DAT	Decision Aiding Taxonomy

DCIEM	Defence and Civil Institute of Environmental Medicine (Canada)
DERA	Defence Evaluation Research Agency (UK)
DDP	Decision Decomposition Protocol
DR	Decision Requirements
DSS	Decision Support Systems
DSTO	Defence Science and Technology Organisation (Australia)
DTD	Display Task Description
EA	Electronic Associate
EID	Ecological Interface Design
FAH	Functional Abstraction Hierarchy
FCBA	Future Carrier-Borne Aircraft
FCM	Fuzzy Cognitive Map
FMOE	Function-based MOE
FMS	Flight Management System
FOAEW	Future Organic Airborne Early Warning
FOAS	Future Offensive Air System
GIM	Global Implicit Measures
GOMS	Goals, Operators, Methods, and Selection rules
GPT	Goal-Plan Tracking
GUI	Graphical User Interface
HADS	Helicopter Aircrew Decision Support
HCI	Human-Computer Interaction
HEC	Human-Electronic Crewmember
HFE	Human Factors Engineering
HGA	Hierarchical Goal Analysis
HMD	Helmet-Mounted Display
HOTAS	Hands-On-Throttle-And-Stick
HSI	Human Systems Integration
HTA	Hierarchial Task Analysis
HUD	Heads-Up Display
IDEF	Integrated computer-aided manufacturing DEFinitions
IP	Information Processing
IPME	Integrated Performance Modeling Environment
IR	Information Requirement
ISD	Instructional Systems Design
IV	Internal Variable
JOE	Joint Operations Environment
KA	Knowledge Acquisition
KBS	Knowledge-Based Systems
MA&D	Micro Analysis and Design (US)
MFTA	Mission, Function, Task Analysis
MHP	Model Human Processor

MMA	Mission Management Aid
MMI	Man-Machine Interface
MOD	Ministry of Defence (UK)
MOE	Measures of Effectiveness
MPT	Mission Plan Tracking
NDM	Naturalistic Decision Making
NLR	National Aerospace Laboratory (The Netherlands)
OSU	Ohio State University (US)
PA	Pilot's Associate
PARI	Precursor, Action, Result, and Interpretation method
PCT	Perceptual Control Theory
PLI	Pilot Load Indicator
R&D	Research and Development
RBR	Rule-Based Reasoning
RISØ	RISØ National Laboratory (Denmark)
RPA	Rotorcraft Pilot's Associate
RPD	Recognition-Primed Decision Making
SA	Situation Awareness
SA	Strategies Analysis
SASS	Situation Assessment Support System
SBIR	Small Business Initiative Research
SME	Subject-Matter Expert
SO	Social Organizational
SOA	Social Organizational Analysis
SRK	Skills, Rules, and Knowledge taxonomy
TBCS	Tactical Battlefield Command System
TCCTA	Temporal Coordination Control Task Analysis
TIM	Task Interface Manager
TTCP	The Technical Cooperative Program
UAV	Unmanned Air Vehicle
UCAV	Unmanned Combat Air Vehicle
UCSD	University of California at San Diego (US)
UI	User Interface
UML	Unified Modeling Language
USAFRL	U.S. Air Force Research Laboratory
WC	Worker Competencies
WCA	Worker Competency Analysis
WDA	Work Domain Analysis
WPAFB	Wright-Patterson Air Force Base (US)
WDAW	Work Domain Analysis Workbench
WSO	Weapon Systems Officer

Author Index

Abdi 298, 310
Abelson 235, 250
Abotel 74
Ainsworth 54, 61, 74
Akkermans 310
Anderson 117, 308
Anesgart 106, 117
Anjewierden 310
Annett 207, 238, 246
Asimov 6-7, 11
Atwood 59, 74
Baird 213, 248
Ball 144, 165, 264, 328, 360
Ball. 144, 165, 264, 328, 360
Banbury 266, 269, 293, 296, 306
Bannon 86, 113
Barbato 303, 307, 309
Barker 99, 113
Baron 99, 116
Bates 320, 359
Baumeister 57, 61-63, 72
Bautsch 104, 108-109, 113, 115
Becker 213, 246
Beevis 201, 209, 217-218, 241, 247, 249
Belyavin 207, 247
Ben Shaul 326, 362
Benda 59, 72, 178, 198
Bennett 148, 164
Berggrund 266, 308
Bernston 277, 360
Besner 213, 247
Beyer 308
Bhatia 213, 248
Bigger 307
Billings 5, 11, 149, 166, 261, 307
Billingsley 318, 359, 361
Bisantz 34, 73
Blaney 359
Boff 73, 90, 117

Bonner 253, 266, 274, 298, 300, 306-307, 310, 363
Bookman 77
Borah 308
Bost 209, 247
Boucek 303, 307
Bourbon 240, 247
Bovair 24, 73
Bower 359
Boyer 165
Bransford 95, 101, 113
Breese 328, 360
Brehmer 268-269, 307
Brickman 266, 308
Broadbent 22, 73
Brown 75, 83, 101, 103, 108, 113-118, 250
Buchanan 87-88, 113
Bunting 258, 311
Burnett 106, 117
Burns 74
Butterfield 213, 249
Byrne 72
Cacioppo 323, 360
Cain 207, 247
Calderwood 74-75, 90, 100, 114
Callister 344, 359-360, 363
Cameron 56, 76, 256, 310
Canamero 359-360
Capaldi 213, 250
Card 19, 21-24, 73, 84, 90-91, 93, 113, 284
Carr 308
Carroll 75, 149, 164, 359-360
Casper 295, 307
Caurana 362
Chaillet 205, 247
Chalmers 144, 164
Chapnik Smith 213, 247
Checkland 204-205, 209, 247
Chernecki 213, 248
Chery 268, 308
Chin 230, 244, 247
Chipman 53, 75-76, 134, 140, 165, 246, 248
Christoffersen 121, 133, 144, 147, 164, 166
Citera 83, 103, 108, 113, 116

