

Electronic Theses and Dissertations, 2004-2019

2011

Development Of A Cognitive Work Analysis Framework Tutorial Using Systems Modeling Language

Wilfred Henry Wells
University of Central Florida

 Part of the [Industrial Engineering Commons](#)
Find similar works at: <https://stars.library.ucf.edu/etd>
University of Central Florida Libraries <http://library.ucf.edu>

This Doctoral Dissertation (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Electronic Theses and Dissertations, 2004-2019 by an authorized administrator of STARS. For more information, please contact STARS@ucf.edu.

STARS Citation

Wells, Wilfred Henry, "Development Of A Cognitive Work Analysis Framework Tutorial Using Systems Modeling Language" (2011). *Electronic Theses and Dissertations, 2004-2019*. 1727.
<https://stars.library.ucf.edu/etd/1727>

DEVELOPMENT OF A COGNITIVE WORK ANALYSIS FRAMEWORK TUTORIAL USING SYSTEMS MODELING LANGUAGE

by

WILFRED HENRY WELLS

B.S. University of Central Florida, 1997

M.S. University of Central Florida, 2004

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Industrial Engineering and Management Systems
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Fall Term
2011

Major Professor: Waldemar Karwowski

ABSTRACT

At the present time, most systems engineers do not have access to cognitive work analysis information or training in terms they can understand. This may lead to a disregard of the cognitive aspect of system design. The impact of this issue is system requirements that do not account for the cognitive strengths and limitations of users. Systems engineers cannot design effective decision support systems without defining cognitive work requirements. In order to improve system requirements, integration of cognitive work requirements into the systems engineering process has to be improved. One option to address this gap is the development of a Cognitive Work Analysis (CWA) framework using Systems Modeling Language (SysML). The study had two phases. The first involved aligning the CWA terminology with the SysML to produce a CWA framework using SysML. The second was the creation of an instruction using SysML to inform systems engineers of the process of integrating cognitive work requirements into the systems engineering process. This methodology provides a structured framework to define, manage, organize, and model cognitive work requirements. Additionally, it provides a tool for systems engineers to use in system design which supports a user's cognitive functions, such as situational awareness, problem solving, and decision making.

This dissertation is dedicated to my mother, Clarice Wells in recognition of her unwavering support, sacrifice, guidance and love.

ACKNOWLEDGMENTS

This dissertation would not be possible without the contributions and support of others. First, I would like to express my gratitude to my advisor, Dr. Waldemar Karwowski, for his guidance, patience, insight, and encouragement during my entire study.

Second, I am grateful to the members of my doctoral dissertation committee. To Dr. Serge Sala-Diakanda, thank you for helping me utilize Systems Modeling Language for this study. To Dr. Ahmad Elshennawy and Dr. Kent Williams, thank you for serving on my committee and providing an academic challenge. To Dr. Tareq Ahram, thank you for being readily available to discuss ideas and answer questions.

Third, I would like to thank all my friends and family for their support and camaraderie, which kept me grounded and sane.

Finally, I would like to thank my wife, Kecia for proofreading and critiquing my writing. To my children Kayla, Isaiah, and Halle, thank you for your patience and understanding during the long hours that was necessary to complete this dissertation.

TABLE OF CONTENTS

LIST OF FIGURES.....	xii
LIST OF TABLES.....	xviii
LIST OF ACRONYMS/ABBREVIATIONS	xix
CHAPTER ONE: INTRODUCTION.....	1
1.1 Problem Description	3
1.2 Purpose of the Study	5
1.3 Significance of the Study	5
CHAPTER TWO: LITERATURE REVIEW	6
2.1 Systems Engineering.....	6
2.1.1 Introduction	6
2.1.2 The “V” Model	8
2.1.2.1 Requirements Analysis	10
2.1.2.2 Functional Analysis.....	10
2.1.2.3 Component Design	11
2.1.2.4 Implementation	12
2.1.2.5 Integration.....	12

2.1.2.6 System Verification	13
2.1.2.7 System Validation	13
2.1.3 The Three Evils of Systems engineering.....	14
2.2 Cognitive Work Analysis	15
2.2.1 Introduction	15
2.2.2 Work Domain Analysis	17
2.2.3 Control Task Analysis	20
2.2.4 Strategies Analysis.....	24
2.2.5 Social Organization and Cooperation Analysis	26
2.2.6 Worker Competencies Analysis	27
2.2.7 Current CWA Tools	28
2.2.8 Cognitive Work Analysis Summary	28
2.3 Systems Modeling Language	30
2.3.1 Introduction	30
2.3.2 SysML Description	33
2.3.2.1 Structure	36
2.3.2.1.1 Block Definition Diagram.....	36
2.3.2.1.2 Internal Block Diagram.....	39

2.3.2.1.3 Package	41
2.3.2.2 Behavior	42
2.3.2.2.1 Sequence Diagram	42
2.3.2.2.2 Activity Diagram	44
2.3.2.2.3 State Machine Diagram.....	47
2.3.2.2.4 Use Case Diagram.....	49
2.3.2.3 Requirements Diagram.....	50
2.3.2.4 Parametric	53
2.3.3 SysML Elements for Cognitive Work Analysis.....	54
2.3.3.1 SysML Elements for Work Domain Analysis.....	54
2.3.3.2 SysML Elements for Control Task Analysis	56
2.3.3.3 SysML Elements for Strategies Analysis	59
2.3.3.4 SysML Elements for Social Organization and Cooperation Analysis.....	60
2.3.3.5 SysML Elements for Worker Competencies Analysis.....	61
2.3.4 SysML Requirement Development Function	64
2.4 Applications of System Modeling Language	66
2.5 Cognitive Work Analysis Framework Modeled using SysML	73
2.5.1 Cognitive Work Analysis Structure Modeled using SysML.....	74

2.5.2 Work Domain Analysis Modeled using SysML	74
2.5.3 Control Task Analysis Modeled using SysML	78
2.5.4 Strategies Analysis Modeled using SysML.....	81
2.5.5 Social Organization and Cooperation Analysis Modeled using SysML	84
2.5.6 Worker Competencies Analysis Modeled using SysML	86
2.6 Literature Review Summary	89
CHAPTER THREE: METHODOLOGY	91
3.1 Development of Cognitive Work Analysis Tutorial using SysML.....	92
3.1.1 Step 1: Create an Outline.....	92
3.1.2 Step 2: Model Introduction Section	92
3.1.2.1 Overview of the Cognitive Work Analysis Framework	92
3.1.3 Step 3: Detailed Description of the CWA Framework.....	96
3.1.3.1 Align CWA Models to SysML Diagrams.....	96
3.1.4 Step 4: Construct CWA Process Flow Chart	98
3.1.4.1 Create CWA Process Flow Chart using Activity Diagram	98
3.1.5 Step 5: CWA Use Case.....	100
3.1.6 Step 6: Construct Template CWA Framework Model.....	100
3.2 Methodology Summary.....	102

CHAPTER FOUR: RESULTS	104
4.1 Cognitive Work Analysis Tutorial Introduction Section.....	104
4.1.1 Title Page.....	105
4.1.2 Outline.....	107
4.2 CWAT Introduction Section	107
4.3 CWAT Detailed Description Section	110
4.3.1 CWAT Sequence and Information Exchange.....	110
4.3.2 CWAT Abstraction Hierarchy	121
4.3.3 CWAT Decision Ladder.....	123
4.3.4 CWAT Information Flow Maps	126
4.3.5 CWAT Use Case.....	128
4.3.6 CWAT Skill, Rule, and Knowledge Inventory	130
4.4 Process Flow Charts Section.....	133
4.4.1 CWAT Process Flow for Constructing a CWA Structure	134
4.4.2 CWAT Process Flow for Constructing an Abstraction Hierarchy.....	140
4.4.3 CWAT Process Flow for Constructing a Decision Ladder	145
4.4.4 CWAT Process Flow for Constructing an Information Flow Maps.....	153
4.4.5 CWAT Process Flow for Constructing a Use Case	160

4.4.6 CWAT Process Flow for Constructing a Skill Rule and Knowledge Inventory	168
4.5 CWA Cognitive Factors Team	175
4.6 CWA Method Example for the Terminal Radar Approach Control System	181
4.6.1 Example CWA method introduction	181
4.6.2 Example of a CWA Structure	183
4.6.3 Example of a CWA Abstraction Hierarchy	185
4.6.4 Example of a CWA Decision Ladder	187
4.6.5 Example of a CWA Information Flow Maps	191
4.6.6 Example of a CWA Use Case	195
4.6.7 Example of a CWA Skill, Rule, and Knowledge Inventory	202
4.7 CWA Framework ATM Example	206
4.7.1 CWA Structure for ATM Example	208
4.7.2 Work Domain Analysis	209
4.7.3 Control Task Analysis	213
4.7.4 Strategies Analysis	218
4.7.5 Social Organization and Cooperation Analysis	221
4.7.6 Worker Competencies Analysis	224

4.8 Preliminary Usability Evaluation.....	228
4.9 Results Summary	232
CHAPTER FIVE: CONCLUSION	233
5.1 Summary of Study	233
5.2 Using SysML to Model a CWA Framework.....	234
5.3 Challenge to the Three Evils of Systems engineering	235
5.4 Future Research	236
5.4.1 Trade-off Analysis Capability.....	237
5.4.2 Human Systems Integration	239
5.5 Concluding Statement	240
APPENDIX : COPYRIGHT PERMISSION	241
REFERENCES.....	243

LIST OF FIGURES

Figure 1: Systems Engineering Process Model.....	9
Figure 2: SysML Architecture	34
Figure 3: A Block Definition Diagram with Directed Composition Relationship	38
Figure 4: A Block Definition Diagram with Flow Relationship	39
Figure 5: An Internal Block Diagram.....	40
Figure 6: A Package Diagram	41
Figure 7: A Sequence Diagram	44
Figure 8: An Activity Diagram.....	47
Figure 9: A State Machine Diagram	48
Figure 10: A Use Case Diagram	50
Figure 11: A Requirements Diagram	52
Figure 12: A Parametric Diagram	54
Figure 13: A Template Abstraction Hierarchy Constructed Using SysML	56
Figure 14: A Template Decision Ladder Constructed Using SysML.....	58
Figure 15: A Template Information Flow Maps Model Constructed Using SysML.....	60
Figure 16: A Template Use Case Diagram Constructed Using SysML	61
Figure 17: A Template Skill, Rule, and Knowledge-Based Inventory Constructed with SysML	63
Figure 18: An Example of a CWA Structure	76

Figure 19: An Abstraction Hierarchy Model Created using SysML.....	77
Figure 20: Internal Block Diagram of one General Function and three Control Tasks ..	79
Figure 21: A Decision Ladder Created using SysML.....	80
Figure 22: A Template of an SA Structure.....	82
Figure 23: A Template of an IFM.....	83
Figure 24: A Template of a Social Organization and Cooperation Analysis Structure...	85
Figure 25: A Template of a CWA Use Case Diagram	86
Figure 26: A Template for a Worker Competencies Analysis Structure	88
Figure 27: A Template of a Skill, Rule, and Knowledge Inventory Model.....	90
Figure 28: Cognitive Work Analysis Tutorial Outline	93
Figure 29: CWAT High-Level View.....	94
Figure 30: Internal Block Diagram Introduction to CWA.....	95
Figure 31: CWAT Framework Process Flow View	97
Figure 32: Work Domain Analysis Tutorial	98
Figure 33: Work Domain Analysis Process Flow Chart.....	99
Figure 34: Template CWA.....	101
Figure 35: Template Abstraction Hierarchy Model	102
Figure 36: Title Page of Cognitive Work Analysis Tutorial.....	106
Figure 37: CWAT Outline	109
Figure 38: CWAT Framework High Level View	110
Figure 39: Introduction to CWA in an Internal Block Diagram	111

Figure 40: CWAT Framework Process Flow View	113
Figure 41: A Model-Based and Text-Based Description of an Abstraction Hierarchy .	122
Figure 42: A Model-Based Description of a Decision Ladder	124
Figure 43: A Text-Based Description of a Decision Ladder	125
Figure 44: A Model Based and Text-basedd Description of an Information Flow Map	127
Figure 45: A Model Based and Text-basedd Description of an Use Case.....	129
Figure 46: A Model Based Description of the Skill, Rule, and Knowledge Inventory...	131
Figure 47: A Text-Base Description of the Skill, Rule, and Knowledge Inventory	132
Figure 48: CWAT Process Flow Overview	135
Figure 49: Process Flow Chart for Building a Cognitive Work Analysis Structure	136
Figure 50: Process Flow Chart Steps for Building a Cognitive Work Analysis Structure	137
Figure 51: A Process Flow Chart for Building an Abstraction Hierarchy.....	141
Figure 52: A Process Flow Chart Steps for Building an Abstraction Hierarchy	142
Figure 53: A Process Flow Chart for Constructing a Decision Ladder	146
Figure 54: Process Flow Chart Steps for Building a Control Task Analysis Structure .	147
Figure 55: Process Flow Chart Steps for Building a Decision Ladder	148
Figure 56: A Process Flow Chart for Constructing an Information Flow Map	154
Figure 57: Process Flow Chart Steps for Building a Strategies Analysis Structure	155
Figure 58: Process Flow Chart Steps for Building a Information Flow Maps.....	156

Figure 59: A Process Flow Chart for Constructing a Social Organization and Cooperation Analysis Use Case.....	162
Figure 60: Process Flow Chart Steps for Building a Social Organization and Cooperation Analysis	163
Figure 61: Process Flow Chart Steps for Building a Use Case	164
Figure 62: A Process Flow Chart for Constructing a Skill, Rule, and Knowledge Inventory using SysML.....	169
Figure 63: Process Flow Chart Steps for Building a Worker Competencies Analysis Structure.....	170
Figure 64: A Process Flow Chart Steps for Building a Skill, Rule, and Knowledge Inventory	171
Figure 65: A Use Case Diagram of the Education and Experience Requirements of a Cognitive Factors Team	177
Figure 66: Background Information of Terminal Radar Approach Control System.....	182
Figure 67: Cognitive Work Analysis Structure Model for TRACON System	184
Figure 68: An Abstraction Hierarchy Model of the TRACON System.....	186
Figure 69: An Example of a ConTA Structure for the TRACON System.	188
Figure 70: An Example of a Decision Ladder for the TRACON User.	190
Figure 71: A Strategies Analysis Structure for the TRACON System.....	192
Figure 72: An Information Flow Map Model of the User's Strategy to Hold One Aircraft	193

Figure 73: An Information Flow Map Model of the User’s Strategy to Reroute One Aircraft.....	194
Figure 74: An Information Flow Map Model of the User’s Strategy to Tweak One Aircraft	195
Figure 75: A Social Organization and Cooperation Analysis Structure for the TRACON System.	197
Figure 76: A Use Case Diagram Showing Allocation of Tasks Between TRACON User and the System for Holding One Aircraft.....	199
Figure 77: A Use Case Diagram Showing Allocation of Tasks Between TRACON User and the System for Rerouting One Aircraft.....	200
Figure 78: A Use Case Diagram Showing Allocation of Tasks Between TRACON User and the System for Tweaking One Aircraft.....	201
Figure 79: A Worker Competencies Analysis Structure for the TRACON System	203
Figure 80: A Use Case Diagram Showing Allocation of Tasks.....	205
Figure 81: Introduction Page for the ATM Example	207
Figure 82: CWA Structure for ATM example	209
Figure 83: A Text-Based Description of the Knowledge Elicitation Process for the WDA Phase.....	211
Figure 84: An Abstraction Hierarchy Model for the ATM Example	213
Figure 85: A Text-Based Description of the Knowledge Elicitation Process for the ConTA Phase.....	215

Figure 86: A Decision Ladder Model for the ATM Example.....	217
Figure 87: A Text-Based Description of the Knowledge Elicitation Process for the SA Phase.....	219
Figure 88: An Information Processing Activity Model for the ATM Example.....	221
Figure 89: A Text-Based Description of the Knowledge Elicitation Process for the SOCA Phase.....	223
Figure 90: A Use Case Diagram for the ATM Example.....	224
Figure 91: A Text-Based Description of the Knowledge Elicitation Process for the WDA Phase.....	226
Figure 92: A Skill, Rule, and Knowledge Inventory Model for the ATM Example	227

LIST OF TABLES

Table 1: Organizations Involved in SysML Development	31
Table 2: SysML Notation of the Different Diagram Types.	36
Table 3: Applications and Disciplines Supported by System Modeling Language	71
Table 4: Work Domain Analysis Operations Description.....	114
Table 5: Work Domain Analysis Attributes Description	115
Table 6: Control Task Analysis Operations Description	116
Table 7: Control Task Analysis Attributes Description.....	117
Table 8: Strategies Analysis Operations Description	117
Table 9: Strategies Analysis Attributes Description.....	118
Table 10: Social Organization and Cooperation Analysis Operations Description	118
Table 11: Social Organization and Cooperation Analysis Attributes Description.....	119
Table 12: Worker Competencies Analysis Operations Description	119
Table 13: Workers' Competencies Analysis Attributes Description	120
Table 14: Cognitive Factors Team Educational Background Description.....	178
Table 15: Cognitive Factors Team Experience Description.....	180
Table 16: Usability Evaluation for User 1	228
Table 17: Usability Evaluation for User 2	229
Table 18: Usability Evaluation for User 3	230
Table 19: Usability Evaluation for User 4	231

LIST OF ACRONYMS/ABBREVIATIONS

AC	Aircraft
ACT	Activity diagram
AH	Abstraction Hierarchy
ATM	Automated Teller Machine
BDD	Block Definition Diagram
CDR	Critical Design Review
CDD	Capability Development Document
CNO	Chief of Naval Operations
ConTA	Control Task Analysis
CWA	Cognitive Work Analysis
CWAT	Cognitive Work Analysis Tutorial
DDG	Guided-Missile Destroyer
DL	Decision Ladder
DoD	Department of Defense
GAO	Government Accountability Office
GOMS	Goal-Operators-Method-Selection Rules
HCI	Human-Computer Interaction
HFE	Human Factors Engineering
HSI	Human System Integration
IBD	Internal Block Diagram
IBM	International Business Machines
IDA	Institute for Defense Analyses
IFFA	Information and Functional Flow Analysis
IFM	Information Flow Map
IMPRINT	Improved Performance Research Integration Tool

INCOSE	International Council of Systems Engineering
JASS	Jack, and Job Assessment Software System
KBB	Knowledge-Based Behavior
LCS	Littoral Combat Ship
NASA	National Aeronautics and Space Administration
NDM	Naturalistic Decision-Making
NIST	National Institute of Standards and Technology
OMG	Object Management Group
OSD	Office of the Secretary of Defense
PAR	Parametric diagram
PEO IWS	Program Executive Office Integrated Warfare Systems
PKG	Package diagram
RBB	Rule-Based Behavior
REQ	Requirements diagram
SA	Strategies Analysis
SBB	Skill-Based Behavior
SD	Sequence diagram
SOCA	Social Organization and Cooperation Analysis
SoS	Systems-of-Systems
SME	Subject Matter Expert
SRK	Skill, Rule, and Knowledge
STM	State machine diagram
SysML	System Modeling Language
TRACON	Terminal Radar Approach Control
UC	Use case diagram
UML	Unified Modeling Language
WCA	Work Competencies Analysis
WDA	Work Domain Analysis

CHAPTER ONE: INTRODUCTION

Developing a set of complete and consistent requirements is the most important step in the systems engineering process. However, it is also becoming more challenging and critical as work environments have become more complex. The evolution of work to incorporate more computerization increases the need for more cognitive skills to effectively complete work. Because industry and government organizations have very limited resources, it is important to establish the correct requirements early in the development process in order to reduce errors and costs throughout the entire system's life cycle.

Cognitive work requirements are vital for defining the system requirements for an effective system. The primary purpose of the cognitive work requirements is to identify user strategies in performing cognitive tasks. Decision making, problem solving, and system monitoring are included in cognitive work requirements. The Surface Warfare Program Manager's Guide claims that requirements typically lack completeness, correctness, consistency, and validity and are often ambiguous (Department of the Navy, 2001). The lack of accounting for cognitive factors during the systems engineering process contributes to incomplete system requirements.

This study will use the Cognitive Work Analysis (CWA) framework implemented within the Systems Modeling Language (SysML) to address the lack of cognitive requirements defined by systems engineers during the systems engineering process. (Stoner, 2006) has stressed the significance of including cognitive strengths and

limitations of users in system designs. The CWA framework was developed by Rasmussen at Riso National Laboratory in Denmark in the 1970s. The primary purpose of the framework was to focus on human-centered design when developing new information systems.

The CWA framework models five different aspects of a system and how the system impacts the worker. It is used to identify how the system will be used, the environment in which work will be performed, the tasks users will perform, and how the user will perform the tasks. Additionally, the framework determines who will be accountable for each task and the level of competency the user will require. The CWA framework functions by integrating all the information provided by the models in each phase of analysis. The results are utilized for design requirements that are used for developing complex sociotechnical systems. The field of CWA is expanding into other applications because it provides a holistic systems approach and a comprehensive evaluation of the work environment. It has been applied to various work domains which include, but are not limited to, the following:

- Air traffic controller training
- F/A-18 pilot training
- Camera interface development
- Design proposal evaluation for military defense systems
- Identification of relevant information needs in an emergency management system

SysML will be employed to construct the cognitive work requirements framework. SysML is a system modeling tool currently used by systems engineers. SysML is vital to the process of the study because it transitions systems engineering from a document-based process to a model-based process. A model-based process provides consistency when exchanging information between product teams. Properly structured requirements are essential for all stakeholders' comprehension. SysML represents requirements as model elements. The formal description of system requirements in the early phases of development improves the understanding of the system requirements and how they answer the users' needs.

SysML is a graphical language for building models of systems that are complex, distributed, and large-scale. It is used to create object-oriented models of systems that incorporate software, people, material, and other physical resources. SysML expresses both structure and behavior for such systems (Huang, Ramamurthy, & McGinnis, 2007). SysML is designed to support the specification, analysis, design, verification, and validation of a broad range of systems. It is capable of graphically illustrating the interaction between all five models of the CWA framework. The interaction between phases of CWA is not typically demonstrated.

1.1 Problem Description

Currently, most systems engineers do not have access to cognitive work analysis information or training. This lack leads to a disregard of the cognitive aspect of system design. Inadequate descriptions of cognitive strengths and limitations of users

contributes to a decrease in system performance (Woods & Roth, 2005). The reasons systems engineers ignore the benefits of cognitive work are that they do not know how to do it, when to do it, or what cognitive analysis methods to use and/or suffer from an inadequate allocation of budget and time to cognitive analysis (Rasmussen, Pejtersen, & Goodstein, 1994).

Since most systems engineers do not have access to the information or training that is needed to apply cognitive work analysis methods to the systems engineering process, cognitive work requirements are usually ignored. The result of this issue is system requirements that do not account for the cognitive strengths and limitations of users. A lack of understanding of users' cognitive strength and limitations leads to imprecise system requirements. In July 2002, a report from the General Accounting Office to the Chairman of the Subcommittee on Technology and Procurement Policy, Committee on Government Reform in the House of Representatives claims that some of the government's largest procurement operations are not always run efficiently, because requirements are not clearly defined (Cooper, 2002). The Surface Warfare Program Manager's Guide affirms that requirements analysis errors constitute a majority of the training objective deficiencies in complex training systems (Department of the Navy, 2001). In 2009, the Chief of Naval Operations (CNO), Admiral Gary Roughead, recognized a direct link between accurate requirements and the total life cycle costs of procuring a new system. The CNO testified before the Subcommittee on Appropriations on June 2, 2009 to the effect that he would continue to demand that the

Navy accurately articulate requirements to deliver effective and affordable systems to the fleet. In order to improve system requirements, integration of cognitive work requirements into the systems engineering process has to be improved.

1.2 Purpose of the Study

The objective of this study is to develop a Cognitive Work Analysis Framework using Systems Modeling Language. The study has two phases. The first is to align the CWA terminology with the SysML to produce a CWA framework using SysML. The second is to create an instruction using SysML to inform systems engineers of the process of incorporating cognitive requirements into their system designs.

1.3 Significance of the Study

The framework developed using SysML provides a structured and standard way to define cognitive work requirements for users of SysML, provides a tool for systems engineers to incorporate the cognitive strengths and limitations of the user using SysML, and contributes to defining more accurate use cases.

CHAPTER TWO: LITERATURE REVIEW

2.1 Systems Engineering

2.1.1 Introduction

The term “systems engineering” was conceived in the early 1940s (Schlager, 1956). In 1995, the International Council on Systems Engineering (INCOSE) was formed to address the need for improvements in systems engineering practices and education (Resp Group, 2010). INCOSE’s definition of systems engineering:

Systems Engineering is an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, then proceeding with design synthesis and system validation while considering the complete problem (INCOSE, 2006).

The Department of Defense defines the systems engineering process as:

The systems engineering process is a technical management and problem-solving process applied through all stages of development to transform needs and requirements into a set of system product and process descriptions (adding value and detail with each level of development)

(Defense Acquisition University, 2009).

Booton and Ramo of TRW Electronic Systems Group defined systems engineering as:

A discipline that concentrates on the design and application of the whole system. It involves looking at a problem in its entirety, taking into account all the facets and all the variables and relating the social to the technical aspect.

(Booton & Ramo, 1984)

Overall, systems engineering aids in better comprehension and management of complex systems throughout the systems' life cycles. The applications of systems engineering have evolved from large, complex military and government systems to more business and consumer-oriented products. Presently, systems engineering is applied to commercial aircraft, energy systems, health care, highway transportation, information technology, manufacturing, medical devices, automobiles, space exploration, telecommunications, agriculture, household appliances, emergency services, Internet banking, modeling and simulation, Internet-based applications, logistics, and many other organizations. Systems engineering can be applied to any system development (Wray, McKinney, & Whalen, 2000).

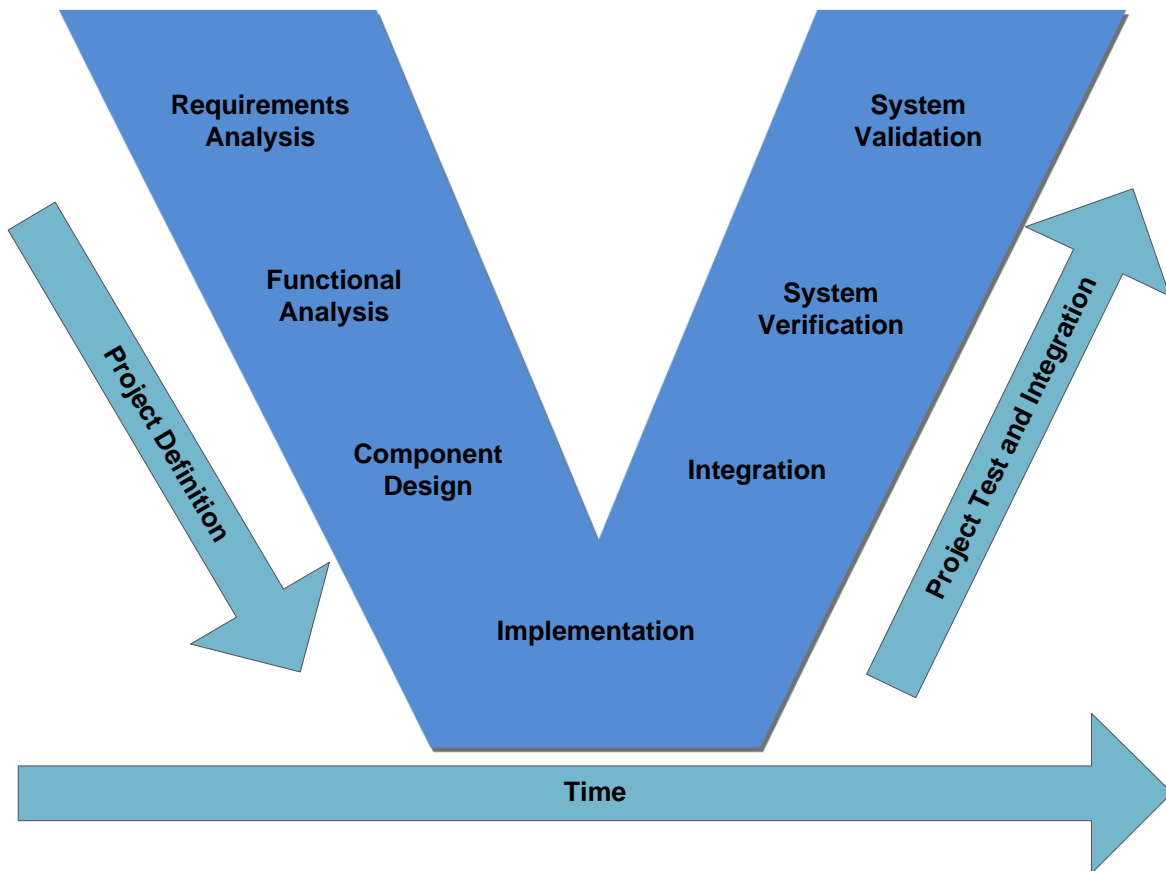
Systems engineering reduces the risk of cost overruns, scheduling delays, and performance deficiencies. It increases the probability that the system will satisfy the user's requirements. Other benefits include stakeholder participation, verified functionality, and better documentation (Boehm, Valerdi, & Honour, 2008). This statement of benefits has been supported by several studies. The studies demonstrated that the utilization of effective systems engineering results in better cost, schedule, and performance (Valerdi, Miller, & Thomas, 2004), (Honour & Valerdi, 2006). Most of the

studies showed a 50% overrun on projects not using systems engineering (Honour, 2006). In addition, the studies confirmed an improvement in the project cost performance when effectively implementing the systems engineering process (Honour, 2004).

A broad range of disciplines are involved in systems engineering. These disciplines include: industrial engineering, requirements engineering, cognitive systems engineering, configuration management, control engineering, interface design, operations research, project management, program management, performance engineering, reliability engineering, safety engineering, software engineering, and any other discipline that is involved in satisfying stakeholders' needs.

2.1.2 The "V" Model

There exist many different systems engineering process models. Each version of the process models specifies the main steps of the systems engineering methodology. The "V" Model has become the standard way to represent systems engineering methodology. Figure 1 shows an adapted "V" Model from Forsberg & Mooz (1992). The "V" Model illustrates each phase of the system life cycle except for the concept development and disposal phases.



Adapted from Forsberg & Mooz (1992).

Figure 1: Systems Engineering Process Model

The horizontal arrow represents time to complete the system. The development process starts on the left side of the model and concludes on the right side. The left side of the “V” Model represents the development of system and functional requirements. The right side of the “V” Model represents system integration, verification, and validation. The “V” Model is composed of several phases. The phases are requirements analysis, functional analysis, component design, implementation, integration, system verification, and system validation.

2.1.2.1 Requirements Analysis

The initial focus of the systems engineering process is to develop a complete set of system and user requirements. Requirements analysis is one of the most critical phases of systems engineering. Positive requirements analysis minimizes design changes through the development process (Blanchard & Fabrycky, 1990). If requirements analysis is done well, project cost will not exceed budget, schedule will not be extended, and system will maintain at least minimum performance constraints. The requirements are determined by the needs of the user or users. There are many different methods for gathering requirements. These knowledge elicitation techniques include, but are not limited to: interviews, case studies, simulations, observation, questionnaires, prototyping, and document analysis (Zowghi & Coulin, 2005). Typically, the users will be interviewed. The results of the interviews should include a user requirements document. The user requirements document should contain the system's purpose, operational constraints, interaction with external systems, functionality, performance parameters, and interface characteristics. The user requirement document will guide system designers in the subsequent systems engineering phases.

2.1.2.2 Functional Analysis

Functional analysis is the process of identifying and describing the functions of a system (Kossiakoff & Sweet, 2003). This is not a physical description of the system. It is a description in terms of functions and performance parameters. Functions are actions

that are necessary to meet system objectives. The functions are performed through the use of resources (i.e., equipment, personnel, facilities, etc.) (Leonard, 1999).

The functional analysis phase starts with identifying the system goals and relating them to functions the system will perform in to order to achieve those goals (Cockburn, 1997). The functions identified should be high-level functions of the system. The high-level functions should be decomposed into lower-level functions of the system. The lower-level functions are the steps that are required by the system to achieve the goals of the system.

“The decomposition can be carried out as deeply as needed to define the transformations that the system must be able to perform.” (Buede, 2009)

2.1.2.3 Component Design

The component design phase of the systems engineering process describes how the components will be developed (Forsberg & Mooz, 1992). The individual hardware and software components are sketched, blueprinted, outlined, or drafted in this phase. Software is modeled and documented with specifications prior to actual coding. Hardware is drawn or modeled and a set of specifications are developed before actual fabrication. This phase concludes with a Critical Design Review (CDR) to get final approval before components are built. All stakeholders are involved in the CDR.

2.1.2.4 Implementation

The actual fabrication of hardware and software components is accomplished in the implementation phase of the systems engineering process. The components are constructed according to the specifications established during the component design phase. After the system components are constructed, they are tested. The deliverables for this phase of the process include hardware and software components that have been tested and are ready for the integration phase. In addition, supporting documentation—which consists of user manuals, maintenance manuals, and/or installation manuals—will be part of the deliverables.

2.1.2.5 Integration

The purpose of the integration phase is to successfully combine hardware and software components. Integration is a highly iterative process. Sub-components are incrementally combined, verified, and then combined into larger sub-components. The larger sub-components are combined, verified, and then combined into larger sub-components until the whole system is finished. The interim verification ensures correct communication and interaction between sub-components and reduces risk and minimize errors (Curtis, Krasner, & Iscoe, 1988). The process of integrating the components of a system requires a plan to get started. The integration plan outlines the assembly order for the sub-components and explains how those sub-components will be integrated with other system components.

2.1.2.6 System Verification

The verification process confirms that the system fulfills all requirements that were specified in the previous phases. The focus of system verification is to make sure the system has been “built right” (Preece, 2001). The process is utilized by stakeholders before accepting the system. Verification should be thorough so that defects are identified early and at the lowest level possible. Isolating a defect early at the component level is important, because it gets more difficult to find the problem when the entire system is built. Any one of many components could be contributing to the defect. As stated before, verification is performed iteratively. So individual components are verified first. Then the sub-systems are verified. Finally, the whole system is verified. In order to start the process of system verification, a verification plan must be created. The verification plan outlines the step-by-step process for verifying each component of the system against the requirements. The verification plans should be written during the requirement analysis phase.

2.1.2.7 System Validation

The objective of the validation phase of the systems engineering process is to confirm that the system fulfills its intended purpose. The focus of system validation is to make sure the “right system has been built” (Sheard, 1996). Validation takes place after the system has been deployed and is in operation. At this phase, systems engineers and systems designers have a good measure of the effectiveness of the system in its

operational environment. The validation process starts with planning. The output of the validation planning process will include the participants, the schedule, the location, the required resources, and what will be validated. The validation plan may be formal or informal. The choice belongs to the system owner. A formal validation plan will be repeatable and well-documented (Boehm, 1986). The results of the validation process will include a report on that satisfactory achievement of the functional purpose as well as any deficiencies in the system. Recommendations to upgrade aspects of the system will be made based on the system deficiencies identified in the report.

2.1.3 The Three Evils of Systems engineering

The “three evils of systems engineering” are complexity, communication, and understanding (Holt & Perry, 2008). Complexity in the systems engineering domain can be defined as a system that has many independent parts that interact and work together toward a common goal (Calvano & John, 2004). The complexity of a system is based on the number of relationships that exist between system elements. The higher the number of relationships, the more complex a system will be.

Lack of understanding can occur in any phase of the systems engineering process. A lack of understanding can lead to the needs of the user not being addressed, problems that are not clearly defined, improper application of systems engineering principles, inaccurate requirements, or incorrect component interactions.

Communication problems can exist between individuals, groups, systems, and organizations (Elm, Goldenson, El Emam, Donatelli , & Neisa, 2007). If three people

read a set of system requirements, more than likely there will be three different interpretations of the meaning (Schindel, 2005). Even different models may have problems communicating, sharing data, tracing requirements, or duplicating work (Doyle & Pennotti, 2005).

These three evils cannot be eliminated in systems engineering, but they can be minimized using model-based systems engineering. Complexity, lack of understanding, and communication are interrelated. Any deficiencies in one will lead to deficiencies in the others evils of systems engineering. However, any improvements to one will lead to improvements in the other factors.

2.2 Cognitive Work Analysis

2.2.1 Introduction

Cognitive Work Analysis (CWA) is described as a formative, constraint-based framework for analyzing complex sociotechnical systems (Rasmussen, Pejtersen, & Schmidt, 1990). There are three categories of work analysis modeling. They are normative, descriptive, and formative work analysis modeling techniques (Vicente, 1999). Normative models describe what a user should do when interacting with a system. Descriptive models describe what a user actually does when interacting with a system. Finally, formative models describe what a user could do when interacting with a system. The formative approach can assist in generating new ways of doing work (Vicente, 1995). Traditional work analysis models fall into the normative or descriptive categories of modeling, which focus on specific tasks and procedures. However, CWA

identifies the constraints of the work environment and the operator, the purpose of the system, and the tasks the user can accomplish within the constraints of the work environment (Fidel & Pejtersen, 2004).

CWA has five phases of analysis. The phases of analysis include: work domain analysis (WDA), control task analysis (ConTA), strategies analysis (SA), social-organizational and cooperation analysis (SOCA), and worker competencies analysis (WCA). Each phase of analysis uses a different modeling technique to describe a different aspect of a system. The modeling techniques most commonly used in CWA include abstraction hierarchies (AH); decision ladders (DL); information flow maps (IFM); and skill, rule, and knowledge-based (SRK) inventories (Naikar, 2006b).

The purpose of the WDA is to determine what can be accomplished with a system without violating the laws of nature or exceeding the capabilities of the system (Crone, Sanderson, & Naikar, 2003). An Abstraction Hierarchy (AH) modeling tool is used to map out the functional properties of a sociotechnical system. The AH has five level of decomposition. The highest level of the model defines the purposes and goals of the system. The lowest levels of the model indicate and describe the physical components (e.g., equipment) of the system.

The second phase of CWA is ConTA. The ConTA phase covers what needs to be done within the limits of the work domain. A Decision Ladder (DL) model is used to show all the tasks that could be accomplished within the limits of the work domain. A DL shows the alternative courses of action for a particular decision.

The third phase of CWA is the SA phase. This phase focuses on how the user performs the control tasks to accomplish the goal. Typically, the same control task can be performed in many ways using different cognitive strategies (Darses, 2001). An Information Flow Map (IFM) model is used to represent the control tasks. IFM is a graphical representation of how the user can reach an end goal. All information processing activities are contained in IFM.

The fourth phase of CWA is the SOCA phase. The SOCA phase determines who will carry out the work and how it is shared. The IFM modeling tool can also be employed to identify who will do what tasks.

The last phase of CWA is the WCA phase. This phase identifies the physical and cognitive demands placed on the operator and the level of competency that the operator will need to function effectively. The Skill, Rule, and Knowledge-based (SRK) inventory model is used to determine how information should be displayed to take advantage of human perception and psychomotor abilities.

2.2.2 Work Domain Analysis

An Internet search of a variety of databases was conducted for this literature review. These databases include: Proquest, Academic Search Premier, LEA Online, Google Scholar, the University of Central Florida online library, and First Search. Relevant dissertations and theses were also included in the literature review. In addition, relevant references located in the reference sections of the journals, dissertations, and

theses were used. Since the field of CWA is rapidly being employed in a variety of applications, the timeframe is limited to the years from 1990 to 2010.

The review of the literature showed very little information available for a five-phase CWA application. The majority of studies focused efforts on the initial phases of CWA, which are Work Domain Analysis and Control Task Analysis (Rehak, Lamoureux, & Bos, 2006). The main reason cited for not using all phases of CWA is that the technique is too time-consuming or there was insufficient funding to complete a full CWA model (Sanderson, Naikar, Lintern, & Goss, 1999).

Naikar implemented the WDA phase of CWA to identify the training needs of military fighter aircraft. She used the AH to compare their functional purpose, abstract function, general function, physical function, and physical form to training objectives, measures of performance, scenario generation, physical functionality, and physical structure, respectively (Naikar, Sanderson, & Lintern, 1999).

The highest level of the AH is defined as the training objectives of a training domain. Training objectives are the primary purpose for the training system's existence (i.e., what it will train). These training objectives can be converted into specifications for an actual training system.

The Abstract Function level of the AH is defined as the performance measures or measures of effectiveness of the training domain. The concept at this layer describes measures of effectiveness for evaluating trainees' performance or for evaluating the

effectiveness of a training program. Measures of effectiveness may be translated into specifications for the data collection capabilities of the simulator (Naikar et al., 1999).

The General Function level of the AH is converted into basic training functions. The basic training functions are used for learning particular tasks and skills that satisfy the overall training objectives (Roth, 2008). The basic training functions can be used to identify specifications for generating scenarios for a training system or used to develop a part task trainer.

The fourth level of the AH identifies the physical devices that the trainee must learn to operate in order to complete the basic training functions to satisfy the overall training objectives. Physical Functionality may be translated into specifications for the functionality of the physical systems that should be available in the training system so that trainees can be trained to utilize this functionality in performing basic work functions (Naikar, Moylan, & Pearce, 2006).

The physical form is the lowest level of the AH; it describes the equipment, tools, and/or resources available in the training domain. Trainees should know the location, appearance, configuration, and other physical properties of these devices in performing the basic work functions of the training system (Naikar, 2006a).

Finally, each level of the AH is connected by a means-end relationship. The means are defined as how a task or function is achieved in the AH. The end is the function or task. On the AH, the lower-level objects are the means and the higher-level objects are the ends. The means-end relationship may be used to train operators to

think and behave in adaptive ways to deal with unexpected or unpredictable situations (Sanderson et al., 1999). The results of a study showed CWA to be opportunistic and flexible when new knowledge elicitation activities arose and when the scope of the project itself expanded significantly (Paradis, Breton, Elm, & Potter, 2002). One study confirmed that a CWA model developed for a military command and control environment did not fail when the scope was expanded to include novel events (Paradis et al., 2002).

2.2.3 Control Task Analysis

The ConTA complements the WDA by identifying what needs to be done to accomplish the system objectives established in the first phase of CWA. Traditional task analysis approaches typically break down an activity into sequences of tasks. ConTA is not concerned with how an activity is carried out, who carries it out, or what skills and training are necessary (Naikar et al., 2006). The answers to those questions are covered in the strategies analysis, social organization and cooperation analysis, and worker competencies analysis phases of the CWA framework. There are three critical aspects of ConTA (Naikar et al., 2006). First, ConTA recognizes that the same goals may be accomplished in different ways in many complex systems. Second, activities identified in ConTA are characterized as a set of work situations and work functions. Finally, ConTA recognizes that decision-making functions or control tasks are required for each work situation and work function (Naikar et al., 2006). Work situations in a ConTA model are associated with the functional purpose and abstract function of the

AH model. Work functions in a ConTA model are interrelated with the general function and physical function of the AH model of the first phase of CWA. Each work situation and work function will be associated with a DL. The boxes on the DL represent information processing activities the user should engage in and the circles represent states of knowledge that are the results of information processing activities (Vicente, 1999).

Rasmussen incorporates work functions and work situations into a matrix showing which activities can occur in which situation (Rasmussen et al., 1990). This is called a Contextual Activity Template. The work situations are located on the horizontal axis of the Contextual Activity Template matrix and the work functions are located on the vertical axis. The dotted boxes in the matrix represent all of the work situations in which a work function can take place.

In traditional task analysis, the information processing activities are completed in sequential order, but in ConTA not all information processing activities have to be completed in order to complete the task. These shortcuts are represented by the arrows in the center of the DL and are called shunts and leaps. The shunts link information processing activities to states of knowledge. The leaps link two knowledge states together. The application of these shortcuts depends on the level of expertise of the system user or operator.

The DL can be broken into three sections. The left side of the DL template is used for representing control tasks related to identifying the system state. The top part

of the decision ladder is used for representing control tasks related to the system goals. The right side of the DL template is used for representing control tasks related to planning and execution. During the planning and execution steps, the proper sequence of control actions is implemented through the process of identifying tasks and resources and scheduling and carrying out actions (Naikar et al., 2006).

The DL was added as a formative element to Naturalistic Decision-Making (NDM) (D. P. Jenkins, Stanton, Salmon, Walker, & Rafferty, 2010). They showed how decision-making can proceed within an environment, independent of situation and actor. The research was applied in a tank warfare environment. In this warfare environment, life-and-death decisions are made in a relatively short period of time. Tank crews have to distinguish between enemies and friendlies before the enemy identifies them. The benefits of integrating the ConTA phase of CWA and NDM produced critical information that assisted in the design of tank interfaces, helmet-mounted displays, and training support and in the development of operating procedures and decisions relating to the allocation of crew functions (Jenkins et al., 2010).

In 2003, Cummings and Guerlain modified the CWA method for designing a decision support system for a non-existent domain. The modified CWA framework was applied to the Navy's new Tactical Tomahawk missile. The Tactical Tomahawk missile is a ship-launched, long-range, land-attack missile. It is employed against land-based air defenses, power plants, communications buildings, and other high-value land-based stationary targets. The new in-flight redirection capability added a new human-

computer interaction (HCI) that did not exist in any legacy missile systems. They deduced that most HCI modeling techniques required some knowledge of an existing domain, tasking procedures, and well-established organizational parameters (Cummings & Guerlain, 2003). Two additional phases were added to the CWA framework. The first addition was the Global Organizational Analysis phase. This phase focused on identifying the “relevant social group” (Bijker, 1997). In general, relevant social groups are all the stakeholders (i.e. users, engineers, SMEs, manufacturers, management, etc.) involved in the development of the sociotechnical system. The Global Organization Analysis is done prior to performing the WDA phase. The next modification to the CWA framework occurs after the ConTA phase. This new phase is titled Creation of Pilot Domain. The pilot domain is added to validate the WDA and ConTA phases. Cognitive modeling, simulation, prototypes, and scenario-based design are techniques used to establish the pilot domain (Cummings & Guerlain, 2003). A prototype of the user interface for the Tactical Tomahawk was developed for this research based on WDA and ConTA.

Another application of ConTA was demonstrated by Neelam Naikar and Alyson Saunders in the aviation training domain. This approach was used to identify specific requirements for training F-111 pilots. The study reviewed and analyzed aircraft accident and incident reports to determine when pilots have crossed the safe boundaries in the past and the problem-solving difficulties they have experienced (Naikar & Saunders, 2003). After identifying when the safe boundary was crossed, they

used the DL to examine the pilot's decision making and extract training requirements for other air crews. The objective is to allow air crews to cross the safe boundaries during training, detect the error made when they crossed the safe boundary, and then execute the appropriate actions to recover from the error. The research confirms how critical decision-making is to reducing errors.

2.2.4 Strategies Analysis

The purpose of the SA phase is to explore the variety of ways in which each of the control tasks could be accomplished. WDA and ConTA phases are prerequisites for the SA phase of CWA. WDA and ConTA can be used independently of the other phases of CWA, but SA is dependent on the results and products of the preceding phases. In other words, you cannot develop a strategy on how to complete an activity without knowing what the activity will be, which comes from the ConTA. Additionally, ConTA is typically produced from the WDA. Therefore, the CWA approach provides an interrelated set of methodologies where different attributes of a system can be analyzed. Jenkins, Stanton, Salmon, Walker, & Young (2008) suggest that there has been little attempt in the literature to extend the CWA framework beyond the first two phases. In other words, there is limited research using SA and the follow-on phases of CWA. (Naikar, 2006b) recognized four key concepts of SA.

- SA is not concerned with defining detailed sequences of actions.
- Several strategies are usually possible for performing a single activity.

- Workers often switch between multiple strategies while performing a single activity.
- It is important to identify the range of strategies that are possible as opposed to the range of strategies that are used by workers.

The selection of a strategy is highly dependent on situation and actor (St-Cyr & Kilgore, 2008). Therefore, it is difficult to identify a particular response to a given control task. Naikar goes on to say that the SA phase identifies potential categories of generic strategies. In summary, SA will derive generic strategies to complete a control task, but it may not be the strategy utilized by the actor. Diverse actors will normally use diverse strategies.