Clark 322, 363
Clore 328, 362
Cohen 328, 360
Coles 213, 249
Collins 12, 328, 362
Cone 307
Cook 30, 73, 77, 144, 164
Cooke 54, 73, 90, 94, 113, 118, 335, 360, 364
Coplen 144, 165
Corder 9, 12
Crandall 49, 74, 84, 114
Cress 308
Crews 96
Cunningham 207, 238, 246
Curry 276, 309
Dahn 228, 230, 247
D'Azzo 22, 73
Dalrymple 360
Damasio 320, 360
Darkow 104, 113
Darnell 209, 247
Dauchy 308
Davidson 321, 323, 360, 363
Davies 258, 282, 307, 312
De Hoog 310
Deckert 316, 360
DeJong 328, 360
Dekker 134, 149, 164, 166
Denning 166
Dennis 308
DeVilbiss 213, 250
Dickson 253, 266, 273, 306, 309, 363
Dietrich 327, 360
Dinadis 58, 73, 196, 198
DiStefano 240, 247
Dixon 308
Dominguez 73
Dornheim 8, 11
Draper 3, 11, 74-75, 116, 309
Dru-Drury 298, 300, 307, 310
Drumm 188, 198
Duncan 102, 116
East 209, 248, 360

Easter 76, 137, 144, 164
Eckberg 307
Edwards 201, 228, 230, 237, 247
Eggleston 15, 34, 51, 56, 61, 73-74, 76, 145, 164, 255-256, 260, 307, 310
Ekman 321, 323, 360, 363
Elliot 320, 325, 328, 360
Elliott 360
Ellifritt 204, 250
Elm 75, 107, 116, 134, 137, 140, 142, 144, 149, 165
Emerson 12, 257, 307
Emery 269, 311
Endsley 115, 325, 330, 360
Entin 360, 363
Ercoline 250
Ericsson 101, 114
Eriksen 213, 248
Eysenck 316, 360
Fahy 213, 250
Farrell 209, 211, 214-216, 220, 222, 241, 248, 268, 300, 307-308
Feehrer 99, 116
Feigenbaum 249
Feldman 249
Fijalkiewicz 360
Finnie 259, 264, 268, 295, 311
Flach 60, 73, 87, 102, 114, 118, 148, 164, 166, 364
Fleming 328, 360
Fletcher 300, 307
Foa 266, 308, 363
Forssell 232, 248
Fowler 213, 248
Fraser 108, 114, 327, 361
Freed 60, 73
Freiberger 221, 249
Freitag 362
Friesen 360
Fuld 4, 11
Fulghum 9, 11
Gagne 4, 11
Garner 213, 248
Geddes 260, 267, 304, 308
Gibson 39, 73, 90-91, 96, 99, 114

Gill 90, 94, 114
Goldman 259, 261-262, 266, 288, 290-293, 309
Gomes 52, 73
Goode 205, 248
Goodman 8, 11
Goodstein 76, 96, 117, 142, 165, 169-170, 199, 238, 250, 255, 309, 316, 363
Gordon 90, 94, 114
Goss 174, 199, 271, 310
Gott 52, 74
Gray 59-60, 74, 206
Green 117, 204, 250
Grossman 307
Gualtieri 76, 137, 145, 164
Guarino 361
Guerlain 258, 266, 288, 290-292, 309
Guha 87, 115
Haas 266, 308
Hajdukiewicz 70, 74
Hake 213, 248
Hall 52, 74, 117, 360
Hammond 100, 114
Hancock 5, 11, 114, 118, 166, 248, 308, 364
Hannen 258, 288, 290-292, 309
Harbor 96, 240, 249
Harper 12, 327, 361
Hart 278, 308
Hatfield 360
Hathaway 322, 361
Haugh 255, 258-259, 311
Hayes-Roth 262, 308
Healy 328-329, 361
Hendrick 86, 114-115, 209, 248
Hendy 201, 207, 209, 211, 213-216, 220-221, 230, 234, 241, 247-248, 268, 308
Herdman 213, 248
Hettinger 266, 308
Hicinbothom 77
Hill 165
Hinsz 90, 114
Hintzman 213, 248
Hipel 108, 114, 361
Hoffman 49, 74, 134

Hollan 43, 74
Hollenbeck 360
Hollnagel 18, 26, 74, 77, 84-85, 91, 93, 114, 134, 142, 144, 166, 254-255, 268-270, 308, 316, 361
Hornof 74
Horowitz 341, 361
Houpis 22, 73
Howell 4, 11
Howells 253, 257-258, 265-266, 274, 281, 306, 308, 310-311, 363
Howie 73
Hoy 259, 311
Hudgins 299, 308
Hudlicka 315, 318, 335, 341, 361
Hugdahl 323
Hughes 213, 248
Hunter 147, 166
Hutchins 42, 74, 88, 90, 114, 125, 134, 326, 361
Hutton 50, 53, 60, 74-75
Isen 316, 341, 361
James 101, 103, 117, 137, 230
Jarret 308
Johannesen 77, 94, 118, 364
John 1, 24, 59-60, 72, 74, 95, 101-102, 114, 150, 247, 359
Johnson-Laird 28, 77, 326, 361
Johnston 103, 117
Kaempf 195, 198-199
Kahneman 325, 363
Kaiser 321, 328-329, 361
Katsuyama 328, 362
Katz 108, 115
Kaufman 73, 307
Keane 213-215, 247
Kellogg 149, 164
Kenney 144, 165
Kerch 144, 165
Kieras. 24, 60, 64, 73-74
Kilgore. 108, 114, 361
Killion 213, 246
King 359-360, 363
Kinkade 205-206, 250
Kirlik 59, 76
Kirwan 54, 61, 74

Klein 8, 11, 48-51, 60, 74-75, 84, 88, 90-91, 94-96, 100, 114, 117, 119, 164-165, 243, 248, 282, 303, 308-309, 325-326, 329, 336, 360-364
Klinger 60, 74-75
Knemeyer 3, 11
Knight 75
Knobel 213, 249
Kramer 213, 249
Kuhme 360
Kuperman 60, 73, 133
Lachman 213, 249
Lave 91, 117
Layton 213, 248, 292, 308
Lazarus 320, 328, 362
Le Mentec 77
LeDoux 316, 320, 341, 362
Leger 308
Lenat 87, 115
Lessons 12, 80, 113, 228, 244, 288
Lester 213, 248, 328
Leveson 174-176, 198
Levine 89, 117, 363
Liao 211, 248, 268, 308
Lichacz 201, 214, 248
Lin 144, 165
Lind 52, 73, 142, 164
Linde 266, 308
Lindsay 92, 115, 326, 362
Lintern 169, 174, 176-177, 189, 191, 193-199, 271, 310
Lipshitz 326, 362
Lizza 7, 12, 101, 118
Logan 213, 249
Long 4, 6, 18, 24, 114, 180, 213, 215, 230, 250
Longinow 103, 118
Lowdon 228, 230, 247
Loyall 359
MacGregor 74, 100, 114
Machol 205, 248
MacLeod 255, 311, 364
MacMillan 360, 363
Maes 327, 362
Mainzer 103, 117
Malik 307

Malin 133, 148, 162, 165
Malinowski 360
Malsch 144, 165
Marks 213, 250
Marr 213-214, 249
Marsh 359-360
Marshak 104, 113
Martin 179, 258, 281, 308
Mathews 364
Mavor 347, 361, 363
McCall 9, 12
McCallum 309
McClelland 100, 114
McCormick 205, 249
McCoy 149, 166, 292, 308
McDermott 362
McFarren 102, 116
McKee 144, 165
McKinley 322, 361
McMillan 209, 249, 295, 308, 311
McNeese 51-52, 75, 77, 79, 82-85, 88-91, 95, 99-109, 111, 113-118, 254, 256, 309, 315, 317, 326-328, 360-364
McNeill 221, 249
Meister 205, 249
Menard 106, 117
Milgram 211, 248, 268, 308
Militello 49-50, 74-75
Miller 7, 12, 74, 99, 116, 253, 258-259, 261-262, 266, 274, 288, 290-293, 300, 303, 307, 309, 363
Milton 253, 274, 310, 363
Mineka 316, 362
Mitchell 213, 248, 327, 362
Montecalvo 303, 307, 309
Moran 21-22, 73, 84, 90-91, 93, 113
Morris 100, 117, 138, 165
Muir 7, 12
Mulgund 327, 329-330, 362
Multer 144, 165
Mumaw 49, 76, 142, 145-146, 154, 165, 290, 310
Mumpower 100, 114
Mycielska 94, 117
Nagaraja 307

Naikar 169, 174, 176-177, 179, 184, 188-189, 191, 193-194, 197-199, 271, 310
Narayanan 104, 107, 109, 113, 115
Neisser 22, 75, 213, 249
Nelson 308
Newell 21-22, 25, 66, 73, 75, 84, 90-91, 93, 113, 231, 249
Nick 327-328, 363
Nordo 247
Norman 18-19, 28, 40-41, 43-45, 74-75, 84-85, 91-94, 115-116, 254, 309, 326, 362
Norris 213, 248
Nosek 108, 116
Oberman 247
O'Hara 282, 309
Onken 257-258, 309-312
Orasanu 75, 90, 96, 114
Orr 303, 307, 309, 323, 359, 362
Ortony 320, 328, 362
Panksepp 363
Pape 106, 117
Papin 247
Paris 213, 250
Pastor 308
Patterson. 60, 77, 134, 164
Pawlak 58, 75
Pearce 188, 198
Peio 102, 116
Pejtersen 76, 96, 117, 142, 165, 169-170, 196, 199, 238, 250, 255, 309, 316, 363
Pelican 259, 261-262, 266, 288, 290-293, 309
Persons 363
Perusich 89, 95, 104-106, 108, 115-116, 328, 362
Peterson 322, 363
Petrushin 328-329, 363
Pew 99, 116, 347, 361, 363
Phillips 240, 249
Picard 321, 323, 328-329, 359, 361-363
Pleydell-Pearce 253, 273, 276-277, 309, 363
Pliske 74-75
Pohl 327-328, 363
Pokorny 52, 74
Polson 24, 73-74
Pongratz 308

Porges 307
Porlier 213, 248
Potter 52-53, 55, 57-58, 60, 69, 75-76, 84, 107-109, 116, 134, 137, 140, 142, 144, 147, 149, 164-165
Powers 202, 209, 211, 232-234, 237, 240, 249, 268, 300, 309
Previc 213, 250
Raeth 7, 12
Rasmussen 18-20, 32-33, 36-40, 48-49, 58-60, 64, 67, 69, 75-76, 84-85, 91, 96-97, 99-100, 103, 107, 117-118, 139, 141-142, 148, 165-166, 169-170, 173, 178, 184, 188, 196, 199, 238, 243, 250, 254-255, 268, 270-271, 309, 316, 363
Reason 2, 9, 21, 29, 51, 64, 70, 84, 87, 94, 111, 117, 138, 180, 232, 275, 277, 280, 360
Redden 303, 307, 309
Redding 195, 198-199
Reed 213-214, 250
Reilly 359
Reinecke 12, 257, 307
Reis 363
Reising 1, 7, 12, 148, 165, 257-258, 261, 268, 307, 309-312, 317, 363
Remington 73
Rentsch 89, 95, 104, 106-108, 115-117, 328, 362
Resnick 89, 117, 326, 363
Retzlaff 322, 344, 360, 363
Rieser 95, 113
Riley 7, 12, 292, 310
Rinkus 362
Roe 308
Roediger 213, 250
Rogoff 91, 117
Rolek 303, 307, 309
Rood 308
Ross 77
Rosson 149, 164
Roth 27, 49, 58, 60, 64, 75-77, 84-85, 94, 107, 116, 118, 134, 137-140, 142, 144-146, 148-149, 154, 162, 164-166, 290, 310
Rothrock 59, 76
Rouse 7, 12, 74, 90, 100, 117, 138, 165, 257, 259-260, 310, 327, 363
Rushton 213, 250