The typical SA modeling tool is an Information Flow Map (IFM). IFM are graphical representations of activities that are necessary to complete a control task. The graphical representation is similar to the ConTA phase. Circles are used to represent knowledge states and rectangles are used to represent information processing activities. Information processing activities are the cognitive and computational actions a trainee employs to complete a task. Knowledge states are defined as the products of the cognitive actions (St-Cyr & Kilgore, 2008). The actor has three options to resolve the situation. They can “hold one aircraft,” “reroute one aircraft” or “tweak one aircraft.” Each of these options is a control task. The process to complete each of the control tasks is represented by circle and rectangle objects in Figure 11. Some of the information from the previous phases is used to fill in the circles and rectangles in the

SA phase of CWA. There are numerous strategies for achieving the functional purpose of the system from the AH.

2.2.5 Social Organization and Cooperation Analysis

The purpose of the Social Organization and Cooperation Analysis (SOCA) phase of the CWA framework is to allocate information processing activities and knowledge states responsibility among the actors in the system. In a training environment, SOCA models determine how tasks are distributed within the team and how the team will communicate. The products from the initial phases of CWA support assigning responsibility among the actors in the SOCA phase. The IFM model used in the SA phase can be used in the SOCA phase. Each actor will be assigned a color, then each information processing activity and knowledge state on the IFM model will be color-coded with the responsible actor. Some information processing activity and knowledge states will be shared by a human operator and computer automation. To distribute each information processing activity and knowledge state between team members and computer automation, the strengths and weaknesses of each team member's functional position and computer automation are compared to each task to establish the best task allocation for the team. Tasks that require high cognitive loads on working memory are normally assigned to the computer. Task that require judgment are normally assigned to the human operator.

There are several key concepts related to the SOCA model (Naikar, 2006b). First, SOCA does not define a single or best organizational structure. Second, the allocation

of work may be defined in relation to the work domain, work situations, work functions, control tasks, and strategies (Naikar, 2006b).

2.2.6 Worker Competencies Analysis

The Worker Competencies Analysis (WCA) phase of CWA is concerned with making the task easier for the end-user. It identifies the competencies required for each team member to effectively complete the work domain's control tasks. This is done by using skill and rule-based behavior when applicable and also supporting the users' knowledge-based behavior when addressing unanticipated events. The modeling technique used in this phase is the Skill, Rule, and Knowledge (SRK) inventory. This taxonomy outlines basic distinctions between the three main psychological processes: Skill-Based Behavior (SBB), Rule-Based Behavior (RBB), and Knowledge-Based Behavior (KBB) (Rasmussen et al., 1990). The SRK inventory is used to outline the competencies that system users must have or must acquire in order to effectively perform control tasks across all three of the behavior types.

A skill-based behavior requires very little conscious effort to perform a task. Using a mouse to move a cursor is an example of a skill-based behavior. Very little conscious effort is required to move the cursor on the screen.

A rule-based behavior is based on the rules and/or procedures established by an organization, for example, user instructions or regulatory authority rules necessary to complete a task or use equipment. Following the procedures to start a plane would be considered a rule-based behavior.

A knowledge-based behavior requires the highest level of conscious effort to complete a task. An example of a knowledge-based behavior is a pilot response to losing both engines due to bird strikes and landing the airplane in the Hudson River. Knowledge-based behavior is used when the situation is unfamiliar or unanticipated. Traditionally, SBB and RBB are executed more quickly, effectively, and effortlessly than KBB. The SRK inventory can be used to determine employee selection, job prerequisites, and training.

2.2.7 Current CWA Tools

Currently, most users of the CWA approach use Microsoft Word, Excel, Visio, post-its and string, flowcharting, and/or paper to illustrate the different phases of CWA. The CWA process is often criticized for being complex and time-consuming (Cummings, 2006). The SysML has many benefits over both Microsoft software and pen and paper. SysML passes important data forward aiding the completion of subsequent representation (Balmelli, 2007). This means that minor changes to text in a diagram will feed through from the initial stages to the subsequent phases. This has particular benefits in the SOCA phase, which reuses the products generated in the previous three phases. This capability increase the speed and accuracy at which CWA models can be developed, edited and reviewed.

2.2.8 Cognitive Work Analysis Summary

In conclusion, the CWA approach consists of five interrelated phases that are used to analyze and explore different aspects of a system. CWA provides a useful

complement to the systems engineering processes. The most useful aspect of CWA as a modeling approach is that it allows the researcher to move from a high-level conceptual view of purpose, intent and goals to a detailed view of functionality and capability (Chin, Sanderson, & Watson, 1999). The CWA framework has been applied to a variety of complex work and training environments, including: revolutionary and first-of-a-kind system development (Naikar, Pearce, Drumm, & Sanderson, 2003), (M. Cummings & Guerlain, 2003); system design (Bisantz et al., 2003); training needs analysis (Naikar & Sanderson, 1999), (Naikar et al., 1999); training system design (Crone et al., 2003), (Lintern & Naikar, 1998); human-system integration, (P. M. Sanderson & Naikar, 2000); interface design and evaluation (Jenkins, Stanton, Walker, Salmon, & Young, 2008), (Vicente, 2000); evaluation of system design proposal (Naikar & Sanderson, 2001); human error management (Naikar & Saunders, 2003); process control (Vicente, 1999); military command-and-control decision making,(Jenkins et al., 2007), (Jenkins, Stanton, Salmon, & Walker, 2009), (Chin et al., 1999), (Burns, Bryant, & Chalmers, 2000), (Paradis et al., 2002); military aviation (Naikar, Lintern, & Sanderson, 2002); health care (Burns, Momtahan, & Enomoto, 2006); and air traffic controllers (Kilgore & St-Cyr, 2006).

2.3 Systems Modeling Language

2.3.1 Introduction

One way to challenge the three evils of systems engineering is to model systems. System modeling helps to simplify understanding of complex systems. This simplification of complex systems can lead to more effective communication among project stakeholders. System Modeling Language (SysML) provides a common language for systems engineers to model complex systems. In 2001, the International Council of Systems Engineering (INCOSE) and Object Management Group (OMG) started the process of adapting Unified Modeling Language (UML) to a systems engineering modeling language. In November 2008, the OMG released SysML Version 1.1. SysML is a visual language that supports model-based design, requirements analysis, verification and validation for a variety of large-scale, multidisciplinary complex systems. SysML graphically models system architecture, behavior, and functionality (Object Management Group, 2008). The language is an expansion of the UML 2.0. UML 2.0 is typically used to model system software. SysML extends UML capabilities to model system hardware, software, personnel, facilities, information, and procedures. In addition to INCOSE and OMG, many other organizations from industry, vendors, government, and academia supported parts of the SysML specifications. Table 1 shows most of the participant organizations in SysML development.

Table 1: Organizations Involved in SysML Development

Industry	Vendors	Government	Academia
American Systems Corporation	ARTISAN Software Tools	DoD/Office of the Secretary of Defense (OSD)	Georgia Institute of Technology
BAE SYSTEMS	Ceira Technologies		
Pivot Point Technology	I-Logix	NASA/Jet Propulsion Laboratory	
Raytheon	Mentor Graphics		
Boeing	Embedded Plus Engineering	National Institute of Standards and Technology (NIST)	
Israel Aircraft Industries			
Lockheed Martin Corporation			Gentleware
	Sparx Systems		
Deere & Company	Vitech		
EADS Astrium	IBM		
Eurostep	Telelogic		
Motorola	Structured Software Systems Limited		
Northrop Grumman			

One of the most important phases of a system's life cycle is requirements analysis. A lack of accurate requirements leads to schedule delays and additional costs. The complexity of modern sociotechnical systems makes requirements analysis more difficult. To simplify the development process, SysML methodology represents complex systems graphically. Requirements analysis is traditionally represented as texts which are accompanied by figures and drawings. The requirements describe all the product

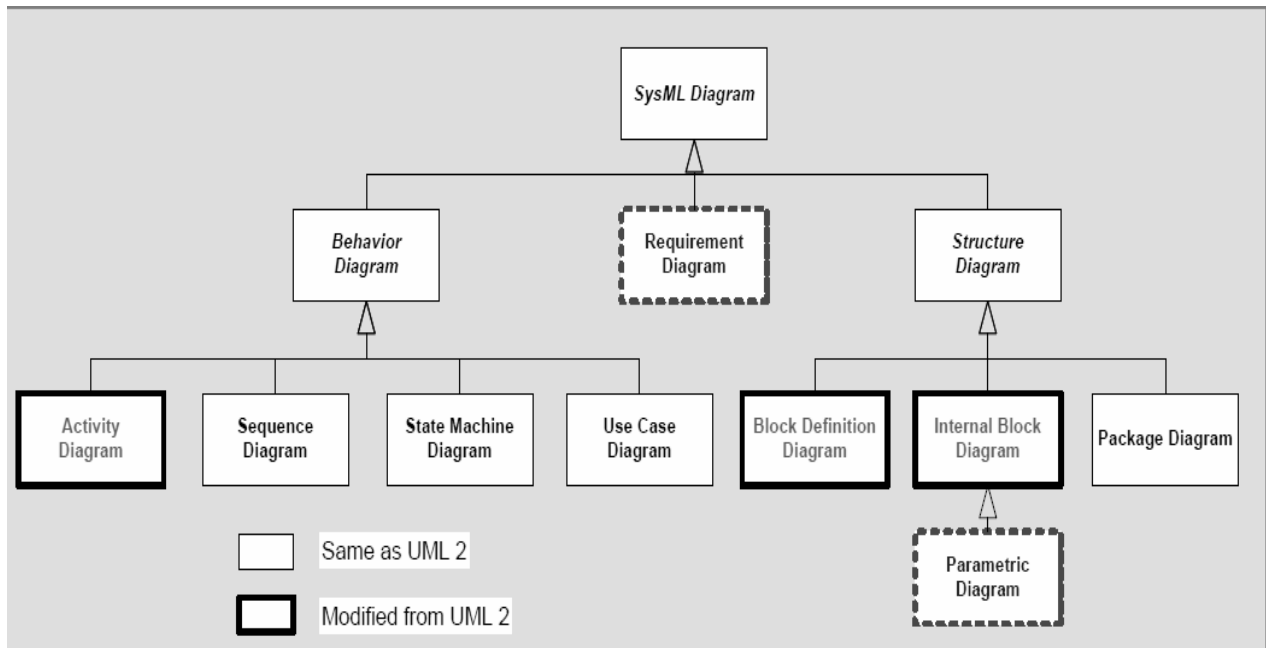
functions and the constraints under which these functions should be achieved (Balmelli, 2007).

There are many challenges that exist when identifying system requirements during the conceptual phase. A lack of comprehensible product architecture obstructs team understanding and communication, which consequently increases the risk of integration issues (Balmelli, 2007). The Boeing 787 Dreamliner was supposed to be a revolutionary step forward in aircraft design. To develop and build the aircraft, Boeing created a sophisticated global manufacturing network. Sections of the plane are constructed by companies in Japan, Russia, Australia, Italy, France, South Carolina, Sweden, India, Washington state, and Kansas. According to Airframer.com, over 300 companies from around the world are involved in the Dreamliner program. The integration of all these different components from an enormous amount of suppliers has created a supply chain debacle. According to a *Seattle Times* report, several specifications from Boeing provided ambiguous instructions and measurements that led mechanics to cut holes too shallow to attach fasteners. The specifications were prepared in Everett by Boeing engineering staff and were supposed to be translated by Boeing planners into easily followed instructions (Gates, 2008). The integration delays are costing Boeing and their suppliers billions of dollars. To address the manufacturing and logistics deficiencies, Boeing issues new design and production modification, which contribute to more confusion and delays by suppliers.

SysML is specifically designed to mitigate these challenges in the early stages of system development and throughout the system life cycle. Previous studies have shown that deficiencies in organization and management are typically responsible for problems associated with the development of a complex sociotechnical system (Sage & Armstrong, 2000). One study concluded that the main factors associated with cost overruns, schedule delays, and customer dissatisfaction are the result of a lack of user input, incomplete requirements, and continual requirement modifications (Hofmann & Lehner, 2001).

2.3.2 SysML Description

SysML has four classifications of diagram used to construct system models. The diagram classifications are structure, behavior, requirements, and parametric relationships. These are the four pillars of Object Management Group (OMG) SysML. System behavior and structure diagrams had previously existed in UML 2.0. System requirements and parametric relationships have been added for the purpose of systems engineering modeling. SysML is designed to represent structure and behavior of systems (Graves, 2009). It is a graphical language which is advantageous for human comprehension (Graves, 2009). Figure 2 shows the SysML Architecture.



From (Friedenthal, Moore, & Steiner, 2008))

Figure 2: SysML Architecture

The system structure diagrams are used to represent the physical structure of the system. The system structure includes the hardware, software, data, procedure, personnel, and facilities components. The basic structural elements in SysML are blocks. A block is a description of the system, subsystem, part, function, human, or process. System components in the structure diagrams are represented by Block Definition Diagrams (BDD) and Internal Block Diagrams (IBD). BDDs are used to describe the hierarchical and component structure of the system. IBDs describe the internal structure of each system component which consists of parts, connectors and flows. The structure diagrams also identify the interconnections between BDDs through the IBDs.

The system behavior diagrams describe the system functionality, component interaction, and processes. The system behavior diagrams contain use case, activity, sequence, and state machine diagrams. Use case diagrams illustrate system functionality. Activity diagrams show the flow of data and information between activities. Sequence diagrams describe the interaction between different parts in the system and the interaction of actors and the system or component of the system. The state machine diagram describes the actions that a system performs in order to complete an event.

The system requirement diagram graphically represents text-based requirements and associates them with related model diagrams that verify the requirements. These diagrams improve requirements visualization by using a graphical approach to system design. The requirement diagram provides traceability that bridges the gap between requirements and system models. The requirement diagrams also addresses the relationships between requirements, system design models and use cases (Hause, 2006). This graphical approach improves communication between stakeholders by facilitating system understanding, explicitly maps requirements relationships, and mitigates design errors.

The system parametric diagram identifies the physical or intentional constraints of the system. Constraint blocks are used to represent constraints in the parametric diagram. Constraint block properties are expressed by mathematical equations within the blocks and help establish mathematical relationships between system properties (Johnson, Paredis, Burkhart, & Jobe, 2007). Parametric diagrams can be used to

identify performance parameters or other mathematical constraints which can be used to support trade-off analysis. Each diagram discussed so far is identified by SysML notation. Table 2 indicates the notation of the different diagram types.

Table 2: SysML Notation of the Different Diagram Types.

Diagram Type	Notation
Activity diagram	act
Block definition diagram	bdd
Internal block diagram	ibd
Package diagram	pkg
Parametric diagram	par
Requirement diagram	req
Sequence diagram	sd
State machine diagram	stm
Use case diagram	uc

2.3.2.1 Structure

2.3.2.1.1 Block Definition Diagram

Block Definition Diagrams (BDD) describe the components of the system. BDD show the components of the system and the relationships between them. Components can be represented by blocks, parts, packages, and constraint blocks in the BDD model. Packages and constraint blocks will be discussed in more detail in the later sections of this dissertation. Block diagrams are used to describe the architecture of a system in terms of systems and subsystems. More detail can be added within each block. Attributes, operations, and/or a description can be added to each block to provide

more detail about the specific block. Part diagrams are also used to describe the composition of block diagrams. Attributes, operations, and/or a description can be added to each part diagram to provide more detail about the specific part. The most commonly used relationships are aggregation, directed composition, dependency, generalization, and association. Aggregation shows a part-whole relationship—a whole system or component of a system made up of different parts. It is represented by a clear diamond shape on the aggregate end and no symbol on the other end of the line. A directed composition is a one-direction composition relationship between a block and another block. It is represented by a solid-color diamond shape on the composite end and an arrow symbol on the directed end of the line. A dependency relationship is shown when one block depends on another block. There is a client and supplier relationship that exists between the two blocks. A dependency is represented by a dashed arrow. A generalization shows how one block is derived from another. This is a parent and child relationship. The parents has a child. A generalization is represented by a clear triangle arrowhead and a solid line. The arrowhead points to the parent. Finally, a flow relationship indicates the flow of data and commands from one block or part in the system to another block or part. The flow relationship expresses that information can be exchanged between blocks, parts, or use cases (Weilkiens, 2008). Flow relationships are notated by a green dashed line with an arrowhead at one end. The arrowhead indicates where the data or commands are going.

Figure 3 and Figure 4 illustrate the models discussed in this section (i.e., section 2.3.2.1.1). Figure 3 has two blocks that have a Directed Composition relationship between them. Block_1 is the whole and block_2 is part of the whole. Block_2 is composed of two part diagrams. The relationship that exists between block_2 and part_1 and part_2 is an aggregation. Which means block_2 is the whole and is composed of part_1 and part_2.

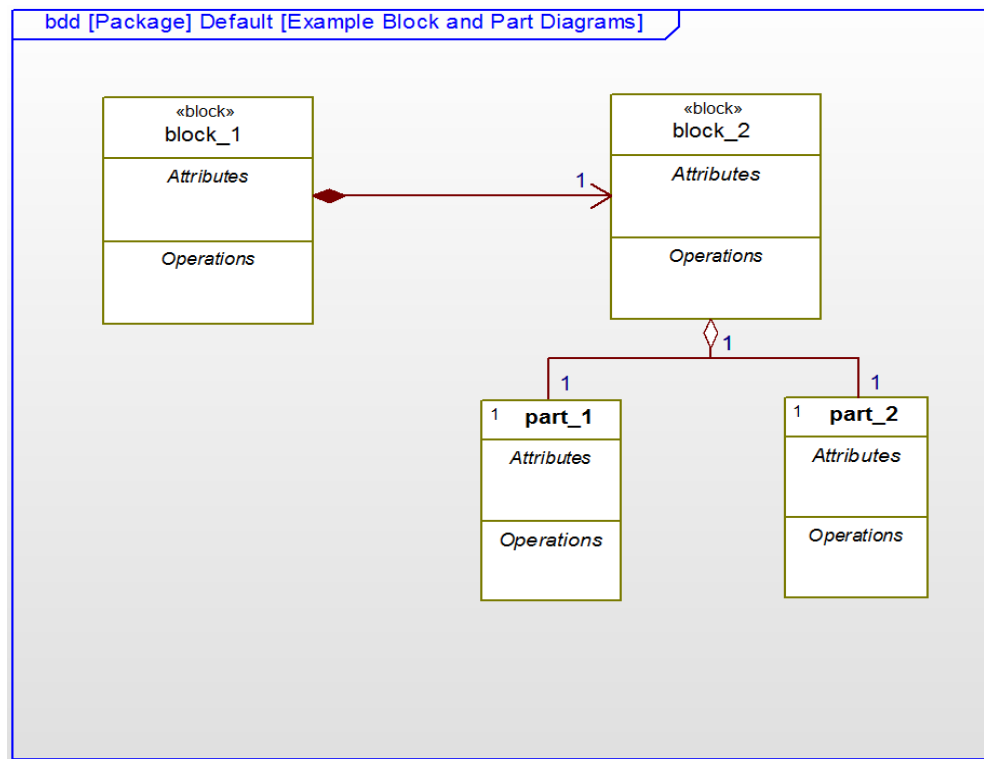


Figure 3: A Block Definition Diagram with Directed Composition Relationship

Figure 4 contains two blocks and two parts. Block_3 and block_4 have the flow type of relationship. This means block_3 provides data or commands to block_4.

Block_4 has a generalization relationship with part_3 and part_4. This means part_3 and part_4 are derived from block_4.

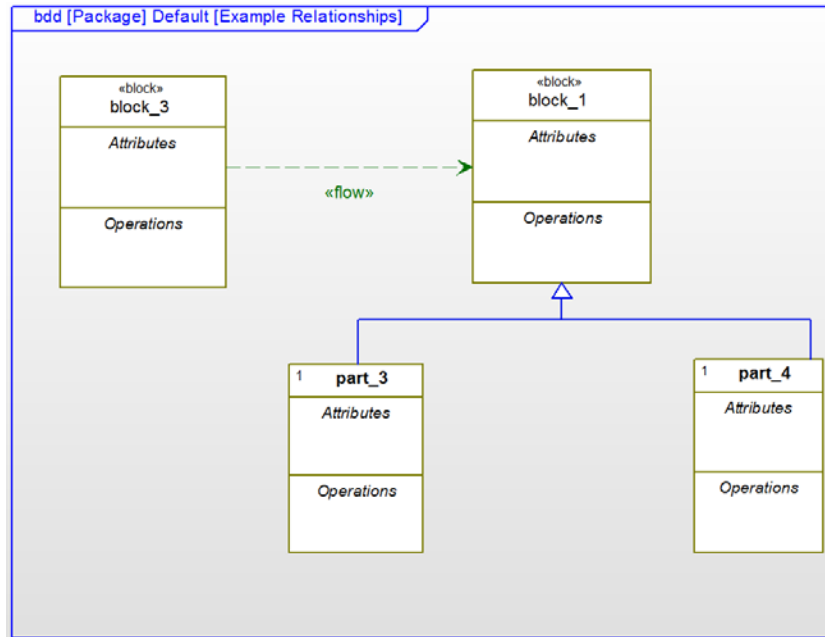


Figure 4: A Block Definition Diagram with Flow Relationship

2.3.2.1.2 Internal Block Diagram

Internal Block Diagrams (IBD) describe the internal structure of each system component represented by a block. IBD consist of parts, blocks, connectors, and ports. Connectors and ports specify interconnections of the parts or blocks. Flow ports and standard ports are the two main types of ports. A flow port is used to show block input and output of materials, data, or energy. A standard port is used to show the exchange of services. A service is a functionality that a block provides or requires.

Figure 5 shows an IBD of block_1 in Figure 3. The IBD is composed of block_5 and four part diagrams. There is an exchange of services between Part_5 and Part_6

through standard ports. Part_5 provides service to part_6. Part_6 requires service from part_5. Part_7 and part_8 exchange physical items (e.g., material, data, energy, etc.) through flow port interaction. Finally, the blue dashed arrow lines indicate a dependency relationship between block_5 and part_5 and part_6. Also, part_6 depends on part_7 and part_8.

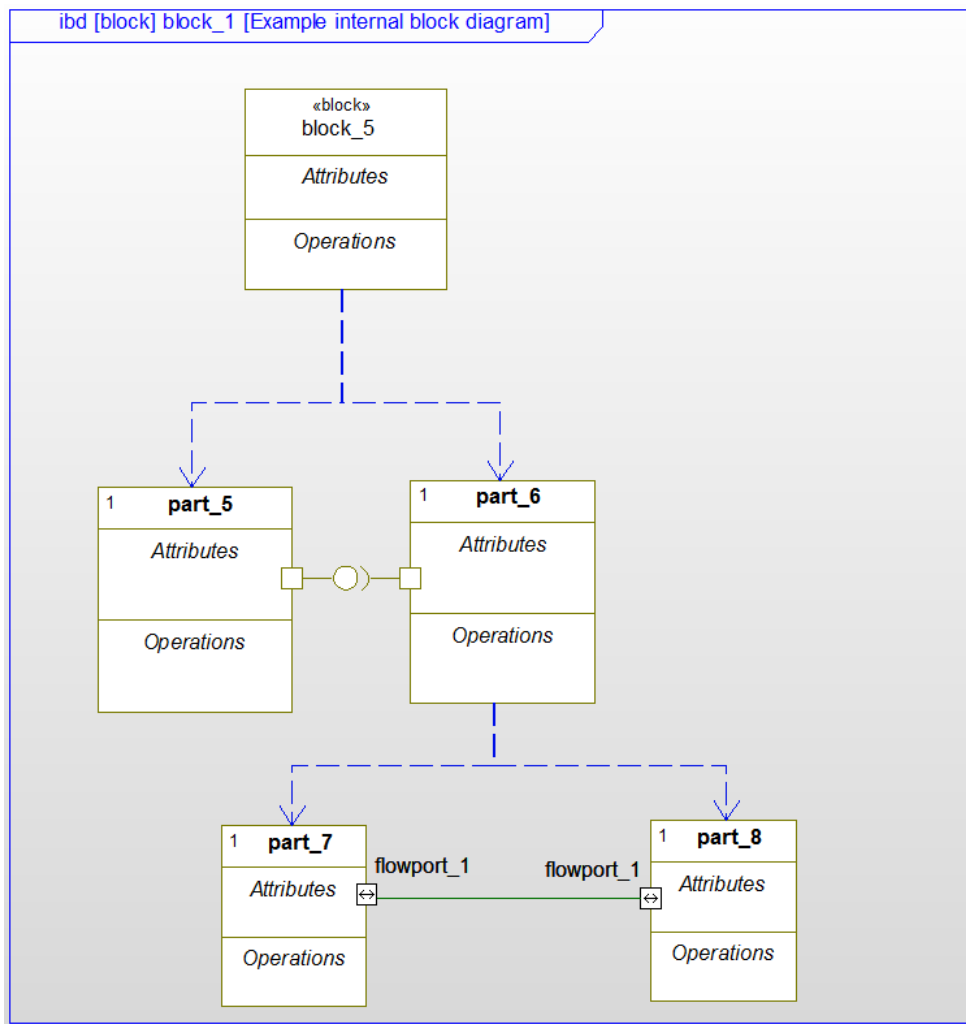


Figure 5: An Internal Block Diagram

2.3.2.1.3 Package

Package diagrams are an organizational feature of SysML for grouping model elements into logical components and associate dependencies between the packages. This organizational advantage is intended to help manage large complex systems. Eventually, most system models become very large and unwieldy, which makes it necessary to structure them into higher-level packages (Peak et al., 2007).

Packages can be used in Block Definition Diagrams or Requirements Diagrams. Figure 6 shows several package diagrams. Package_2 has a dependency relationship with package_1 and package_3.

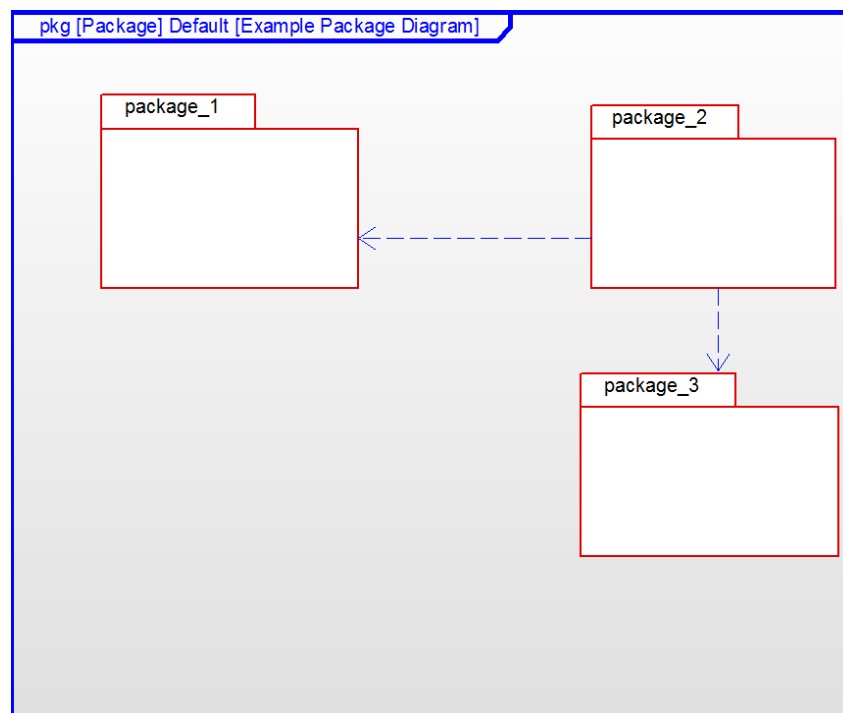


Figure 6: A Package Diagram

2.3.2.2 Behavior

Behavior diagrams describe the system functionality, component interaction, and processes. The system behavior diagrams include use case, activity, sequence, and state machine diagrams. Use case diagrams illustrate system functionality. Activity diagrams show the flow of data and information between activities. Sequence diagrams describe the interaction between different parts of the system and the interaction between actors and the system or components of the system. The state machine diagram describes the actions that a system performs in order to complete an event. Activities are the basic unit of behavior used in activity, sequence, and state machine diagrams (Hause, 2006).

2.3.2.2.1 Sequence Diagram

Sequence diagrams describe the interaction between different parts in the system and the interaction of actors and the system or component of the system. Sequence diagrams and state machines are widely used to model control flow (Viehl, Schönwald, Bringmann, & Rosenstiel, 2006). Sequence diagrams allow a graphical representation of interactions between the system and the user or interactions between different components of the system. Figure 7 shows a sequence diagram. It is composed of SysML system borders, messages and timeouts. The system borders can be used to represent the end user, environment, system components, and/or system interface. The SysML notation is a column of diagonal lines with a rectangular box on

top. Message arrows are used to show the exchange of communication or interaction between the system and the user or between individual system components. The SysML notation for messages is a green arrow. The tip of the arrow indicates where the communication is received and the end of the arrow line indicates the origins of the communication. Timeouts are used to show processing time. SysML uses half a square with a small square on the top end and an arrow on both ends. See Figure 7 for an example of a timeout. Figure 7 shows a sequence diagram that represents the flow of exchange in chronological order between an end-user and part_1 and part_2 of the system. The scenario shows the end-user interacting with part_2. Part_2 processes the request by the end-user. Tm(7) shows part_2 processing the request. After part_2 processes the request, it send a message to the end-user. The end-user then sends a message to part_1. Part_1 receives the message and responds with a message back to the end-user.

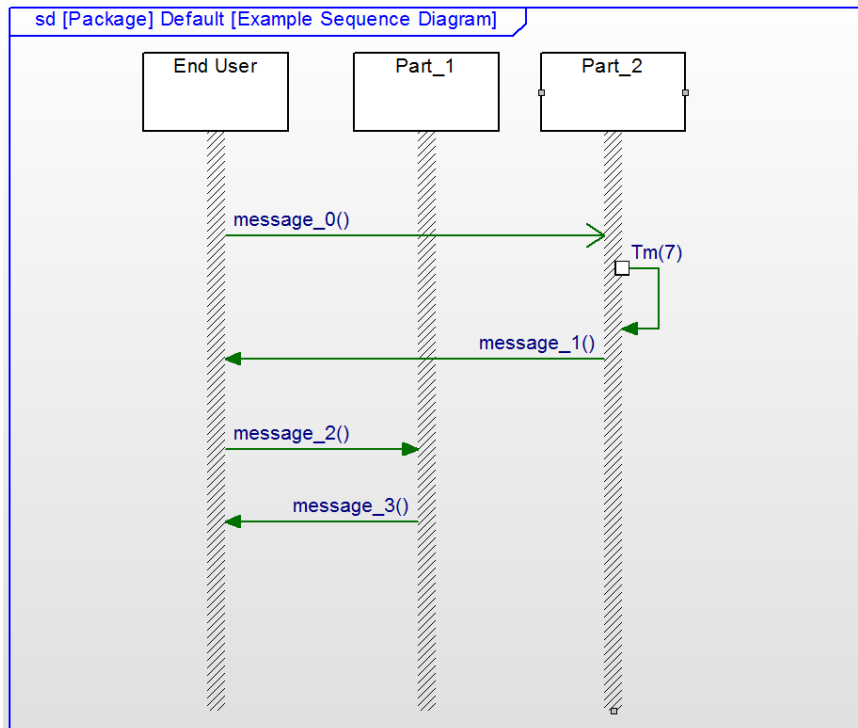


Figure 7: A Sequence Diagram

2.3.2.2.2 Activity Diagram

Activity diagrams are used to model system process flows and describe operational step-by-step workflows. Activity diagrams can also indicate required inputs by actions and outputs produced by actions (Ahmad, 2007). In other words, activity diagrams serve as flowcharts for system processes. Activity diagrams are essential for behavioral modeling in SysML. Activity diagrams are typically constructed with action elements, send action elements, decision nodes, join nodes, fork nodes, swimlanes, and control flow lines. An action element is a unit of system functionality. The functionality represents a process in the modeled system. Actions elements are necessary to construct activity diagrams. The SysML notation for an action is a rounded

corners rectangle. The send action elements are used to transmit a procedure to an activity element. The send action elements can be used in state machine diagrams and activity diagrams. The SysML notation for a send action element is a rectangle with a triangle attached to one end.

A decision node provides the capability to choose between two or more possible paths in an activity diagram (Jarraya, Debbabi, & Bentahar, 2009). A decision node has one input flow and multiple output flows. Decision nodes are represented by a diamond shape in SysML.

Join and fork nodes are used to show process or workflow concurrency. A join node is used to show the combining of two or more input flows into a single output flow. The SysML notation for a join node is a horizontal bar with two red arrows pointing down on the top half of the bar and one red arrow pointing down on the bottom of the bar. A fork node is used to show the division of a single input flow into two or more output flows. The SysML notation for a fork node is a horizontal bar with one red arrow pointing down on the top half of the bar and two red arrows pointing down on the bottom of the bar.

Swimlanes are used to allocate responsibilities of actions between blocks, parts, or actors. A swimlane is a large rectangle located in the background of an activity diagram. The individual lanes are separated by vertical lines and can be labeled with the appropriate name of the block, part, or actor.

A control flow is used to connect activity diagram elements together. Control flows provide a path to follow through an activity diagram. Control flows are represented by red dashed arrows in SysML.

Figure 8 shows an activity diagram. The red solid line arrow with the dot on the end indicates the start of the activity diagrams. Control flow runs from the start of the activity to the end in order. The swimlanes show who is responsible for completing each decision or action. The decision nodes show a condition before proceeding to the next action in the activity diagram. The condition is normally a yes or no question. The decision nodes will have one control flow input and two control flow outputs. A fork node is used after action_2 which leads to action_4 and decisionnode_2. A join node is used for decisionnode_2 and action_5 which leads to action_6. The activity diagram is concluded after action_6. The encircled black circle represents the end of the activity diagram.

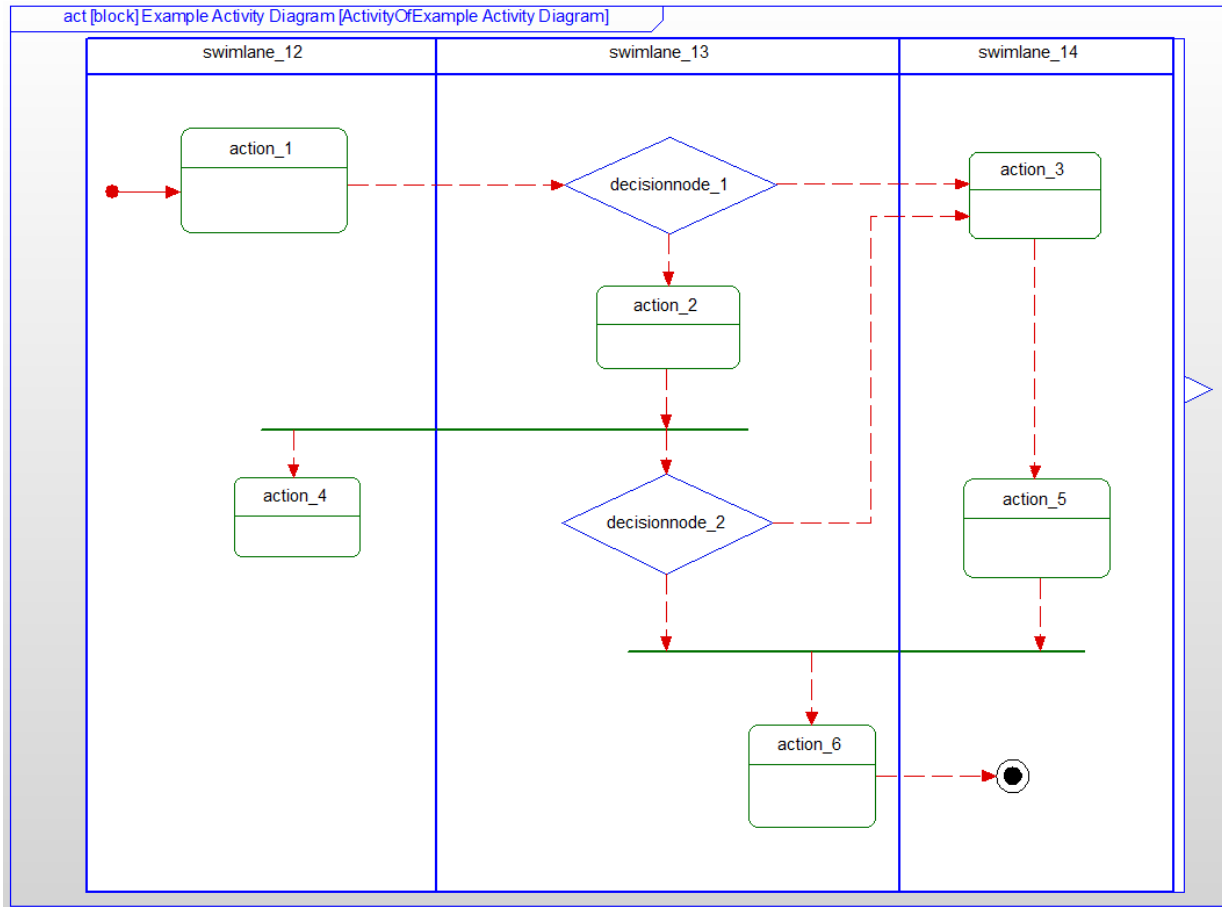


Figure 8: An Activity Diagram

2.3.2.2.3 State Machine Diagram

The state machine diagram describes a system transition between different states when it is performing actions to complete an event. System behavior is represented in terms of transitions and states. States and transitions are the main elements of a state machine diagram (Weilkiens, 2008). State machine diagram elements consist of states, send actions, and transitions. State elements are used to show the system's life cycle path as it transitions from one state to another. The SysML

notation for a state is a rectangle with rounded corners and a line through the top half. The send action elements are used to transmit a procedure for changing the state of the system. The SysML notation for a send action element is a rectangle with a triangle attached to one end. Finally, transitions provide an ordered path to follow through a state machine diagram. Transitions are represented by solid red arrows in SysML.

Figure 9 shows a state machine diagram. A default transition red arrow with a dot on the end shows the start of the state machine diagram. The system is currently in state_0. The sendaction_1 elements transmit a procedure to change the state of the system. The new state of the system is state_1. The encircled black circle represents the end of the state machine diagram.

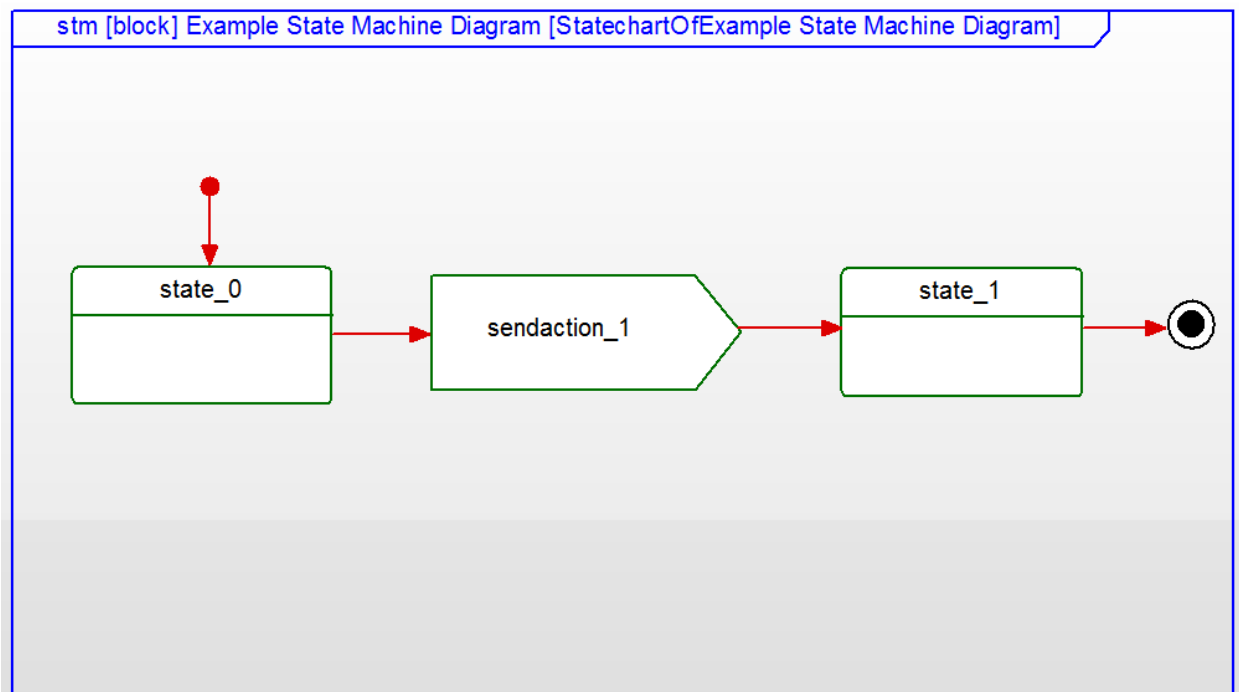


Figure 9: A State Machine Diagram

2.3.2.2.4 Use Case Diagram

Use case diagrams illustrate system functionality in terms of an actor's goals. A use case describes the interaction between an actor and the system. It also shows which system functions are performed by which actor or actors. Use case diagrams are typically composed of actors, use cases, associations and generalization. The actor(s) represent a user that interacts with the system. The SysML notation for an actor is an outline of a person. A use case is used to describe the functionality of a system. The notation for a use case is an oval.

An association shows interaction between an actor and a use case. Associations are represented by solid red lines. A generalization is another form of relationship in use case diagrams. A generalization is used when a use case has common properties and behaviors with a more general use case. The SysML notation for a generalization is a solid blue line with a clear triangle at the end. Include and extend is another form of relationship in use case diagrams. An include relationship provides more detail to describe one use case. Include means the use cases are required. An extend relationship expands on the capabilities of a use case. Extend means the use case is optional. The notation for include is a dashed blue arrow with the label "include" next to it. The notation for extend is a dashed blue arrow with the label "extend" next to it.

Figure 10 shows a use case diagram. The actor is associated with usecase_1 and usecase_2 elements. Usecase_3 and usecase_4 are required for usecase_1. This

is an include relationship. Usecase_5 is optional for usecase_2. This illustrates the extends relationship.

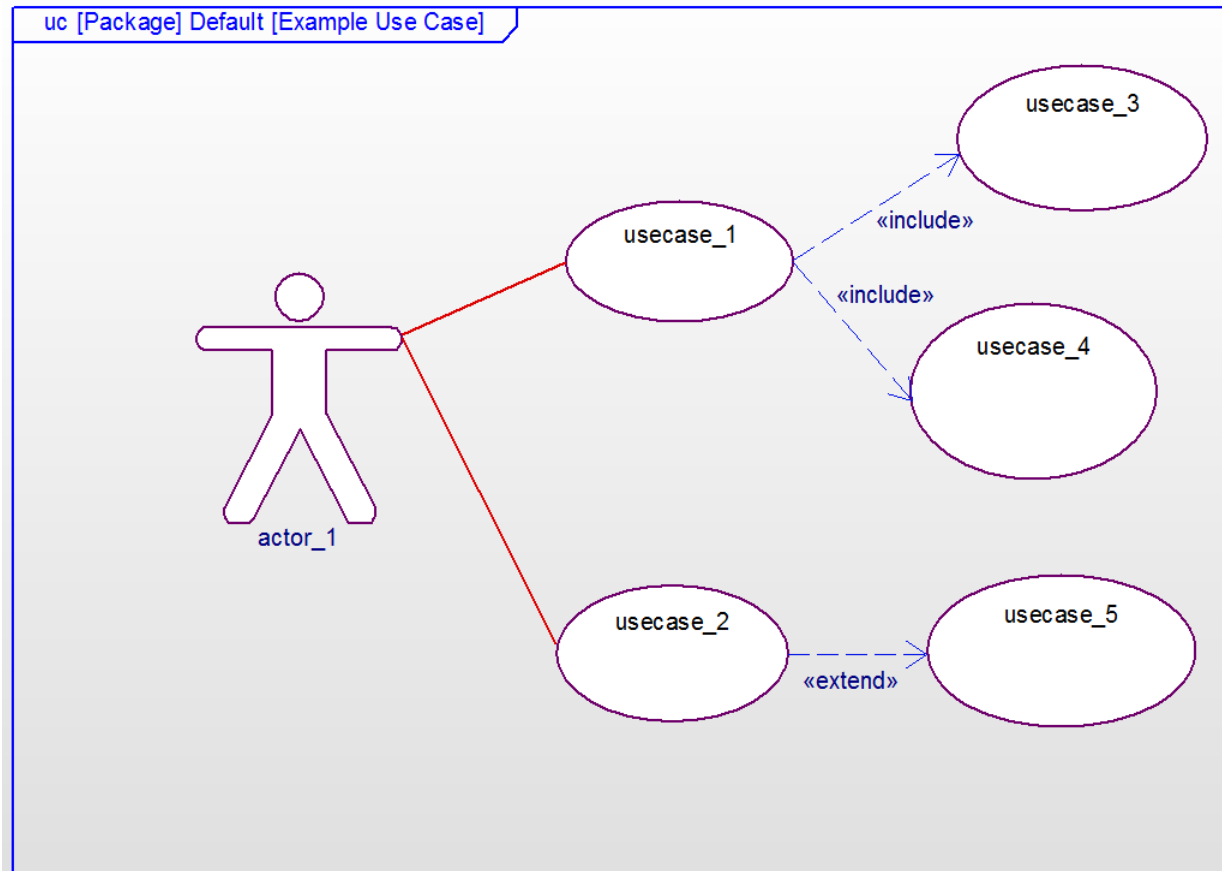


Figure 10: A Use Case Diagram

2.3.2.3 Requirements Diagram

Requirements diagrams graphically represent text-based requirements. In addition, requirements diagrams provide traceability between requirements and system models (Herzog, Pandikow, & Syntell, 2005). This capability to relate model diagrams to system requirements is critical for system verification. One of the significant improvements of SysML over UML is the ability to represent requirements and relate

them to the model of a system (Vanderperren & Dehaene, 2005a). Requirements consist of requirement blocks and derive, satisfy, and verify relationships.

A requirement block is used to describe one or more properties of a system that have to be met by the system. Requirement blocks are represented by rectangular blocks with a requirement stereotype in the top part of the block. Satisfy relationships are used to show that a model element satisfies a particular requirement. The notation for a satisfy relationship is a blue dashed arrow with the label “satisfy” next to it. Verify relationships are used to determine whether a model element fulfilled a requirement. The notation for a verify relationship is a blue dashed arrow with the label “verify” next to it. Derive relationships are used to show a dependency between two requirements. One requirement is the source and the other is the derived requirement. The notation for a derive relationship is a blue dashed arrow with the label “derive” next to it.

Figure 11 shows a requirement diagram. The requirement diagram shows that package_6 satisfies requirement_1. Requirement_1 derives requirement_2. Which means that requirement _1 and requirement_2 are dependent on each other. Finally, in the last relationship shown, testcase_1 is used to verify requirement_2.