Ryder 77
Salas 77, 115, 165, 209, 249, 360
Sanderson 56-57, 59, 72, 76, 84-85, 91, 101-103, 108, 117, 148, 165, 169, 174, 176-179, 183-184, 188-189, 191, 197-199, 230, 247, 256, 271, 310
Sandry 351, 364
Sarter 77, 94, 118, 126, 134, 149, 166, 364
Saul 307
Schank 235, 250
Scherer 320, 363
Schiflett 360
Schmidt 150, 329, 361
Schmitter-Edgecombe 213, 250
Schneider-Hufschmidt 360
Schraagen 53, 75-76, 134, 140, 165, 246, 248
Schreckenghost 148, 162, 165
Schreiber 256, 282-283, 310
Schwartz 77
Scott 84, 103, 107, 117, 137, 230, 247, 359
Schuffle 247
Seamster 195, 198-199
Seidler 101, 117
Selcon 258, 311
Self 204, 250
Selvaraj 103, 113, 118
Serfaty 325, 360-361, 363
Shadbolt 49, 74, 253, 274, 282, 309-310, 363
Shadrake 258-259, 311
Shafto 60, 73
Shalin 53, 75-76, 134, 140, 165, 246, 248
Shannon 22, 76
Shattuck 164, 292, 310
Shaughnessy 213, 249
Shaw 308, 359
Sheridan 91, 117, 188, 199, 300, 309-310
Sherwood 95, 113
Shortliffe 87-88, 113
Simon 22, 25, 66, 75-76, 101, 114, 138, 165, 213, 231, 249-250
Sinclair 228, 237, 247
Sipes 250
Skilton 56, 76, 256, 310
Small 7, 12, 80, 107, 161, 183, 195, 258, 325, 351, 369

Smith 117, 149, 166, 213, 247, 292, 307-308, 310
Snyder 51-52, 73, 77, 83, 91, 101-102, 104, 108, 114, 116-118, 308, 327, 361, 364
Solso 213-214, 250
South 72, 76, 158, 308
Staveland 278, 308
Sternberg 213, 250
Stone 307
Streets 247
Strub 249
Stubberud 240, 247
Suchman 89-91, 117
Sugibayashi 144, 165
Sutton 249, 316, 362
Sykes 217, 250
Szczepkowski 77
Talleur 196, 198
Tanabe 58, 77
Taylor 12, 74, 77, 117, 247, 253, 255, 257-259, 261, 264-266, 268-269, 274, 295, 298, 300, 306-312, 317, 327, 363
Teasley 89, 117, 363
Tellegen 322, 363
Tennison 253, 273-274, 310-311, 363
Thomas 73, 144, 165, 198, 359
Thordsen 90, 117
Thorndsen 303, 309
Tindale 90, 114
Tremblay 269, 311
Tversky 325, 363
Van Breda 249
Van Cott 205-206, 250
Van de Vegte 211, 250
Van De Velde 310
Van Der Molen 307
Van Heijst 282, 309
Verplanck 188, 199, 300, 310
Vicente 10, 12, 20, 33-34, 36, 38, 40, 48, 58, 66, 73-77, 85, 91, 94, 117-118, 139-140, 142, 147-148, 164, 166, 169-170, 175, 178, 183-184, 196, 198-199, 203-204, 208-209, 218-219, 238-244, 250, 255, 266, 271, 311, 364
Vidulich 295, 311, 351, 362-364
Vollrath 90, 114
Vye 95, 113

Vyzas 323, 363
Waite 196, 198
Walters 144, 164
Warren 99, 116
Wason 28, 77
Watanabe 103, 117
Watson 230, 247, 257, 265, 311, 322, 363
Watts 134, 364
Weaver 22, 76
Wegner 90, 118
Wehrle 329, 361
Weick 326, 364
Weiland 77
Wellens 101, 108, 117-118, 325, 361, 364
Whitaker 3, 11, 75, 83, 91, 103, 108-109, 113, 116, 118
Whitecross 253, 363
Wickens 5, 12, 351, 364
Wielinga 310
Wiener 87, 118
Wilensky 238, 250
Williams 240, 247, 249, 307, 316, 341, 364
Wilson 101, 108, 118
Wolf 303, 309
Wood 74, 107
Woods 18-21, 26-27, 29-30, 60, 73-75, 77, 84-85, 88, 91, 93-96, 101, 107, 109, 114, 116, 118, 121, 126, 133-134, 138-140, 142, 144, 147-149, 164-166, 254, 308, 316, 326, 361, 364
Woodson 99, 116, 205, 250
Woodworth 104, 113
Yauch 213, 250
Yingling 3, 11
Young 3, 11, 88, 90-91, 95, 101, 108, 118, 326, 364
Zabowski 362
Zacharias 347, 359, 361-362, 364
Zachary 48, 50-51, 58, 77
Zaff 51, 75, 77, 83-85, 91, 101-104, 108, 113, 116-118, 327, 364
Zanconato 258, 282, 312
Zenyuh 7, 12
Zhang 28, 77
Zsambok 75, 88, 90-91, 114, 119, 326, 361-364

Subject Index

Abstraction-Decomposition
171, 173, 176, 266, 297
Abstraction Hierarchy
32-34, 48, 58, 65, 69, 73, 103, 107, 139, 141-142, 151-153, 162, 164, 171, 173, 180-181, 183, 189-194, 218-219, 236, 241, 243-244, 367-368
Action
22-23, 36, 40, 42, 44, 50-52, 54, 64, 67-68, 75, 92-94, 96, 114, 116, 138, 140, 143, 150, 170-171, 188, 191, 195, 197, 202, 215, 217, 226, 228, 232-236, 240-241, 244-245, 259, 266, 269, 286, 288, 300-301, 316, 325, 336, 340, 346, 369
Activating Knowledge
95
Activity Analysis
188, 197, 272
Activity Context
184-185
Adaptation
72, 85, 128, 138-139, 170, 215, 233, 244, 257, 260, 266, 299, 304, 306, 311, 315-318, 321, 327-335, 342-344, 347-358, 362
Adaptive Automation
254, 258-259, 264, 268, 287-288, 302, 306-307
Adaptive Interface
246, 268, 275, 315, 317, 327, 356, 361, 367
Adaptive Methodology
315, 317-318, 330-331, 335, 352, 354-357
Advanced Knowledge and Design Acquisition Methodology (AKADAM)
75, 91, 101-104, 108, 116, 367
Aiding Concept
149-150
Affect and Belief Adaptive Interface System (ABAIS)
315, 317-326, 330-333, 335-340, 342, 344-350, 352-359, 361, 367
GUI Adaptation
315, 318, 331, 349-350, 353, 356
DSS Adaptation
330, 333, 349

Impact Prediction
318, 330-331, 338-340, 343, 348-349, 358
Strategy Selection
318, 330-331, 349, 358
User State Assessment
315, 318, 330-331, 342, 345
Affective States
315-317, 320-326, 329, 331, 336-337, 341, 347-349, 356, 358
Affordances
39, 59, 84, 90-91, 101, 178
Aircraft
1-2, 4-6, 8-9, 11, 58, 73, 88, 96, 113, 118, 169, 179, 184-185, 188, 191, 193, 198, 217, 226, 257-260, 264-267, 273-275, 277, 279-283, 287-288, 290, 297, 302-303, 308, 330-334, 337, 339-340, 345-348, 352, 368
Airborne Early Warning and Control (AEW&C)
169, 174, 179-186, 188, 271, 367
Airborne Warning and Control System (AWACS)
75, 81, 89, 95, 334, 340, 367
C-130
6
Eurofighter
257-258, 297
F-4 Phantom
6
F-15E Strike Eagle
6
F-22
110
F/A-18
163, 172-173, 185-188
Fighter
9, 84, 101, 104, 113, 292, 303, 309, 318, 333, 344-346, 353
Future Carrier-Borne Aircraft (FCBA)
257, 282-283, 368
Future Offensive Air System (FOAS)
254, 257, 266, 282-283, 292, 297, 302, 306, 368
Future Organic Airborne Early Warning (FOAEW)
281-282, 284, 307, 368
GR-1 Tornado
6

Applied Cognitive Task Analysis (ACTA)
49-50, 75, 198-199, 307, 367
Artificial Intelligence (AI)
19, 26, 87, 93, 99, 113, 230, 247, 257, 260, 367
Assessment of Affective States
321
Automation
1-5, 7, 11, 26, 88, 99, 116-117, 132, 173, 227, 242, 254-255, 258-262, 264, 266-268, 270-272, 274-275, 280, 287-293, 295-296, 300, 302-304, 306-307, 309-311
Automation Philosophy
2-5
Autopilot
4
Belief States
315-318, 320, 322, 324-326, 328, 330-332, 334-338, 340, 342, 344, 346, 348-350, 352, 354-358, 360, 362, 364, 366
Biases
109, 145, 215, 315, 317, 324-326, 331, 337, 340, 344, 349, 357, 362-363
COGENT
91, 99, 102-103, 108, 117
C3 Operator Performance Engineering (COPE)
3, 7, 28, 95, 100, 104, 240, 255, 367
Closed-Loop Negative Feedback Control
240
Cockpit Assistant System
250
CAMMA
258
CASSY
258
Cockpit Task Adaptation
257
Cogminutia Fragmentosa
1-2, 5, 11, 80, 362-363
Cognition in the Wild
90-91, 94, 114, 326, 361
COGPIT Simulation Test Environment (COGSIM)
275, 284, 293-296, 300, 303, 367

Cognition-in-Context Framework
27
Cognition Monitor (COGMON)
273-280, 287-288, 293, 297, 301-303, 306, 309, 367
Cognitive Affective Personality Task Analysis (CAPTA)
318, 334-341, 347, 349, 357-359, 361, 367
Cognitive Analysis
48, 54, 76, 137-138, 140-150, 152, 154, 156, 158, 160, 162-164, 166, 168, 341, 360
Cognitive Cockpit (COGPIT)
253-254, 256-258, 260, 262-276, 278, 280, 282, 284-288, 290-298, 300, 302, 304, 306-308, 310-312, 314, 363, 367
Cognitive Compatibility
258, 261, 311
Cognitive Design
110, 254-257
Cognitive Engineering (CE)
1-2, 4, 6, 8, 10, 12, 14-16, 57, 74-77, 80, 82-84, 86-89, 94-96, 103, 105-106, 114-118, 137-138, 140, 142, 144, 146, 148, 150, 152, 154, 156, 158, 160, 162, 164-166, 168, 256, 258, 259, 264, 266, 269, 293, 309, 316, 318, 320, 322, 324, 326, 328, 330, 332, 334, 336, 338, 340, 342, 344, 346, 348, 350, 352, 354, 356, 358, 360, 362-364, 366
Cognitive Field Studies
81, 107, 109
Cognitive Modeling
22, 81, 98, 107-108, 110, 112-113, 164, 256, 317
Cognitive Network of Tasks (COGNET)
50-51, 295, 367
Cognitive Representation
96
Cognitive Schemata
316, 341
Cognitive Science
18, 21, 75, 77, 81, 91, 98-99, 101, 112, 320, 359
Cognitive Systems
1-2, 11, 15-44, 46, 48, 50, 52-56, 58, 60, 62, 64, 66, 68-86, 88, 90-100, 102, 104, 106-110, 112, 114, 116-118, 120-123, 132-134, 163-165, 170, 199, 203, 238, 250, 254, 256, 259, 268, 277, 304, 308, 315-316, 318, 336-338, 354, 357, 359, 361-364, 367