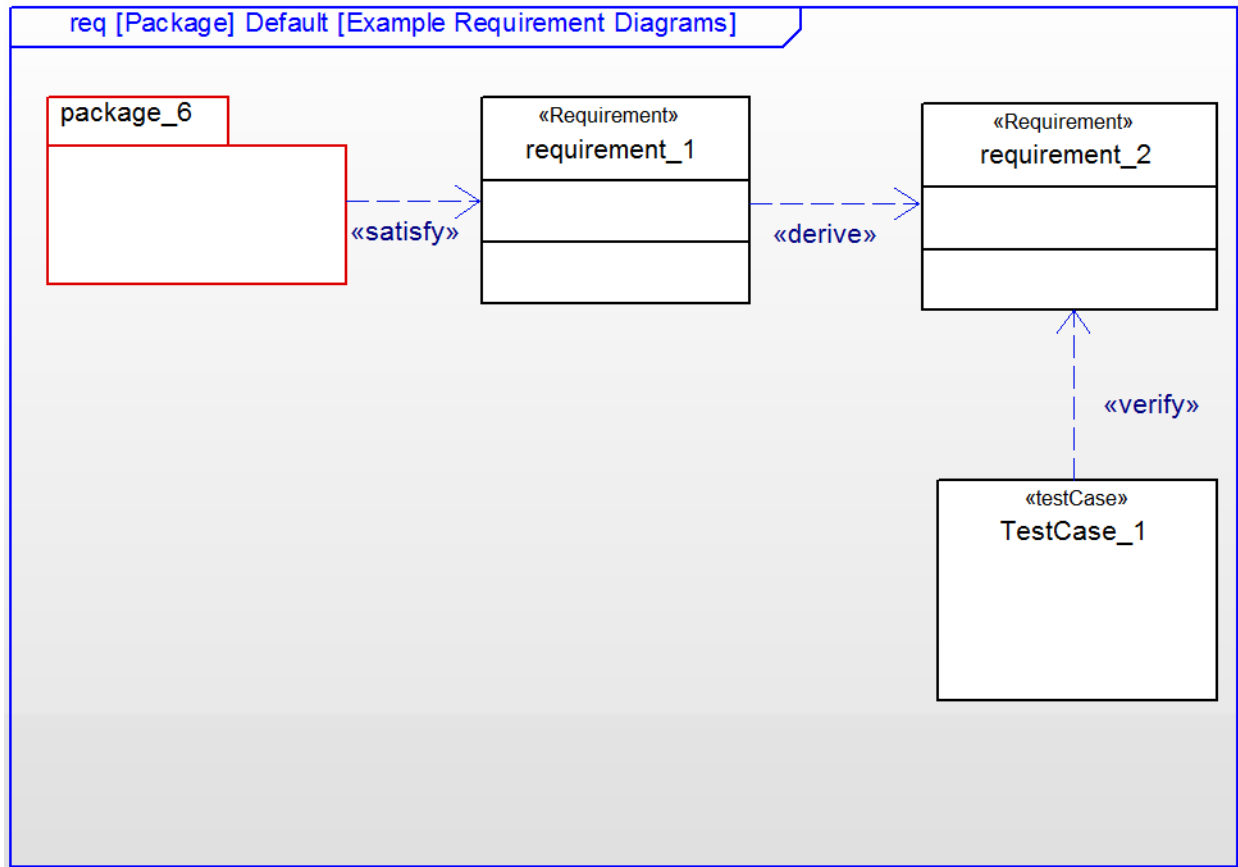


Figure 11: A Requirements Diagram

In summary, Hause and Thorn observed that systems engineers dealt with many different categories of requirements throughout the development life cycle. SysML requirements models provide systems engineers with a greater visibility of requirements, a direct means of traceability, and a holistic system view to conduct impact analysis (Hause, 2007).

2.3.2.4 Parametric

The parametric diagram graphically illustrates the physical or intentional constraints of the system. A constraint is an operation boundary that cannot be exceeded. Parametric diagrams are used to identify system performance parameters, quantitative analysis, and trade-off analysis. The main element of parametric diagrams is constraint blocks which are used to represent system constraints. Mathematical equations within the constraint blocks are used to express system constraints. Standard math symbols are used to represent relationships between properties of different model elements, but no formal language is used to define these relationships (Herzog et al., 2005). Constraint parameters represent system boundaries and limitations. The notation for constraint parameters is a small box located within the constraint block. The relationship that exists between constraints is a binding connector. A binding connector between two constraints indicates that the properties at both ends of the connector have equivalent values. The parameters used in a constraint block can be linked to the properties of another block or constraint block using binding connectors (Johnson, Paredis, & Burkhart, 2008). Figure 12 shows a parametric diagram.

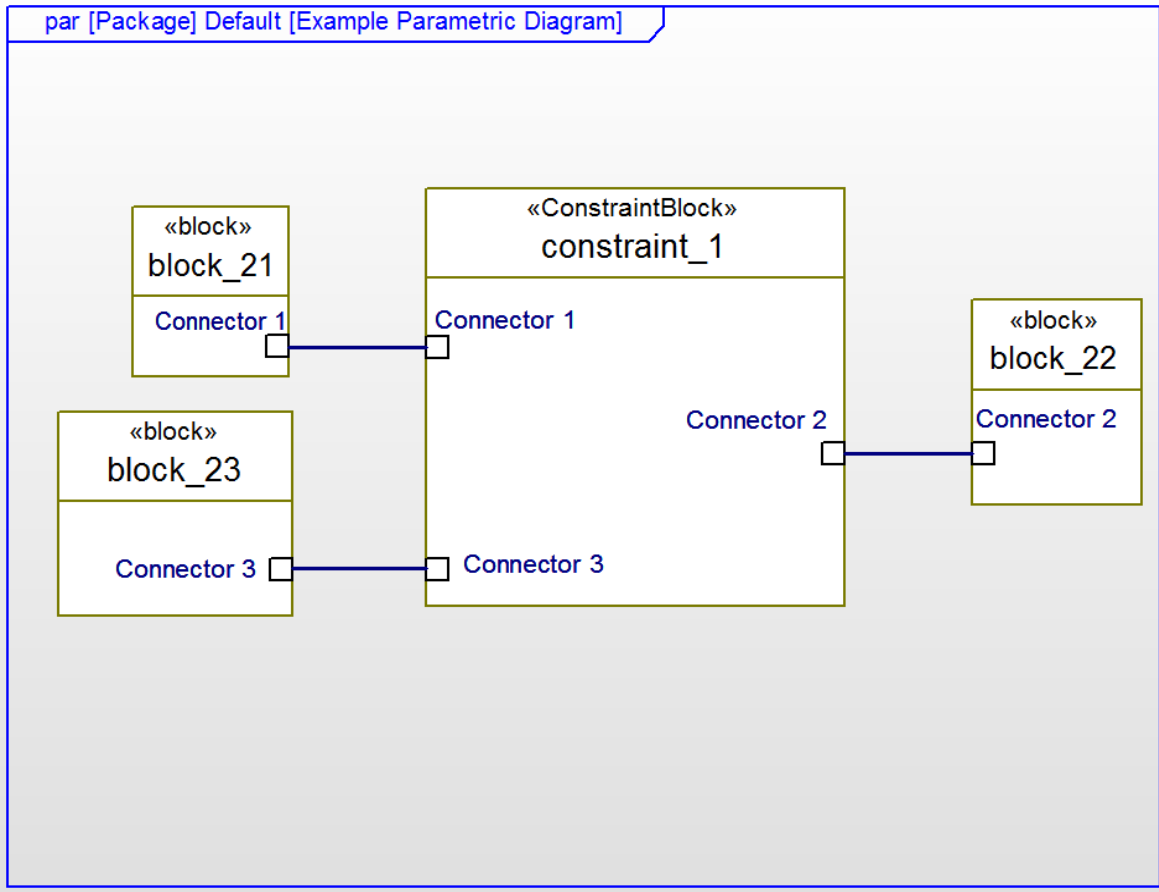


Figure 12: A Parametric Diagram

2.3.3 SysML Elements for Cognitive Work Analysis

2.3.3.1 SysML Elements for Work Domain Analysis

The first phase is Work Domain Analysis (WDA). This phase contains the physical and/or intentional constraints. The WDA determines what can be accomplished within the boundaries of the sociotechnical system. Figure 13 illustrates the use of the Abstraction Hierarchy (AH) tool to determine the functional purpose and physical components of a system. Figure 13 was created using SysML. The AH will be

constructed in a Block Definition Diagram (BDD). BDDs are used to describe the hierarchical and component structure of the system. The BDDs also can be used to identify the interconnections and relationships between blocks. Elements at the highest level of the AH model define the purposes and goals of the system. Elements at the lowest levels of the model indicate and describe the physical components of the system. The lowest levels of the AH, Levels 4 and 5, are represented by part diagrams. Level 3, the General Function, is represented by block diagrams in the BDDs. Level 2, the Abstract Function on the AH, is represented by constraint blocks within SysML. Constraint blocks can be used to identify performance parameters or other mathematical constraints which can be used to support trade-off analysis. Level 1, the Functional Purpose of the system, is represented by blocks. The AH is linked to the Control Task Analysis (ConTA) model through the General Functions.

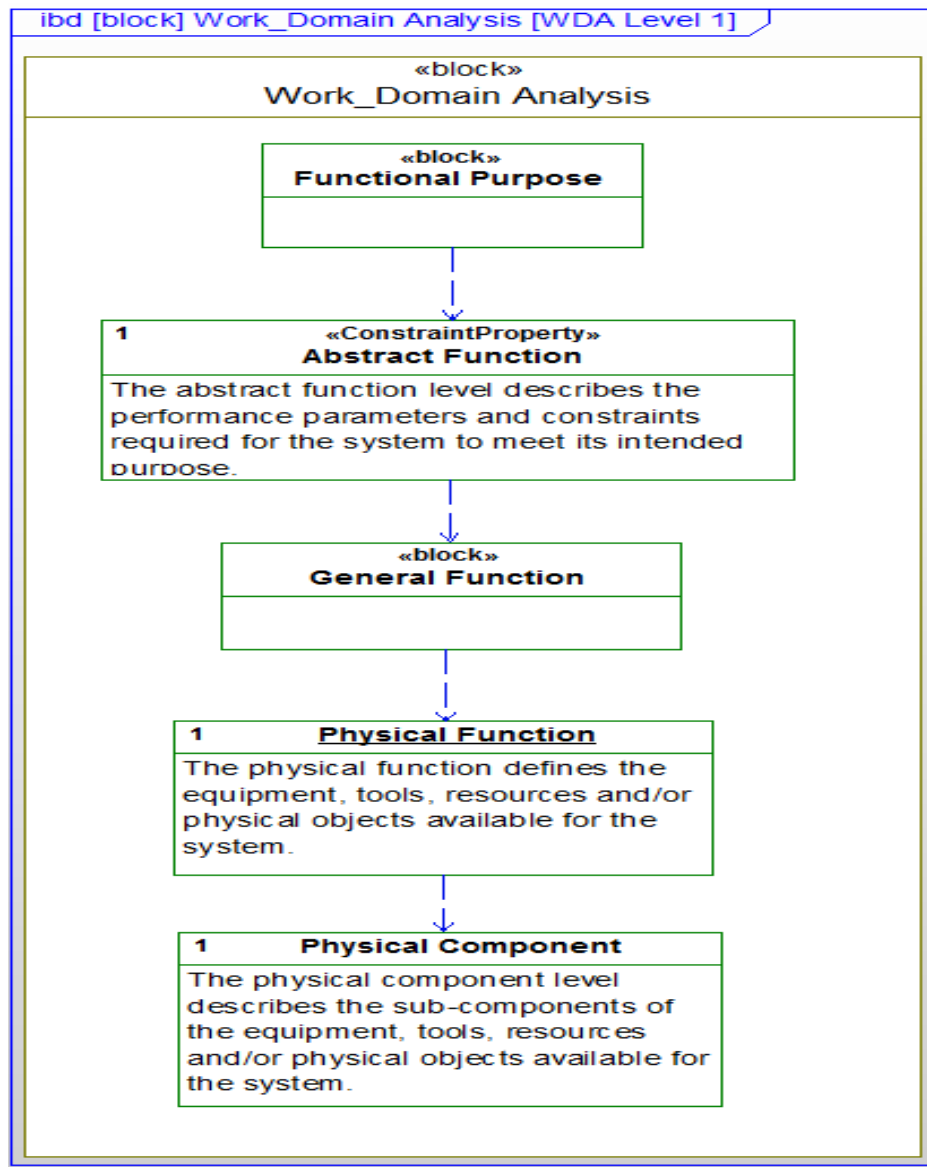


Figure 13: A Template Abstraction Hierarchy Constructed Using SysML

2.3.3.2 SysML Elements for Control Task Analysis

The second phase of the CWA framework is Control Task Analysis (ConTA). ConTA covers what needs to be done within the limits of the work domain. The Decision Ladders (DL) tool is used to model the second phase. A DL shows the alternative

courses of action for a particular decision. It is a useful way of representing detailed process knowledge. A decision ladder constructed with SysML is shown in Figure 14. In order to complete the General Function, the tasks in the DL have to be completed, and the user will use the physical functions and components to accomplish the tasks. Every step in the DL process does not have to be accomplished in order to execute the completion of the General Function. This provides flexibility to the user for unanticipated events or adaptive learning (Lui, Watson, & Queensland, 2002). Normally, rectangles represent information processing activities and the circles represent resultant knowledge states required to complete component tasks. In SysML, information processing activities are represented by send action elements and knowledge states are represented by state elements. The ConTA phase is linked to the Strategies Analysis through the “Formulate Procedures” send action element in Figure 14.

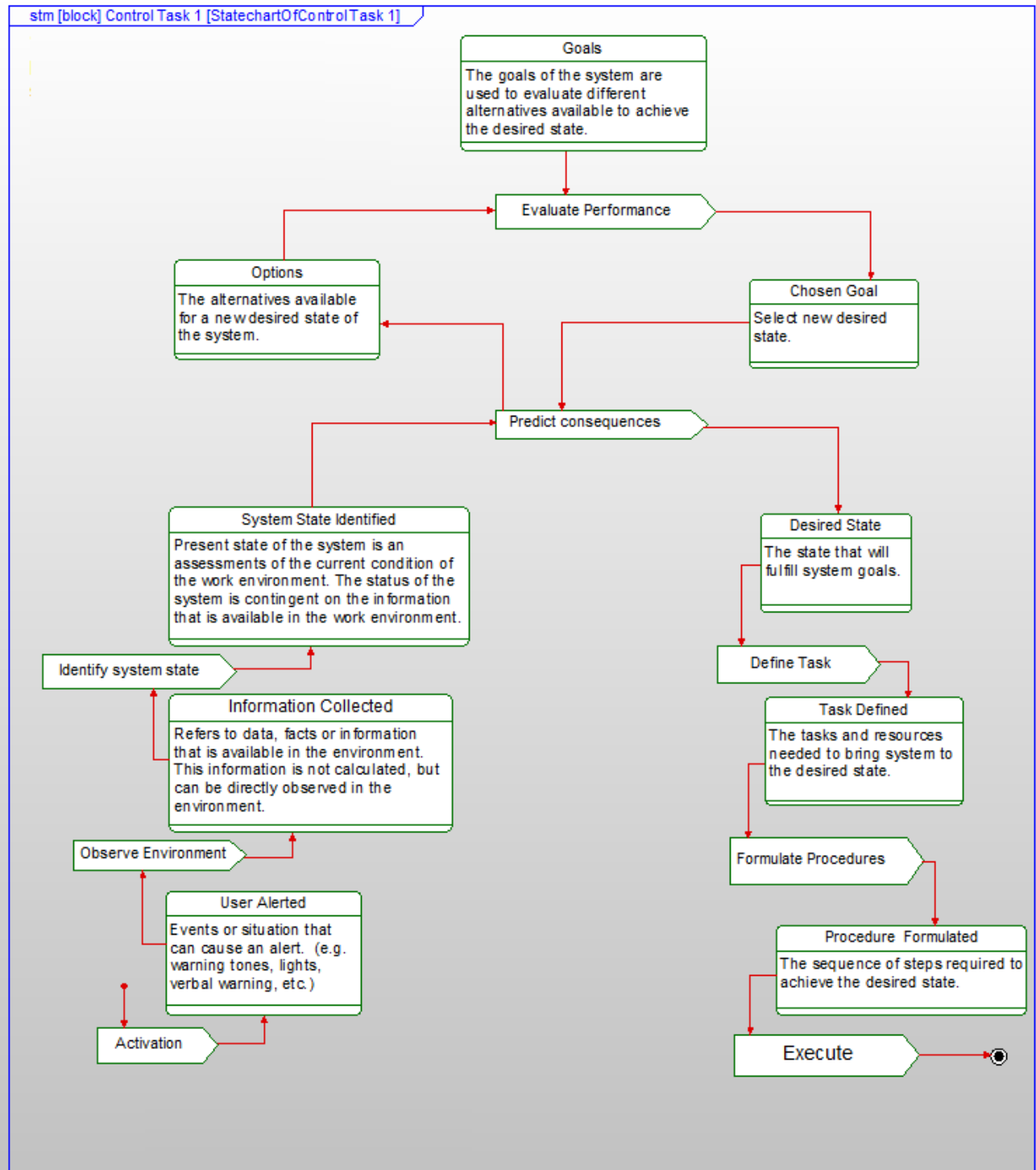


Figure 14: A Template Decision Ladder Constructed Using SysML

2.3.3.3 SysML Elements for Strategies Analysis

Strategies Analysis (SA) is the next phase in the CWA framework. The purpose of SA is to determine which actions are necessary to achieve the control tasks. The same control task can be completed in a variety of ways using different cognitive strategies (Jenkins et al., 2008). Information Flow Maps (IFM) are a graphical representation of information processing activities that depict how a user can perform a sequence of tasks to reach an end goal. Figure 15 illustrates the sequence of tasks and one way to represent an IFM in SysML. Figure 15 shows one strategy that is available to the system user to complete a task. Normally, there will be other user strategies modeled using the IFM. By performing one of the strategies, the user can complete the Control Task. By completing the Control Tasks, the user satisfies the Functional Purpose of the system. Typically, rectangles represent information processing activities and circles represent resultant knowledge states which are required to complete control tasks. State machine diagrams are used to construct the IFMs. Send action elements will represent information processing activities and state elements will represent knowledge states.

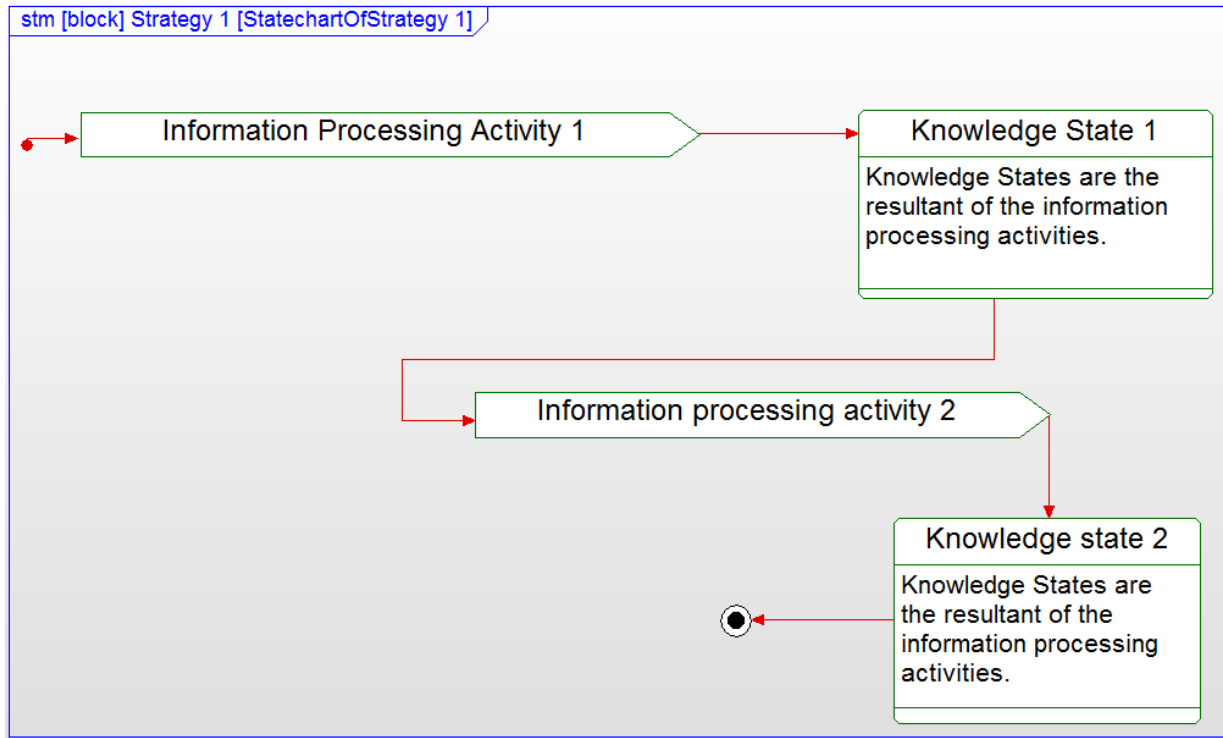


Figure 15: A Template Information Flow Maps Model Constructed Using SysML

2.3.3.4 SysML Elements for Social Organization and Cooperation Analysis

The Social Organization and Cooperation Analysis (SOCA) phase determines who will carry out the work and how it is shared. The IFM tool is typically used in the SOCA phase of the CWA framework. The IFM identifies the actors and their roles. In general, the IFM answers the question of who will do what tasks by allocating responsibility among actors. In SysML, use case diagrams are used to show distribution of tasks among actors and represent the SOCA phase modeling tool. Figure 16 illustrates how a use case diagram represents the allocation of responsibilities for completing a specific task.

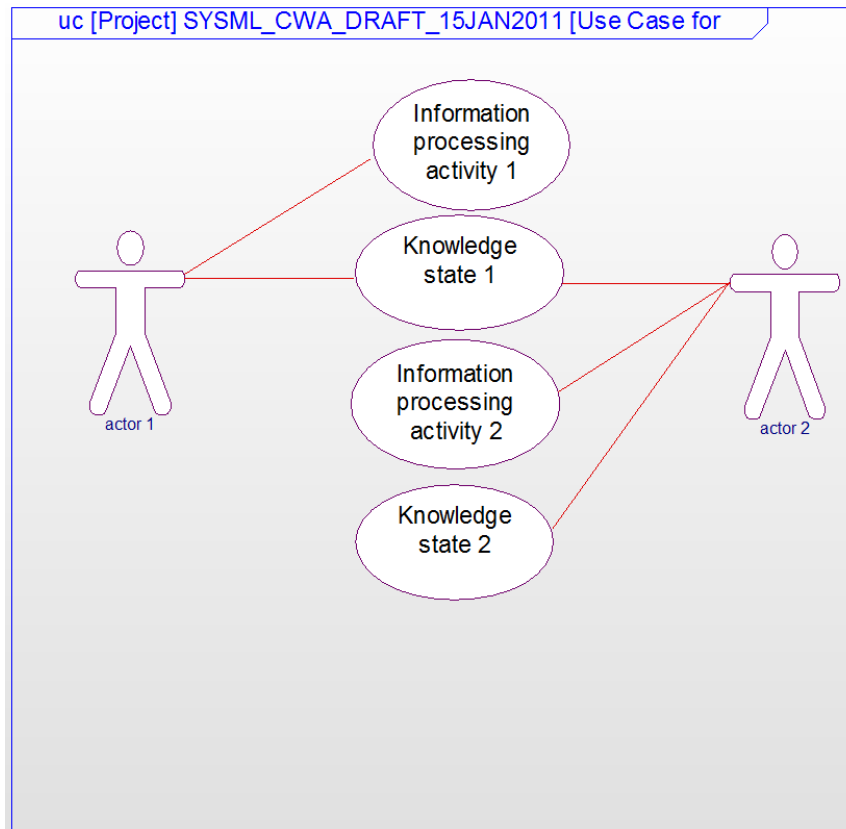


Figure 16: A Template Use Case Diagram Constructed Using SysML

2.3.3.5 SysML Elements for Worker Competencies Analysis

The final phase of the CWA framework is the Worker Competencies Analysis phase, which contains the level of conscious effort required to complete an information processing activity. The level of conscious effort measures the physical, perceptual, and cognitive demands placed on the worker using the system (Jenkins et al., 2008). Skill, Rule, and Knowledge-based (SRK) inventory is the modeling tool used in this phase. SRK ascertains the level of conscious effort an individual uses when processing information. The skill-based behavior (SBB) category requires almost no conscious

effort when completing a task. For example, when a person is approaching a red traffic light while driving, the person instinctually steps on the brakes. The rule-based behavior (RBB) category is centered on the rules of the organization or the proper procedures learned in training. Unlike the skill-based category, the knowledge-based behavior (KBB) category requires a high level of conscious effort to complete a task. Examples include when a person learns to drive a car for the first time or when an operator encounters an unanticipated event. Figure 17 shows the different levels of effort required by the system user to complete a task and how each task is represented in SysML. The WCA phase is linked to each information processing activity on the Decision Ladder (DL) in the Control Task Analysis (ConTA) phase and on the Information Flow Maps (IFM) in the Strategies Analysis (SA) phase of CWA. An activity diagram is utilized to construct the SRK inventory in SysML. The send action elements represent the information processing activity. SysML action diagrams represent skill, rule, and knowledge based-behaviors. Finally, swimlane elements are used to organize the action diagrams into behavior categories.

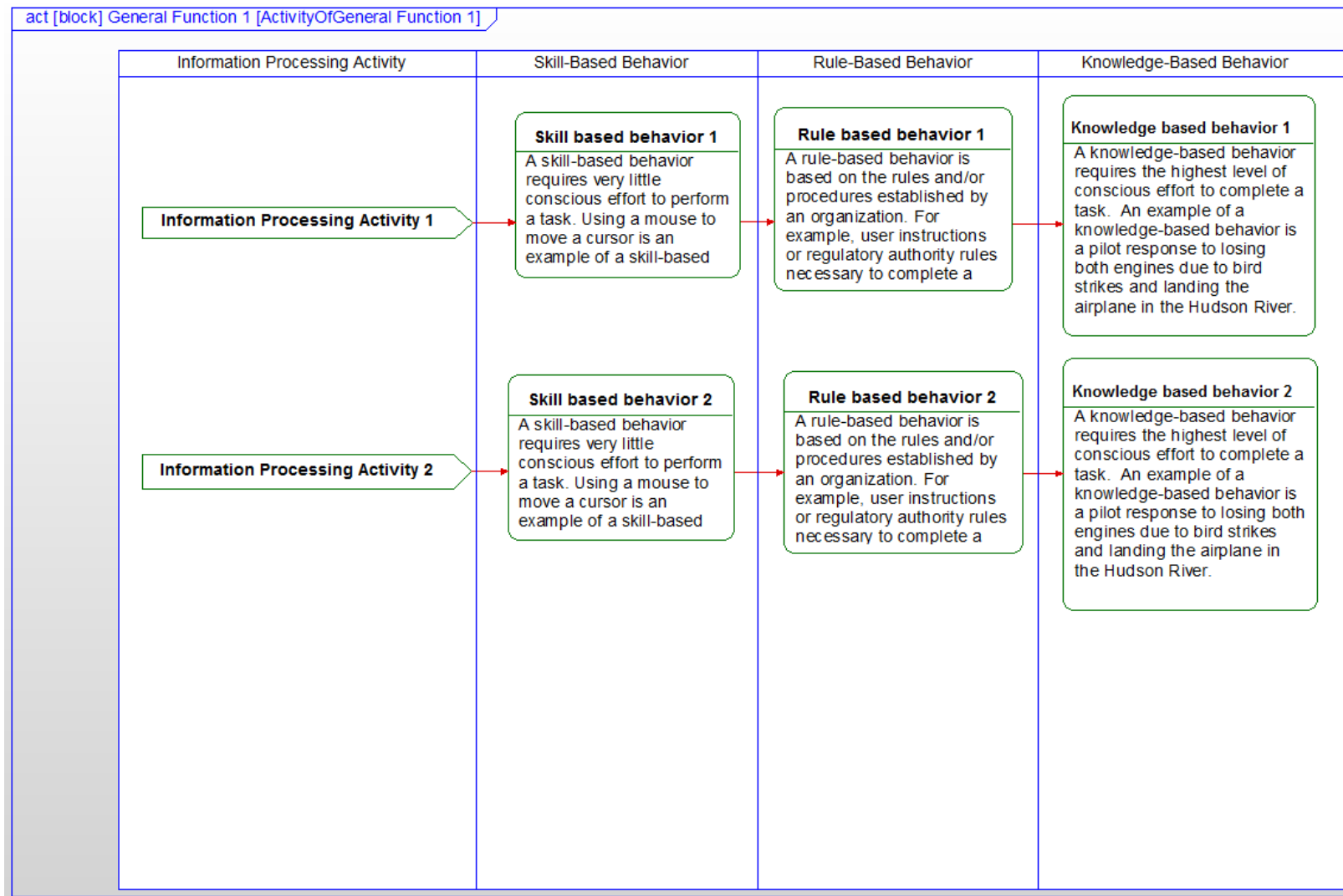


Figure 17: A Template Skill, Rule, and Knowledge-Based Inventory Constructed with SysML

2.3.4 SysML Requirement Development Function

SysML employs use case diagrams, requirements diagrams, and requirements tables to define system requirements. A use case is a defined task of interest to the user. It contains the user intentions and system responsibilities in the course of accomplishing that task (Constantine & Lockwood, 2001). Use cases provide a means for describing basic functionality in terms of usages of the system by users. The main advantage of use cases is their simplicity. The process can be described as: find actors, find use cases, and describe the use cases (Armour & Miller, 2001).

When modeling system requirements, use cases have several disadvantages according to Soares & Vrancken (2008). First, SysML use case diagrams as well as requirements diagrams are not consistent in format or structure, which may lead to differences in interpretations by stakeholders (Soares & Vrancken, 2008). SysML use case diagram structure is not standard and normally varies between different organizations because the process for developing use cases varies across organizations. There are no generally accepted and well-defined notational standards existing for requirements engineering in SysML (Insfrán, Pastor, & Wieringa, 2002).

Secondly, current use cases narratives are not sufficient to support requirement development of larger and more complex sociotechnical systems (Constantine & Lockwood, 2001). Expressions of large, complex sociotechnical system models in UML and SysML are currently difficult to read and comprehend. Constantine and Lockwood also state:

“Integrating actors with use case relationships in a single model leads to bewildering jumbles of lines for all but the most trivial problems of the sort found in books, articles, and tutorials.” (Constantine & Lockwood, 2001)

The third disadvantage of the undefined use case process is the issue of too many use case models. This situation is referred to as a use case explosion. It detracts from the user goals and describes trivial interactions or incidental actions of the users. Furthermore, not knowing when to stop developing use cases can lead to use case explosion (Lilly, 2000).

2.4 Applications of System Modeling Language

Systems engineers are rapidly adopting Systems Modeling Language (SysML) as the new standard for modeling systems (Friedenthal et al., 2008). SysML is methodology- and tool-independent (Friedenthal, Moore, & Steiner, 2006). Since SysML is methodology- and tool-independent, it can be and has been applied to a variety of domains. These domains include safety engineering, requirements engineering, process plant control systems, manufacturing control systems, satellite communication, Army weapons systems, Human Systems Integration, software project management, and many others.

SysML has been implemented in modeling cognitive handoffs of wireless networks (Gonzalez-Horta, Enriquez-Caldera, Ramirez-Cortes, Martinez-Carballido, & Buenfil-Alpuche, 2010). A cognitive handoff provides a quality, seamless, and secure transition between mobile wireless networks. The proposed theoretical framework is based on a functional decomposition method and scientific problem solving. SysML was chosen for this domain because it shows the dynamic behavior of communications systems and how the wireless network will transition between states (Buede, 2009).

SysML has also been implemented in the safety engineering domain (Hause, 2007). The authors state that system safety needs to be incorporated into all aspects of system development and operation. The study demonstrated how SysML allowed different engineering disciplines to model the different aspects of a system together while working individually. They used an integrated database and ergonomic profile to

support a single holistic model of the system. The main focus of the study was the utilization of SysML elements to support risk identification, risk management, and risk mitigation.

The next paper focused on modeling discrete processes in the production domain. In the past, there has been an issue with simulating complex systems and verifying that the simulation model adequately describes the system (Law & Kelton, 2000). SysML has been used by many to develop production models capable of being transformed with simulation modeling tools into simulated systems (Schonherr, 2009). The research led to the development of an automated approach for model generation in a simulator-based environment based on SysML models. Another study explored the use of SysML for modeling a system and then simulating that system automatically (Huang et al., 2007).

Leon McGinnis and Volkan Ustun also contributed to the process of automating the conversion of SysML models to Arena simulations (McGinnis & Ustun, 2009). They described how Object Management Group (OMG) SysML can be used to create a conceptual model of the system, which can then be automatically translated into a simulation program. In this case the simulation program was Arena. McGinnis and Ustun also demonstrated how SysML model-driven architecture provides a formal approach for developing a conceptual model. They also demonstrated how Extensible Markup Language (XML) and XML Metadata Interchange (XMI) interchange standards

can be used to automatically translate the modeling language into a simulation language.

Another application for SysML is in the mechanical design domain. A case study on the design of a passenger car's luggage compartment showed how many different SysML elements are suitable for mechanical concept design (Wölkl & Shea, 2009). The study compared traditional document-based model techniques against model-based techniques. Typically, document-based techniques produce incompatible documents because a different method is used to describe each aspect of the system (Helms, Shea, & Hoisl, 2009). In addition, computational support for modeling is limited to the office software, which has very little reuse capability (Hirtz, Stone, McAdams, Szykman, & Wood, 2001). In contrast, SysML diagrams and their meaning present a formal modeling approach and capability to integrate a variety of models.

As stated in previous chapters, systems are become more complex and system operators currently have a more difficult job of operating, maintaining, and managing them (Aarts & Roovers, 2003). Dobre, et al. investigated an approach based on Ambient Intelligence to improve the interaction between operators and sociotechnical systems (Dobre, Morel, Pétin, & Bajic, 2008). SysML was used to model the shared activities of the human operator and the automated system.

SysML has been used to model continuous system dynamics (Johnson et al., 2007). Johnson et al. modeled these continuous system dynamics in SysML using differential algebraic equations. Continuous system dynamics models represent energy

or signal exchange between system components. To demonstrate the method, the authors modeled a hydraulic pump.

The next application of SysML is in the software safety domain. There exist deficiencies in traceability between safety requirements and software design. (Briand, Falessi, Nejati, Sabetzadeh, & Yue, 2010). Others contend that software and hardware interface development processes have not matured to a high level of safety confidence (Kaiser, Klaas, Schulz, Herbst, & Lascych, 2011). Another study proposed a framework to enable efficient software design inspections during safety certification (Nejati, Sabetzadeh, Falessi, Briand, & Coq, 2011). This approach helped to reduce cognitive load and errors attributed to overlooked safety issues.

SysML has been integrated with other modeling languages. One of those languages is SOPHIA. SOPHIA is a modeling language for representing safety-related concepts and relationships to system models (Cancila et al., 2009). The study by Cancila et al. focused on the challenges that exist when integrating safety engineering and system design. Another modeling language that was integrated with SysML is Petri Net (Wang & Dagli, 2008). Petri Net is a mathematical modeling language used to describe distributed systems. Wang and Dagli applied SysML and Petri Net to a C4 network system, but it could also be used for general system design. The integration of the modeling languages enables static and dynamic system analysis and verification of system behavior and functionality.

SysML has been used in the maritime domain (Ruegger, 2008). Ruegger modeled a Maritime Domain Awareness system using SysML. The purpose of the system was to monitor the oceans and waterways for any security violations. The system received information from multiple sources. The sources included sensors, databases, and intelligence information. The sources provided a big picture view for the user's situational awareness. The model-based engineering method with SysML assisted in the development of a system with direct links between sources of information and the command-and-control center. This reduced the time it took for an operator to establish the big picture and have a high quality of situational awareness.

The next paper reviewed focuses on applying SysML to system-on-chip (SoC) and network-on-chip (NoC) development (Vanderperren & Dehaene, 2005b). There has been a need for model-driven development with the increased complexity of SoCs and NoCs (Ma & Sun, 2008). Through model-based design, researchers are able to model and develop more capabilities with less power consumption into smaller chips.

The design challenges faced by the aerospace industry present a perfect opportunity for SysML. SysML notation was applied to an aerospace project at Saab Aerosystems (Andersson, Herzog, Johansson, & Johansson, 2009). The authors describe their use of use case, sequence, and activity diagrams in the development of unmanned aerial vehicles (UAV).

In summary, Model-Based Systems Engineering is becoming an accepted method for designing systems in industry, academia, and government. SysML is becoming the standard language to accomplish Model-Based Systems Engineering. Table 3 shows a list of other domains impacted by SysML that have not been discussed in this section.

Table 3: Applications and Disciplines Supported by System Modeling Language

REFERENCE TITLE	AUTHORS	DOMAIN
ACCIDENT PREVENTION BY CONTROL SYSTEM RECONFIGURATION	(Luis de la Mata, J. & Rodríguez, 2009)	Process Plant Control Systems
ON THE SUITABILITY OF MODELING APPROACHES FOR RE-ENGINEERING DISTRIBUTED CONTROL SYSTEMS	(Luder, Hundt, & Biffi, 2009)	Manufacturing Control Systems
COMPLEX TERMINAL SYSTEMS DESIGN: MINIMIZING TIME TO DEPLOYMENT	(Rittenbach, Kovarik, Krause-Aiguier, & Stewart, 2010)	Satellite Communication
ARMY STRATEGIC SOFTWARE IMPROVEMENT PROGRAM STUDY OF REAL-TIME SAFETY-CRITICAL EMBEDDED SOFTWARE-INTENSIVE SYSTEMS ENGINEERING PRACTICES	(Feiler & De Niz, 2008)	Army Weapons System
HUMAN SYSTEMS INTEGRATION SOFTWARE DEVELOPMENT BASED ON SYSML/UML AND IBM RATIONAL UNIFIED SOFTWARE DELIVERY PLATFORM	(Ahram & Karwowski, 2009)	Human Systems Integration
A METAMODELING APPROACH FOR REASONING ABOUT REQUIREMENTS	(Goknil, Kurtev, & Van den Berg, 2008)	Web Architectures for Services Platforms
ARCHITECTING A NET-CENTRIC OPERATIONS SYSTEMS OF SYSTEMS FOR MULTI-DOMAIN AWARENESS	(Ruegger, 2008)	Maritime System of Systems
A NOVEL PROJECT MANAGEMENT THEORY AND ITS APPLICABILITY	(Erguner, 2008)	Software Project Management
MULTI-VIEW MODELING TO SUPPORT EMBEDDED SYSTEMS ENGINEERING IN SYSML	(Shah, Kerzhner, Schaefer, & Paredis, 2010)	Embedded Systems

SYSTEMS ENGINEERING IN THE PRODUCT LIFE CYCLE	(Bock, 2005)	Systems Engineering
ON THE MEANING OF SYSML ACTIVITY DIAGRAMS	(Jarraya et al., 2009)	Verification and Validation
AUTOMATIONML. THE GLUE FOR SEAMLESS AUTOMATION ENGINEERING	(Drath, Luder, Peschke, & Hundt, 2008)	Manufacturing Engineering

2.5 Cognitive Work Analysis Framework Modeled using SysML

This section of the dissertation will expound on the Cognitive Work Analysis (CWA) framework development using SysML. It will explain how the CWA framework is developed with SysML, what SysML model elements are used, how the model elements interact with each other, and what it should look like when it is completed. The SysML figures in this section are approximations of what the modeling framework will look like at the completion of this study.

The CWA approach consists of five interrelated phases. These five phases are:

1. Work Domain Analysis (WDA)
2. Control Task Analysis (ConTA)
3. Strategies Analysis (SA)
4. Social Organization and Cooperation Analysis (SOCA)
5. Worker Competencies Analysis (WCA)

Within each phase of CWA, there is a modeling technique that is commonly used to model cognitive work. These modeling techniques are:

1. Abstraction Hierarchy (AH)
2. Decision Ladders (DL)
3. Information Flow Map (IFM)
4. Information Flow Map (IFM)
5. Skill, Rule, and Knowledge-based Inventory (SRK)

2.5.1 Cognitive Work Analysis Structure Modeled using SysML

The first step in constructing a System Modeling Language (SysML) Cognitive Work Analysis (CWA) framework is to build the CWA structure. The CWA structure provides a way to organize the analysis. The CWA structure is created in a Block Definition Diagram (BDD) using blocks to represent the five phase of CWA. In addition, flow arrows between the block diagrams show the exchange of information between each phase. Also, each block includes a description of the attributes of each CWA phase. Within each of the blocks is an Internal Block Diagram (IBD). More detail about each of the CWA phases is decomposed in the IBD. Figure 18 shows a CWA structure. The CWA structure illustrates the five CWA phases and shows the information exchanged between phases.

2.5.2 Work Domain Analysis Modeled using SysML

The first phase is Work Domain Analysis (WDA). This phase contains the physical and/or intentional constraints. The WDA determines what can be accomplished within the boundaries of the sociotechnical system. The Abstraction Hierarchy (AH) is the method used to determine the functional purpose and physical components of a system. Elements at highest level of the AH model define the purposes and goals of the system. Elements at the lowest levels of the model indicate and describe the physical components of the system. Figure 18 presents users with an overview of the CWA

structure. A description of the attributes of each phase of the CWA process is located in each block. Within the Work Domain Analysis block is a window that contains the

This is a template model of the **Cognitive Work Analysis Structure**.

Copy, paste and modify these models to customize Cognitive Work Analysis structure.

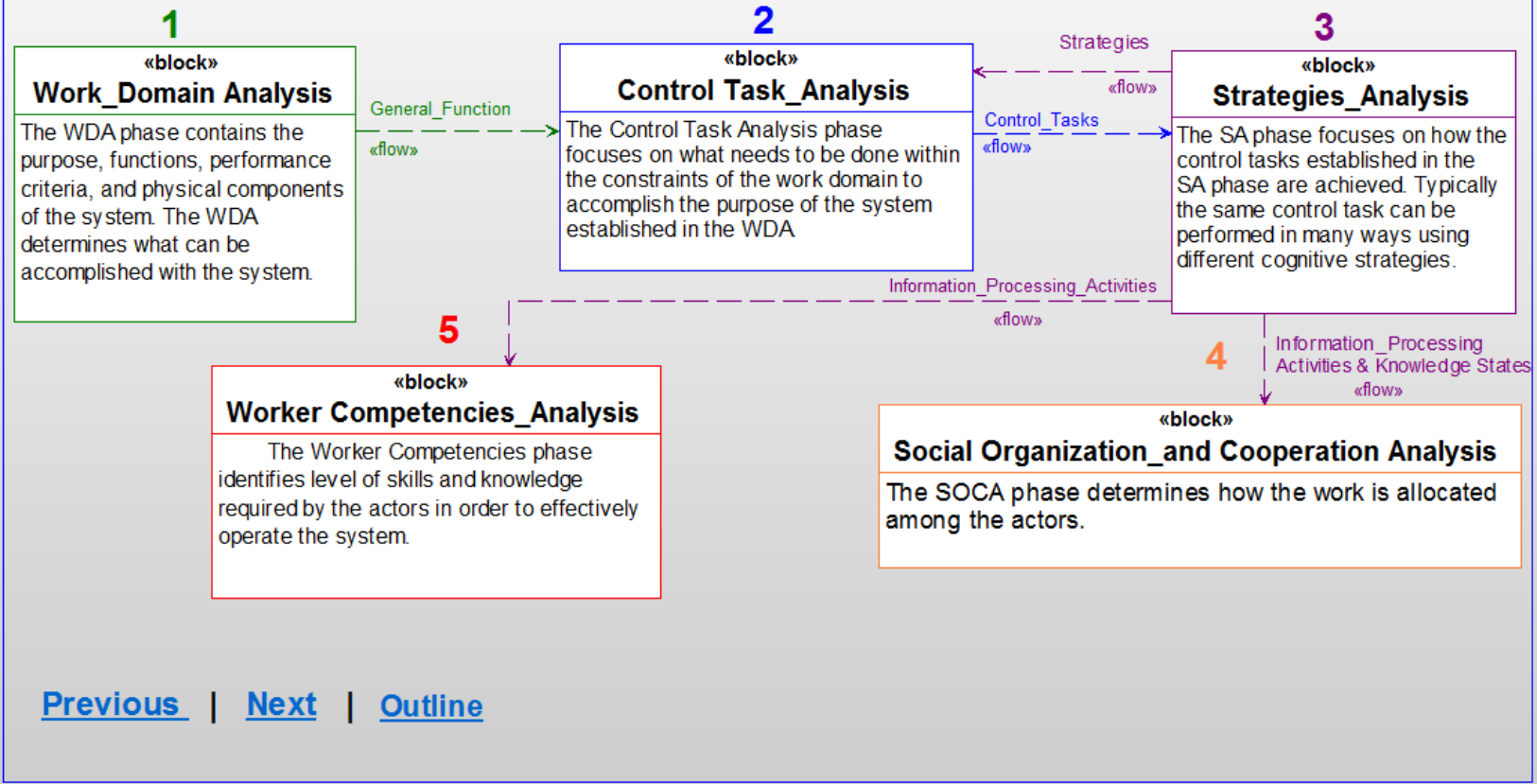


Figure 18: An Example of a CWA Structure

Abstraction Hierarchy (AH), which is the model that is used in the WDA phase. Figure 19 shows an AH model created using SysML. Figure 19 was modeled with SysML blocks, constraint property, and part diagrams. More blocks, constraints property, and parts will be added to actual systems being designed.

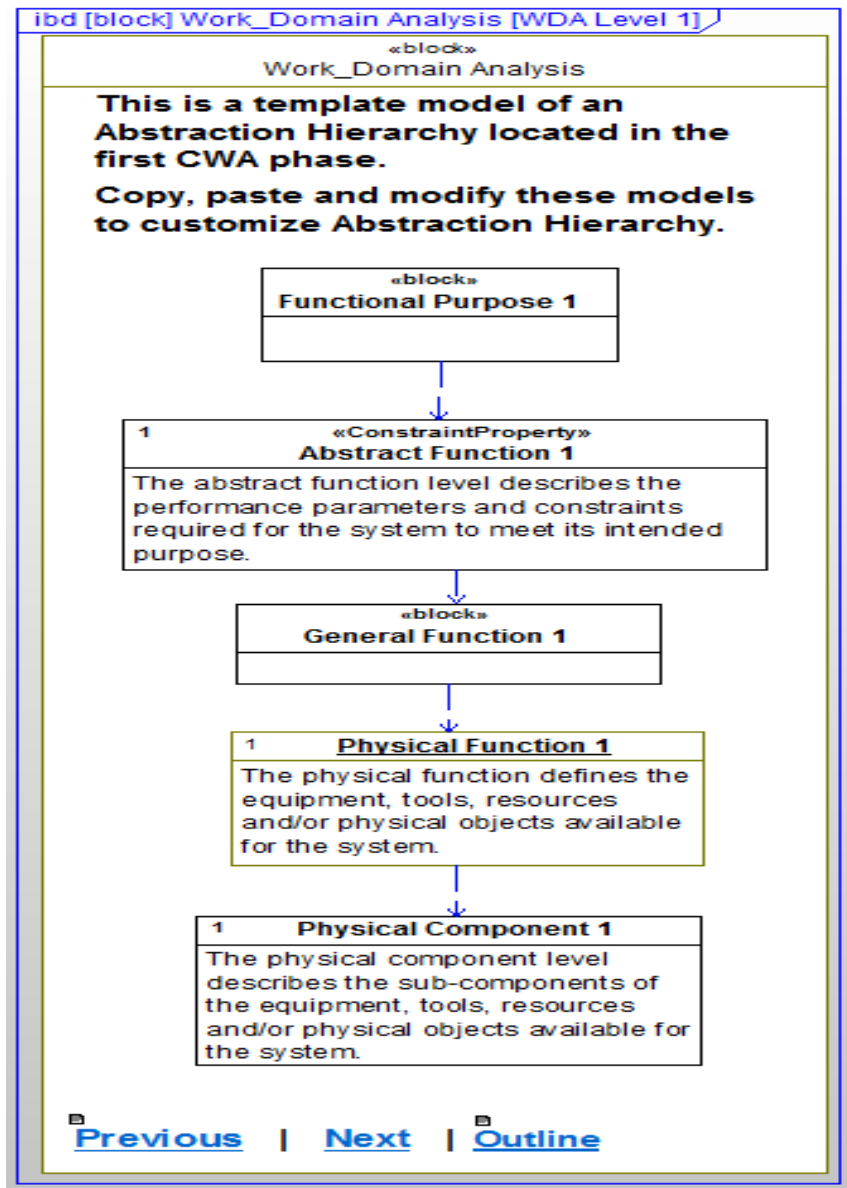


Figure 19: An Abstraction Hierarchy Model Created using SysML

2.5.3 Control Task Analysis Modeled using SysML

The second phase of the CWA framework is Control Task Analysis (ConTA). ConTA covers what needs to be done within the limits of the work domain. The Decision Ladders (DL) tool is used to model the second phase. A DL shows the alternative courses of action for a particular decision. It is a useful way of representing detailed process knowledge. An example of a decision ladder is shown in Figure 21. In order to complete the General Function, the tasks in DL have to be completed. The user will use the physical functions and components from the AH to accomplish the tasks. Not every step in the DL process has to be accomplished in order to execute the completion of the General Function. This provides flexibility to the user for unanticipated events or adaptive learning.