Cognitive Systems Engineering (CSE)
1-2, 11, 15-81, 88-105, 107-117, 121-123, 133, 163-165, 170, 199, 203, 238, 250, 254, 256, 259, 304, 308, 315-316, 318, 326-327, 354, 359, 361-363, 367
Analysis of Cognitive Activities
97, 101
Engineering of Cognitive Systems
83, 97
First-of-a-Kind Artifacts
97-98
Work in Context
26, 97, 99
Cognitive Task Analysis (CTA)
8, 11-12, 16-17, 46, 48-49, 52, 54, 74-77, 90-91, 94, 96, 102, 104, 107, 112, 114, 134, 137, 140, 164-165, 198-199, 203-204, 246, 248, 256, 295, 310, 318, 321, 324, 331, 334-335, 347, 367
Cognitive Tools
26, 85, 94-95
Cognitive Triad
27-28, 30, 40, 48
Cognitive Work Analysis (CWA)
10, 12, 55, 76, 81, 94, 117, 142, 166, 169-182, 184, 186, 188, 190, 192, 194-196, 198-200, 203-204, 230, 238-239, 241-247, 250, 256, 271-272, 304, 310-311, 364, 367
Cognitive Work Domains
142
Collaboration
84-85, 88-90, 103, 105, 113, 118, 125, 133, 144-145, 150, 166, 265-266, 281, 295, 364
Collaborative Design Technology Laboratory (CDTL)
103, 367
Collaborative System
100
Collaborative Work Settings
95
Combat Power
140, 150-162
Command and Control (C2)
9, 89, 95, 100, 115, 143-144, 151-152, 164, 169, 179, 197, 228-229, 247, 334, 360, 367

Command, Control, and Communication (C3)
100, 268, 367
Command, Control, Communication, and Intelligence (C3I)
89, 104, 107, 367
Commercial Aircraft
2
Complementarity
121-123, 125-127, 132-133
Component Processes
206
Computer-Aided Cognitive Systems Engineering (CACSE)
55-58, 72, 75, 107-109, 116, 367
Computer-Supported Collaborative Work (CSCW)
133, 367
CommonKADS
256, 283-284, 286, 304, 306, 310
Concept Mapping
51, 102, 104-105, 256
Concept Maps
90, 102, 104-105, 327, 335
Context-Sensitive Aiding
253-254, 256-258, 260, 262, 264, 266, 268, 270, 272, 274, 276, 278, 280, 282, 284, 286, 288, 290, 292, 294, 296, 298, 300, 302, 304, 306, 308, 310, 312, 314
Contextual Constraints
316
Control Loops
202, 211, 216-217
Constraints
1, 21, 24, 27-28, 32-33, 35, 37-38, 51, 59, 67, 80, 84, 89, 95, 104-110, 125-126, 128, 148, 152-153, 162, 170-171, 173, 175, 178, 184-185, 189, 191, 204-206, 211, 218, 220, 238, 241, 244-246, 255, 287, 292, 295, 316, 325, 356-358
Control Strategy (CS)
32, 37, 329, 367
Control Task (CT)
32, 35-37, 39, 74, 171, 173, 175-179, 184-185, 188, 199, 241-243, 249, 256, 271, 299, 367, 369
Control Task Analysis (CTA)
171, 173, 175-179, 184, 199, 241-243, 256, 271, 367, 369

Copilote Electronique
367
Critical Decision Method (CDM)
49-50, 74, 91, 114, 367
Cues-Situation Mapping
338-339
DARPA Command Post of the Future
150
Decision Aids
12, 100, 138, 150, 162-163, 227, 259, 266, 296, 310
Decision Decomposition Protocol (DDP)
50-57, 368
Decision Ladder
36-38, 90, 173, 188, 242-243, 271
Decision Requirements (DR)
50, 56, 140, 146, 154, 359, 368
Decision Support
18, 26, 30, 48-50, 54, 60, 70, 75, 77, 105, 114, 116, 138, 140-141, 145, 148, 150, 155, 163-164, 226, 228, 253-254, 256-260, 262, 264-266, 268-270, 272, 274, 276, 278, 280-284, 286, 288, 290, 292, 294, 296, 298, 300, 302, 304, 306-308, 310, 312, 314, 317, 331, 361, 363, 368
Decision Support System (DSS)
70, 138-139, 150, 154, 157, 161, 163, 281-283, 312, 330-331, 333, 349, 352, 354-356, 368
Defensive Aids Suite (DAS)
284, 302-303, 367
Display-Group Interface
108
Distributed Cognition
85, 94, 121-122, 326
Distributed Innovation
121-122, 133
Display Task Description (DTD)
56, 148-150, 155-156, 161, 368
Dynamic Worlds
94-95
Ecological Interface
58, 75-76, 91, 118, 148, 150-151, 153, 155, 157, 159, 161-162, 165-166, 199, 256, 368
Ecological Interface Design (EID)
58, 75-76, 91, 118, 148, 165-166, 199, 256, 368

Ecological Psychology
90-93, 96, 113-114, 166
Effort-after-Coordination
129-130
Electronic Associate (EA)
1, 5-11, 368
Embedded Software Mediators
106
Engine of Frustration
131
Engine of Innovation
130
Balanced Cycles
130
Envisioned World Problem
110, 127-128, 149, 164
Modes of Operation
128
Plurality
127-128
Underspecification
128
Epistemological Engineering
87
Feedback
73, 93, 105-106, 109, 145, 151, 202, 208, 215, 219, 227, 230-231, 240, 244, 247, 249-250, 266, 269, 272, 278, 287, 295, 299-300, 306, 348, 357, 359
Fighter Pilot
84, 318, 333, 344, 346, 353
Flight Management System (FMS)
226, 368
Fly-by-Wire
4
Force Wheel Steering
4
Function Allocation
4, 11, 201-202, 247, 258, 268, 308, 310
Functional Abstraction Hierarchy (FAH)
48-49, 56, 58, 64, 69, 139, 141-145, 147-148, 150-155, 162-163, 368
Functional Decomposition
102, 204, 209, 256, 291, 298, 304