Figure 18, Figure 20, and Figure 21 shows the ConTA phase of CWA created using SysML. Figure 18 presents users with the CWA structure. Within the ConTA on the CWA structure is a Internal Block Diagram (IBD) that contains all the control tasks for one general function. Each general function from the Abstraction Hierarchy will have an IBD created in the ConTA block on the CWA structure (i.e., Figure 18).

Figure 20 represents the general function and its associated control tasks. The model in Figure 20 is called the control task structure. The structure is modeled with blocks and a dependency relationship. Each control task block will have a state machine diagram (STM) attached to it. The STM icon will be located in the upper right corner of the block. The STM is used to model the DL.

Figure 21 shows the DL modeled using SysML. The DL is composed of information processing activities and knowledge states. Information processing activities are the mental or cognitive activities system operators must utilize to complete a task. Knowledge states are the results of the information processing activities. Send action and state elements are used to represent information processing activities and knowledge states, respectively.

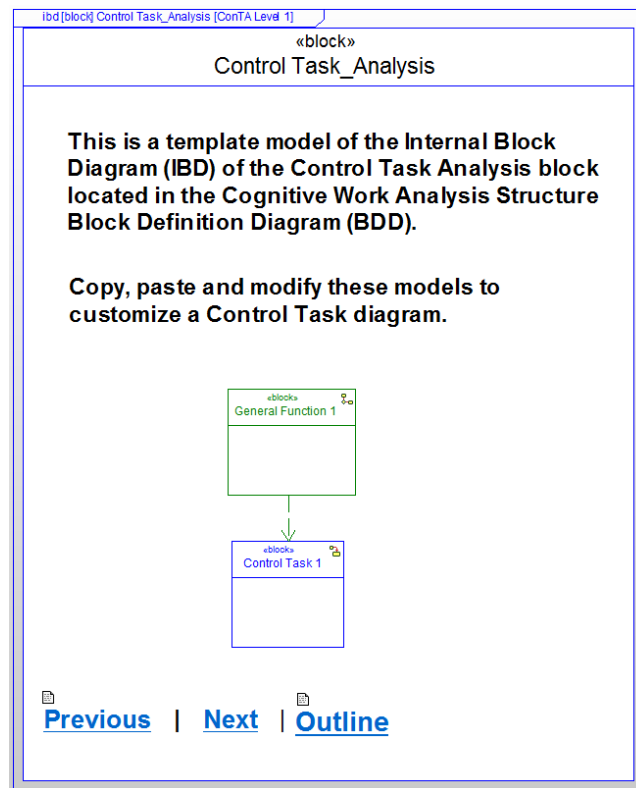


Figure 20: Internal Block Diagram of one General Function and three Control Tasks

This is a template model of the Decision Ladder located in the second CWA phase.

Copy, paste and modify these models to customize Decision Ladder

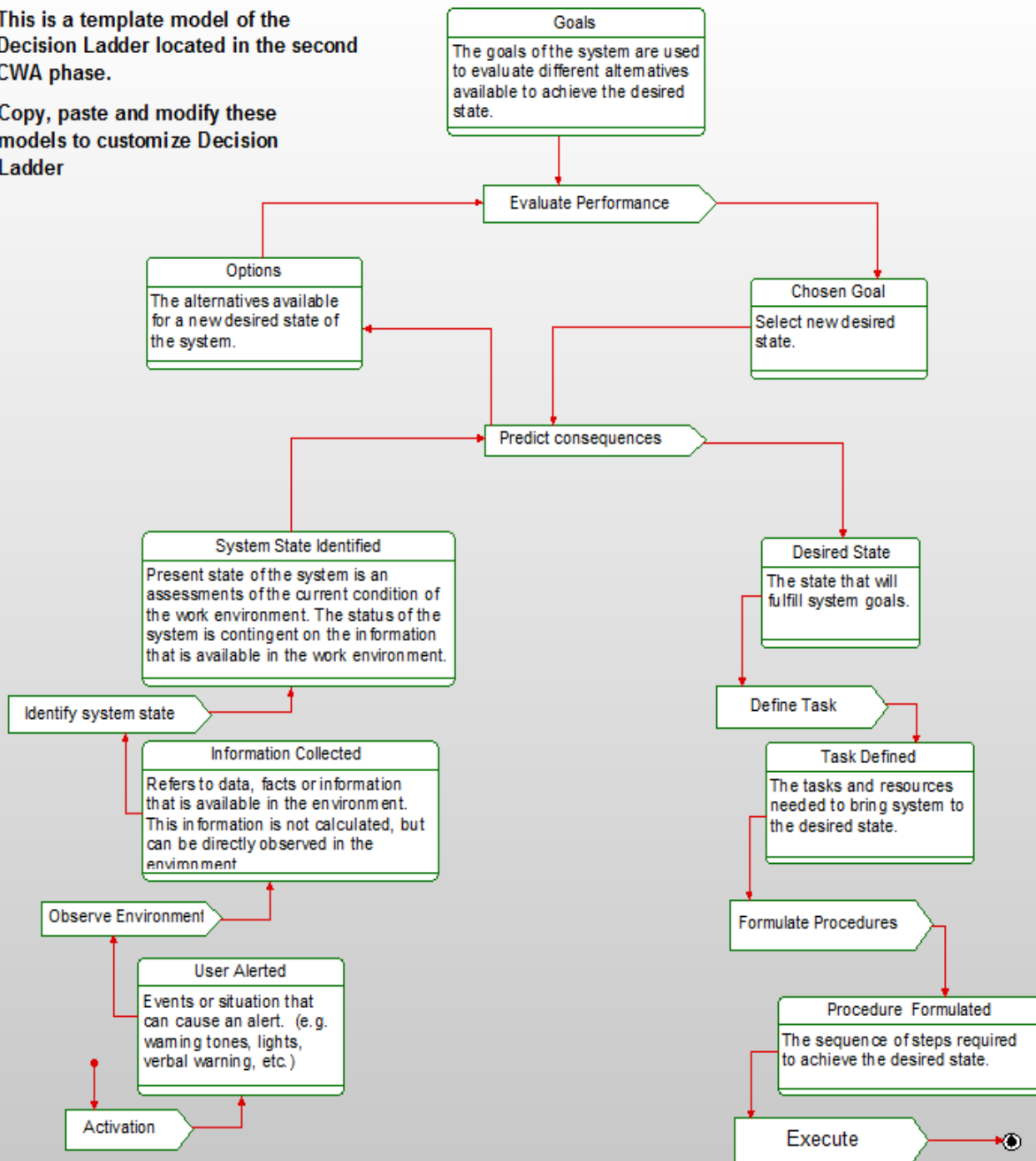


Figure 21: A Decision Ladder Created using SysML

2.5.4 Strategies Analysis Modeled using SysML

The Strategies Analysis is the next phase in the CWA framework. The purpose of Strategies Analysis is to determine which actions are necessary to achieve the control tasks. The same control task can be completed in a variety of ways using different cognitive strategies. Information Flow Maps (IFM) are graphical representations of information processing activities that depict how a user can perform a sequence of tasks to reach an end goal. The successful execution of a strategy allows for the completion of a control task. By performing one of the strategies, the user completes the control task. By completing the control tasks, the user satisfies the functional purpose of the system.

Figure 22 and Figure 23 show the Strategies Analysis (SA) phase of the CWA framework created using SysML. Within the SA phase on the CWA structure (i.e., Figure 18) is an IBD that contains a general function, control tasks and user strategies. The IBD is called the SA structure. The SA structure provides the capability to organize and manage all general functions, user control tasks, and strategies. This capability becomes more critical for a larger, more complex analysis and is one of the major advantages of using SysML.

Figure 22 shows a generic SA structure. Each general function from the AH will be an Internal Block Diagram (IBD) to the SA block on the CWA structure on Figure 18. The general function is decomposed into control tasks. The control tasks are decomposed into strategies. SysML block elements are used to represent the general

functions, the control tasks, and the strategies. Dependency arrows are used to represent the relationships between the general function and the control task blocks. Flow lines are used to represent the relationship between control tasks and the strategy blocks. Each strategy block has a State Machine Diagram (STM) attached to it. The STM icon is located in the upper right corner of the block. The STM is used to model the Information Flow Maps.

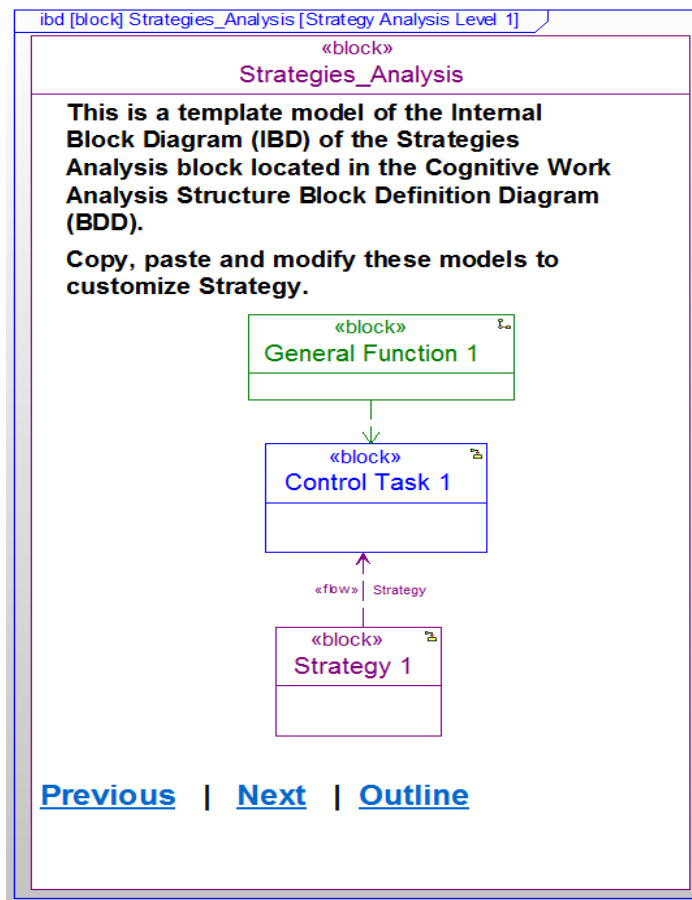


Figure 22: A Template of a SA Structure

Figure 23 shows a generic Information Flow Map (IFM) created using SysML. The third step is to add a State Machine (STM) diagram to each strategy block. The IFM will be constructed in the STM diagram. Send action elements and state elements are used to represent the information processing activities and knowledge states, respectively.

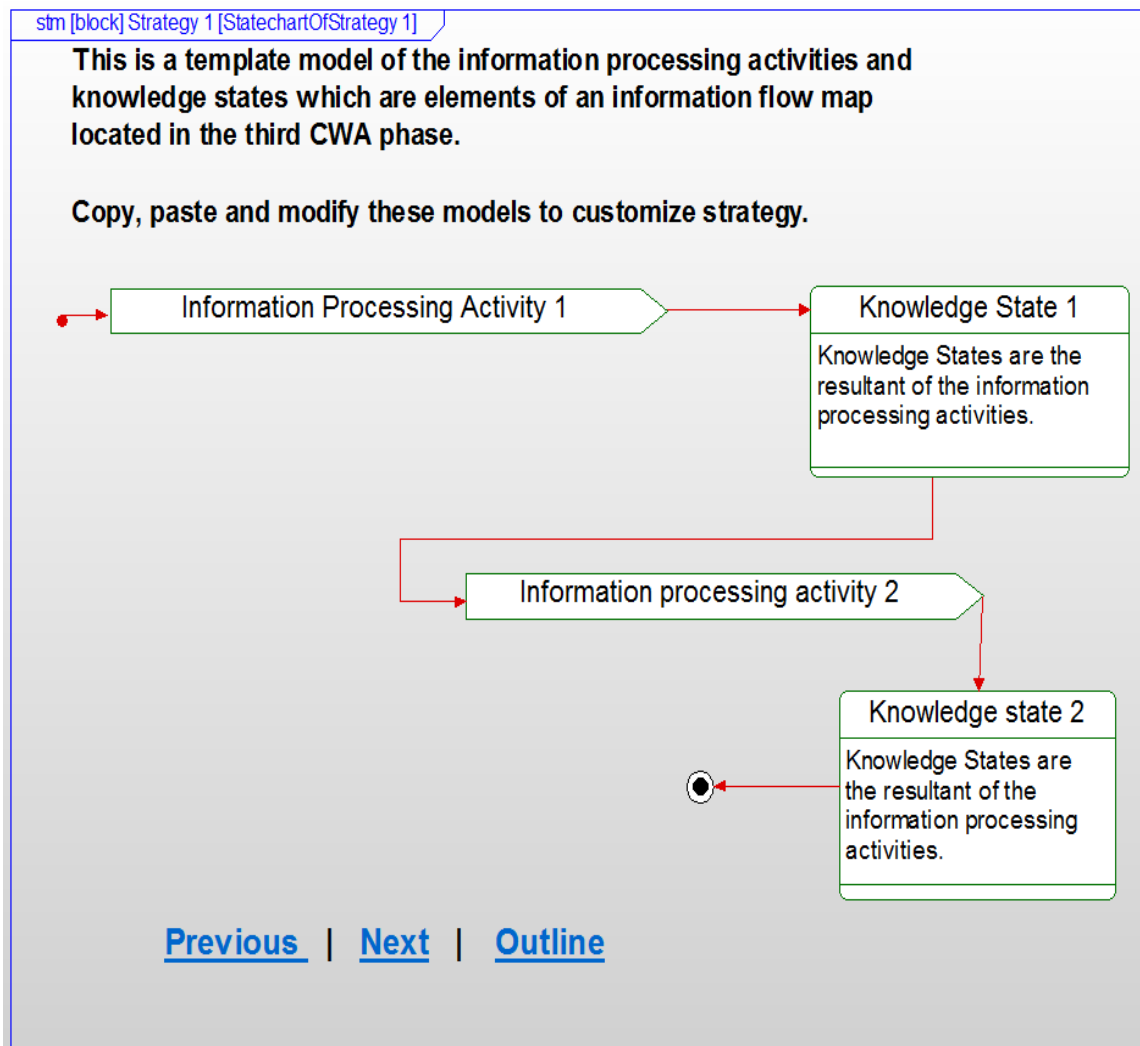


Figure 23: A Template of an IFM

2.5.5 Social Organization and Cooperation Analysis Modeled using SysML

The Social Organization and Cooperation Analysis (SOCA) phase determines who will carry out the work and how it will be shared. IFM identifies the users and their roles. In general, the SOCA answers the question of who will do what tasks by allocating responsibility among actors. The IFM tool is typically employed in this phase of the CWA framework. In the CWA framework, use case diagrams are applied to the SOCA phase.

Figure 24 and Figure 25 shows the Social Organization & Cooperation Analysis (SOCA) phase of the CWA framework created using SysML use case elements. Within the SOCA phase of the CWA structure in Figure 18 is an IBD that contains a generic general function, a control task, and a strategy for completing one control task. Each general function from the AH will be an Internal Block Diagram (IBD) of the SOCA block of the CWA structure on Figure 18. The general function will be decomposed into control tasks. The control tasks will be decomposed into strategies. SysML block elements are used to represent the general functions, the control tasks, and the strategies. Dependency arrows will be used to represent the relationships between the general function and the control task blocks. Flow lines will be used to represent the relationship between control tasks and the strategy blocks. Each control task and strategy block will have a State Machine Diagram (STM) attached to it. The STM icon will be located in the upper right corner of the block. The STM is utilized to model the Decision Ladder (DL) and the Information Flow Maps (IFM). As stated before, the

ConTA, SA, and now SOCA structure are all information management tools and critical for large, complex system analysis.

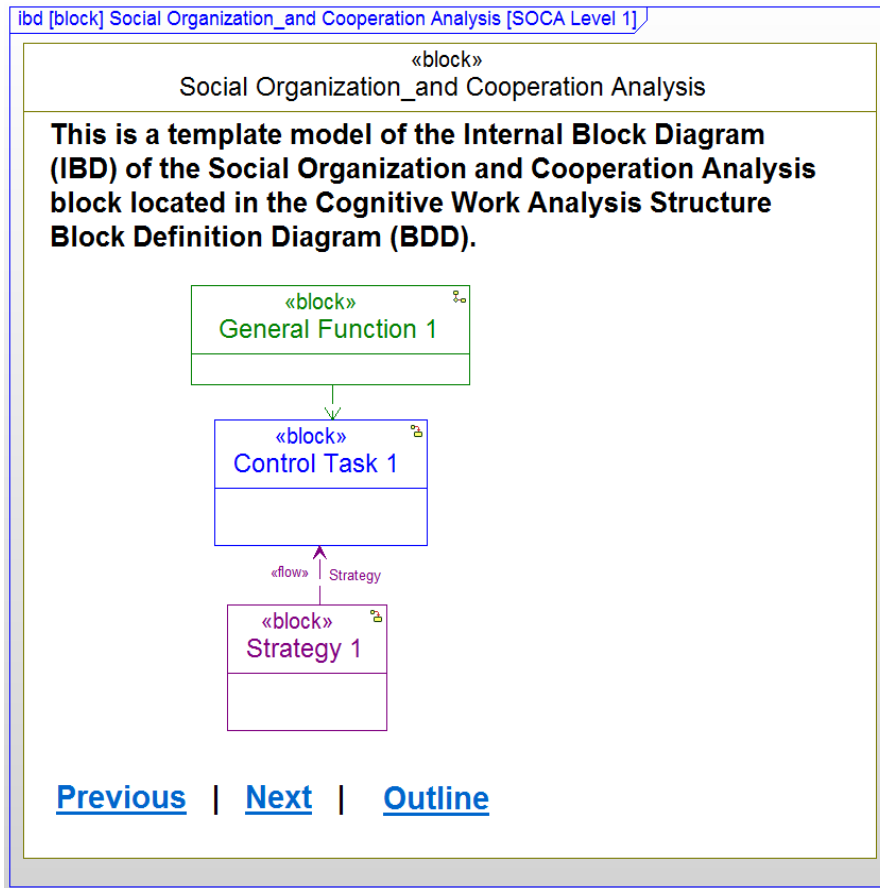


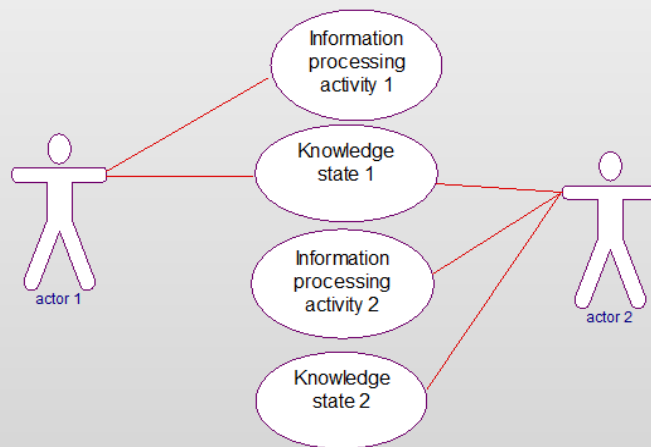
Figure 24: A Template of a Social Organization and Cooperation Analysis Structure

The use case diagram will be modeled with information processing activities and knowledge states from the Strategies Analysis phase. The actors will be based on information acquired from interviews with subject matter experts. The model elements will include actors and use cases to represent the user and the system, respectively.

Figure 25 shows a generic CWA Use Case Diagram.

This is a template model of information processing activities and knowledge states which are elements of the use case located in the third CWA phase. It is related to the strategy developed in the Strategies Analysis phase.

Copy, paste and modify these models to customize use case.



[Previous](#) | [Next](#) | [Outline](#)

Figure 25: A Template of a CWA Use Case Diagram.

2.5.6 Worker Competencies Analysis Modeled using SysML

The final phase of the CWA framework is the Worker Competencies Analysis (WCA) phase, which contains the level of conscious effort of the users when completing an information processing activity. The level of conscious effort determines the physical, perceptual, and cognitive demands placed on the worker using the system. Skill, Rule, and Knowledge-based (SRK) inventory is the modeling tool used in this phase. SRK

ascertains the level of conscious effort an individual uses when processing information. The skill-based category requires almost no conscious effort when completing a task. For example, when a person is approaching a red traffic light while driving, they instinctually step on the brakes. The rule-based category is centered on the rules of the organization or the proper procedures learned in training. The knowledge-based category requires a high level of conscious effort to complete a task. Examples include when a person learns to drive a car for the first time or when a worker encounters an unanticipated event.

Figure 26 and Figure 27 show the Workers Competencies Analysis (WCA) phase of the CWA framework created using SysML. Within the WCA phase of the CWA framework is an IBD that contains a general function, a control task, and a user strategy. Each general function from the AH will be an Internal Block Diagram (IBD) to the WCA block of the CWA structure on Figure 18. The general function will be decomposed into control tasks. The control tasks will be decomposed into strategies. SysML block elements are used to represent the general functions, the control tasks, and the strategies. Dependency arrows will be used to represent the relationships between the general function and the control task blocks. Flow lines will be used to represent the relationship between control tasks and the strategy blocks. Each control task and strategy block will have a State Machine Diagram (STM) attached to it. The STM icon will be located in the upper right corner of the block. The STM is utilized to model the Decision Ladder (DL) and the Information Flow Maps (IFM). As stated before, the

ConTA, SA, and now SOCA structure are all information management tools and critical for large, complex system analysis. Finally, the general function block of the WCA structure is attached to a activity diagram (ACT). The ACT icon will be located in the upper right corner of the block. The ACT is utilized to model the Skill, Rule, and Knowledge (SRK) inventory.

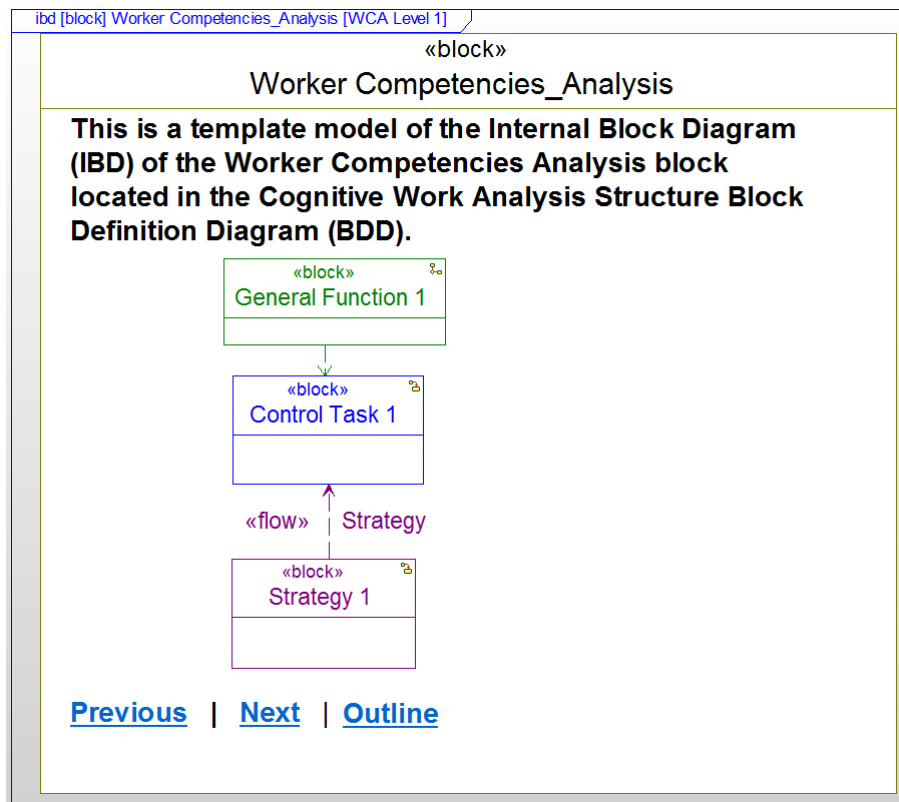


Figure 26: A Template for a Worker Competencies Analysis Structure

Figure 27 shows a generic SRK inventory model. There should be an activity diagram for each general function. SysML swimlanes, send actions, and action elements will be utilized to represent the level of cognitive behavior (i.e., Skill-Based Behavior (SBB), Rule-Based Behavior (RBB), Knowledge-Based Behavior (KBB)), the

information processing activities from the Control Task Analysis and Strategies Analysis phases, and the level of knowledge required by the user. The SysML swimlanes represent levels of cognitive behavior. The send action elements are used to represent information processing activities from the control task analysis and strategies analysis phases. Finally, the action elements are used to represent the level of knowledge required by the user.

2.6 Literature Review Summary

In conclusion, the CWA framework approach provides an interrelated set of methodologies that describe the different attributes of a system. SysML visually demonstrates the interrelated links between each phase of the CWA framework. Chapter two reviewed the literature materials related to systems engineering, CWA, and SysML. The literature review for systems engineering reveals the need to develop tools and methods to address the challenges of complexity, understanding, and communication that exist in system design. The literature review of SysML demonstrates a need for a method of defining cognitive work requirements using SysML. None of the numerous published materials contain the capability to bridge the gap of incorporating cognitive work requirements into the systems engineering process.

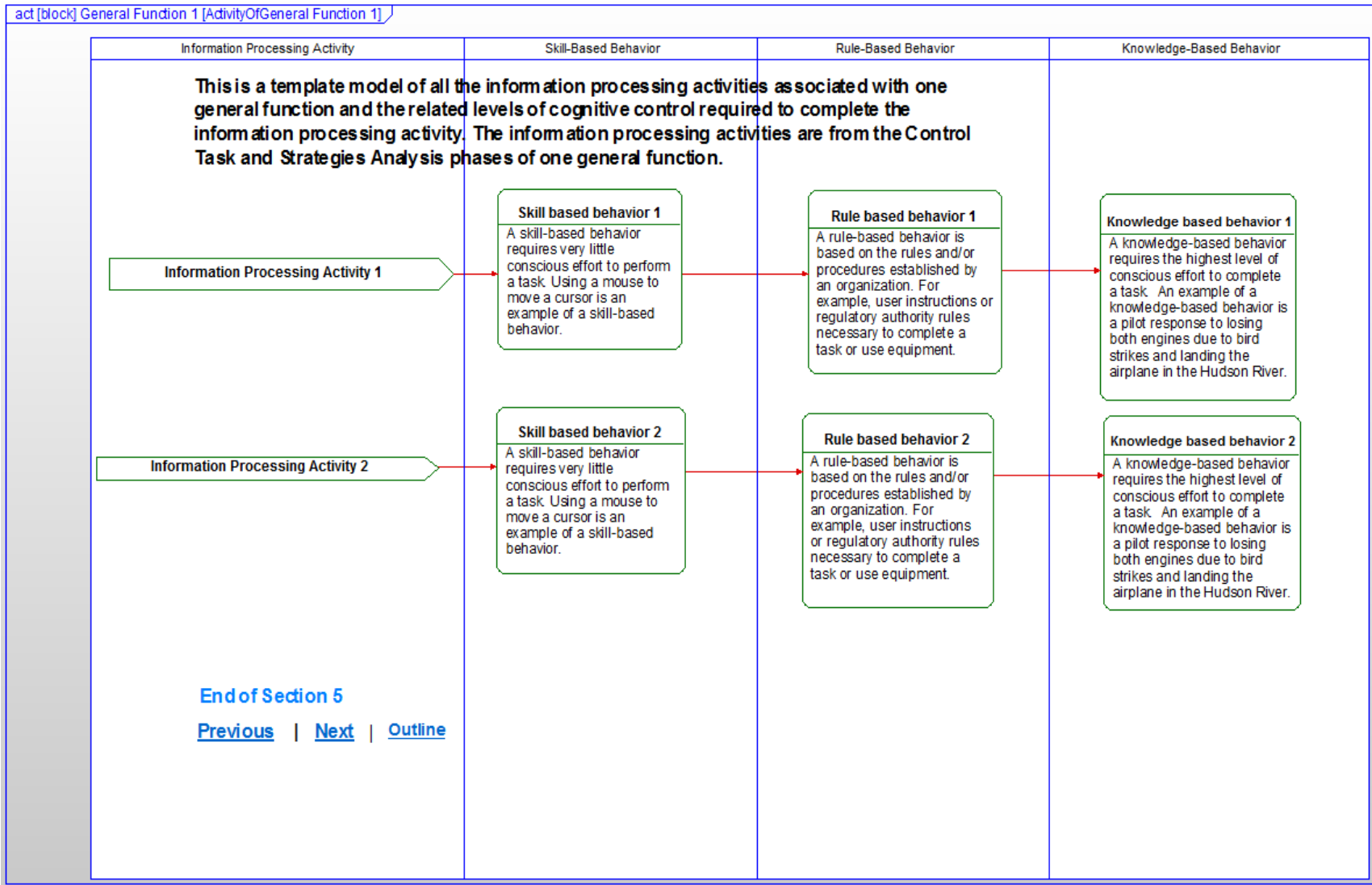


Figure 27: A Template of a Skill, Rule, and Knowledge Inventory Model

CHAPTER THREE: METHODOLOGY

Human work is becoming more cognitive and less physical. Evolving technologies such as smartphones, cloud computing, and enterprise resource planning have increased the number of cognitive tasks a person will perform in the work environment. The evolution of the work environment amplifies the need for cognitive analysis in the systems engineering process. The lack of cognitive factors in system design leads to systems that do not fully leverage the cognitive strengths of the human user or compensate for their limitations (Stoner et al., 2006). The proposed framework can fill this gap by providing systems engineers with a holistic tool that will guide them through the process of incorporating cognitive work requirements into their system designs.

The addition of a CWA framework implemented within SysML ensures that the roles and needs of human users are addressed during system development. Because of the diversity of available tools, it can be overwhelming for systems engineers to locate and select the right tools. One option would be to minimize the number of tools needed, while maximizing the number of areas of interest that the tools would address. CWA can potentially increase the number of areas of interest evaluated by one tool.

3.1 Development of Cognitive Work Analysis Tutorial using SysML

3.1.1 Step 1: Create an Outline

The tutorial is intended to provide a set of guidelines to assist systems engineers with the integration of cognitive work requirements into the systems engineering process.

The first step in the process of developing this tutorial is to make an outline. The outline will form the navigation element in the CWA Tutorial (CWAT). The outline will identify the main topics of the tutorial. Figure 28 shows a draft outline of the tutorial created in SysML. A SysML Block Definition Diagram (BDD) is used to represent the content page of the tutorial. Within each of the blocks is an Internal Block Diagram (IBD) or BDD that provides more information on the section the user wants to review.

3.1.2 Step 2: Model Introduction Section

3.1.2.1 Overview of the Cognitive Work Analysis Framework

The introduction will provide a summary of the CWA framework. The summary will use SysML diagrams to present the user a description of CWA, the applications CWA, the objectives of this tutorial, and a list of acronyms. Figure 29 is the first page of the introduction section. It shows a description of each phase of CWA.

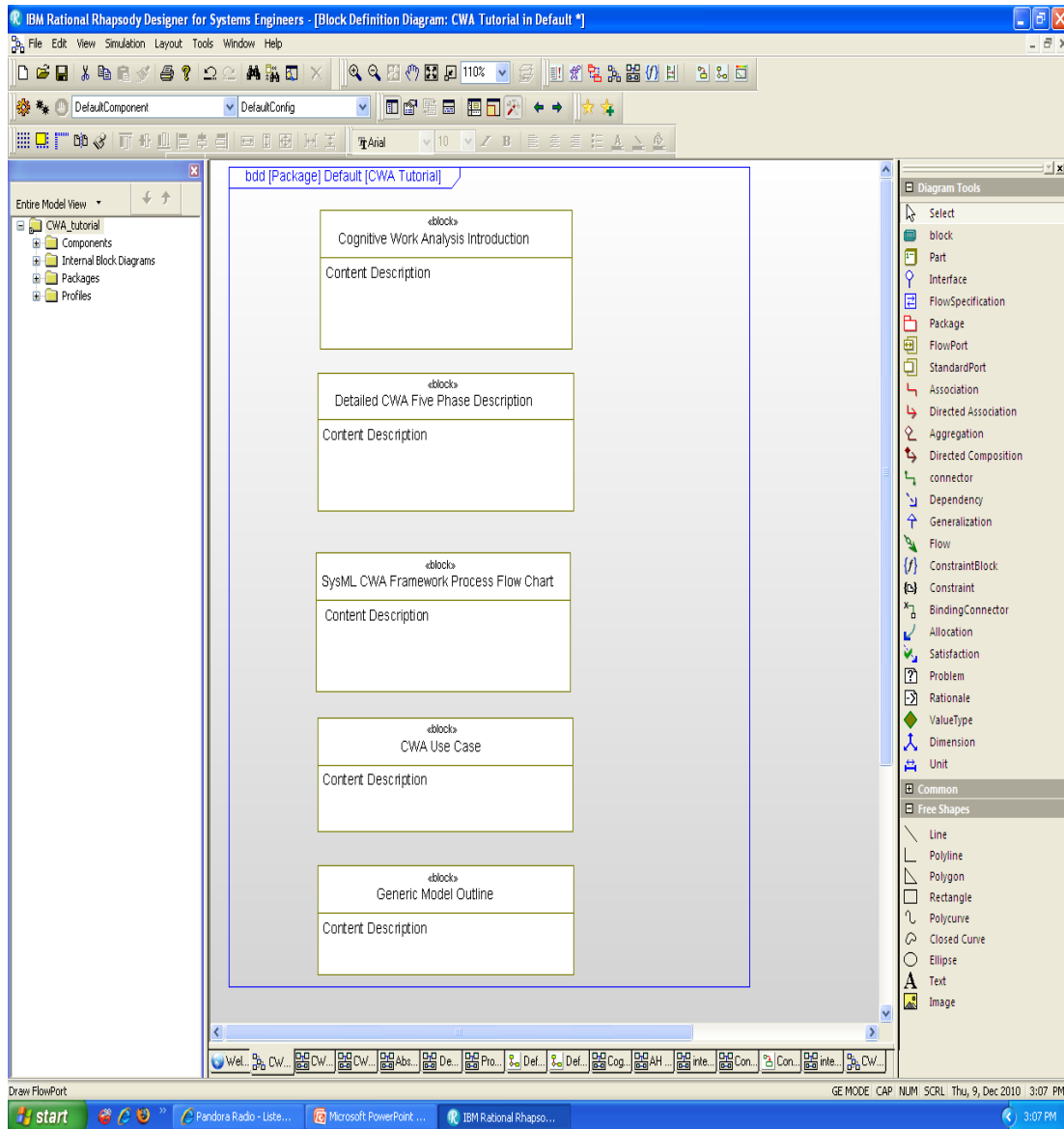


Figure 28: Cognitive Work Analysis Tutorial Outline

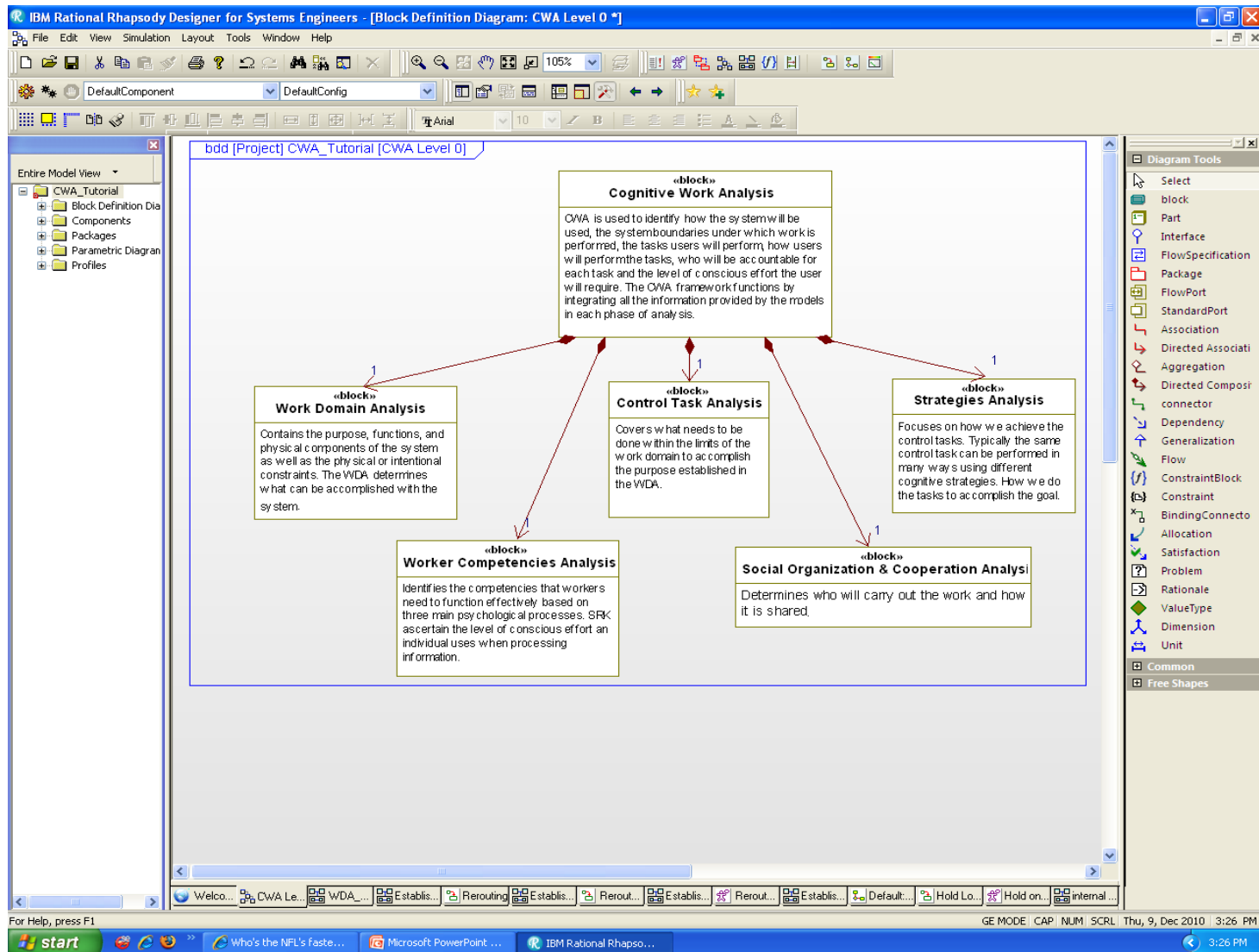


Figure 29: CWAT High Level View

A SysML BDD is used to represent a high-level view of CWA. Blocks are used to represent each CWA phase. The diamond- and arrow-tipped lines are directed composition relationships. These relationships show that WDA, ConTA, SA, SOCA and WCA are part of CWA, which is the whole. The number 1 or 0 is used to represent a state existence. The number 1 means that the part end must always be part of the whole end to exist. The number 0 means that the part end can exist without the whole end.

Within the CWA block on Figure 29 is an IBD that contains information about what is CWA, tutorial objectives, uses for CWA framework, a list of acronyms and definitions. Figure 30 shows the IBD of the CWA block of Figure 28. Figure 30 will contain text and graphics. No SysML diagrams will be used in this IBD.

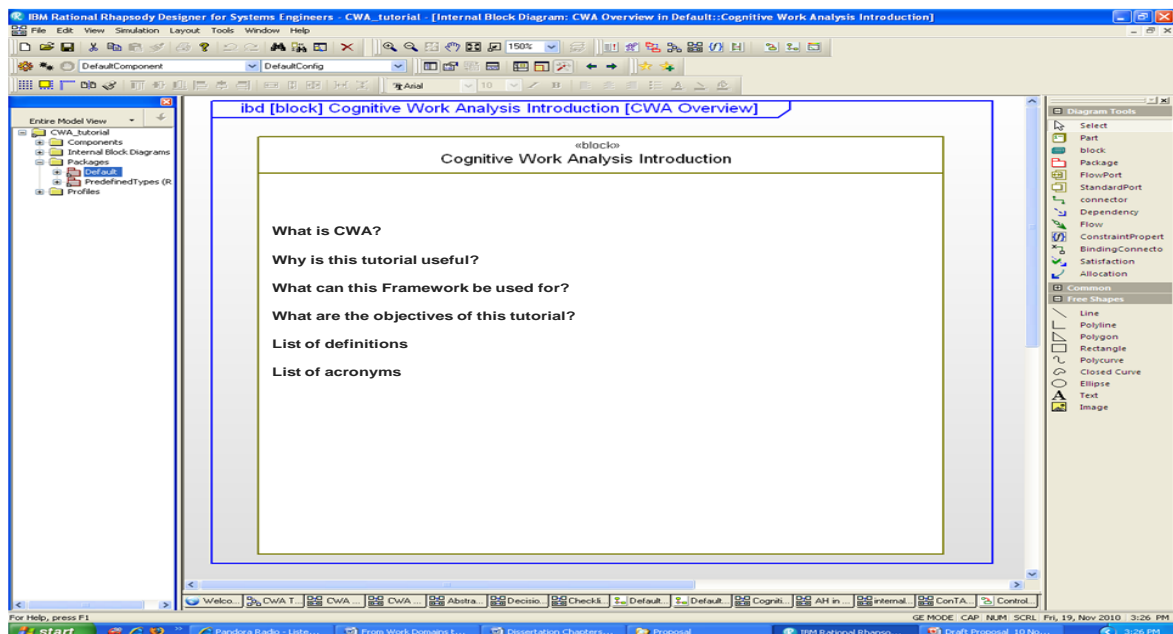


Figure 30: Internal Block Diagram Introduction to CWA

The CWA block in Figure 31 has another IBD, which contains the flow of information between the different phases of CWA and the sequence in which the phases should be completed. Figure 31 uses blocks and flow arrows to illustrate the flow of information between CWA phases. Each flow arrow contains the information being transferred to each CWA phase. The numbers above the blocks represent the order in which each phase should be done.

3.1.3 Step 3: Detailed Description of the CWA Framework

3.1.3.1 Align CWA Models to SysML Diagrams

The next section of the tutorial provides a detailed description of CWA and translates CWA terminology into SysML diagrams. A tutorial will not be useful to systems engineers if it is full of technical language that only cognitive engineers understand. The translation of CWA into SysML will enhance a systems engineer's ability to grasp the CWA process. When the user enters this section from the contents page, he will see an IBD diagram similar to Figure 31. The user can select any block and view the IBD of the each CWA phase. For example, Figure 32 shows the IBD for the WDA block in Figure 31. The user will get a text and image description for WDA. In addition, SysML diagrams will be used to represent the components of the model of each phase. In this case, Figure 32 will have an Abstraction Hierarchy (AH) converted into SysML diagrams. To build the AH in SysML, blocks, constraints block, and part diagrams as well as association lines will be used.

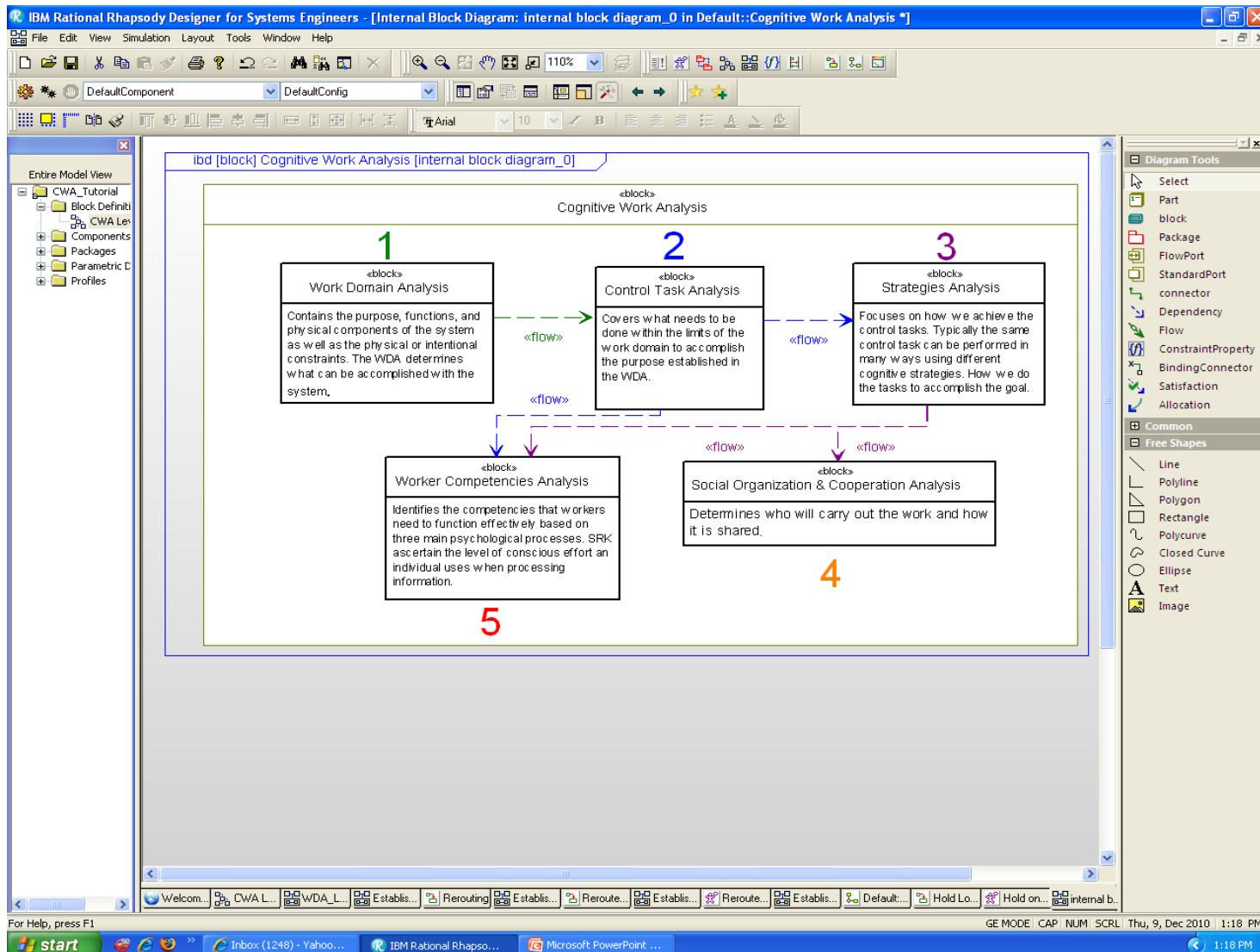


Figure 31: CWAT Framework Process Flow View

Each SysML diagram will be embedded with a description or definition that will inform systems engineers of the diagram's uses and significance.

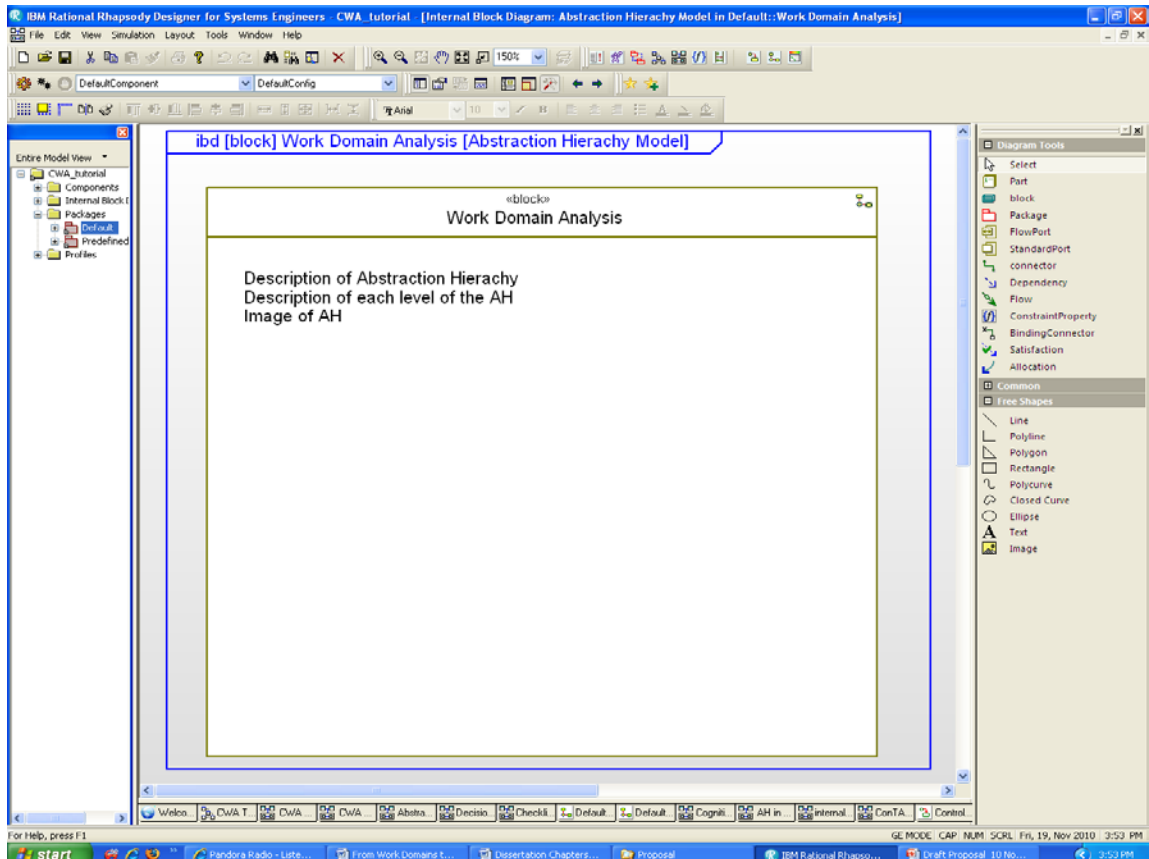


Figure 32: Work Domain Analysis Tutorial

3.1.4 Step 4: Construct CWA Process Flow Chart

3.1.4.1 Create CWA Process Flow Chart using Activity Diagram

The process flow section of the tutorial begins with a CWA framework process flow view similar to Figure 32. A process flowchart will promote understanding of the CWA model building process by using graphical symbols to depict the flow of the steps

in the process. The user can review any CWA phase process flow chart by entering into the corresponding activity diagram. For example, the user can click on the activity diagram symbol on the WDA block on CWA framework process flow and view Figure 33.

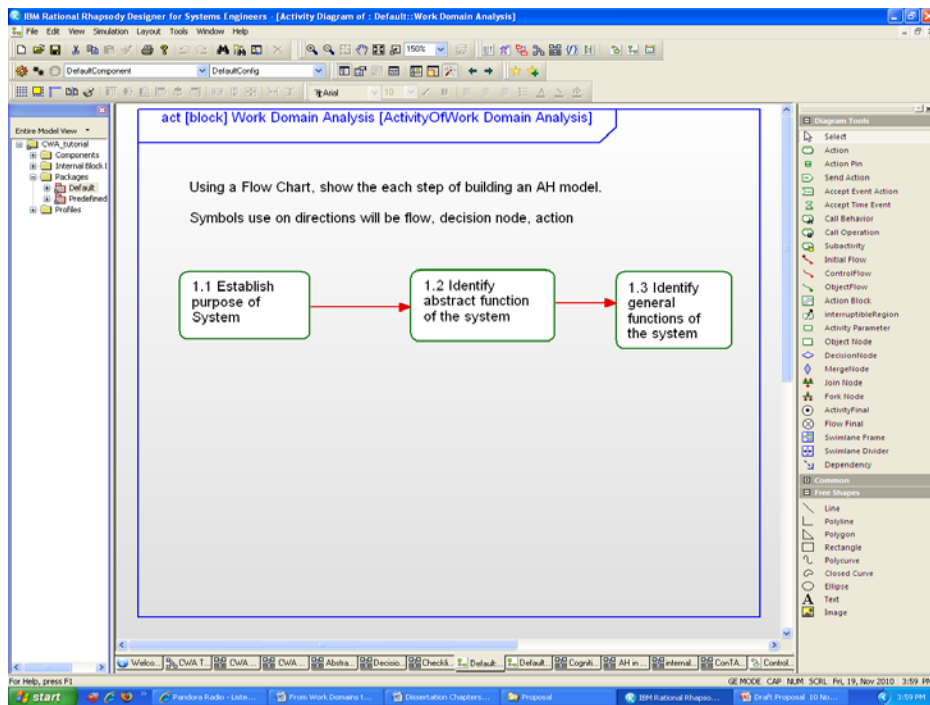


Figure 33: Work Domain Analysis Process Flow Chart

Figure 33 shows an activity diagram. It is composed of initial flow symbol, action blocks, control flow arrows, and an activity final symbol. The initial flow represents the start of the process. The action blocks represent functions of the process. The control flow arrows represent the transition from state to another. The activity final symbol represents the completion of the process. In addition, each action will contain sample questions for the knowledge elicitation aspect of CWA. Usually the user will have to

interview a domain expert about the domain of interest and how the expert performs his tasks. The success of the interview session depends on the questions asked.

3.1.5 Step 5: CWA Use Case

The CWA use case section will provide systems engineers with knowledge requirements to do CWA. This section will help systems engineers select the appropriate personnel to conduct a CWA. Additionally, this tutorial will inform and support a system engineer's ability to coordinate the efforts of the CWA team.

3.1.6 Step 6: Construct Template CWA Framework Model

The final section of the CWAT will use appropriate SysML diagrams to represent CWA models. Figure 34 shows the first screen the user will view. From this screen, the user can navigate to the CWA phase of interest to view the template models.

Figure 35 shows a template AH model. It is the IBD of the WDA in Figure 34. It is composed of SysML block, constraint property, dependency lines and part diagrams. The template models can be copied and pasted into a model that the user is developing. Once the template models are copied, the user can modified the models to the system being developed.

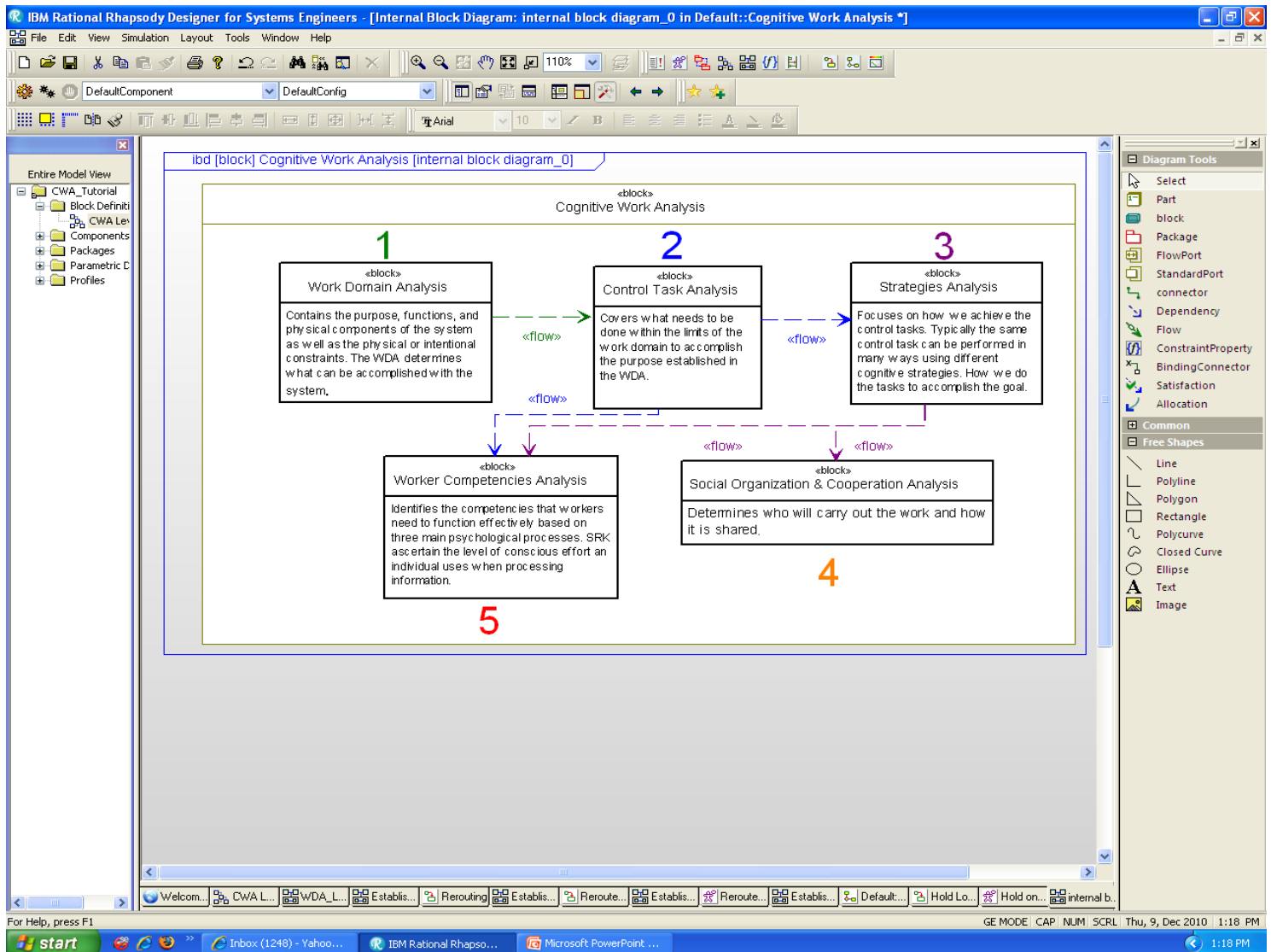


Figure 34: Template CWA

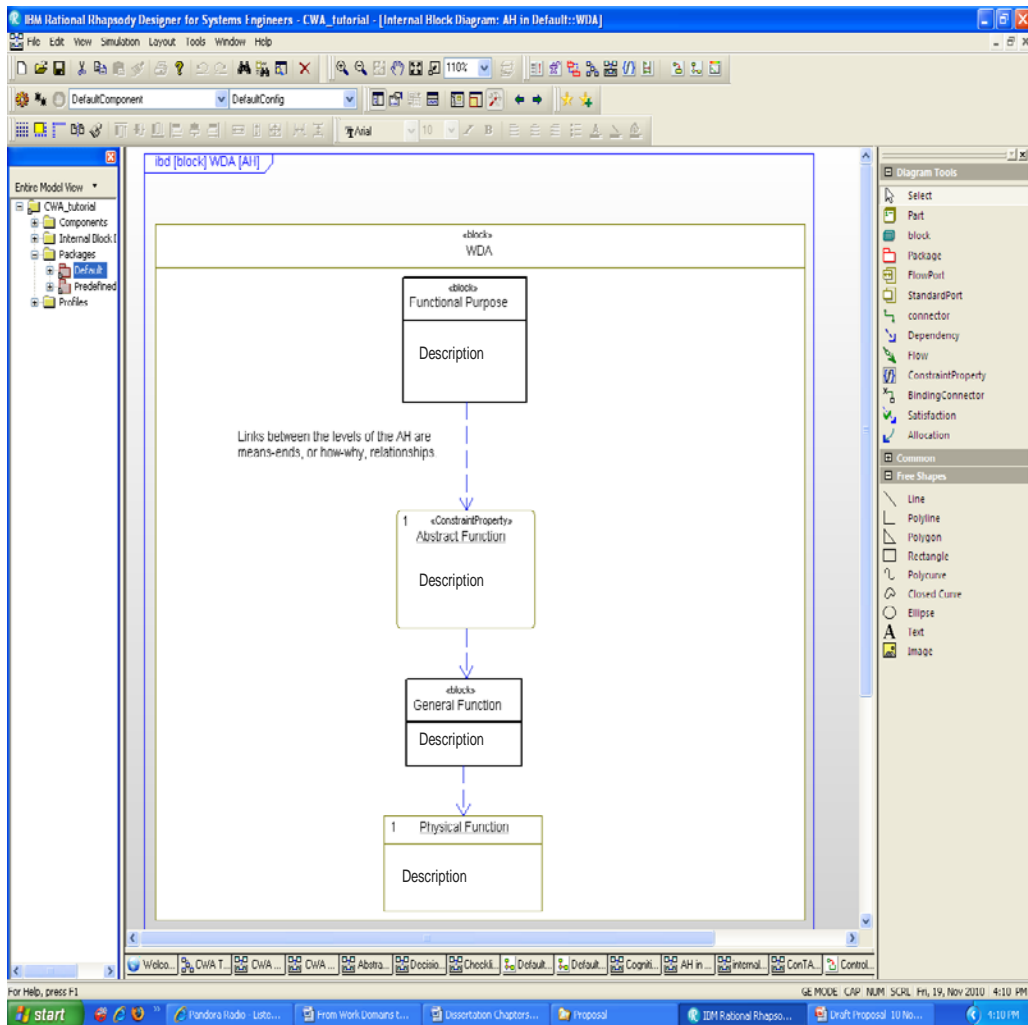


Figure 35: Template Abstraction Hierarchy Model

3.2 Methodology Summary

The goal of this research is to provide a set of guidelines to assist systems engineers with the integration of cognitive work requirements into the systems engineering process. Chapter 3 illustrates how a tutorial will be developed to inform users about the purpose of CWA, the SysML diagrams used to construct the CWA framework in SysML and the CWA construction process flow. The tutorial will be

constructed in SysML using appropriate diagrams to translate CWA terminology to SysML terminology for systems engineers to comprehend.

CHAPTER FOUR: RESULTS

This chapter describes and illustrates the Cognitive Work Analysis Tutorial (CWAT) that was developed using System Modeling Language. A walk-through of tutorial screenshots will be used to describe the CWAT.

4.1 Cognitive Work Analysis Tutorial Introduction Section

The CWAT is composed of seven sections. The seven sections are as follows:

1. An Introduction to Cognitive Work Analysis
2. A Detailed Five-Phase Description
3. Cognitive Work Analysis Process Flow Chart
4. CWA Terminal Radar Approach Control Example
5. CWA Automated Teller Machine Example
6. Cognitive Team Competency Requirements
7. Tutorial References

The introduction section contains a general overview of what CWA is and how it is used. The detailed five-phase description contains a model-based and text-based description of the five phases of CWA as well as a translation of CWA terminology into SysML diagrams. The third section contains a process flow chart for conducting CWA. The CWA Terminal Radar Approach Control (TRACON) example section demonstrates the methodology on an existing system. The fifth section focuses on the ability to collect information that is necessary to construct a CWA framework using SysML. The CWA

competency requirements contain information on qualifications for assembling a cognitive factors team. The final section provides additional sources of information about Cognitive Work Analysis. Furthermore, elements of the references were used to create the CWAT.

4.1.1 Title Page

The title page states the purpose of the CWAT as well as how to navigate and view the tutorial. The tutorial was constructed to provide a set of guidelines to assist systems engineers and other system designers with the integration of cognitive work requirements into the systems engineering process. Figure 36 shows a screenshot of the title page. The title page was constructed using a Block Definition Diagram (BDD). The Next button, located at the bottom left of the page, is hyperlinked to the section outline page, which is the next page in the tutorial.

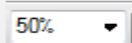
TUTORIAL FOR CONSTRUCTING A COGNITIVE WORK ANALYSIS FRAMEWORK USING SYSTEM MODELING LANGUAGE

The purpose of this tutorial is to provide a set of guidelines to construct a Cognitive Work Analysis (CWA) framework using System Modeling Language (SysML). This framework will assist system engineers and other system designers with the integration of cognitive work requirements into the systems engineering process.

Use the Next, Previous, and Outline text located on the button left of each page to navigate the tutorial.

Next - Continue to next page
Previous - Go back to last page
Outline - Go to section outline page

To Zoom in & out go to "View" -> "Tool bars" ? "Zoom" --> select desired percentage or use



on the toolbar



**Created by
Wilfred Wells**

[Next](#)

Figure 36: Title Page of Cognitive Work Analysis Tutorial

4.1.2 Outline

The outline identifies the main topics of the tutorial. Figure 37 illustrates the tutorial outline created in SysML. The outline was constructed in a SysML Block Definition Diagram (BDD). Block diagrams are used to represent the each section of the CWAT. Within each of the blocks is an Internal Block Diagram (IBD) that provides more information on the section the user wants to review. The section numbers are hyperlinked to the corresponding section. To navigate the CWAT, use the Previous, Next, and Outline buttons located at the bottom left of the page. The buttons are hyperlinked to the last page viewed, the ensuing page in the tutorial, and the outline page, respectively.

4.2 CWAT Introduction Section

The first section of the CWAT is the introduction section, which provides a model-based high-level view of the CWA structure and a text-based summary of the CWA framework. The high-level view is constructed in a BDD. Block diagrams are used to represent CWA and the composition of CWA. Figure 38 illustrates the high-level view of the CWA structure. The relationship that exists between CWA and the CWA phase is a directed composition. A directed composition in SysML is represented by a line with an arrow on one end and a diamond shape on the other. A directed composition is a relationship that exists between a block and a block that is part of that block. Directed

meaning “one-direction” relationship and composition meaning “composed of.” The diamond- and arrow-tipped line represents a directed composition relationship.

[bdd \[Project\]SYSML_CWA_DRAFT_15JAN2011](#)

The tutorial outline provides a summary of the content of each section in the tutorial.

Click next to continue in order or click a section # to skip to that section.

Section 1	<p>«block» Introduction to Cognitive Work Analysis</p> <p>This section contains a general overview of what cognitive work analysis (CWA) is and how it is used.</p>
Section 2	<p>«block» Detailed Five Phase Description</p> <p>This section contains a detail description of the five phases of CWA and a translation of CWA terminology to System Modeling Language (SysML) diagrams.</p>
Section 3	<p>«block» CWA Process Flow Chart</p> <p>This section contains a process flow chart for conducting CWA.</p>
Section 4	<p>«block» SysML CWA TRACON Example</p> <p>This section contains an example of a Cognitive Work Analysis Framework constructed using System Modeling Language.</p>
Section 5	<p>«block» Automated Teller Machine Example</p> <p>The second example focuses on performing cognitive work analysis.</p>
Section 6	<p>«block» CWA Competency Requirements</p> <p>This section contains competency requirements for a cognitive factors team.</p>
Section 7	<p>«block» Tutorial References</p> <p>This section provides additional sources of information about Cognitive Work Analysis. Furthermore, elements of the references were used to create this tutorial.</p>

[Previous](#) | [Next](#)

Figure 37: CWAT Outline

This relationship shows that WDA, ConTA, SA, SOCA, and WCA blocks are parts of the CWA block, which is the whole. To navigate the CWAT the end user will utilize the Previous, Next, and Outline buttons located at the bottom left of the page. The buttons are hyperlinked to the last page viewed, the ensuing page, and the outline page in the tutorial, respectively.

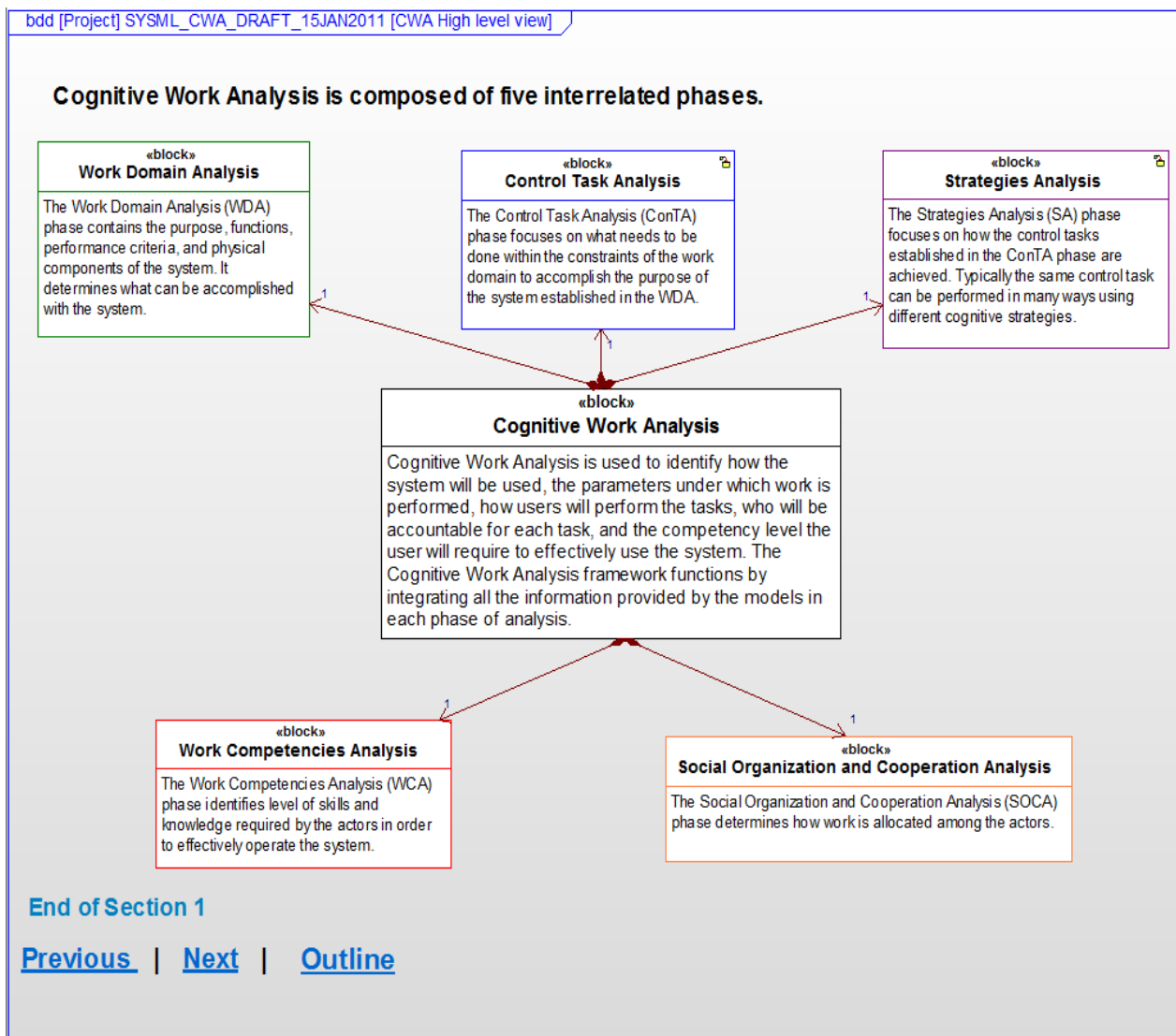


Figure 38: CWAT Framework High Level View

The CWA text-based summary is an IBD of the CWA block in Figure 39. The CWA summary presents the user with a description of CWA, the applications of CWA, and the objectives of this tutorial. Figure 39 shows the text-based description of CWA.

4.3 CWAT Detailed Description Section

Section two of the CWAT provides a detailed description of each phase of CWA and translates CWA terminology into SysML diagrams. A tutorial will not be useful to systems engineers if it is full of technical language that only cognitive engineers understand.

4.3.1 CWAT Sequence and Information Exchange

Figure 40 shows an IBD for the CWA block in Figure 38. This IBD contains the flow of information between the different phases of CWA and the sequence in which the phases should be completed. Figure 40 uses blocks and flow arrows to illustrate the flow of information between CWA phases. Each phase is represented with a different color. Each flow arrow contains the information being transferred to each CWA phase. The information being transferred is written above the flow line. The numbers above the blocks represent the order in which each phase should be done. Each block contains attributes and operations for each CWA phase. The attributes describe the components of each phases. The operations describe the task that should be performed for each

phase. Table 4 to Table 13 contain all the attributes and operations descriptions used within

Read Cognitive Work Analysis Introduction

The objective of this tutorial is to improve communication between the systems engineering and cognitive engineering domains. Most system engineers do not have access to usable information about cognitive work requirements that can be incorporated into their project designs, simply because cognitive factors personnel have difficulty translating their findings. This leads to a disregard of the cognitive aspect of system design. The impact of this problem leads to system requirements that do not account for the cognitive strengths and limitations of its users. This inevitably leads to poor system and human performance.

The Cognitive Work Analysis(CWA) framework analyzes the tasks people perform, the work they do, the decisions they make and how they solve problems in order to satisfy the intended purpose of system. This framework was developed in the 1970's by Rasmussen at Risø National Laboratory in Denmark. It provides a structure for the analysis of human-information interaction. CWA was initially used for interface design.

The CWA framework consist of five interrelated phases. The phases are:

1. Work Domain Analysis (WDA)
2. Control Task Analysis (ConTA)
3. Strategies Analysis (SA)
4. Social Organization and Cooperation Analysis (SOCA)
5. Workers Competencies Analysis (WCA).

Within each of the five phases is a modeling technique used to account for several dimensions of the system. The modeling techniques are:

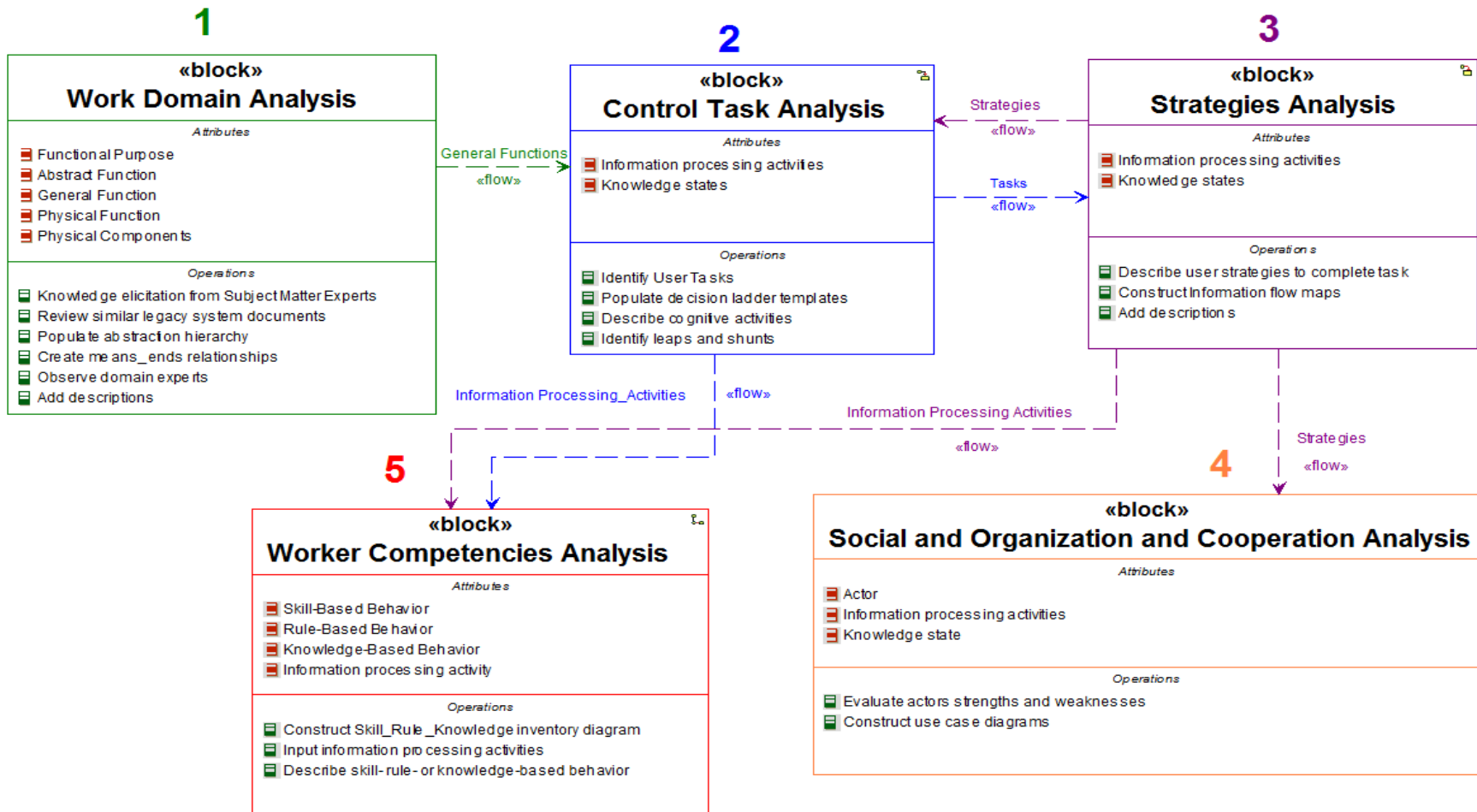
1. Abstraction Hierarchy (AH)
2. Decision Ladders (DL)
3. Information Flow Map (IFM)
4. Use Case (UC)
5. Skill, Rule, and Knowledge-based inventory (SRK).

First, the WDA phase contains the purpose, functions, performance criteria, and physical components of the system. It basically determines what can be accomplished with the system. Second, is the ConTA phase which focuses on what needs to be done within the constraints of the work domain to accomplish the purpose of the system established in the WDA. Third, is the SA phase which focuses on how the control tasks established in the ConTA phase are achieved. Typically the same control task can be performed in many ways using different cognitive strategies. Fourth, is the SOCA phase which determines how the work is allocated among the actors. Fifth, the WCA phase identifies level of skills and knowledge required by the actors in order to effectively operate the system.

Since CWA introduction, it has been applied to a variety of complex work and training environments. Some include: military command and control decision making, air traffic controllers, military aviation, health care, training needs analysis, evaluation of system design proposal, interface design and evaluation, training system design, human-system integration, human error management, and process control. The limits of what CWA could be used for is left to the imagination of the user.

Figure 39: Introduction to CWA in an Internal Block Diagram

the SysML diagrams. In addition, part of the operation descriptions contains sample questions for the knowledge elicitation aspect of CWA. Usually the user of the tutorial will have to conduct interviews of domain experts. The user will ask questions about the domain of interest and how the experts perform tasks. The success of the interview session depends on the questions asked.



[Previous](#) | [Next](#) | [Outline](#)

Figure 40: CWAT Framework Process Flow View

Table 4: Work Domain Analysis Operations Description

WDA Operations	Description
Knowledge elicitation from SMEs	<p>Interviews with subject matter experts will have to be performed to construct each level of the Abstraction Hierarchy. Questions may include, but are not limited to, the following:</p> <ul style="list-style-type: none"> o What are the main goals of the expected system? o What might get in the way of achieving set goals? o What do you have to do to obtain the goals? o What resources are required to help reach goals? o What regulations/policies are necessary in the work domain?
Review similar legacy system documents	<p>Reviewing legacy system references creates a starting point for system designers. It helps to identify the physical equipment, the goals, the functions, and the policy constraints of the system. These documents include, but are not limited to, instructions and operating manuals for the system.</p>
Observe domain experts	<p>Observe domain experts engaged in activities that could be associated with the new system.</p>
Populate abstraction hierarchy	<p>Once interviews and documentation reviews are completed, the Abstraction Hierarchy can be populated with the appropriate data for each level. SysML blocks are used to represent the data at the different levels of the Abstraction Hierarchy.</p>
Create means/ends relationships	<p>Each level is connected by means-ends relationships. The means are a level below the ends. For example, the general function is the means for the abstract function. The lower levels describe the actions, components, or parameters that are necessary for achieving the ends or upper levels of the AH. After the Abstraction Hierarchy blocks are filled with the appropriate data, each block will be connected by SysML dependency lines.</p>
Add descriptions	<p>A detailed description should be added to each diagram.</p>

Table 5: Work Domain Analysis Attributes Description

WDA Attributes	Description
Functional Purpose	The functional purpose describes the reasons the system exists.
Abstract Function	The abstract function level describes the performance parameters required for the system to meet its intended purpose.
General Function	The general function level describes the basic work functions of the system.
Physical Function	The physical function defines the equipment, tools, resources, and/or physical objects available for the system.
Physical Components	The physical component level describes the sub-components of the equipment, tools, resources, and/or physical objects available for the system.

Table 6: Control Task Analysis Operations Description

ConTA Operations	Description
Identify user tasks	<p>Use interviews and other knowledge elicitation methods with subject matter experts to construct each level of the Decision Ladder. The most common knowledge elicitation method is directly questioning domain experts on how they conduct their jobs and the tasks necessary to successfully complete their jobs. Questions may include, but are not limited to, the following:</p> <ul style="list-style-type: none"> o What are some of the steps taken to achieve a task? o What kinds of events can act as alerts? o What kinds of data or facts are available? o What kinds of assessments about the system's condition or situation are possible with the information? o What kinds of choices or alternatives are available for the system's desired or target state? o What kinds of aims or objectives can be relevant or influence decisions? o What kinds of target states are possible? o What kinds of tasks are necessary and what kinds of resources are available? o What kinds of procedures or sequences of steps are necessary?
Describe cognitive activities	Interview domain experts to describe cognitive activities required to complete a system task.
Identify leaps and shunts	During subject matter experts interviews, identify shortcuts experts would use when completing a task.
Populate decision ladder templates	Once interviews are completed, the Decision Ladder can be populated with the appropriate data for each step on the ladder. Use SysML state machine diagrams. Send Action and State diagrams are used to represent the information processing activities and knowledge states at the different steps in the Decision Ladder.
Add descriptions	A detailed description should be added to each diagram.

Table 7: Control Task Analysis Attributes Description

ConTA Attributes	Description
Information processing activities	Information processing activities are the mental or cognitive activities system operators must utilize to complete a task.
Knowledge states	States of knowledge are the result of information processing activities.

Table 8: Strategies Analysis Operations Description

SA Operations	Description
Describe user strategies to complete task	Use interviews and other knowledge elicitation methods with subject matter experts to construct each level of the Decision Ladder. The most common knowledge elicitation method is directly questioning domain experts on the course of action used to complete a task. Questions may include, but are not limited to, the following: <ul style="list-style-type: none"> o What are some of the possible strategies that can be used to complete a task? o Which of the strategies mentioned before would most system operators use to complete a task? o What steps would a system novice use to complete a task? o What steps would a system expert use to complete a task?
Construct Information flow maps	Use data collected during interviews to construct information flow maps. Use SysML state machine diagrams. Send Action and State diagrams are used to represent the information processing activities and knowledge states respectively.
Add descriptions	A detailed description should be added to each diagram.

Table 9: Strategies Analysis Attributes Description

SA Attributes	Description
Information processing activities	Information processing activities are the mental or cognitive activities system operators must utilize to complete a task.
Knowledge states	States of knowledge are the result of information processing activities.

Table 10: Social Organization and Cooperation Analysis Operations Description

SOCA Operations	Description
Evaluate actors' strengths and weaknesses	Use interviews and other knowledge elicitation methods with subject matter experts to identify actors and assign task responsibilities. The most common knowledge elicitation method is directly questioning domain experts about who will do what tasks. The tasks are the result of the Strategies analysis phase. Questions may include, but are not limited to, the following: <ul style="list-style-type: none"> o Describe the various teams using the system? o How do you allocate responsibilities for each person? o Who depends on who for help to complete a task? o What is the specific role of each team member? o How are decisions usually made?
Construct use case diagrams	Use data collected during interviews and information processing activities and knowledge states from the Strategies Analysis phase to construct use case diagrams. Use SysML use case diagrams. Actors and Use case diagrams are used to represent the system users, information processing activities, and knowledge states.

Table 11: Social Organization and Cooperation Analysis Attributes Description

SOCA Attributes	Description
Information Processing Activities	Information processing activities are the mental or cognitive activities system operators must utilize to complete a task.
Knowledge States	States of knowledge are the result of information processing activities.
Actors	Specify a role played by a person or thing when interacting with a system.

Table 12: Worker Competencies Analysis Operations Description

WCA Operations	Description
Describe skill-, rule-, or knowledge-based behavior	Use interviews and other knowledge elicitation methods with subject matter experts to identify the level of knowledge required by the user to complete information processing activities. The most common knowledge elicitation method is direct questioning of domain experts. The information processing activities are the result of the Control Task Analysis and Strategies Analysis phases. Questions may include, but are not limited to, the following: <ul style="list-style-type: none"> o What information does the user have to know in order to complete the information processing activities? o What rules, regulations, or policies does the user need to know? o What problem solving procedures will the user have to be familiar with?
Construct Skill, Rule Knowledge inventory diagram	The information processing activities come from the Control Task Analysis and Strategies Analysis phases.

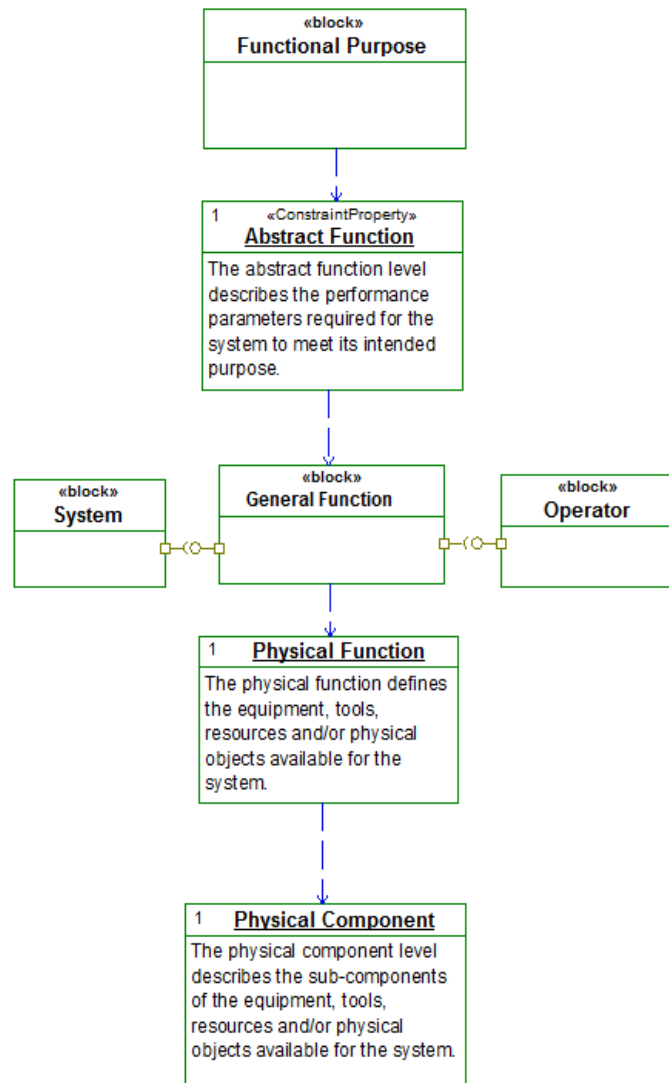
Information processing activities input	Use data collected during interviews and information processing activities from the Control Task Analysis and Strategies Analysis phases to construct a Skill, Rule and Knowledge Inventory diagram. Use SysML swimlanes, Send Action diagrams and Action diagrams to represent the level of cognitive behavior (i.e., Skill-Based Behavior (SBB), Rule-Based Behavior (RBB), Knowledge-Based Behavior (KBB)), information processing activities from the Control Task Analysis and Strategies Analysis phases, and level of knowledge required by the user, respectively.
---	--

Table 13: Workers' Competencies Analysis Attributes Description

WCA Attributes	Description
Skill-Based Behavior (SBB)	A skill-based behavior requires very little conscious effort to perform a task. Using a mouse to move a cursor is an example of a skill-based behavior.
Rule-Based Behavior (RBB)	A rule-based behavior is based on the rules and/or procedures established by an organization. For example, user instructions or regulatory authority rules necessary to complete a task or use equipment.
Knowledge-Based Behavior (KBB)	A knowledge-based behavior requires the highest level of conscious effort to complete a task. An example of a knowledge-based behavior is a pilot response to losing both engines due to bird strikes and landing the airplane in the Hudson River.
Information processing activity	Information processing activities are the mental or cognitive activities system operators must utilize to complete a task.

4.3.2 CWAT Abstraction Hierarchy

Figure 41 shows a detailed view of the Abstraction Hierarchy (AH) modeling tool available in the WDA phase of CWA. The AH has several levels of abstraction and requires a variety of diagrams to build. To build the AH in SysML, blocks, constraints properties, and part diagrams as well as dependency association lines are used. The functional purpose and general function levels is represented with block diagrams. The abstract function level is represented with constraint property diagrams. The physical function and physical component levels are represented with part diagrams. The SysML diagrams are embed with a description that informs systems engineers of the diagram uses and significance. In addition, the higher level of abstraction depends on the lower level of abstraction, therefore a dependency association is used to show the relationship. The dependency association is represented with dash lines with arrows at the end. Finally, there is also a text-based description of the AH and a translation of AH terminology to SysML diagrams.



Abstraction Hierarchy

The Abstraction Hierarchy (AH) describes the work domain. There are five levels of abstraction used to describe the work domains. They are:

- Functional purpose
- Abstract function
- General function
- Physical function
- Physical components

The functional purpose describes the reasons the system exist. The abstract function level describes the performance parameters required for the system to meet its intended purpose. The general function level describes the basic functions of the system. The physical function describes the functional capabilities by the physical objects of the system. The physical component level describes the equipment, tools, resources and/or physical objects available in the work domain.

Finally, each level is connected by means-ends or how-why relationships. The means is a level below the ends. For example, the general function is the means for the abstract function. The lower levels describe the actions, components or parameters that are necessary for achieving the ends or upper levels of the AH.

SysML Translation

AH is constructed in a Block Definition Diagram (BDD)

Block diagrams should be use to represent the functional purpose and general function levels.

Part diagrams should be used to represent the physical function and component levels.

Constraint property diagrams should be used to represent the abstract function level.

Dependency lines are used to represent the means-ends relationships.

Figure 41: A Model-Based and Text-Based Description of an Abstraction Hierarchy

4.3.3 CWAT Decision Ladder

The next topic in the tutorial is the Control Task Analysis (ConTA) phase of CWA. Figure 42 shows a detailed description of the Decision Ladder (DL) modeling tool available in the ConTA phase of CWA. Figure 42 illustrates the SysML diagram representation of the DL. It is constructed using a state machine diagram. The DL is composed of information processing activities and knowledge states. They are represented by send action and state diagrams respectively. The blue numbers in Figure 42 shows the process steps' potential order in the DL. These numbers are a guide, not a strict sequence to follow. Different levels of user expertise and knowledge will yield different courses of action or shortcuts taken by the user. For example, an expert may do steps 3, 4, 7, and 8. In contrast, a novice may do all the steps. The red arrows are transition lines and illustrate the transitional relationships between the information processing activities and knowledge states. Finally, there is the text-based description of the DL and a translation of DL terminology into SysML diagrams. Figure 43 shows a text-based of the ConTA phase and the DL. It is located on the same screen as Figure 42.

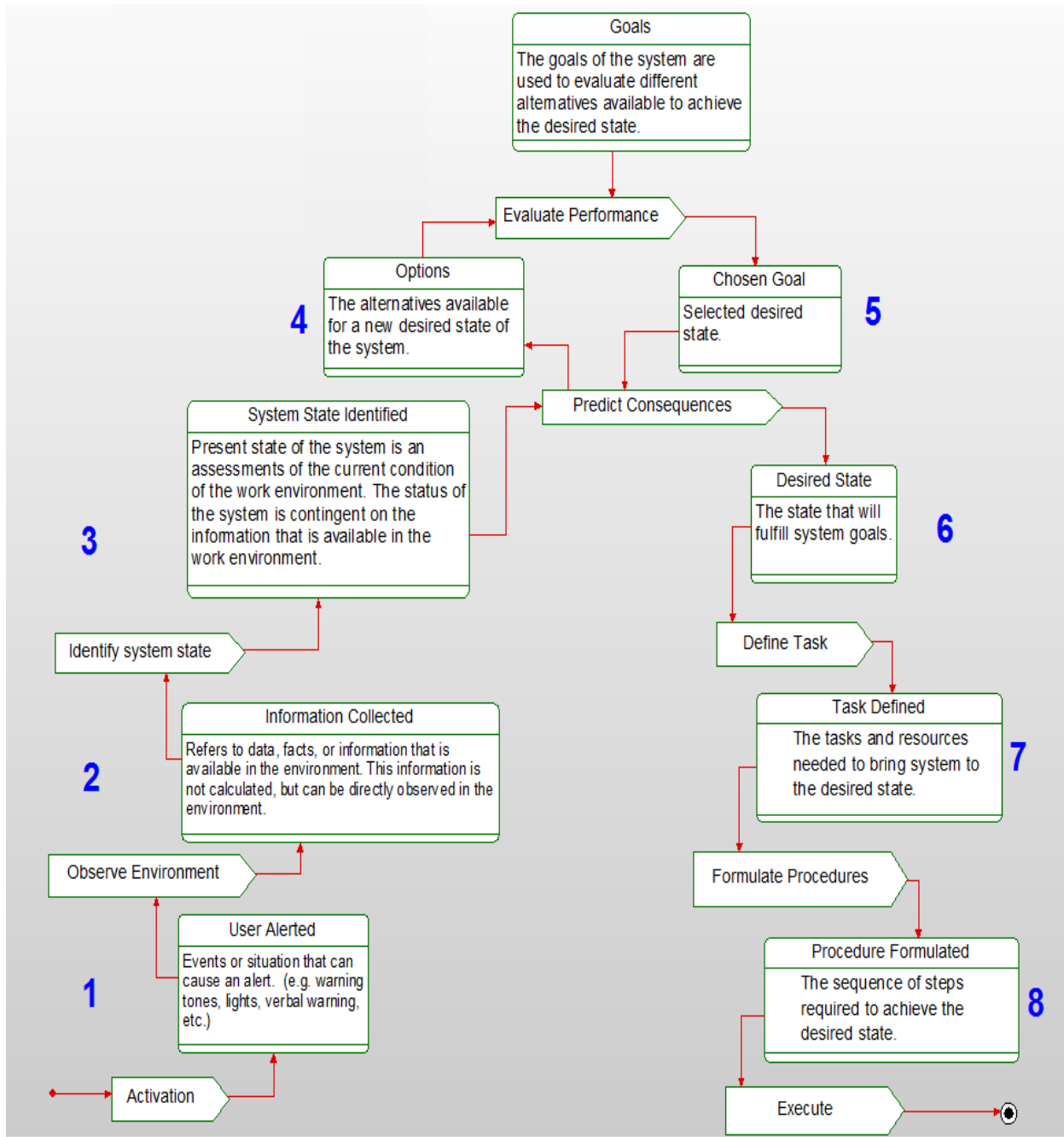


Figure 42: A Model-Based Description of a Decision Ladder

Decision Ladder

The modeling technique used in the ConTA is a Decision Ladder (DL). The DL is composed of information processing activities and knowledge states. Information processing activities are the mental or cognitive activities system operators must utilize to complete a task. Knowledge states are the resultant of the information processing activities.

The DL receives input from the Abstraction Hierarchy of the first phase. The functional purpose is the goals of the DL and the general functions are decomposed into the control tasks. The DL could be divided into three sections. The left side of the DL template are tasks the user will implement to identify the current system state. The top part of the DL template represents the system goals. The right side of the DL template are tasks the user will implement to identify resources, scheduling and carrying out actions.

The ladder is reused for each control tasks. Information processing activities and knowledge states are highlighted for different users. For example an expert will use his experience and knowledge to skip steps on the DL. These shortcuts are called shunts and leaps. They show actor shortcuts from one part of the decision ladder to another. Shunts connect an information processing activity to a state of knowledge. Leaps connect two knowledge states.

SysML Translation

The DL is built with a State Machine Diagram.

Default Transition arrow indicates the start of the DL.

Information processing activities are represented by Send Actions diagrams.

State diagrams are used to represent the knowledge states.

Transition arrows are used to show the transition from information processing activities to knowledge states.

Termination state is used to represent completion of the Control Task.

Shunts and Leaps are represented by free shape lines.