Functional State Assessment
253-254, 256-258, 260, 262, 264, 266, 268, 270, 272, 274, 276, 278, 280, 282, 284, 286, 288, 290, 292, 294, 296, 298, 300, 302, 304, 306, 308-310, 312, 314, 363
Fuzzy Cognitive Map (FCM)
104-106, 368
Fuzzy Controllers
240
Genotypes
16-17, 20-21, 23, 25, 27, 29, 31-33, 35, 37, 39-41, 43-48, 50, 54, 57, 66, 71
Carnegie Mellon University (CMU)
20-21, 25, 30-33, 39-40, 43-45, 48, 54, 59, 367
Ohio State University (OSU)
20-21, 26, 28-32, 39-40, 43-45, 48-50, 52, 54-55, 369
RISO National Laboratory (RISO)
18, 20, 31-33, 35-36, 38-39, 43-45, 48-49, 51, 54-59, 61, 64, 67, 70, 72
University of California at San Diego (UCSD)
20, 40-45, 54, 57-58, 75, 369
Global Hawk
8
Goal-Plan Tracking (GPT)
228, 258, 291, 368
Goals, Operators, Methods, and Selection (GOMS) Framework
21-25, 31, 40-41, 44, 47-48, 51, 57, 59-61, 63-66, 72, 74, 84, 367-368
Goal Structure
50, 217-218, 228
Graphical User Interface (GUI)
290, 292-293, 315, 318, 330-334, 349-356, 368
Hard Systems Thinking
204
Heads-Up-Display (HUD)
332, 334, 340, 351-352, 368
Helicopter Aircrew Decision Support (HADS)
281, 284, 368
Hierarchical Decomposition
201-202, 217, 241-242
Hierarchical Goal Analysis (HGA)
201-202, 209, 217-219, 238-239, 241-245, 368
Hierarchical Task Analysis (HTA)
61-65, 238, 368

Hierarchies
90, 144, 219, 230, 233-234, 236, 269, 327
Human-Centered Automation
5
Human-Computer Interface (HCI)
47, 61, 70, 80, 99, 104, 164, 359, 368
Human-Computer Interface Mil-Prime Standard
99
Human-Electronic Crewmember (HEC)
257-258, 264, 368
Human Engineering
86, 88, 183, 197, 201-202, 204-206, 216, 241, 245, 247, 249-250, 266
Human Error
77, 85, 94, 117-118, 255, 270, 317, 364
Human Factors Engineering (HFE)
16, 18, 22, 47, 84, 86-87, 112, 203-205, 208, 216, 245, 368
Human Systems Analysis
203-205, 207, 238-239, 245, 247, 304
Human Systems Integration (HSI)
184-186, 188, 208-209, 245-246, 368
IEEE-1220 (*Application and Management of the Systems Engineering Process*)
248
IDEF
102, 368
Information Processing (IP) Model
211, 213-216, 245, 248
Information Requirements (IR)
37, 56, 77, 87, 140, 146-148, 150, 152, 154-155, 221, 297, 299, 310
Information Spaces
99
Information Work Spaces
169, 195
Integrated Performance Modeling Environment (IPME)
216, 220-222, 230, 243, 368
Intelligent Cockpits
101, 115, 260, 268, 307, 362
Joint Cognitive System
26, 268
Joint Operations Environment (JOE)
156, 158-162, 164, 368

Knowledge Acquisition (KA)
56, 77, 88, 90, 95, 100, 104, 118, 146, 151, 195, 197, 256, 258, 282, 284-286, 290, 302-306, 309, 364, 368
Knowledge-as-Design
83
Knowledge-Based Assessment
321, 324, 343
Knowledge Engineering
73, 86-88, 93, 101, 111, 114, 256, 283, 304, 306, 310
Knowledge Model
138-139, 283-284
Knowledge Representation
104, 118, 195, 197, 363
Knowledge-Based Systems (KBS)
254, 258, 264, 270-272, 274, 281, 283, 304, 306, 310, 358, 368
Land Forces Tactical Battlefield Command System (TBCS)
228-229, 369
Large-Group Display
100
Laws of Robotics
6-7
Level of Adaptation
351
Display Level
351
Icon Level
351
Notification Level
351
User Interface Level
351
Living Laboratory
100, 109, 115
MacSHAPA
91, 103, 108, 117
Mental Models
30, 75, 100, 117, 138, 165, 211, 215, 244, 256, 326-327, 361-362
Mental Simulation
49, 84, 96, 114
MIL-HDBK-46855A (*Human Engineering Program Process and Procedures*)
249

Mission, Function, Task Analysis (MFTA)
205, 208-209, 216, 219, 226, 228-230, 238-242, 244-245, 368
Mis-synchronized Cycles
129
Model Human Processor (MHP)
21-22, 25, 73, 368
Model of Support
148
Monte Carlo Simulation
180
Naturalistic Decision Making (NDM)
48, 50, 74, 87, 90-91, 96, 114, 119, 121-122, 139, 165, 308, 326, 361-364, 369
Operational Event Sequences
206
Perceptual Control Theory (PCT)
201-248, 250, 252, 268, 300, 308, 369
Physiological Data
323, 325, 345, 356
Physiological Sensing
315, 318, 321-324, 329, 345, 359
Pilot's Associate (PA)
7-8, 12, 75, 91, 101, 108, 116, 165, 258, 288, 309, 327, 362, 369
Pilot Control Force Steering
4
Pilot State
273-275, 278-280, 288-289, 345, 348
Practice-Centered Research and Design
121-122, 124, 126, 128, 130, 132-134, 136
Precursor, Action, Result, and Interpretation (PARI) Method
52, 74, 369
Predator
8
Principles of Interaction
7
Problem Driven
94
Protocol Analysis
90, 101, 103, 109, 114, 117, 357
Rapid Prototype
156, 162