Figure 43: A Text-Based Description of a Decision Ladder

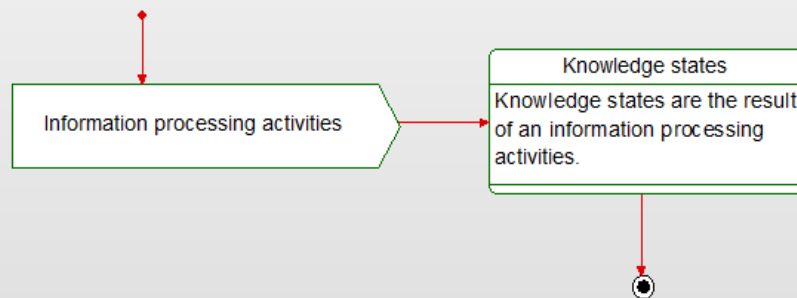
4.3.4 CWAT Information Flow Maps

The next topic in the tutorial is the Strategies Analysis (SA) phase of CWA. Figure 44 shows a detailed description of the Information Flow Map (IFM) modeling tool available in the SA phase of CWA. Figure 44 illustrates a SysML diagram representation of the IFM. It is constructed using a state machine diagram. The purpose of the IFM is to investigate the different ways in which each of the control tasks from the ConTA phase could be accomplished. The IFM is composed of information processing activities and knowledge states. They are represented by send action and state diagrams respectively. The red arrows are transition lines and illustrate the transitional relationships between the information processing activities and knowledge states. Finally, there is the text-based description of the IFM and a translation of IFM terminology into SysML diagrams.

Section 2

A model-based and text-based description of the Information Flow Maps modeling tool for the Strategies Analysis phase of CWA is provided as well as a System Modeling Language(SysML) translation.

Double click on send action or action and go to the “description” tab for more detail.



[Previous](#) | [Next](#) | [Outline](#)

Information Flow Maps

The strategy analysis (SA) phase of CWA can be used to identify general strategies to complete a control task. There is usually a variety of ways to complete a control task. An actor's experience and expertise will play a major role in how the control tasks in their work domain is completed. The SA phase will not identify every strategy an actor will use. Only the most commonly recommended or practical strategies are defined in the SA phase.

Overall, SA phase should support the actor's ability to effectively use the system regardless of the strategy chosen. Information flow maps (IFM) is the modeling technique use in the SA phase. The IFM is composed of information processing activities and knowledge states of single strategy used to complete a control task. Information processing activities are the mental or cognitive activities system operators must utilize to complete a task. Knowledge states are the result of the information processing activities. The IFM of the SA phase provide more detail about the "Formulate Procedure" information processing activities that is on the DL in the second phase.

SysML Translation

The DL is built with a State Machine Diagram.

Default Transition arrow indicates the start of the DL.

Information processing activities is represented by Send Actions.

State diagrams are used to represent the knowledge states.

Transition lines a used to represent the transition from information processing activities to knowledge states.

Termination state is used to represent completion of the Control Task.

Figure 44: A Model Based and Text-based Description of an Information Flow Map

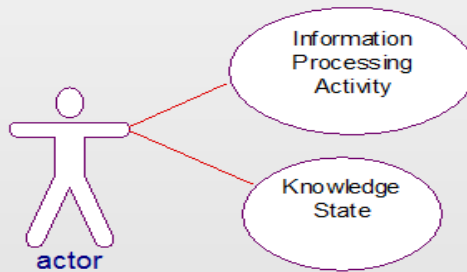
4.3.5 CWAT Use Case

The next topic in the tutorial is the Social Organization and Cooperation Analysis (SOCA) phase of CWA. Figure 45 shows a detailed description of the SysML use case (UC) modeling tool available in the SOCA phase of CWA. Figure 45 illustrates a SysML use case diagram of the IFM from the SA phase. The UC is composed of information processing activities, knowledge states, and actors. UC diagrams are used to represent the information processing activities and knowledge states, while actor diagrams are used to represent the actors that interact with the system. The red lines are used to show the relationships between actors and the UC. Association lines are used to show an interaction between two actors and a use case. Several other types of relationships can be used in the SysML UC diagrams. There are generalization, specialization, extend, include, and flow relationships. Finally, there is the text-based description of the UC and a translation of SOCA phase terminology into SysML diagrams.

Section 2

A model-based and text-based description of use case models for the Social Organization and Cooperation Analysis phase of CWA is provided as well as a System Modeling Language(SysML) translation.

Double click on the actor or use case and go to the “description” tab for more detail.



[Previous](#) | [Next](#) | [Outline](#)

Use Case

The Social Organization and Cooperation Analysis (SOCA) phase determines who will carry out the work and how it is shared. Typically, the Information Flow Map modeling techniques is used to represent the allocation of work. In SysML a use case diagram will be used to represent the allocation of tasks between actors. Each strategy will be represented by a use case. The use case diagram receives its input from the third phase of CWA. Each of the use case diagrams will have to be evaluated to determine which actor is more effective at completing the use case. For example, a human actor should be assigned the decision making and problem solving tasks and a computer system actor should be assigned the data and computationally intensive tasks.

SysML Translation

Use case will be used to represent the allocation of tasks between actors.

Each actor is represented by an actor diagram.

Each information processing activity and knowledge state is represented by a use case diagram.

The relationship between actors and use case diagrams are association links.

Figure 45: A Model Based and Text-based Description of an Use Case

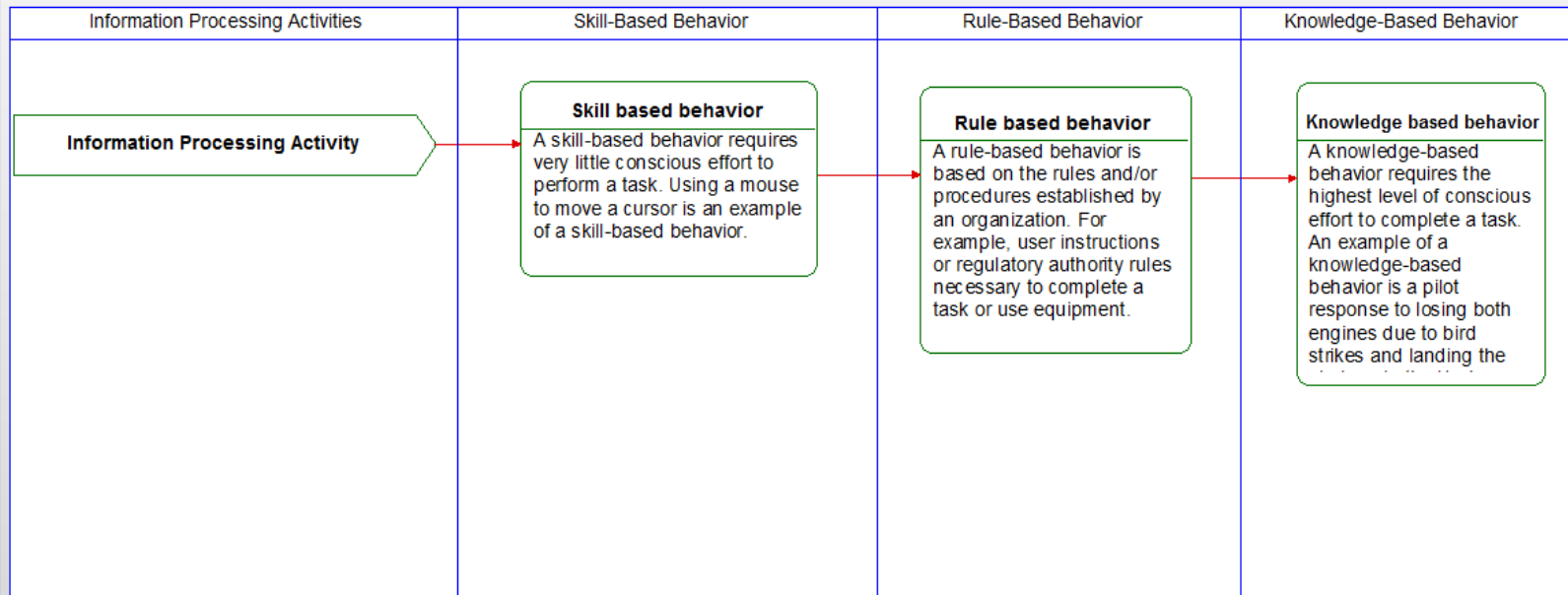
4.3.6 CWAT Skill, Rule, and Knowledge Inventory

The final topic in section 2 of the CWAT is the Worker Competencies Analysis (WCA) phase of CWA. Figure 46 illustrates an Activity diagram that provides a model-based description of the Skill, Rule, and Knowledge (SRK) inventory modeling tool. This modeling illustrates the knowledge and skill required to complete an information processing activity. The SRK inventory is composed of information processing activities and three psychological processes (i.e., skill-, rule-, or knowledge-based behavior). The SRK inventory is created using the swimlane diagrams available in SysML. The swimlanes are divided into information processing activities and skill-based, rule-based, and knowledge-based behavior lanes. Send action diagrams are used to represent the information processing activities. While action diagrams are used to represent the skill-, rule-, and knowledge-based behaviors. The red lines are control flow lines. The control flow lines show the relationship between the information processing activity and the Skill-, Rule-, and Knowledge-Based Behaviors. The control flow lines show the transition from information processing activity to the three psychological processes. Each psychological process is a level of competency required by the system end user to complete an information processing activity.

Section 2

A model-based and text-based description of Skill, Rule, and Knowledge Inventory modeling tool for the Worker Competencies Analysis phase is provided as well as a System Modeling Language(SysML) translation.

Double click on the send action or action diagrams and go to the “description” tab for more detail.



End of Section 2

[Previous](#) | [Next](#) | [Outline](#)

Figure 46: A Model Based Description of the Skill, Rule, and Knowledge Inventory

Finally,

Figure 47 shows a text-based description of the SRK inventory and a translation of SRK inventory terminology into SysML diagrams.

<p>Skill Rule & Knowledge Inventory</p> <p>The final phase of the CWA framework is the Worker Competencies Analysis (WCA) phase. It outlines the competencies that the system users must have or acquire in order to effectively perform control tasks. Skill, Rule, and Knowledge-based (SRK) inventory is the modeling tool used in this phase. SRK ascertain the level of conscious effort an individual uses when completing an information processing activity. The SRK inventory can also be used for personnel selection and training.</p> <p>The skill based category requires almost no conscious effort when completing a task. For example when a person is approaching a red traffic light when driving, they instinctually step on the brakes.</p> <p>The rule based category is centered on the rules of the organization or the proper procedures learned in training.</p> <p>The knowledge based category requires a high level of conscious effort to complete a task. For instance, when a person learns to drives a car for the first time or when a worker encounters an unanticipated event.</p>
<p>SysML Translation</p> <p>Swimlanes are used to represent the SRK inventory model for the 5th phase.</p> <p>Action diagram are used to represent the tasks at the different levels of psychological behaviors.</p> <p>Send Action is used to represent the information processing activities.</p>

Figure 47: A Text-Based Description of the Skill, Rule, and Knowledge Inventory

4.4 Process Flow Charts Section

The process flow charts section of the tutorial begins with a CWAT process flow overview. Figure 48 illustrates the CWAT process flow overview. The process flow chart promotes understanding of the CWA model-building process by using graphical symbols to depict the flow of the steps in the process. The user can review any CWA phase process flow chart by entering into the corresponding activity diagram. For example, the user can click on the activity diagram symbol on the WDA block on CWAT process flow and view Figure 48.

The CWAT process flow chart overview is constructed similarly to Figure 40 with the exception of the activity diagrams located in the upper right side corner of each CWA phase block and the addition of the CWA block. Figure 40 illustrates an IBD for the CWA Process Flow Chart block in Figure 37. This IBD contains the flow of information between the different phases of CWA and the sequence in which the phases should be completed. Figure 48 uses blocks and flow arrows to illustrate the flow of information between CWA phases. Each phase is represented by a different color. Each flow arrow contains the information being transferred to each CWA phase. The information being transferred is written above the flow line. The CWA block has a dependency relationship with the five phases. The numbers above the blocks represent the order in which each phase should be done. Each block contain attributes and operations for each CWA phase. The attributes describe the components of each phase. The operations describe the task that should be performed for each phase. In addition,

each action diagram will contain sample questions for the knowledge elicitation aspect of CWA. Usually the user will have to interview a domain expert about the domain of interest and how the expert performs tasks.

4.4.1 CWAT Process Flow for Constructing a CWA Structure

The first step in section three of the CWAT is the construction of a CWA structure. Figure 49 illustrates an activity diagram that shows the process for building a CWA structure with SysML. It is composed of initial flow symbol, action blocks, control flow arrows, and an activity final symbol. The initial flow represents the start of the process. The action blocks represent functions of the process. The control flow arrows represent the transition from one state to another. The activity final symbol represents the completion of the process. Each action block contains a number that corresponds to a step in the process. Each step number matches a step in the example image in Figure 50. Additionally, there are step-by-step instructions located within the description section of each action block on the process flow chart.

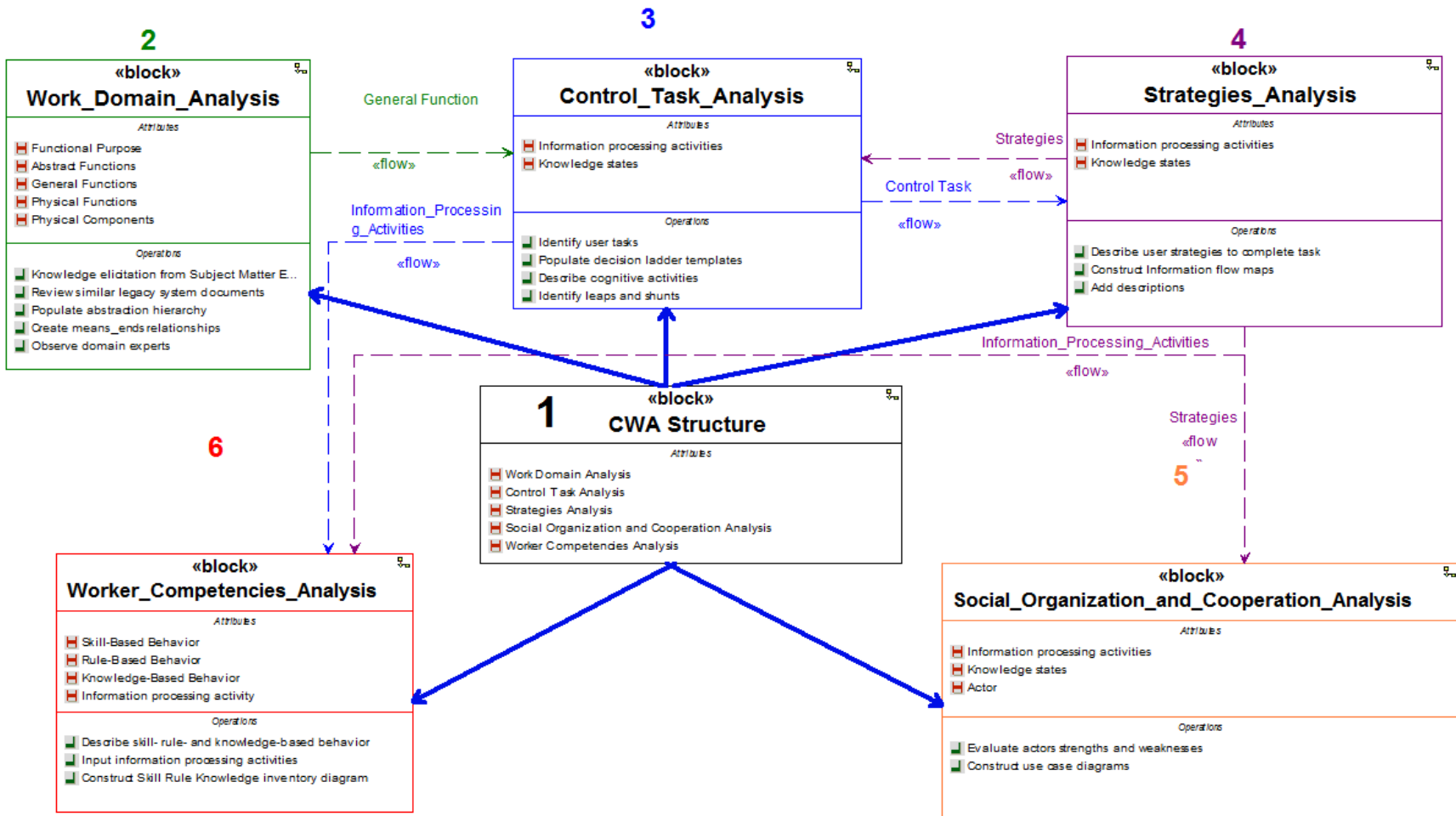


Figure 48: CWAT Process Flow Overview

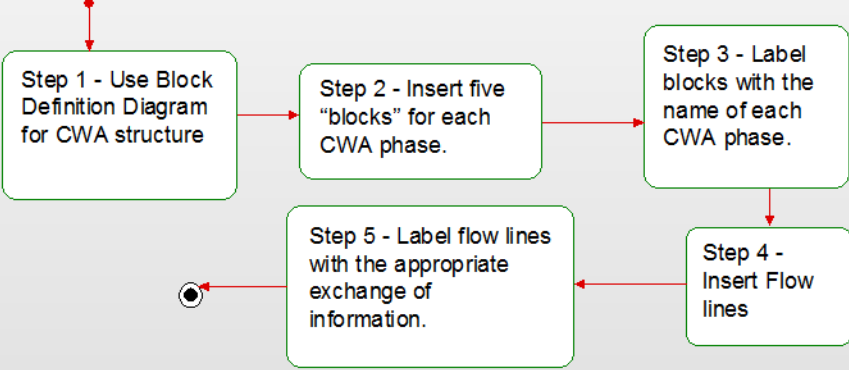
act [block] CWA Structure [ActivityOfCWA Structure]

Section 3

The Activity diagrams show the step by step process of constructing a CWA framework.

The steps in figure 1 corresponds to the steps in the Activity diagram.

Double click on any action diagram and go to the “description” tab for more detail.



```
graph TD; Start(( )) --> Step1[Step 1 - Use Block Definition Diagram for CWA structure]; Step1 --> Step2[Step 2 - Insert five "blocks" for each CWA phase.]; Step2 --> Step3[Step 3 - Label blocks with the name of each CWA phase.]; Step3 --> Step4[Step 4 - Insert Flow lines]; Step4 --> Step5[Step 5 - Label flow lines with the appropriate exchange of information.]; Step5 --> End(( ))
```

[Previous](#) | [Next](#) | [Outline](#)

Figure 49: Process Flow Chart for Building a Cognitive Work Analysis Structure

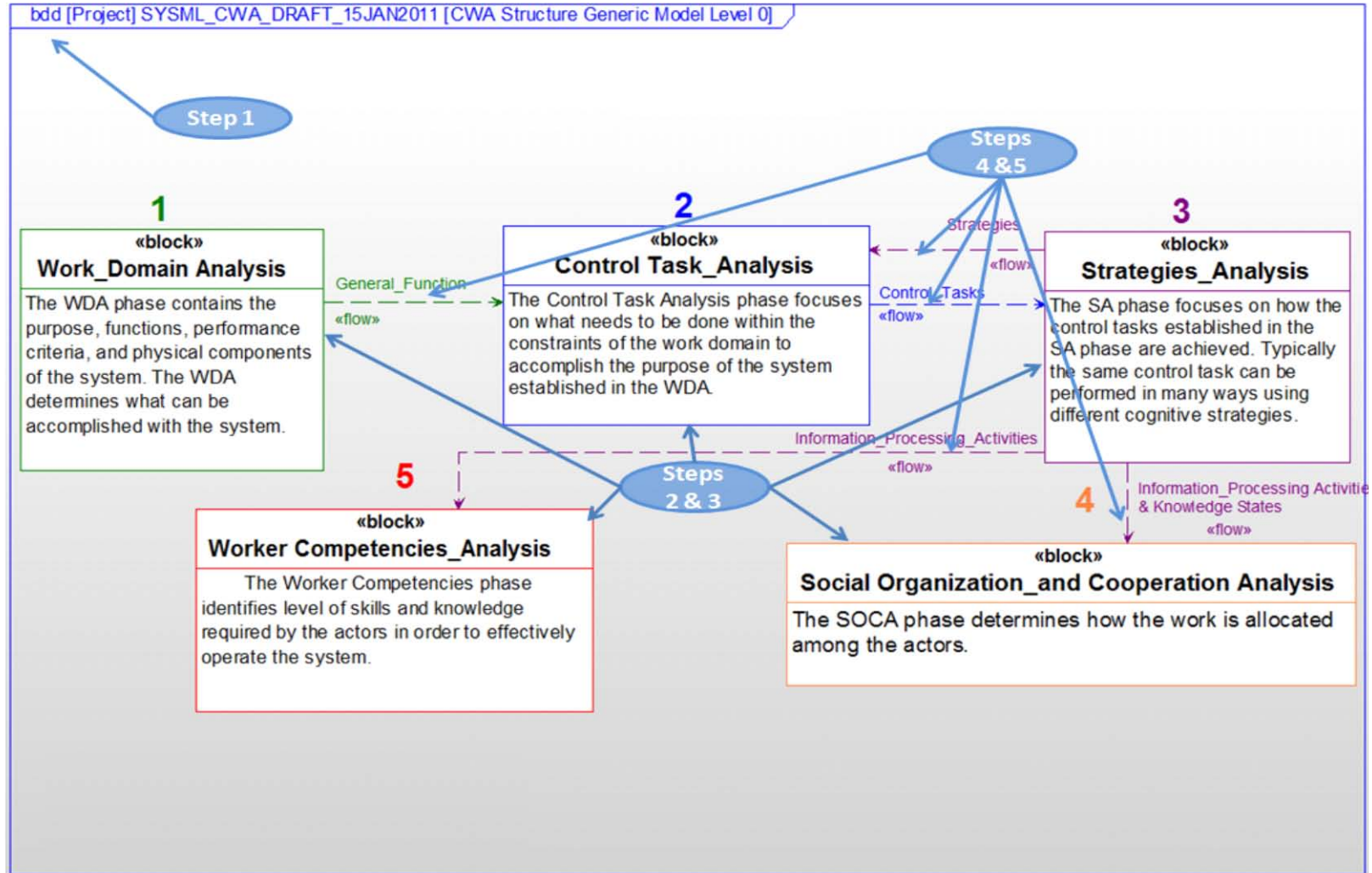


Figure 50: Process Flow Chart Steps for Building a Cognitive Work Analysis Structure

Each action block on Figure 49 contains more detail on how to complete the process step associated with that block. The detailed process steps are located in the description tab of each action block. The following detailed process steps are contained in each action block for constructing a CWA structure:

1. Use Block Definition Diagram for Cognitive Work Analysis structure.
 - a. Go to Tools-->Add New Diagram-->Block Definition Diagram.
 - b. Enter name of new diagram.
2. Insert 5 “blocks” for each CWA phase.
 - a. Select the Block icon on the right-side menu.
 - b. Move cursor over to the diagram frame area and left-click on mouse.
 - c. Repeat step four more times.
 - d. To customize block, move cursor over to block and right-click on mouse, then select Format and/or Display Options.
3. Label blocks with names of each phase.
 - a. Double-click on block and enter the following phases per block in the Name area of the pop-up window:
 - Work Domain Analysis (WDA)
 - Control Task Analysis (ConTA)
 - Strategies Analysis (SA)

- Social Organization and Cooperation Analysis (SOCA)
- Worker Competencies Analysis (WCA)

4. Insert flow lines.

- a. Select the Flow icon on the right-side menu.
- b. Move cursor over to the Work Domain Analysis block.
- c. Left-click mouse,
- d. Move cursor over to the Control Task Analysis.
- e. Left-click mouse.
- f. Repeat step for other flow lines (From ConTA to SA and WCA, from SA to SOCA and WCA.)

5. Label flow lines with the appropriate exchange of information:

- a. from WDA to ConTA = General Functions;
- b. from ConTA to SA = Control Tasks;
- c. from ConTA to WCA = Information Processing Activities (IPA);
- d. from SA to SOCA = Information Processing Activities and Knowledge States (IPA & KS);

- e. from SA to WCA = Information Processing Activities (IPA); and
- f. from SA to ConTA = Information Processing Activities and Knowledge States (IPA & KS).

4.4.2 CWAT Process Flow for Constructing an Abstraction Hierarchy

The second step in section three of the CWAT is the construction of the Abstraction Hierarchy (AH). Figure 51 illustrates an activity diagram that shows the process for building an AH using SysML. It is composed of the same diagrams as Figure 49 (i.e., initial flow symbol, action blocks, control flow arrows, and an activity final symbol). The numbers contained in the action blocks in Figure 51 correspond to the steps in Figure 52. Each process step number matches a step number in Figure 52. To view the diagrams used to create the AH, refer to the CWA detailed description section of this dissertation. Additionally, there are step-by-step instructions located within the description section of each action block on the process flow chart. Within each action block is an explanation of which SysML menus and diagrams to select to complete the models.

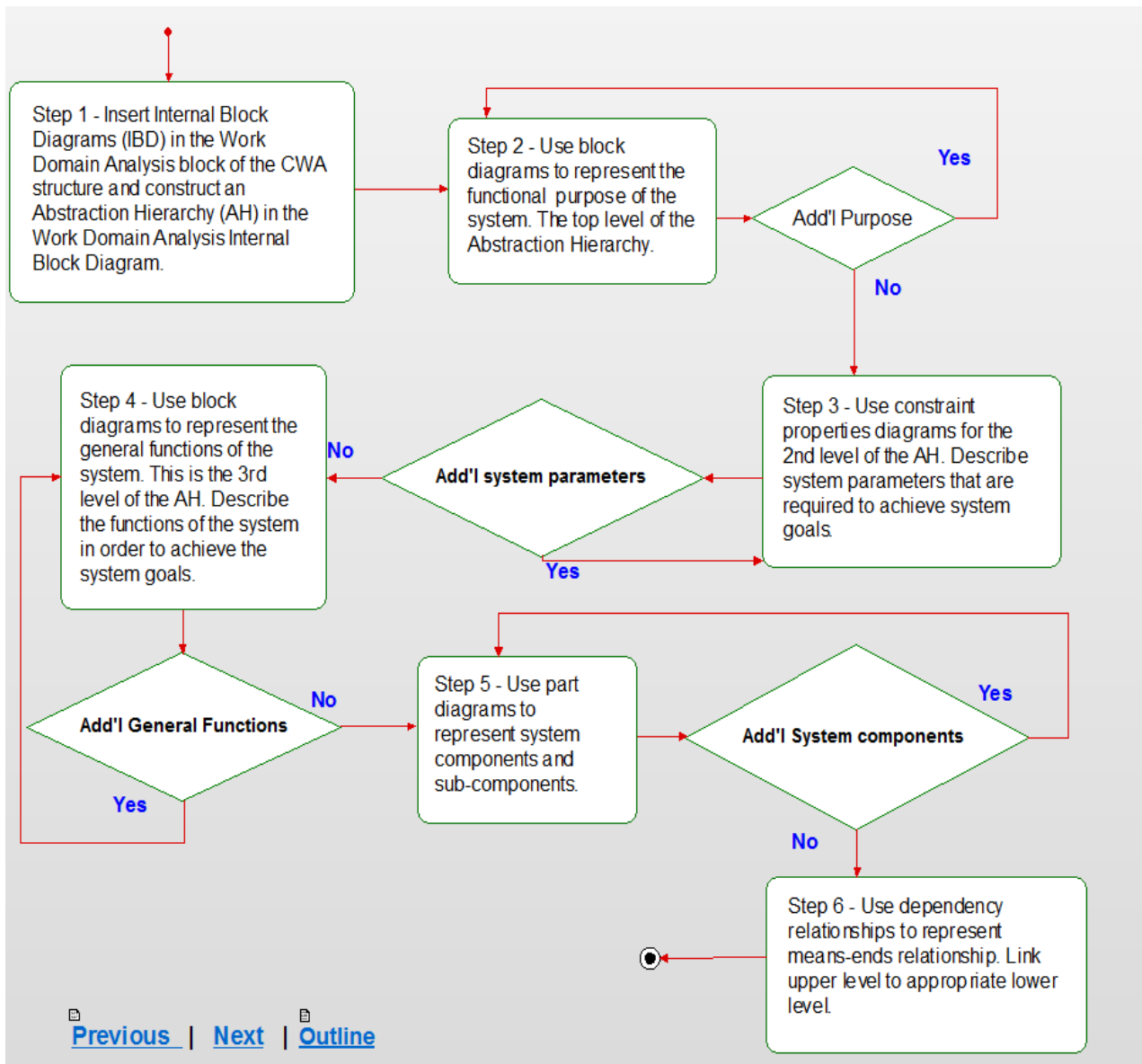


Figure 51: A Process Flow Chart for Building an Abstraction Hierarchy

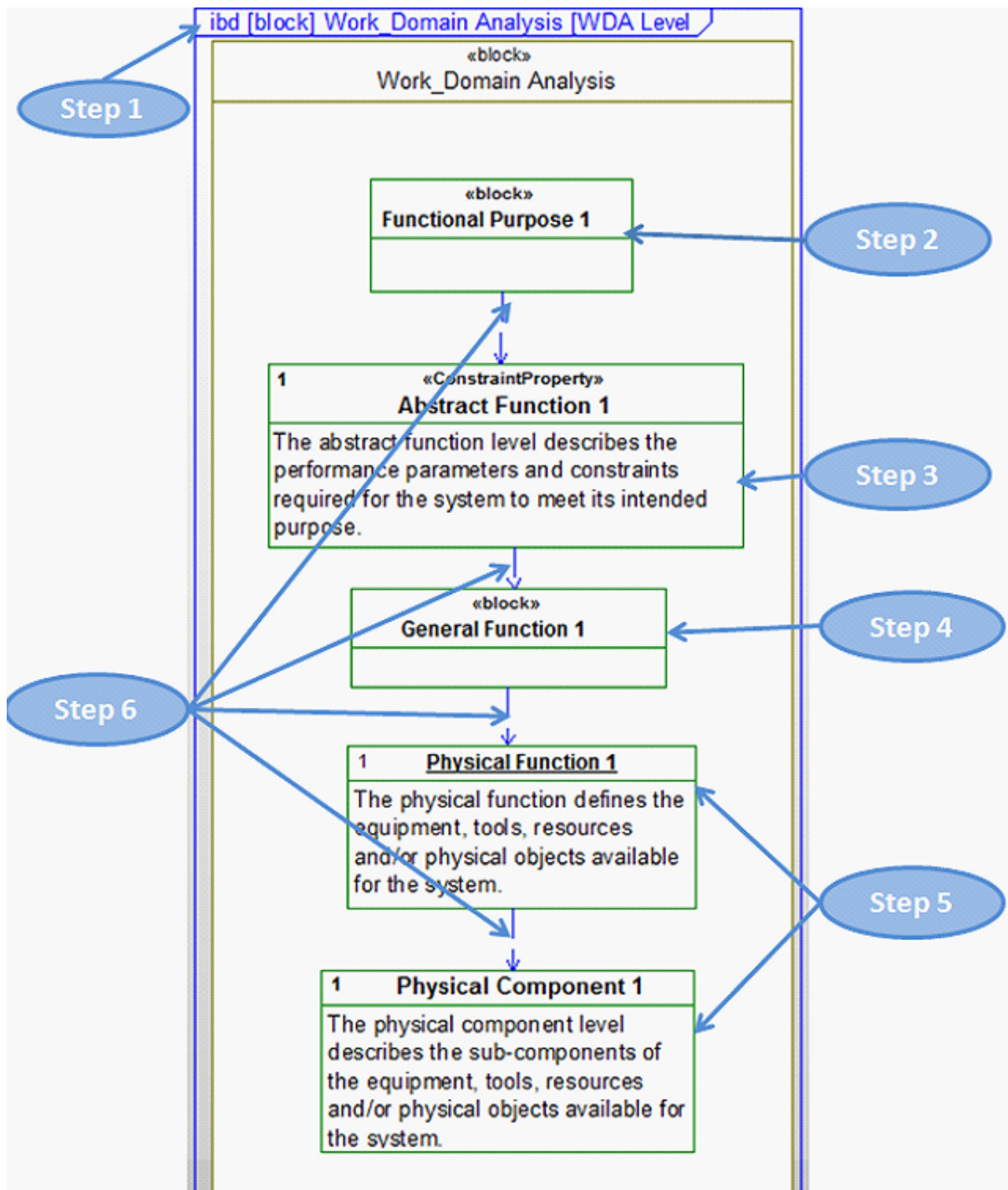


Figure 52: Process Flow Chart Steps for Building an Abstraction Hierarchy

Each action block on Figure 51 contains more detail on how to complete the process step associated with that block. The detailed process steps are located in the description tab of each action block. The following detailed process steps are contained in each action block for constructing an Abstraction Hierarchy:

1. Insert Internal Block Diagrams (IBD) in the Work Domain Analysis block of the CWA structure and construct an Abstraction Hierarchy (AH) in the Work Domain Analysis Internal Block Diagram.
 - a. Move cursor over Work Domain Analysis block.
 - b. Right-click mouse.
 - c. Select Add New --> Internal Block Diagram
2. Use block diagrams to represent the purpose of the system (the top level of the AH).
 - a. Select the Block icon on the right-side menu.
 - b. Move cursor over to the diagram frame area and left-click on mouse.
 - c. Name and describe the primary goals or intention of the system.
 - d. Repeat step for additional system purposes.
3. Use constraint properties diagrams for the 2nd level of the AH.
 - a. Select the Constraint Properties icon on the right-side menu.
 - b. Move cursor over to the diagram frame area and left-click on mouse.

- c. Name and describe system parameters that are required to achieve system goals.
 - d. Repeat step for additional system constraints.
- 4. Use block diagrams to represent the general functions of the system. This is the 3rd level of the AH.
 - a. Select the Block icon on the right-side menu.
 - b. Move cursor over to the diagram frame area and left-click on mouse.
 - c. Name and describe the functions of the system in order to achieve the system goals.
 - d. Repeat step for additional general function.
- 5. Use part diagrams to represent system components and sub-components.
 - a. Select the Part icon on the right-side menu.
 - b. Move cursor over to the diagram frame area and left-click on mouse.
 - c. Name and describe the components of the system.
 - d. Repeat step for additional components and sub-components.
- 6. Use dependency relationships to represent means-ends relationship.
 - a. Select the Dependency icon on the right-side menu.

- b. Move cursor over to the Purpose block.
- c. Left-click mouse.
- d. Move cursor over to the Constraint Properties icon.
- e. Left-click mouse.

4.4.3 CWAT Process Flow for Constructing a Decision Ladder

The third step in section three of the tutorial is the construction of the Decision Ladder (DL). Figure 53 illustrates an activity diagram that shows the process for building a DL using SysML. It is composed of an initial flow symbol, action blocks, control flow arrows, and an activity final symbol. The numbers contained in the action blocks in Figure 53 correspond to the steps in Figure 54 and Figure 55. In other words, each process flow chart step number matches a step number in Figure 54 and Figure 55. To view the diagrams used to create the DL, refer to the CWA Detailed Description section of this dissertation. Additionally, there are step-by-step instructions located within the description section of each action block on the process flow chart. Within each action block is an explanation of which SysML menus and diagrams to select to complete the models. Figure 53 contains the instruction within each process flow chart action block.

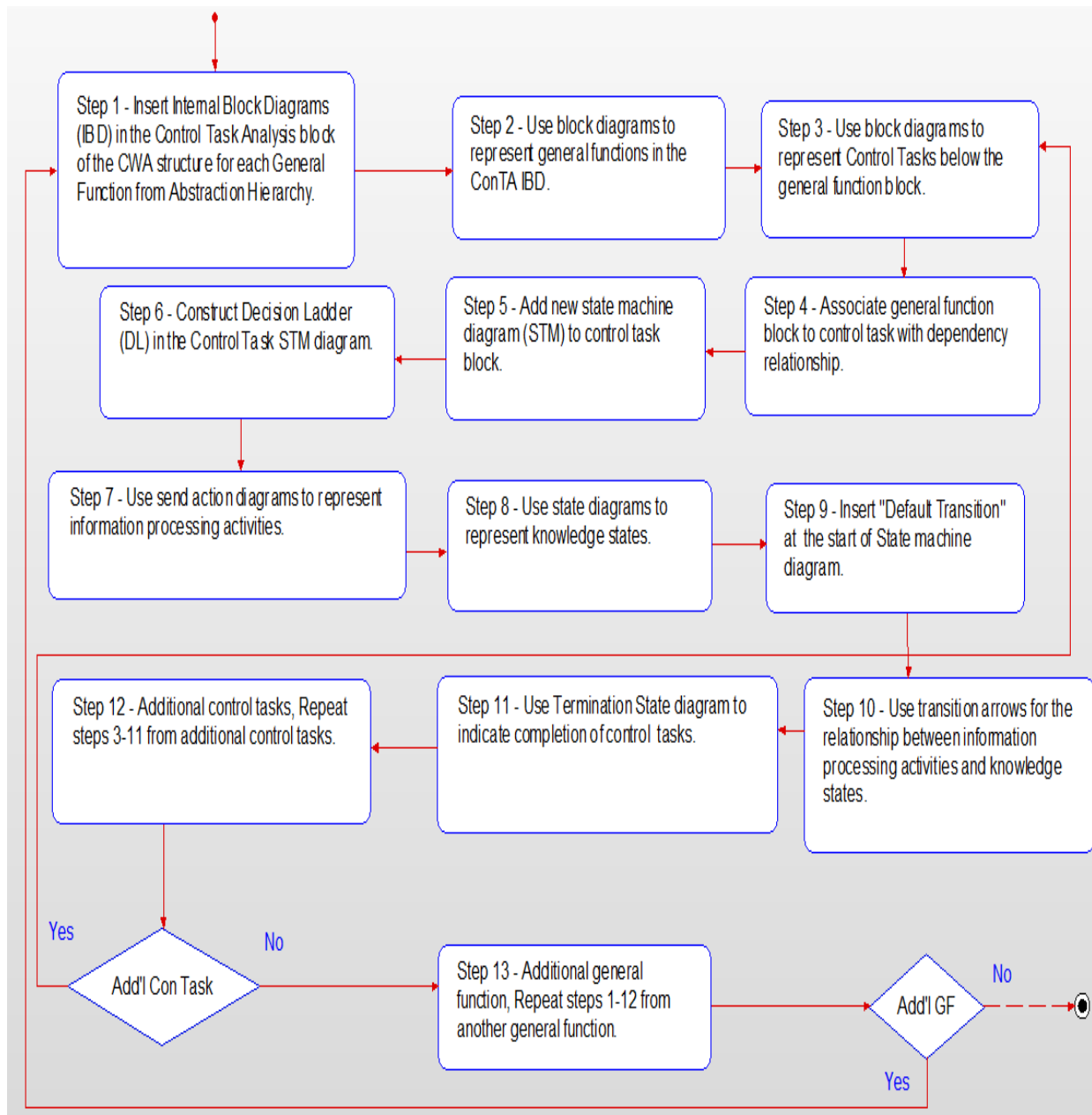


Figure 53: A Process Flow Chart for Constructing a Decision Ladder

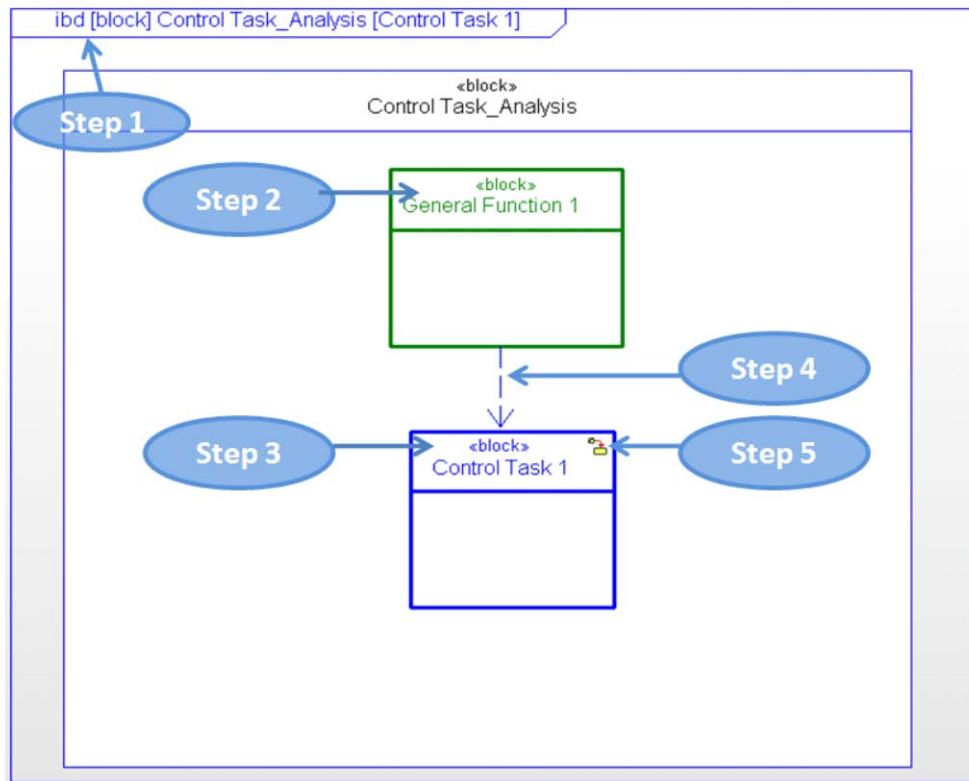


Figure 54: Process Flow Chart Steps for Building a Control Task Analysis Structure

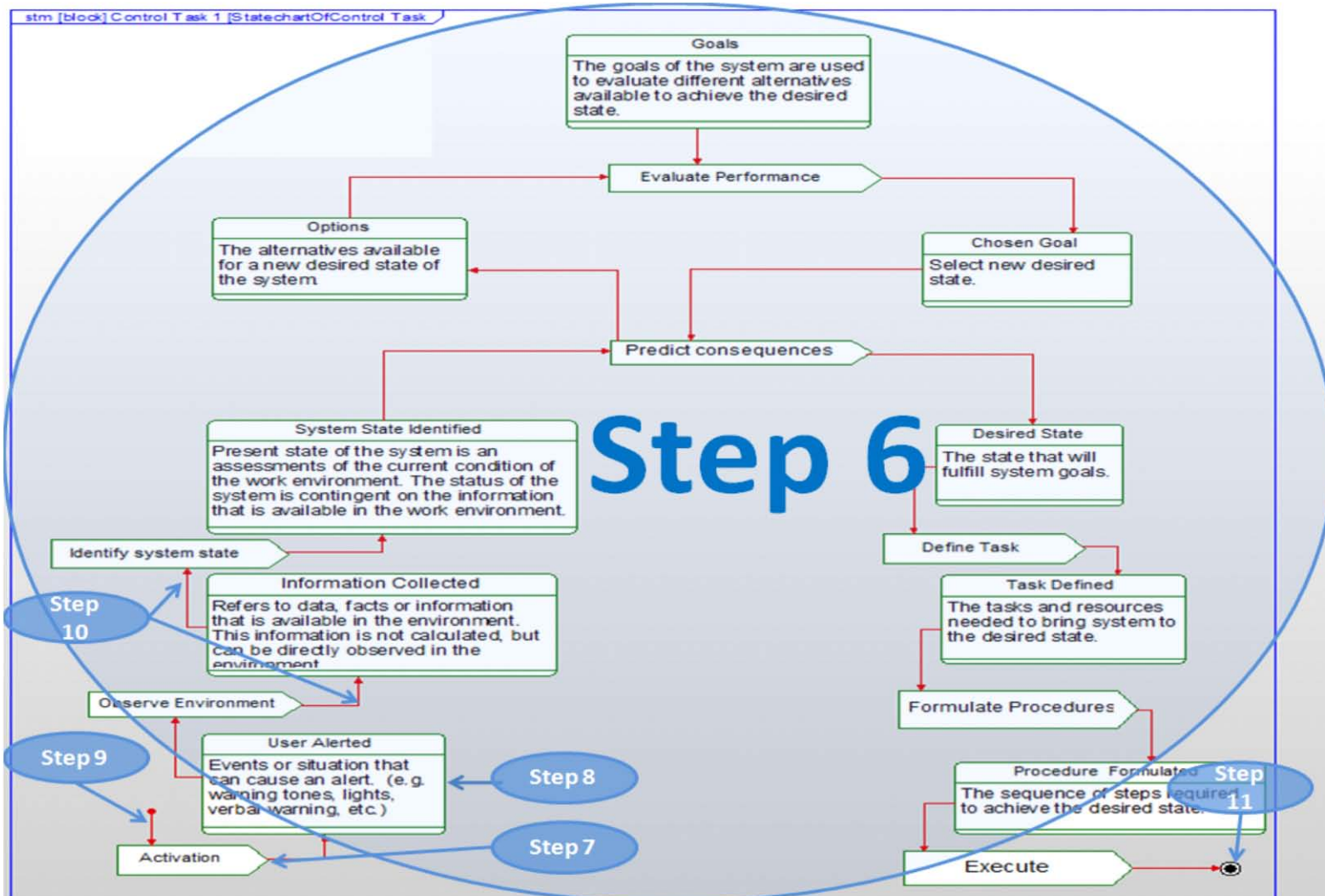


Figure 55: Process Flow Chart Steps for Building a Decision Ladder

Each action block on Figure 53 contains more detail on how to complete the process step associated with that block. The detailed process steps are located in the description tab of each action block. The following detailed process steps are contained in each action block for constructing a Decision Ladder:

1. Insert Internal Block Diagrams (IBD) in the Control Task Analysis block of the CWA structure for each general function from the Abstraction Hierarchy.
 - a. Move cursor over the Control Task Analysis block.
 - b. Right-click mouse,
 - c. Select Add New --> Internal Block Diagram.
2. Use block diagram to represent general function in the ConTA IBD.
 - a. Select the Block icon on the right-side menu.
 - b. Move the cursor over to the diagram frame area and left-click on mouse.
 - c. Name and describe the general function.
3. Use block diagram to represent Control Task below the general function block.
 - a. Select the Block icon on the right-side menu.
 - b. Move cursor over to the diagram frame area and left-click on mouse.
 - c. Name and describe the Control Task.
 - d. Repeat step for additional Control Tasks.

4. Relate general function block to control task with dependency relationship.
 - a. Select the Dependency icon on the Diagram Tools menu located on the right side of screen.
 - b. Move cursor over to the General Function block.
 - c. Left-click mouse.
 - d. Move cursor over to the Control Task block.
 - e. Left-click mouse.
 - f. Repeat step for each additional Control Task.
5. Add state machine diagram (STM) to Control Task block.
 - a. Move cursor over Control Task block.
 - b. Right-click on mouse.
 - c. Select Add New --> State Chart.
6. Construct Decision Ladder (DL) in the Control Task STM diagram.
 - a. Refer to Figure 4.
7. Use Send Action diagrams to represent information processing activities.
 - a. Select the Send Action icon on the right-side menu.
 - b. Move cursor over to the diagram frame area and left-click on mouse.
 - c. Name and describe the Send Action diagram.
 - d. Repeat step for each additional information processing activity.
8. Use State diagrams to represent knowledge states.

- a. Select the State icon on the right-side menu
 - b. Move cursor over to the diagram frame area and left-click on mouse.
 - c. Name and describe the State diagram.
 - d. Repeat step for each additional knowledge state.
9. Insert Default Transition at the start of the DL (i.e., Activation Send Action diagram).
- a. Select the Default Transition icon on the right-side menu.
 - b. Move cursor over to the diagram frame area and left-click on mouse.
 - c. Connect Default Transition to first Send Action diagram.
10. Use transition arrows for the relationship between information processing activities and knowledge states.
- a. Select the Transition icon on the right-side menu.
 - b. Move cursor over to the Send Action diagram.
 - c. Left-click mouse.
 - d. Move cursor over to the State diagram.
 - e. Left-click mouse.
 - f. Repeat step for each additional Send Action and State diagram.
11. Use Termination diagram to indicate completion of determining appropriate tasks to satisfy overall goal for the system.
- a. Select the Termination State icon on the right-side menu.