Recognition-Primed Decision-Making (RPD)
114, 325, 361, 369
Recognition Model of Decision Making
96
Reference Signal
231-233, 235, 237
Research and Development (R&D)
3, 11-12, 19-20, 73-75, 77, 108, 113-118, 121-122, 126, 129-130, 134, 164-166, 247-250, 307-311, 360-364, 369
Robot
6, 11
Rotorcraft Pilot's Associate (RPA)
8, 258, 288, 295, 307, 309, 327, 369
Rule-Based Reasoning
331, 348, 369
Self-Reports
315, 318, 322, 328-329, 332, 342
Seven-Stage Model
40-44
SHAPA
101, 103, 117
Shared Task Model
287, 290, 292, 306
Simulation
49, 51, 73, 84, 96, 104, 114, 147, 176, 180, 220-221, 225, 248, 275, 284, 293, 295-297, 299, 301, 303-304, 307, 318, 332, 344-345, 352, 356, 358, 367
Situated Context
88, 106
Situation-Action Mappings
338-339
Situation Assessment
49, 144, 254, 257-259, 271, 274, 280-282, 300, 311, 316-317, 325-326, 336, 338-339, 341, 346-348, 356, 369
Situation Assessment Support System (SASS)
274, 280, 311, 369
Situation Awareness (SA)
9-10, 30, 50, 99, 106, 115, 165, 175, 215, 220-221, 225, 243, 248, 250, 258, 264, 266, 275, 279-280, 295, 307, 311, 320, 325, 330, 346, 360, 364, 369

Skill-Rules-Knowledge (SKR) Taxonomy
36, 38, 269-271, 369
Social-Organizational Analysis (SOA)
167, 175, 369
Social/Organizational (SO) Framework
1, 4-5, 32, 42, 48, 55, 85, 91, 93, 108, 110, 142-143, 150, 178, 180, 188, 193, 202, 215, 221, 230, 233, 236, 238-239, 255, 261, 263, 266, 268, 277, 296, 299, 304, 306, 344, 369
Storyboarding
102
Strategies Analysis
173, 175-177, 197, 369
Strategy Selection
318, 330-331, 349, 358
Synchronization
121-122, 130, 132-133, 225
Systems Analysis
16, 73, 203-207, 209, 215-217, 219, 221, 223, 227-231, 233, 235, 238-239, 242, 244-247, 304
Systems Engineering
1-2, 11, 15-44, 46-48, 50, 52-56, 58, 60, 62, 64, 66, 68-81, 83-86, 90-94, 96, 98-99, 104, 107-108, 112, 114, 116-117, 121-123, 133, 163-165, 170, 180, 199, 203-206, 208, 230, 238-239, 241, 244, 246-248, 250, 254-256, 259, 304, 308, 315-316, 318, 354, 359, 361-363, 367
System Functions
4, 177, 206, 259-260, 273
System Life Cycle
174-175, 177, 199, 310
TAKE
4, 6, 9, 19, 38, 47, 53, 71, 80-81, 84, 89-91, 95, 102-103, 107, 139, 142, 153, 159, 171, 207, 213-214, 230, 236, 260, 268, 273, 290-291, 317, 325, 331, 340, 351
Task Analysis
8, 11-12, 16-17, 46-49, 52-54, 61-64, 74-77, 90-91, 94, 96, 102, 104, 107-108, 112, 114-115, 134, 137, 140, 145, 164-165, 171, 173, 175-179, 184, 198-199, 201-205, 208-209, 219, 238-239, 241-243, 245-246, 248, 256, 271, 290, 295, 308, 310, 318, 321, 324, 331, 334-337, 339, 341, 347, 361, 367-369

Task Network Modeling
102, 104
Task Networks
102
Tasking Interface Manager (TIM)
274, 280, 287-293, 297-303, 306-307, 359, 369
Team Analysis
100
Teamwork
4-5, 103, 106-108, 110, 115-116, 188, 257-258, 264, 268, 309-312, 325, 362
Team Coordination
96, 207
Team Error Correction
208
Team Maintenance
208
Temporal Coordination Control Task Analysis (TCCTA)
173, 178, 184, 199, 369
Three Mile Island
18
Training System Design
198
Trust
6-7, 12, 208, 257-258, 292, 311
Unmanned Air Vehicles (UAV)
88, 97, 104
Unmanned Combat Aerial Vehicle (UCAV)
1, 6, 8-10, 73, 369
User-Centered Design
43, 81, 307
User-Centered Knowledge
88, 102
Visualization
58, 138, 140-141, 145, 148-150, 153-158, 162-163, 165
Weapons System Officer (WSO)
369
Work Domain (WD)
16, 25, 32-36, 38-39, 42-45, 47-49, 54, 56, 58-59, 61-62, 64-67, 69-70, 72, 74, 76, 97, 103, 138-139, 142-145, 148-150, 155, 162-163, 170-171, 173-180, 183-

186, 188-189, 191, 193-198, 241, 243-244, 246-247, 256, 310, 369
Work Domain Analysis (WDA)
56, 74, 76, 103, 142-143, 162, 171, 174-180, 183-186, 188-189, 191, 194-196, 198, 241, 256, 310, 369
Functional Purposes
171, 176-177, 183, 191
Physical Form
33, 35, 67, 171, 176, 189, 193
Physical Functions
35, 65, 171, 180, 188, 193-194
Priorities and Values
171, 176, 180, 183, 185, 191, 193
Purpose-Related Functions
171, 176, 183, 185, 193
Work Domain Analysis Workbench (WDAW)
56, 76, 310, 369
Work Domain Representation
97
Worker Competencies (WC)
32, 38-39, 173, 175-177, 369
Worker Competencies Analysis (WCA)
173, 175-177
Workload
4, 8, 12, 87, 205, 207, 214, 220, 229, 241-242, 248, 255, 258-259, 261-262, 272-273, 275-279, 282, 292, 299, 301, 308, 322, 324, 327, 330