- b. Move cursor over to the diagram frame area and left-click on mouse.
 - c. Connect Termination State to last State diagram.
12. Repeat steps 3–11 for each additional control task.
13. Repeat steps 1–12 for each additional general function.

4.4.4 CWAT Process Flow for Constructing an Information Flow Maps

The fourth step in the process flow chart section of the tutorial is the construction of an Information Flow Map (IFM). Figure 56 is an activity diagram that presents a process flow chart for building an IFM using SysML. The process flow chart is composed of the same diagrams used in the previous steps. Finally, each process flow chart step number in Figure 56 matches each step number in Figure 57 and Figure 58. To view the diagrams used to create the IFM, refer to section 4.3.4 CWAT Information Flow Maps. Additionally, there are step-by-step instructions located within the description section of each action block on the process flow chart. Within each action block is an explanation of which SysML menus and diagrams to select to complete the models.

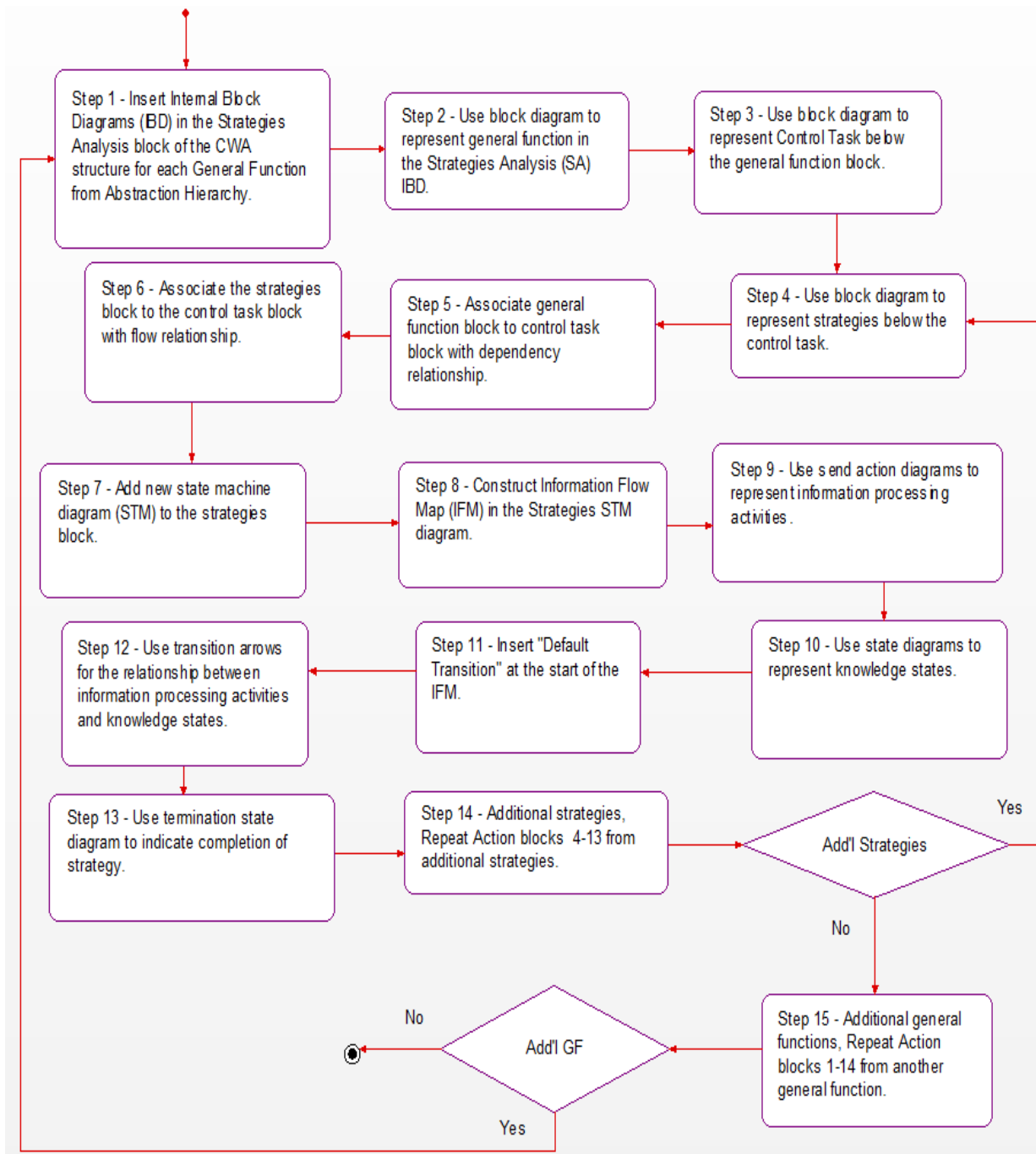


Figure 56: A Process Flow Chart for Constructing an Information Flow Map

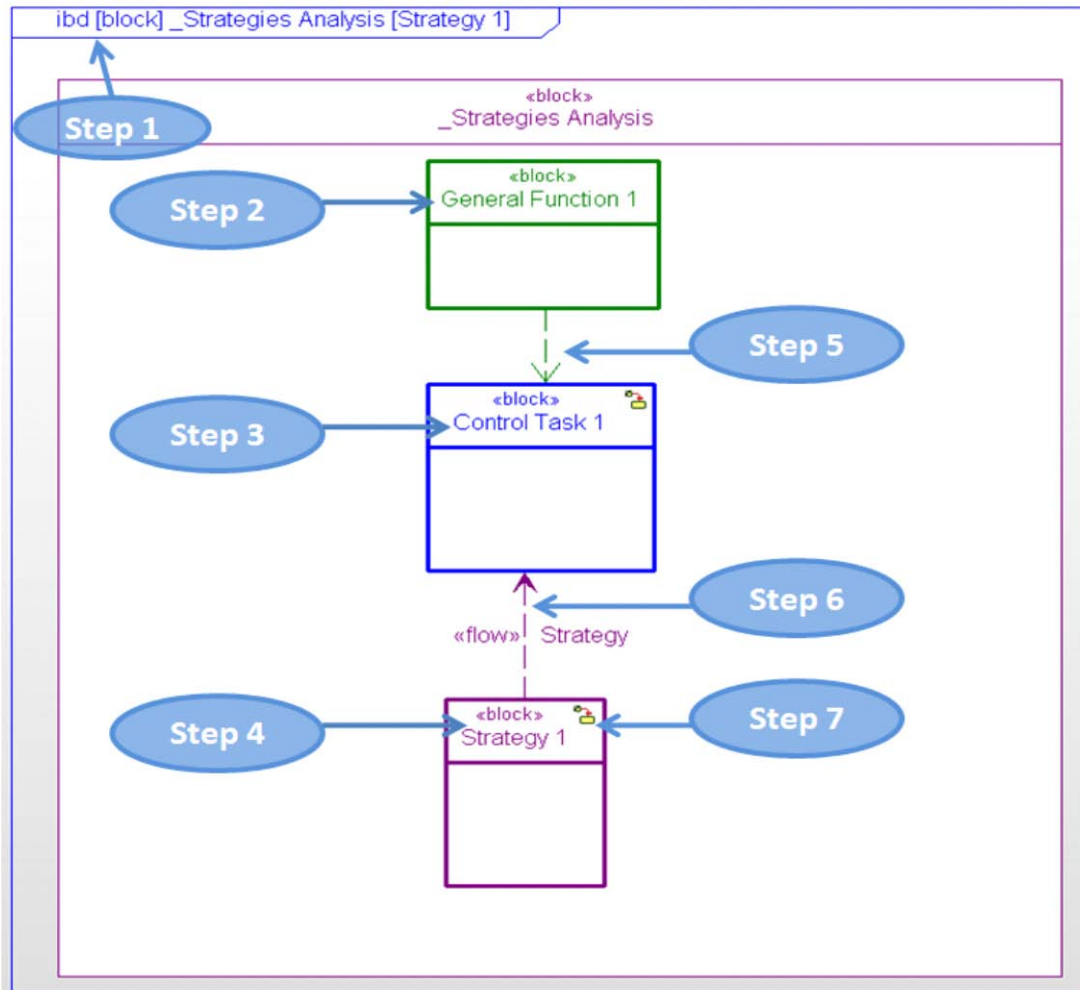


Figure 57: Process Flow Chart Steps for Building a Strategies Analysis Structure

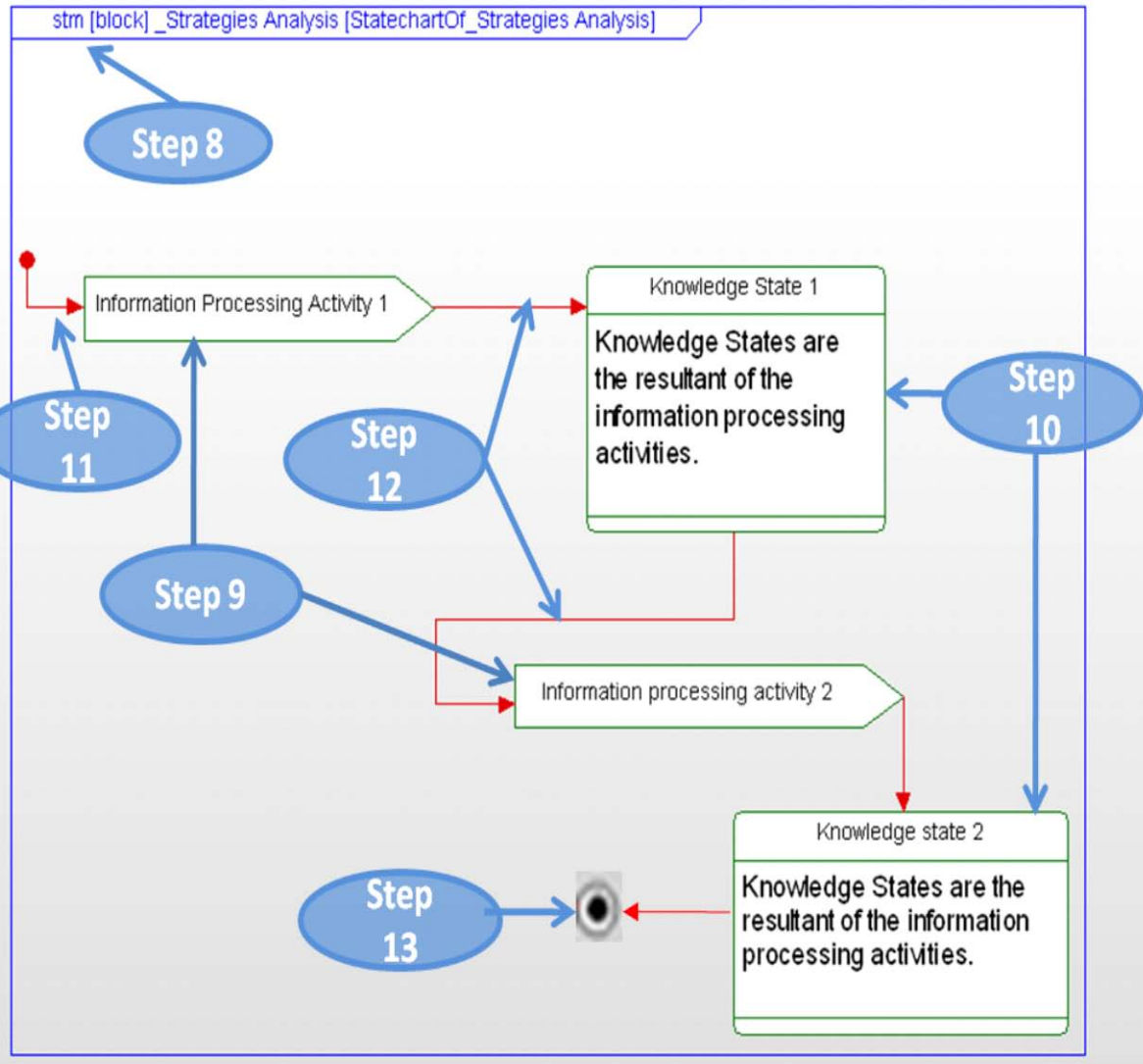


Figure 58: Process Flow Chart Steps for Building a Information Flow Maps

Each action block on Figure 56 contains more detail on how to complete the process step associated with that block. The detailed process steps are located in the description tab of each action block. The following detailed process steps are contained in each action block for constructing an Information Flow Map:

Strategies Analysis

1. Insert Internal Block Diagrams (IBD) in the Strategies Analysis block of the CWA structure for each general function from the Abstraction Hierarchy.
 - a. Move cursor over Strategies Analysis block.
 - b. Right-click mouse.
 - c. Select Add New --> Internal Block Diagram.
2. Use block diagram to represent general function in the Strategies Analysis (SA) IBD.
 - a. Select the Block icon on the right-side menu.
 - b. Move cursor over to the diagram frame area and left-click on mouse
 - c. Name and describe the general function
3. Use block diagram to represent Control Task below the general function block.
 - a. Select the Block icon on the right-side menu.
 - b. Move cursor over to the diagram frame area and left-click on mouse.
 - c. Name and describe the Control Task.
 - d. Repeat step 3 for each additional Control Task.
4. Use block diagram to represent strategies below the control task.
 - a. Select the Block icon on the right-side menu.
 - b. Move cursor over to the diagram frame area and left-click on mouse.
 - c. Name and describe the Control Task.
 - d. Repeat step 4 for each additional strategy.

5. Relate general function block to control task block with dependency relationship.
 - a. Select the Dependency icon on the right-side menu.
 - b. Move cursor over to the General Function block.
 - c. Left-click mouse.
 - d. Move cursor over to the Control Task block.
 - e. Left-click mouse.
 - f. Repeat step for each additional Control Task.
6. Associate the strategies block to the control task block with flow relationship.
 - a. Select the Flow icon on the right-side menu.
 - b. Move cursor over to the strategy block.
 - c. Left-click mouse.
 - d. Move cursor over to the Control Task block.
 - e. Left-click mouse.
 - f. Repeat step for each additional strategy block.
7. Add new state machine diagram (STM) to the strategies block.
 - a. Move cursor over Control Task block.
 - b. Right-click on mouse.
 - c. Select Add New --> State Chart.

8. Construct Information Flow Map (IFM) in the Strategies STM diagram.
 - a. Refer to figure.
9. Use send action diagrams to represent information processing activities.
 - a. Select the Send Action icon on the right-side menu.
 - b. Move cursor over to the diagram frame area and left-click on mouse.
 - c. Name and describe the Send Action diagram.
 - d. Repeat step for each additional information processing activity.
10. Use state diagrams to represent knowledge states.
 - a. Select the State icon on the right-side menu.
 - b. Move cursor over to the diagram frame area and left-click on mouse.
 - c. Name and describe the State diagram.
 - d. Repeat step for additional knowledge states.
11. Insert Default Transition at the start of the DL (i.e., Activation Send Action diagram).
 - a. Select the Default Transition icon on the right-side menu.
 - b. Move cursor over to the diagram frame area and left-click on mouse
 - c. Connect Default Transition to first Send Action or State diagram
12. Use transition arrows for the relationship between information processing activities and knowledge states.

- a. Select the Transition icon on the right-side menu.
 - b. Move cursor over to the Send Action diagram.
 - c. Left-click mouse.
 - d. Move cursor over to the State diagram.
 - e. Left-click mouse.
 - f. Repeat step for each additional Send Action and State diagram.
13. Use termination state diagram to indicate completion of strategy.
- a. Select the Termination State icon on the right-side menu.
 - b. Move cursor over to the diagram frame area and left-click on mouse.
 - c. Connect Termination State to last State diagrams.
14. Repeat steps 4–13 for each additional strategy.
15. Repeat steps 1–14 for each additional general function.

4.4.5 CWAT Process Flow for Constructing a Use Case

The fifth step in the process flow chart section of the CWAT is the construction of a Use Case diagram for the Social Organization and Cooperation Analysis (SOCA) phase of CWA. Figure 59 is an activity diagram that presents a process flow chart for building a UC using SysML. The process flow chart is composed of the same diagrams used in the previous steps. Each process flow chart step number in Figure 59 matches

each step number in

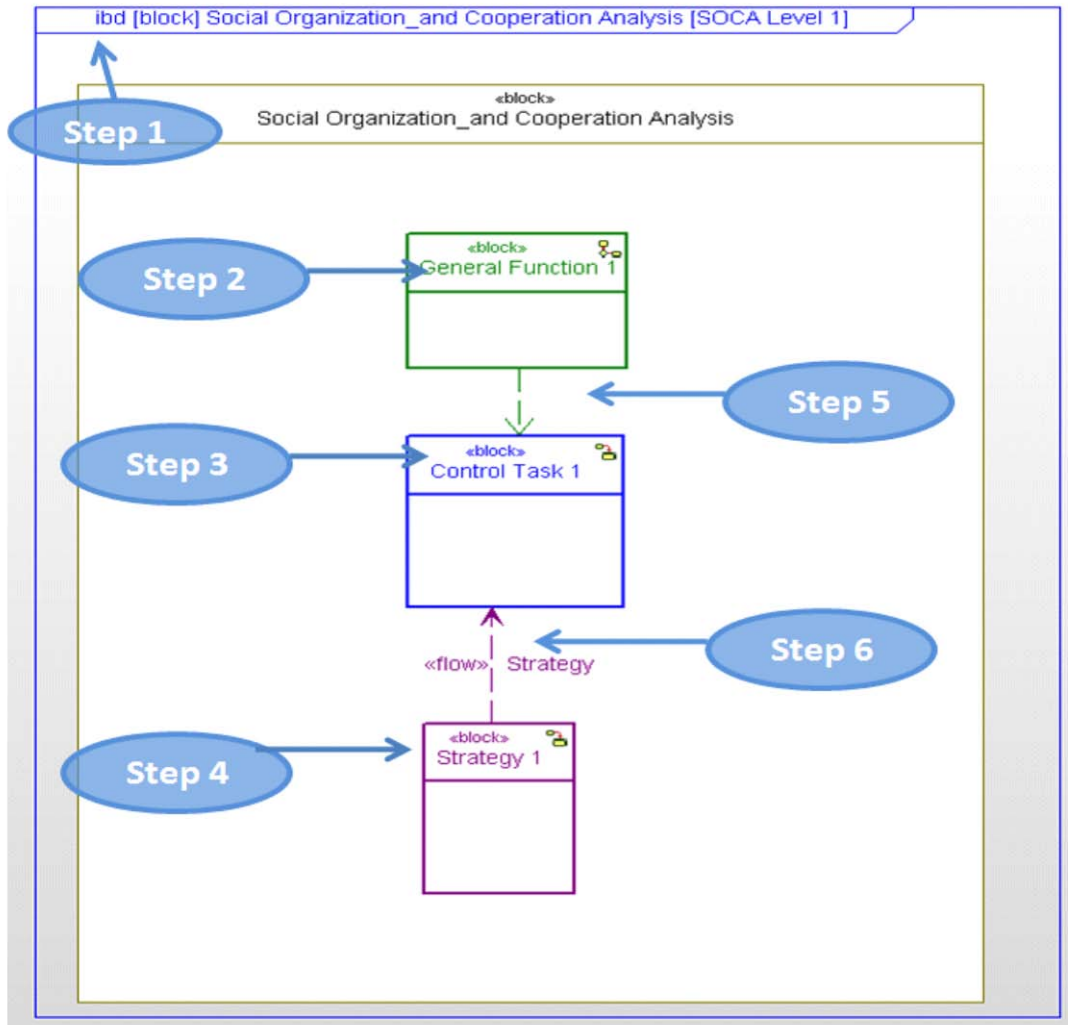


Figure 60 and Figure 61. To view the diagrams used to create the UC, refer to the CWA Detailed Description section of this dissertation. Additionally, there are step-by-step instructions located within the description section of each action block on the process flow chart. Within each action block is an explanation of which SysML menus and diagrams to select to complete the models.

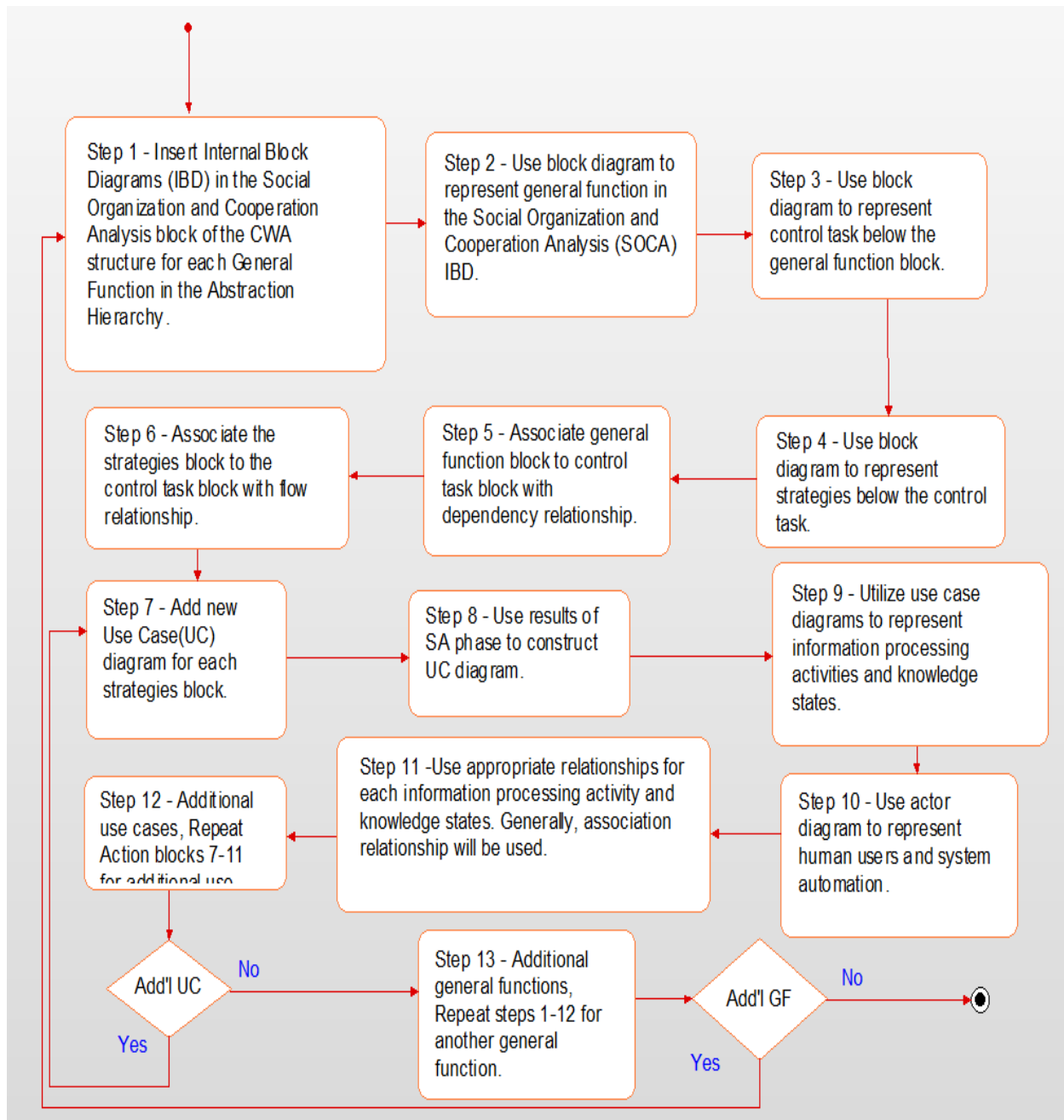


Figure 59: A Process Flow Chart for Constructing a Social Organization and Cooperation Analysis Use Case

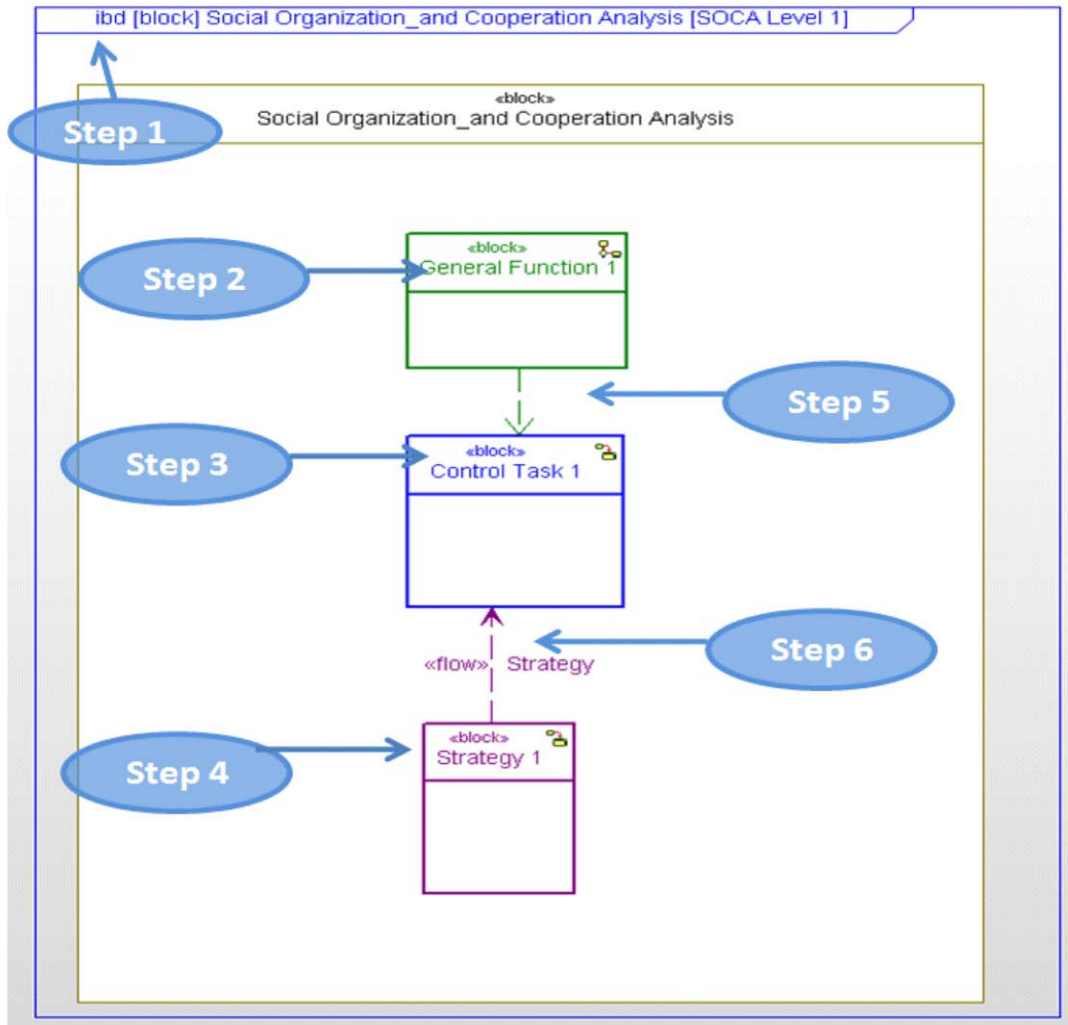


Figure 60: Process Flow Chart Steps for Building a Social Organization and Cooperation Analysis

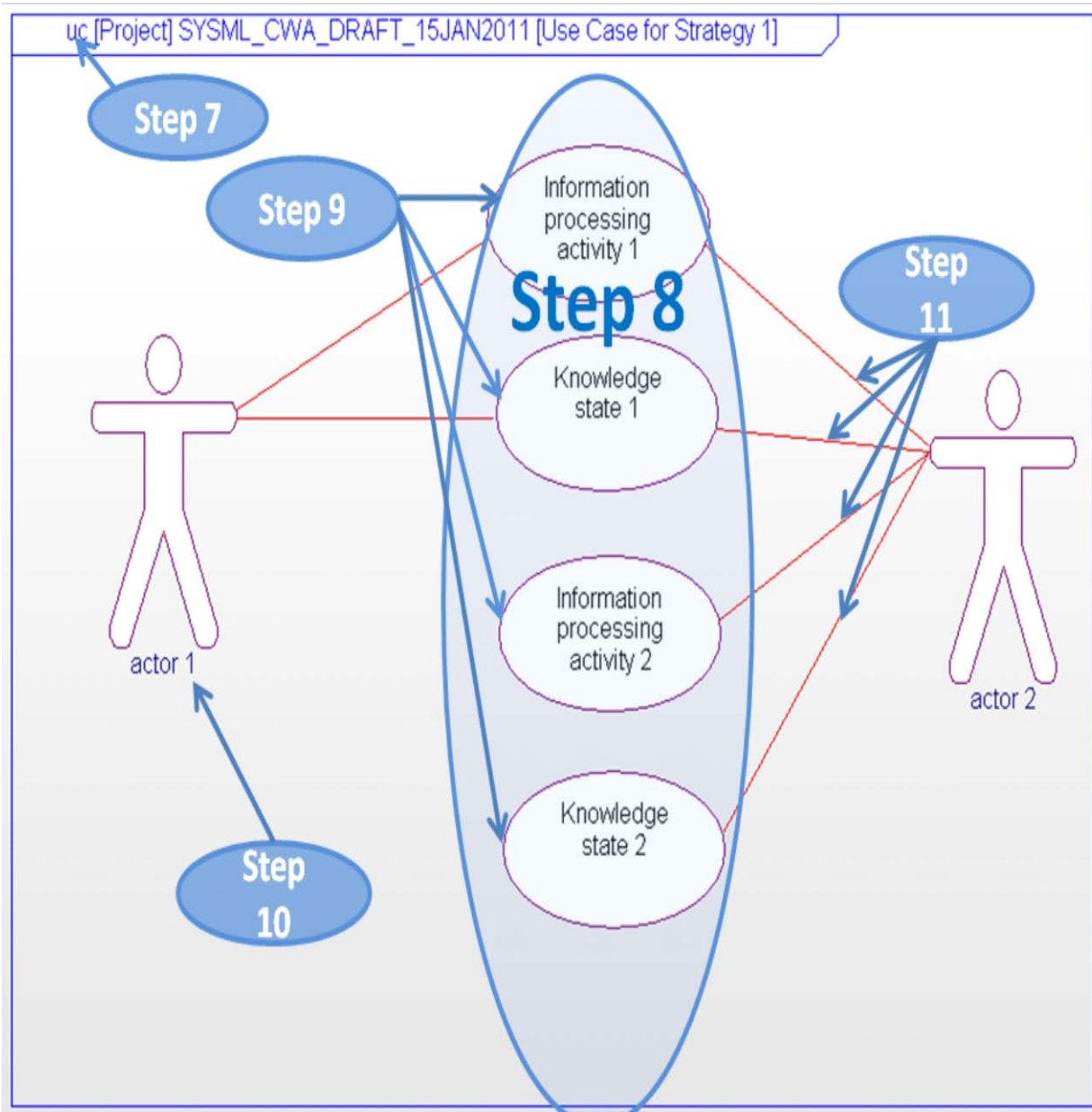


Figure 61: Process Flow Chart Steps for Building a Use Case

Each action block on Figure 59 contains more detail on how to complete the process step associated with that block. The detailed process steps are located in the

description tab of each action block. The following detailed process steps are contained in each action block for constructing a Use Case Diagram:

1. Insert Internal Block Diagrams (IBD) in the Social Organization and Cooperation Analysis block of the CWA structure for each general function in the Abstraction Hierarchy.
 - a. Move cursor over Social Organization and Cooperation Analysis block.
 - b. Right-click mouse.
 - c. Select Add New --> Internal Block Diagram.
2. Use block diagram to represent the general function in the Social Organization and Cooperation Analysis (SOCA) IBD.
 - a. Select the Block icon on the right-side menu.
 - b. Move cursor over to the diagram frame area and left-click on mouse.
 - c. Name and describe the general function.
3. Use block diagram to represent control task below the general function block.
 - a. Select the Block icon on the right-side menu.
 - b. Move cursor over to the diagram frame area and left-click on mouse.
 - c. Name and describe the Control Task.
 - d. Repeat step 3 for each additional Control Task.
4. Use block diagram to represent strategies below the control task.
 - a. Select the Block icon on the right-side menu.

- b. Move cursor over to the diagram frame area and left-click on mouse.
 - c. Name and describe the strategy.
 - d. Repeat step 4 for each additional strategy.
5. Associate general function block to control task block with dependency relationship.
 - a. Select the Dependency icon on the right-side menu/
 - b. Move cursor over to the General Function block.
 - c. Left-click mouse.
 - d. Move cursor over to the Control Task block.
 - e. Left-click mouse.
 - f. Repeat step for each additional Control Task.
6. Relate the strategies block to the control task block with flow relationship.
 - a. Select the Flow icon on the right-side menu.
 - b. Move cursor over to the strategy block.
 - c. Left-click mouse.
 - d. Move cursor over to the Control Task block.
 - e. Left-click mouse.
 - f. Repeat step for each additional strategy block.
7. Add new use case diagram (UC) for each strategy block.

- a. Go to Tools menu and select Add New --> Use Case
8. Use results of SA phase to construct UC diagram.
 - a. Refer to result of Strategies Analysis phase.
9. Utilize use case diagrams to represent information processing activities and knowledge states.
 - a. Select the Use Case icon on the right-side menu
 - b. Move cursor over to the diagram frame area and left-click on mouse.
 - c. Name and describe the use case.
 - d. Name and description come from the information processing activities and knowledge states in the Strategies Analysis phase.
10. Use actor diagram to represent human users and system automation.
 - a. Select the Actor icon on the right-side menu.
 - b. Move cursor over to the diagram frame area and left-click on mouse.
 - c. Name and describe the actor.
11. Use appropriate relationship for each information processing activity and knowledge state. Generally, an association relationship will be used.
 - a. Select the Generalization, Dependency, Flow, or Association icon on the right-side menu.
 - b. Move cursor over to the Actor diagram.
 - c. Left-click mouse.

d. Move cursor over to the Use Case diagram.

e. Left-click mouse.

f. Repeat step for each additional actor and use case diagram.

12. Repeat steps 7–11 for each additional use case.

13. Repeat steps 1–12 for each additional general function.

4.4.6 CWAT Process Flow for Constructing a Skill Rule and Knowledge Inventory

The sixth and final step in the process flow chart section of the tutorial is the construction of a Skill, Rule, and Knowledge (SRK) Inventory. Figure 62 is an activity diagram that presents a process flow chart for building an SRK inventory using SysML. The process flow chart is composed of the same diagrams used in the previous steps. Each process flow chart step number in Figure 62 matches each step number in Figure 63 and Figure 64. To view the diagrams used to create a SRK Inventory, refer to the CWA Detailed Description section of this dissertation. Additionally, there are step-by-step instructions located within the description section of each action block on the process flow chart. Within each action block is an explanation of which SysML menus and diagrams to select to complete the models.

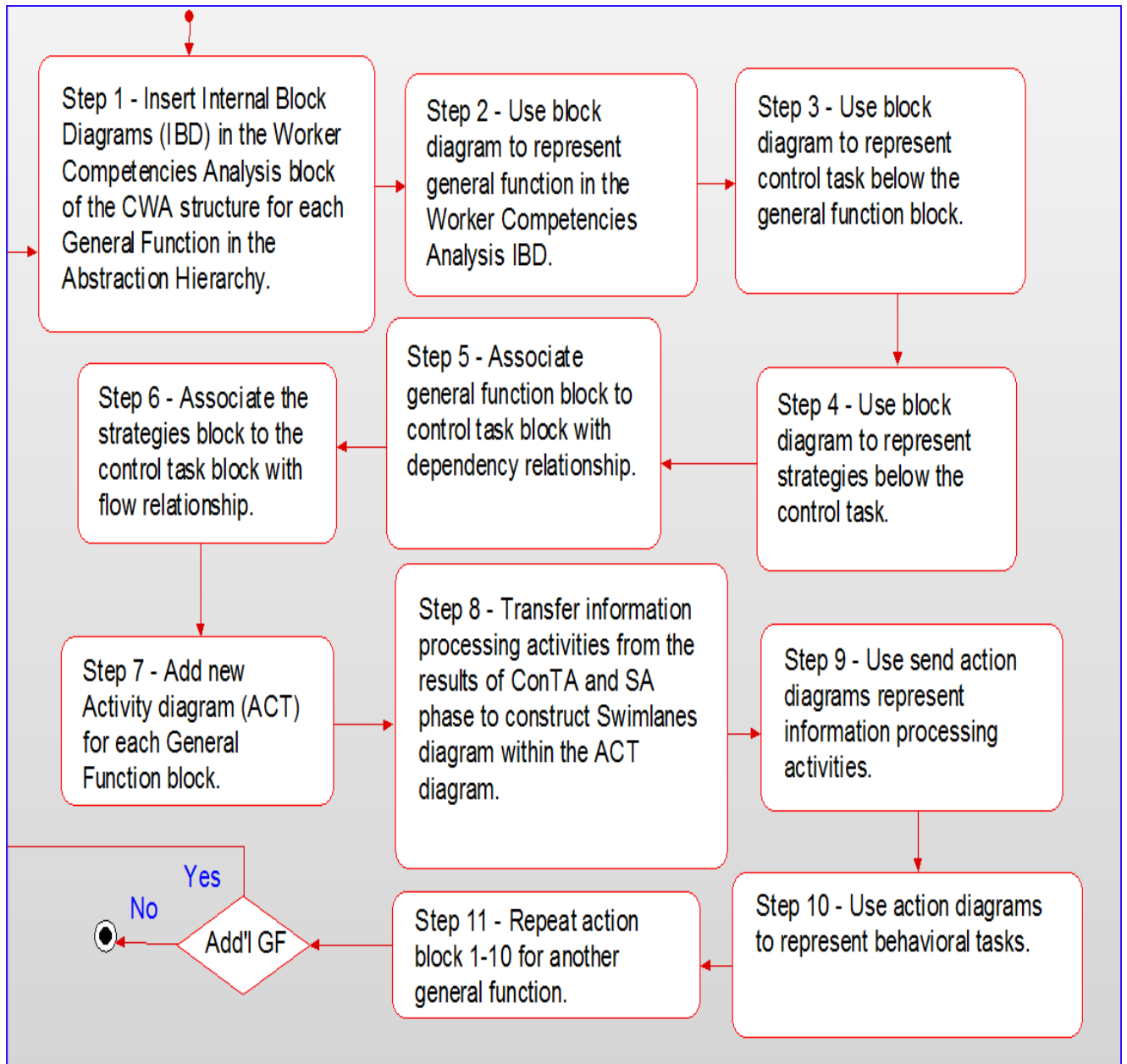


Figure 62: A Process Flow Chart for Constructing a Skill, Rule, and Knowledge Inventory using SysML

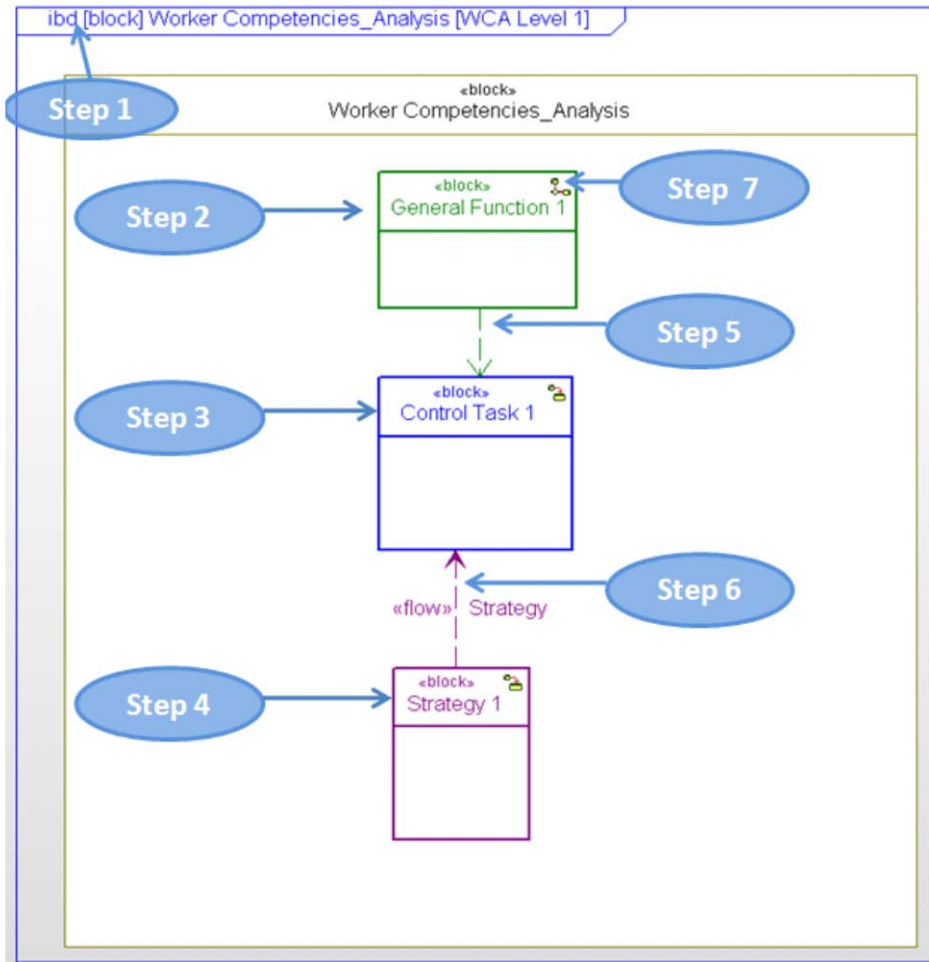


Figure 63: Process Flow Chart Steps for Building a Worker Competencies Analysis Structure

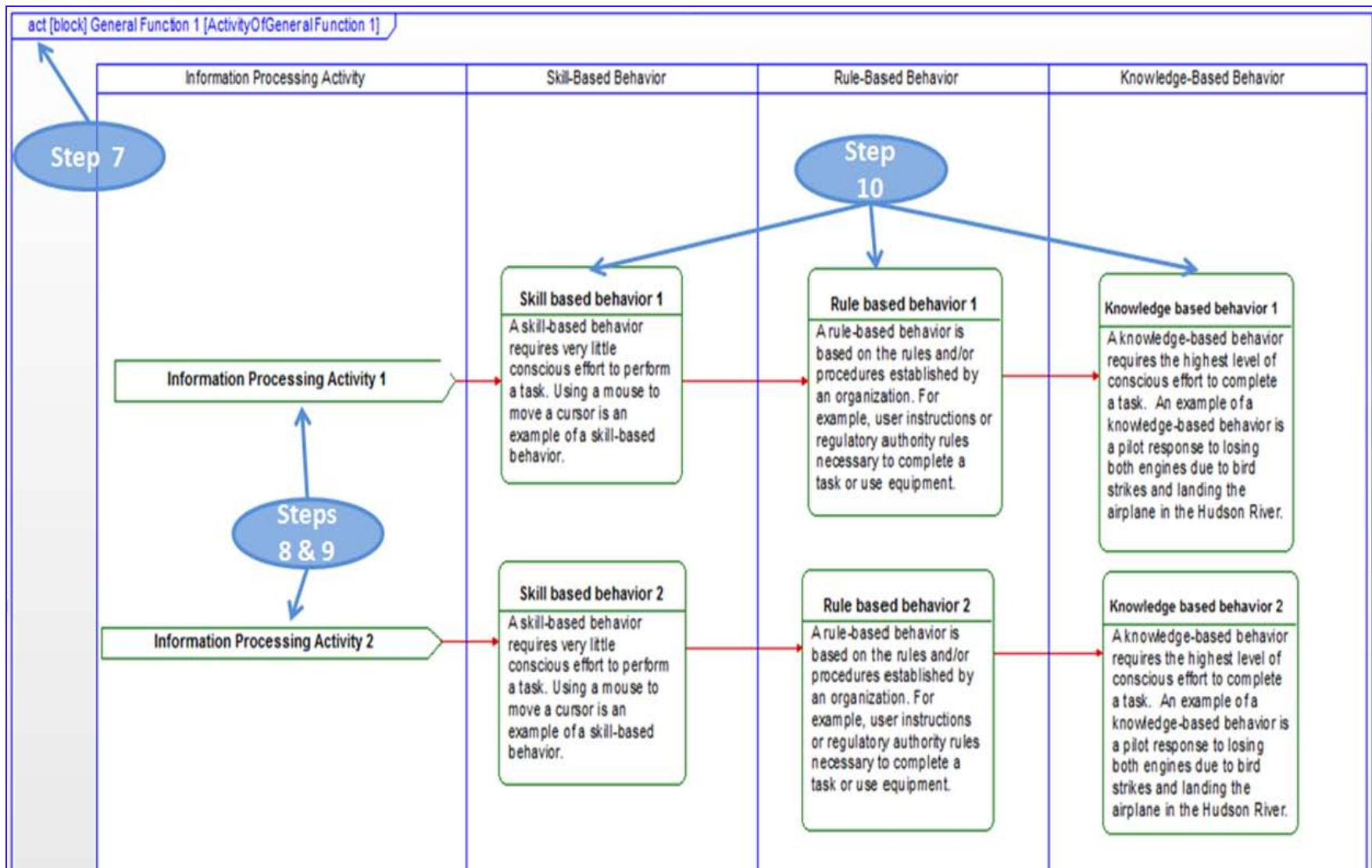


Figure 64: Process Flow Chart Steps for Building a Skill, Rule, and Knowledge Inventory

Each action block on Figure 62 contains more detail on how to complete the process step associated with that block. The detailed process steps are located in the description tab of each action block. The following detailed process steps are contained in each action block on for constructing a Skill, Rule and Knowledge Inventory:

1. Insert Internal Block Diagrams (IBD) in the Worker Competencies Analysis block of the CWA structure for each General Function in the Abstraction Hierarchy.
 - a. Move cursor over Worker Competencies Analysis block.
 - b. Right click mouse.
 - c. Select Add New --> Internal Block Diagram.
2. Use block diagram to represent general function in the Worker Competencies Analysis IBD.
 - a. Select the Block icon on the right-side menu.
 - b. Move cursor over to the diagram frame area and left-click on mouse.
 - c. Name and describe the General Function.
3. Use block diagram to represent control task below the general function block.
 - a. Select the Block icon on the right-side menu.
 - b. Move cursor over to the diagram frame area and left-click on mouse.
 - c. Name and describe the Control Task.
 - d. Repeat step 3 for each additional Control Task.
4. Use block diagram to represent strategies below the control task.

- a. Select the Block icon on the right-side menu.
 - b. Move cursor over to the diagram frame area and left-click on mouse.
 - c. Name and describe the strategy
 - d. Repeat step 4 for each additional strategy.
5. Associate general function block with control task block in a dependency relationship.
- a. Select the Dependency icon on the right-side menu.
 - b. Move cursor over to the General Function block.
 - c. Left-click mouse.
 - d. Move cursor over to the Control Task block.
 - e. Left-click mouse.
 - f. Repeat step for each additional Control Task.
6. Associate the strategies block to the control task block in a flow relationship.
- a. Select the Flow icon on the right-side menu.
 - b. Move cursor over to the strategy block.
 - c. Left-click mouse.
 - d. Move cursor over to the Control Task block.
 - e. Left-click mouse.
 - f. Repeat step for each additional strategy block.

7. Add new Activity diagram (ACT) for each General Function block.
 - a. Move cursor over General Function block.
 - b. Right-click on mouse.
 - c. Select Add New --> Activity.
8. Use results of ConTA and SA phase to construct Swimlanes diagram within the ACT diagram.
 - a. Select Swimlane Frame icon on the right-side menu.
 - b. Move cursor over to the diagram frame area and left-click on mouse.
 - c. Extend Swimlane Frame.
 - d. Select Swimlane icon on the right-side menu.
 - e. Move cursor over to the Swimlane Frame area and left-click on mouse. This step divides the lane into two lanes.
 - f. Repeat step “e” two more times.
 - g. Label each swimlane, from left to right with “Information Processing Activity”, “Skill-Based Behavior”, “Rule-Based Behavior”, and “Knowledge-Based Behavior”.
9. Use send action diagrams to represent information processing activities.
 - a. Select the Send Action icon on the right-side menu.
 - b. Move cursor over to the diagram frame area and left-click on mouse.
 - c. Name and describe the Send Action diagram.

- d. Name and description come from the information processing activities in the Control Task and Strategies Analysis phases.
 - e. Repeat step for each additional information processing activity.
10. Use action diagrams to represent behavioral tasks.
- a. Select the Action icon on the right-side menu.
 - b. Move cursor over to the diagram frame area and left-click on mouse.
 - c. Name and describe the skill, rule, or knowledge based behavior.
 - d. Repeat step 10 for each skill, rule, or knowledge based behavior.
11. Repeat steps 1–10 for each additional general function.

4.5 CWA Cognitive Factors Team

The Cognitive Factors Team section of the tutorial will provide systems engineers with a description of the educational background and experience members that a cognitive factors team should possess in order to employ CWA and other human factor assessment methods. This section will help systems engineers select the appropriate personnel to conduct a CWA. Additionally, the UC section will inform and support a system engineer's ability to coordinate the effort of the cognitive factors team. The UC section is composed of an actor, use cases, association relationships, and generalization relationships. The actor diagram represents the cognitive factors team. A cognitive factors team is a group of experts that study the problem solving, decision making and information processing activities of a human being interacting with a system. The use case diagrams represent the educational background and experience of the

cognitive factors team. The educational background describes the knowledge and skills a cognitive factors team member should possess. Member should have at least a bachelor's degree in one or more of the recommended fields of study. The experience use cases describe the knowledge and skill the cognitive factors members gained through involvement in cognitive factors domain. Figure 65 illustrates a use case diagram of the education and experience requirements a member of a cognitive factors team should have. Table 14 and Table 15 show the contents of the description section of each educational background and experience use case.

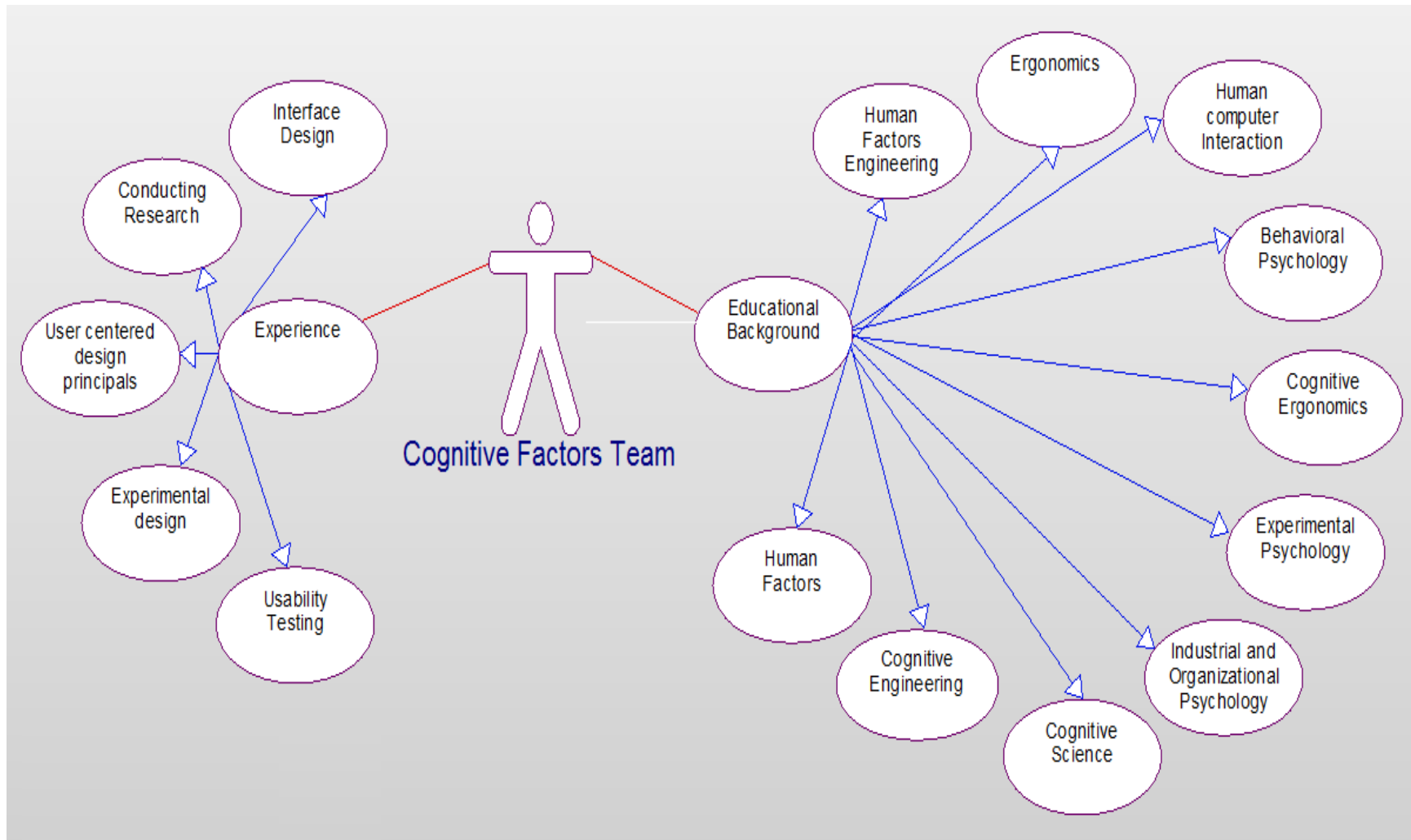


Figure 65: A Use Case Diagram of the Education and Experience Requirements of a Cognitive Factors Team

Table 14: Cognitive Factors Team Educational Background Description

Educational Background	Description
Human Factors Engineering	Human Factors Engineering is the application of knowledge about human beings' physical and cognitive strengths and weaknesses to the design of systems, processes, and work environments. The objective of Human Factors Engineering is to improve human and system performance, improve ease of use, and increase user satisfaction (Wickens, Gordon, & Liu, 2004).
Human-Computer Interaction	Human-computer interaction is the study of the interactions between human users and computers. Human-computer interaction focuses on the human interaction with the computer interface.
Behavioral Psychology	Behavioral psychology is the study of how human behaviors are acquired by interaction with the environment (Skinner, 1984).
Experimental Psychology	Experimental psychology is an area of psychology that utilizes scientific methods to research the cognitive processes and behavior (Khaleefa, 1999).
Industrial & Organizational Psychology	Industrial and organizational psychology is concerned with the study of workplace behavior. The objective of industrial and organizational psychologists is to increase workplace productivity, employee selection and training programs, and system testing (Ones & Viswesvaran, 2003).

Cognitive Science/Psychology	Cognitive science is the scientific study of how human perception, language, and reasoning of information are represented and transformed (Thagard, 2004).
Cognitive engineering	Cognitive engineering is a field of study focused on user-centered design that promotes effective human system interaction.
Cognitive ergonomics	Cognitive ergonomics focuses on analyzing human cognitive processes such as decision making and planning. Cognitive ergonomic professionals develop training programs and information technology systems that support cognitive tasks. This helps to improve human performance of cognitive tasks. For example, designing of a software interface or an airplane cockpit (Vicente, 1999).
Ergonomics	Ergonomics is the study of designing equipment and devices that fit the human body (i.e., body movements and cognitive abilities). Ergonomist apply theories, principles, and methods to design in order to optimize human well-being and overall system performance (Stanton, Hedge, Hendrick, Salas, & Brookhuis, 2004).
Human factors	Human factors is a multidisciplinary field incorporating contributions from psychology, engineering, industrial design, statistics, operations research and anthropometry. The study of human factor focuses on the physical or cognitive property of an individual or group when interacting with a system (Stanton, Salmon, Walker, Baber, & Jenkins, 2005).

Table 15: Cognitive Factors Team Experience Description

Experience Type	Description
Interface Design	Designing cognitively and/or perceptually-based interfaces.
Conducting Research	Conducting research to develop methods of understanding factors affecting human performance.
User Centered Design Principals	An applied knowledge in a variety of human system integration tools and user centered design principals.
Experimental design	Familiar with experimental design, data collection, cognitive walkthroughs and analysis.
Usability Testing	Human factors engineering experience with system interface design and usability testing to determine and assess total system performance.

4.6 CWA Method Example for the Terminal Radar Approach Control System

4.6.1 Example CWA method introduction

To demonstrate the Cognitive Work Analysis (CWA) method, an example was created within the tutorial. The example is adapted from Human Factors and Ergonomics Society Annual Meeting Proceedings. It is titled “A five-phase CWA for air traffic control: Example from a Terminal Radar Approach Control (TRACON) Microworld” (St-Cyr & Kilgore, 2008). TRACON is a software program that simulates an air traffic control environment on a personal computer. TRACON provides a training environment for air traffic controllers to direct aircraft during the departure, descent, and approach phases of flight. Air traffic controllers direct the movement of aircraft by monitoring a radar screen and maintaining voice contact with pilots. The example shows what the CWA framework should look like when applied to an actual system. Figure 66 illustrates the introduction to the TRACON example. Background information about the system’s capabilities, resources, and users is written in this screen of the tutorial.

An example of a Cognitive Work Analysis Framework constructed using System Modeling Language.

Terminal Radar Approach Control (TRACON) is a software program that simulates an air traffic control environment on a personal computer. TRACON provides a training environment for air traffic controllers to direct aircraft during the departure, descent and approach phases of flight. Air traffic controllers direct the movement of aircraft by monitoring a radar screen and maintaining voice contact with pilots. This example is adapted from the book Applications of CWA edited by Ann M. Bisantz and Catherine M. Burns and A five-phase CWA for air traffic control: Example from a TRACON microworld paper presented at the Human Factors and Ergonomics Society Annual Meeting Proceedings in 2008.

TRACON consists of an airspace surrounding a major airport and its regional satellite airports (St-Cyr & Kilgore, 2008). The simulated environment contains radars, aircraft, airports, runways, radio beacons, instrument landing systems, and other navigation aids. The simulator trains user to track many aircraft simultaneously in a dynamic environment. Aircraft must be separated by 1,000 vertical feet and three miles horizontal separation. An alarm sounds if aircraft separation is not

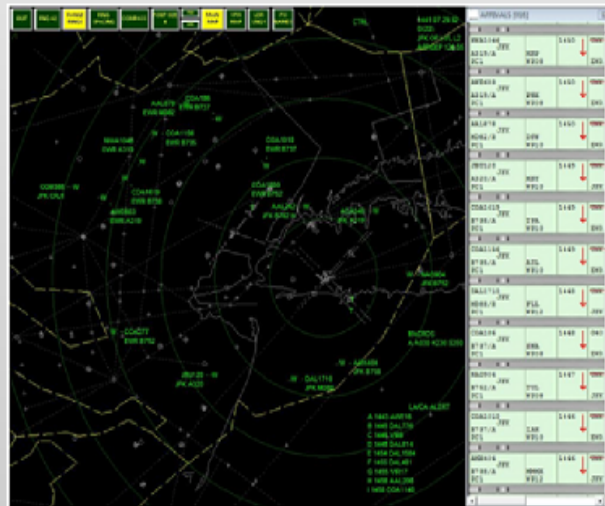


Figure 11: Interface of Terminal Radar Approach Control (TRACON) Air Traffic Control Environment

[Previous](#) | [Next](#) | [Outline](#)

4.6.2 Example of a CWA Structure

The first step in constructing the TRACON CWA framework is to build the CWA structure. The CWA structure provides a way to organize the analysis. The CWA structure is created in a Block Definition Diagram(BDD) using blocks to represent the five phase of CWA. In addition, flow arrows between the block diagrams show the exchange of information between each phase. Also, each block includes a description of the attributes of each CWA phase as it relates to the TRACON system and the Air Traffic Controller (ATC) who uses the system. Within each of the blocks is an Internal Block Diagram (IBD). More detail about each of the CWA phases is decomposed in the IBD. Figure 67 illustrates a model of the CWA structure composed for the TRACON example.

An overview of Cognitive Work Analysis Structure for TRACON system.

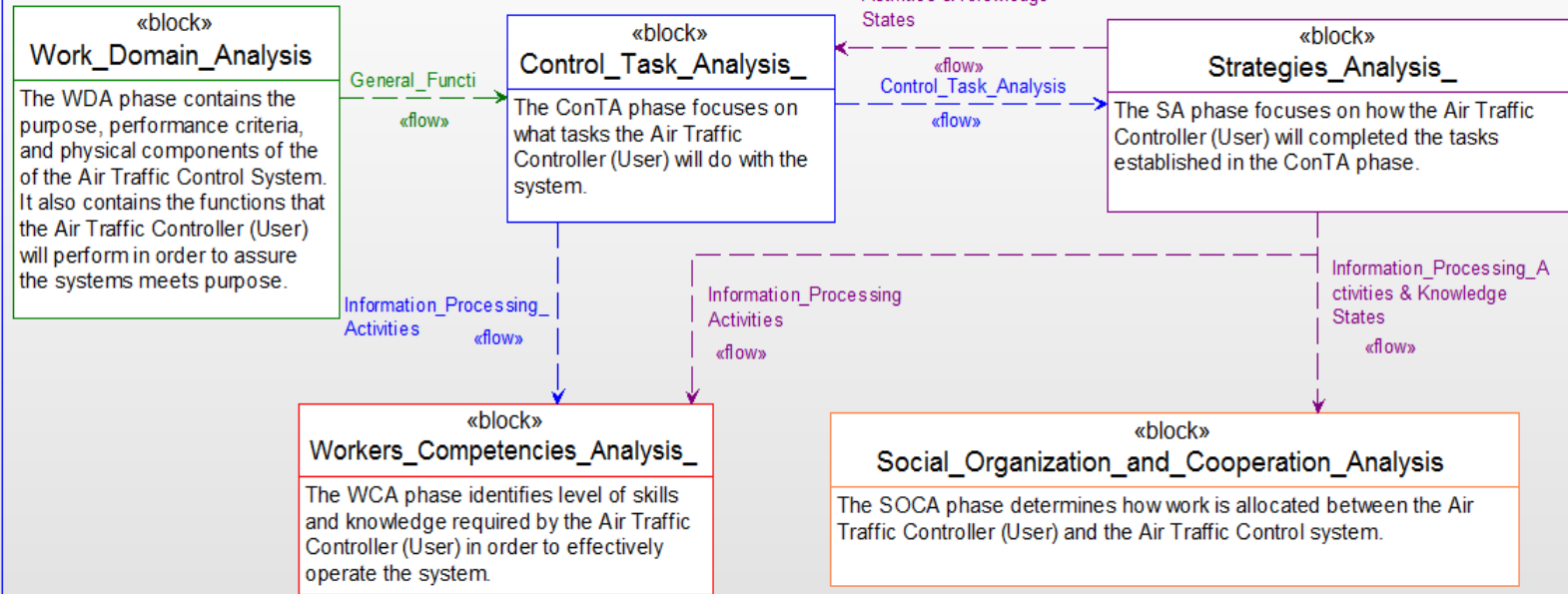


Figure 67: Cognitive Work Analysis Structure Model for TRACON System

4.6.3 Example of a CWA Abstraction Hierarchy

The next step is to construct the Abstraction Hierarchy (AH), which is the modeling technique used in the first CWA phase. The AH is composed of blocks, constraint properties, and part diagrams. Figure 68 shows an AH model of the TRACON system. The model is an Abstraction Hierarchy of the TRACON domain. The highest level is the function purpose of the TRACON system. In this case, the TRACON was developed to train users to route aircraft safely and efficiently. The second level from the top identifies the constraints and performance parameters of the TRACON system. The constraints include: aircraft responsibility, pilot situation awareness, scheduling demands, performance abilities of individual aircraft, passenger comfort parameters, and maintenance of a field of safe travel. The third level from the top contains the general functions the user will perform when interacting with the system. The general functions include: negotiating with neighboring regional ATC centers, establishing and updating aircraft flight paths, locomotion of aircraft through the sector, and the transition between airspace. The fourth level from the top holds the components of the TRACON work domain. The components of the TRACON work domain include systems that the TRACON system interact with: for example, aircraft, external regional ATC centers, and landings. Finally, the fifth level contains the sub-components of the TRACON work domain components.

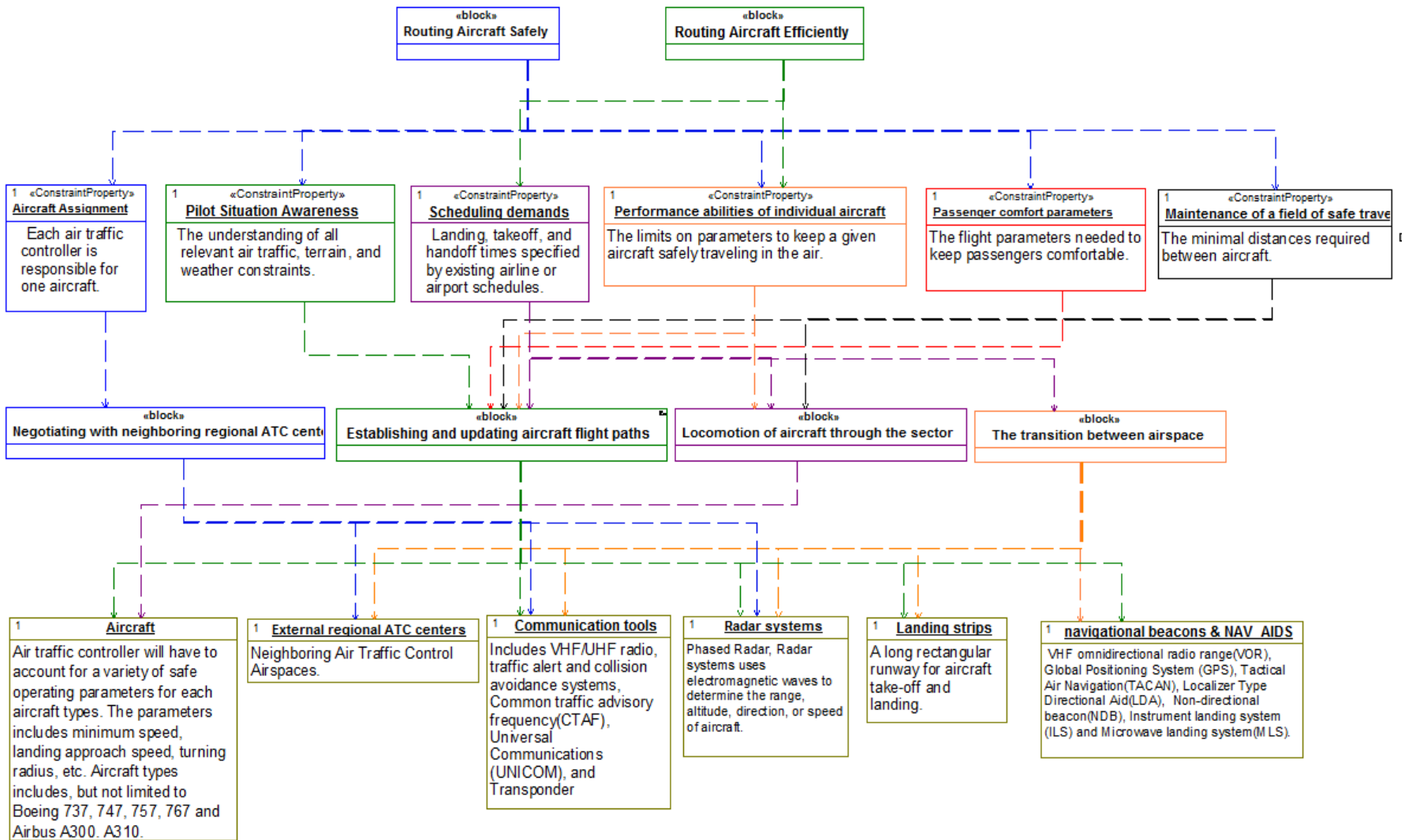


Figure 68: An Abstraction Hierarchy Model of the TRACON System

4.6.4 Example of a CWA Decision Ladder

The next step in conducting CWA is to identify the tasks that the user will do with the system. This aspect is covered in the second phase of CWA. The first step in constructing the second phase of CWA is to build a Control Task Analysis (ConTA) structure. This structure provides the capability to organize and manage all general functions and user control tasks. This capability becomes more critical for a larger, more complex analysis and is one of the major advantages of using SysML.

Figure 69 shows an example of a ConTA structure. The ConTA structure is built in an IBD of the Control Task Analysis block on Figure 67. The input to the ConTA phase is the general functions from the AH. Each general function will have a ConTA structure created in an IBD of the Control Task Analysis block on Figure 67. In this case, there would be four IBDs for the TRACON system (i.e., negotiating with neighboring regional ATC centers, establishing and updating aircraft flight paths, locomotion of aircraft through the sector, and the transition between airspace). One general function will be represented by a block diagram in each IBD. The example will focus on the “establishing and updating aircraft flight paths” general function as illustrated by Figure 69. The “establishing and updating aircraft flight paths” general function is further decomposed into three control tasks. The control tasks are approaching, receiving, and rerouting. The definition of each is located in the description tab of each block. To access the definitions, double-click the block diagram for the control task. The control tasks are represent by block diagrams. The general function is linked to the control

tasks by a dependency relationship. Completion of the control task satisfies the general function. The general function can only be accomplished if the control tasks are successfully completed.

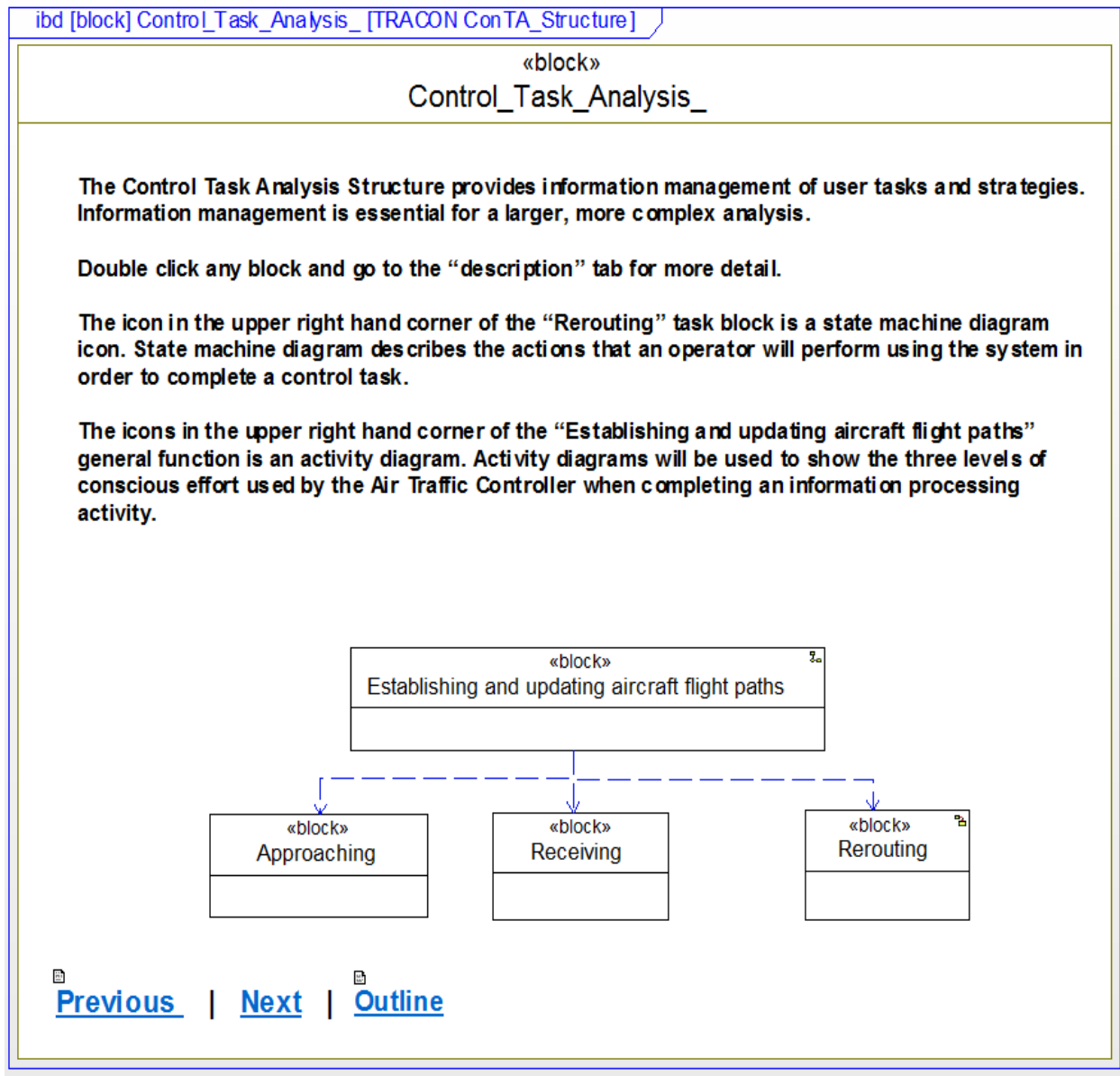


Figure 69: An Example of a ConTA Structure for the TRACON System.

The second step in the ConTA phase is to construct a Decision Ladder (DL). A State Machine (STM) diagram is used to construct the DL. The STM is attached to the “rerouting” control task block on Figure 69. The STM is the icon located on the upper right side of the rerouting block. Figure 70 shows an example of a DL for the TRACON user. Send action and state diagrams are used to represent information processing activities and knowledge states, respectively. The green highlighted send action and state diagrams show a potential expert pathway through the DL. Other DLs for other potential pathways will have to be constructed within the same STM diagram for the “rerouting” control task. The example only includes one potential pathway. Each control task will have a STM diagram.

TRACON Decision Ladder(DL) of the rerouting task.

The green highlighted Send Action and State diagrams are one potential expert pathway through the DL.

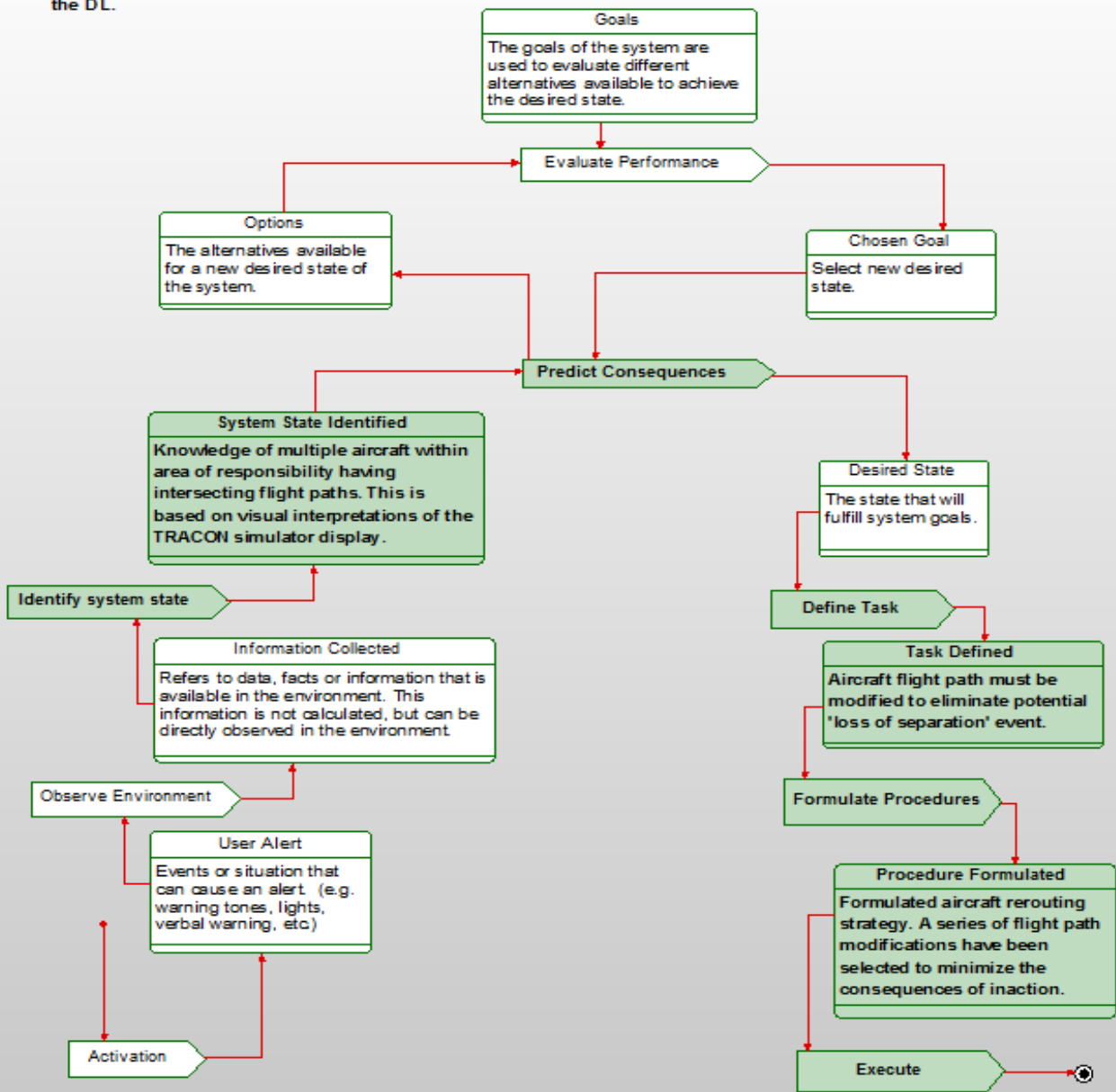


Figure 70: An Example of a Decision Ladder for the TRACON User

4.6.5 Example of a CWA Information Flow Maps

The goal of the third CWA phase is to identify the course of action the user will take to complete the control tasks established in the second CWA phase. The first step in the third phase is to construct a Strategies Analysis (SA) structure. This structure provides the capability to organize and manage all general functions, user control tasks, and user strategies. This capability becomes more critical in a larger, more complex analysis and is one of the major advantages of using SysML. Figure 71 shows an example of a SA structure for the TRACON system. The SA structure is built in an IBD of the Strategies Analysis block on the CWA structure (i.e. Figure 67). The input to the SA phase is the control tasks from the ConTA phase. The SA structure is built with the same diagrams as the ConTA structure, with the exception of the strategies that are added to this phase. Each general function will have an SA structure created in an IBD of the Strategies Analysis block on Figure 67. For the purpose of the TRACON example, only one general function will be spotlighted. The “establishing and updating aircraft flight paths” general function is further decomposed into three control tasks. The “rerouting” control task is further decomposed into three courses of action. The strategies include: hold one aircraft, reroute one aircraft, and tweak one aircraft. Definitions of each strategies are located in the description tab of each block. To access the definitions, double-click the block diagram for the strategy. The strategies are represented by block diagrams. The strategies are linked to the control tasks by an information flow relationship. The strategies provide the course of action required to

complete a control task which satisfies the general function. Each control task will have strategies linked to it. So the framework will increase exponentially as more general functions, control tasks, and strategies are added. The potential expansion of the CWA framework emphasizes the importance of the organizational features of SysML.

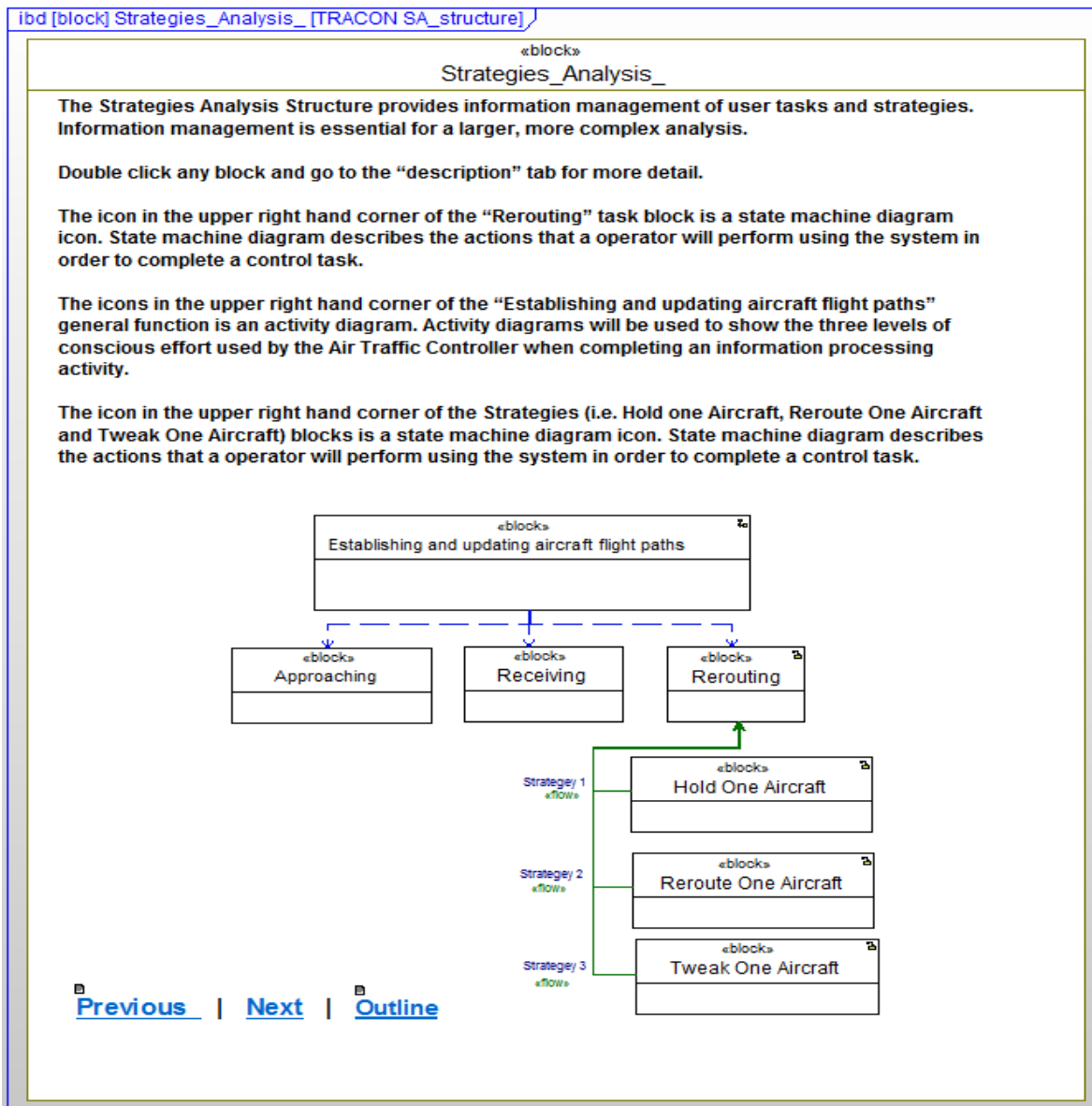


Figure 71: A Strategies Analysis Structure for the TRACON System

The second step in the third CWA phase is to construct an Information Flow Map (IFM). A State Machine (STM) diagram is used to construct the IFM. An STM is attached to each strategy on Figure 71. The STM is the icon located on the upper right side of each strategy block. Figure 72, Figure 73, and Figure 74 are examples of IFMs for the TRACON user.

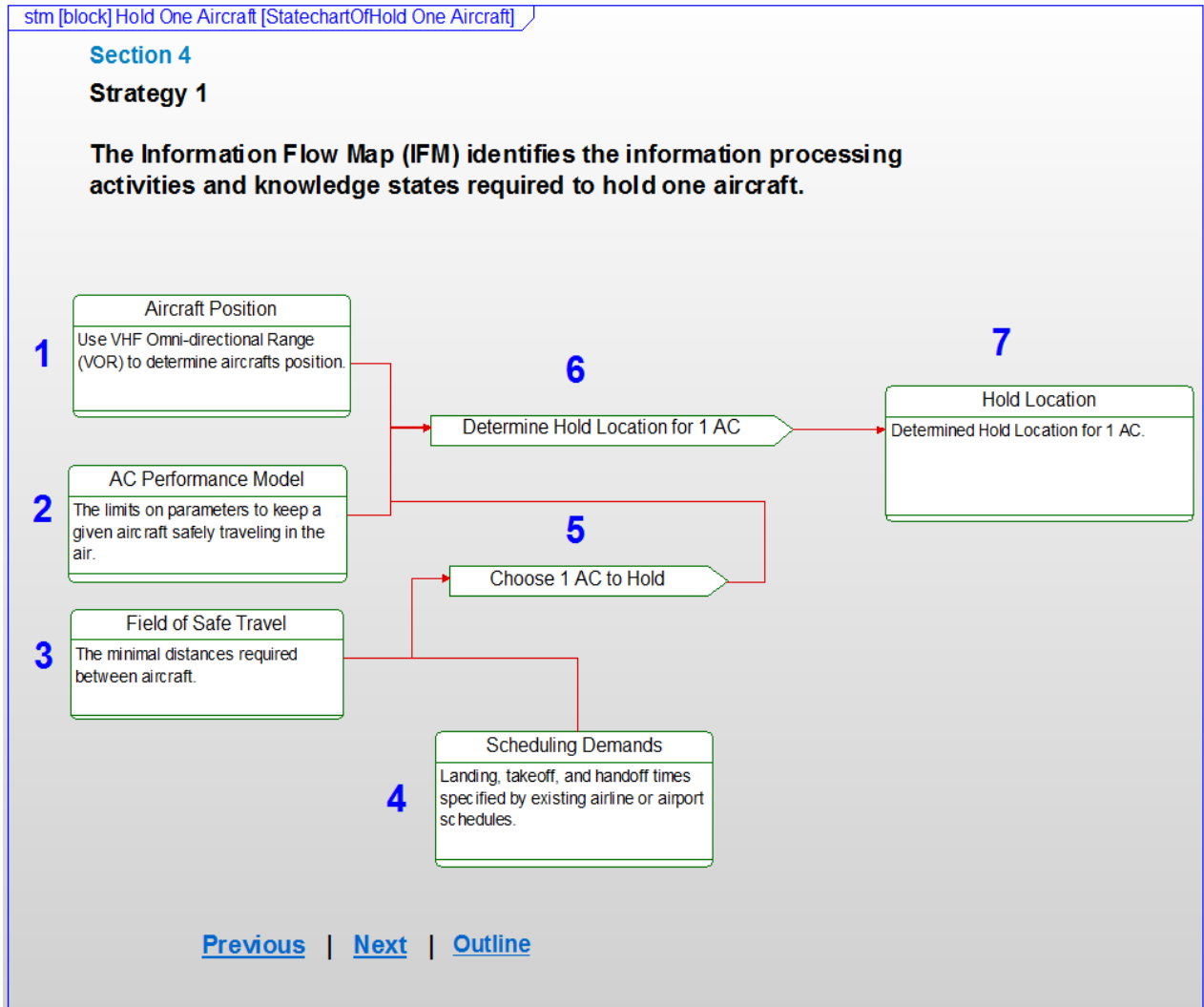


Figure 72: An Information Flow Map Model of the User's Strategy to Hold One Aircraft

Send action and state diagrams are used to represent information processing activities and knowledge states, respectively. The send action diagrams are the arrow-shaped diagrams and the state diagrams are the rectangle-shaped diagrams.

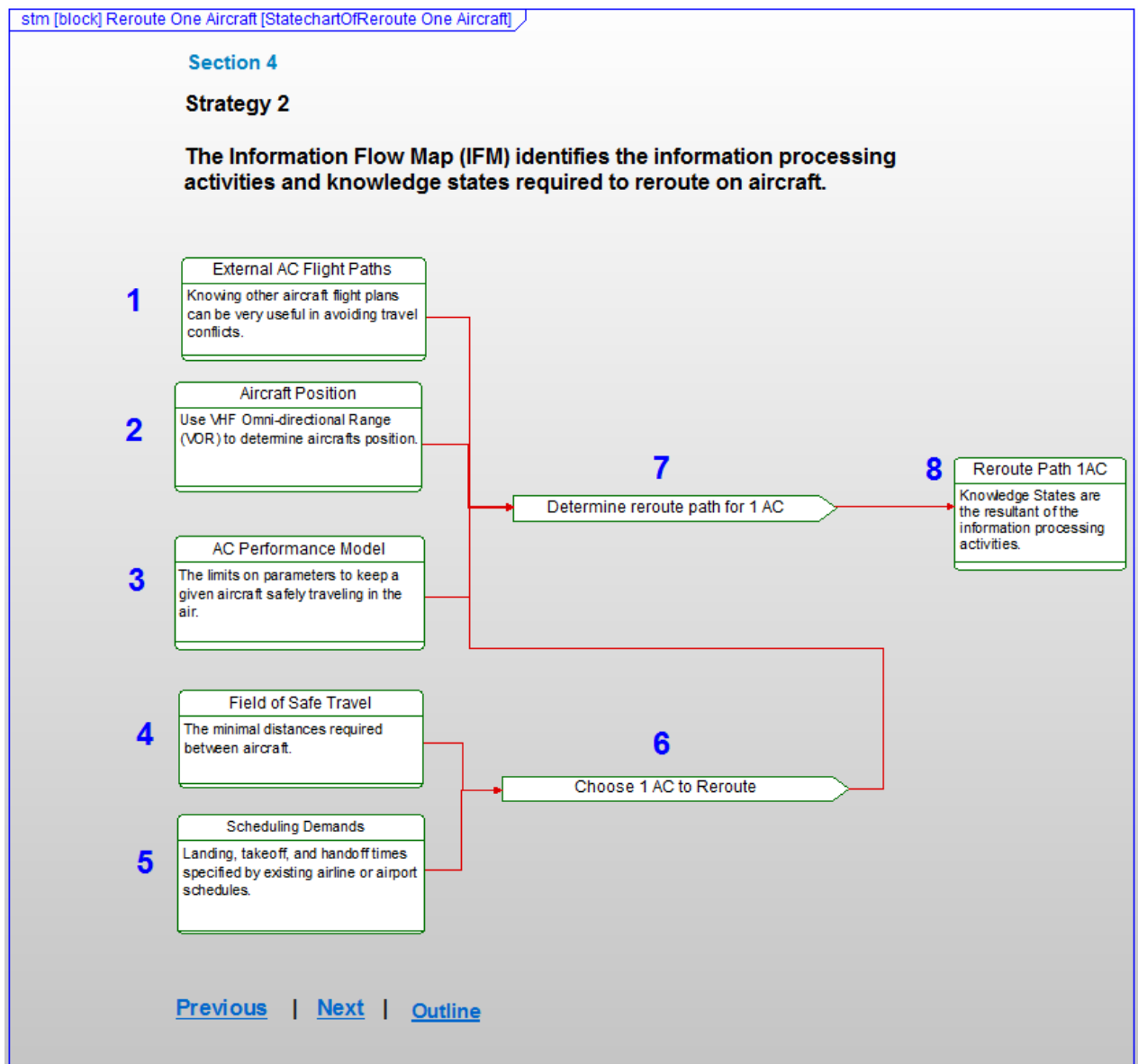


Figure 73: An Information Flow Map Model of the User's Strategy to Reroute One Aircraft

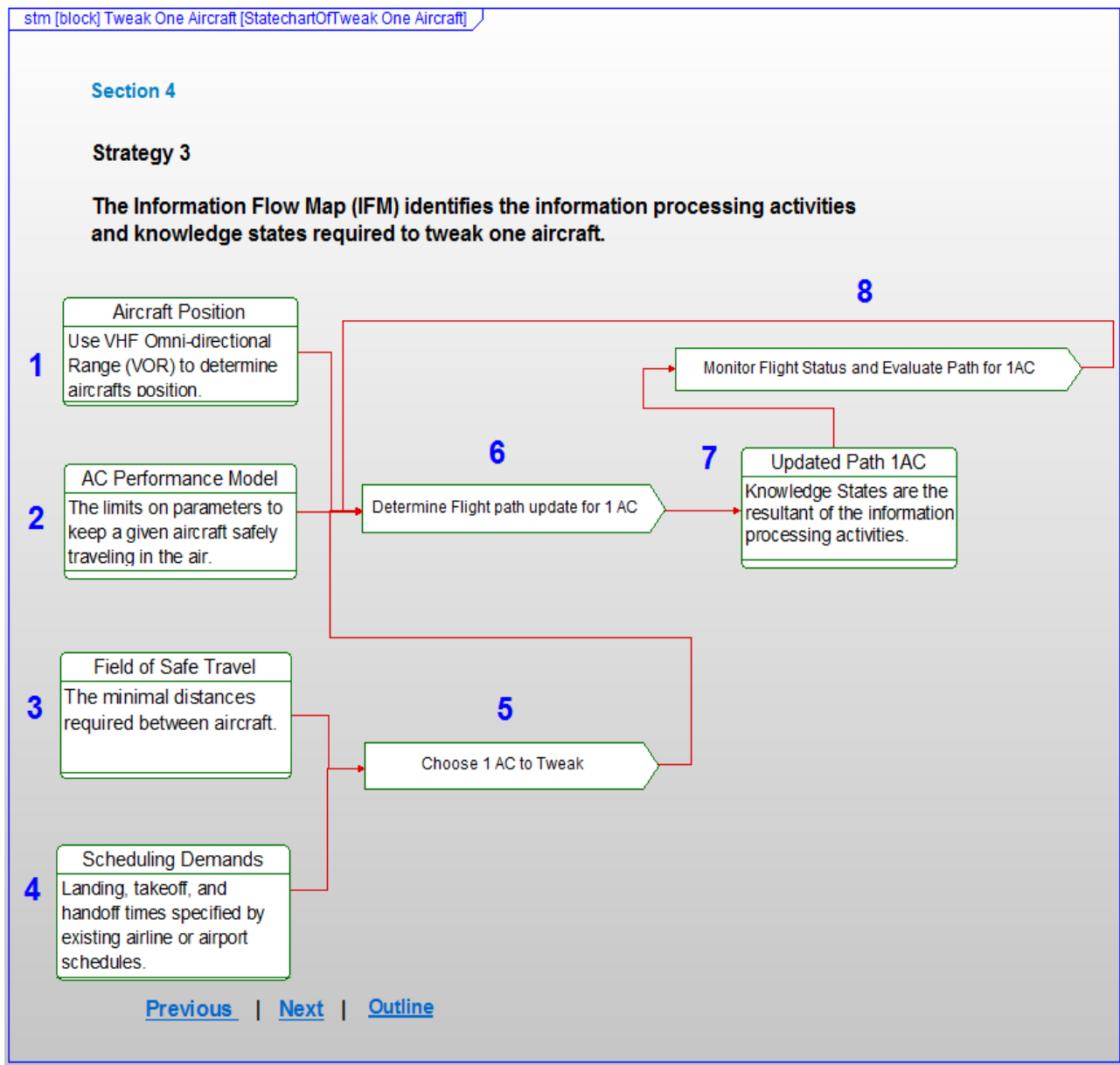


Figure 74: An Information Flow Map Model of the User's Strategy to Tweak One Aircraft

4.6.6 Example of a CWA Use Case

The goal of the Social Organization and Cooperation Analysis (SOCA) phase is to allocate task responsibilities between the actors which interact with the system. The

first step in the SOCA phase is to construct a SOCA structure. This structure provides the capability to organize and manage all general functions, user control tasks, and user strategies. Figure 75 shows an example of a SOCA structure for the TRACON system. The SOCA structure is built in an IBD of the Social Organization and Cooperation Analysis block on the CWA structure (i.e., Figure 67). The input to the SOCA phase is the strategies from the SA phase. The SOCA structure is built with the same diagrams as the SA structure. Each general function will have an SOCA structure created in an IBD of the Social Organization and Cooperation Analysis block on the CWA structure (i.e., Figure 67). For the purpose of the TRACON example, only one general function is spotlighted. The “establishing and updating aircraft flight paths” general function is further decomposed into three control tasks. The “rerouting” control task is further decomposed into three strategies. The strategies are represented by block diagrams. The strategies are linked to the control tasks by information flow relationships. The strategies provide the course of action required to complete a control task which satisfies the general function.

«block»
Social_Organization_and_Cooperation_Analysis

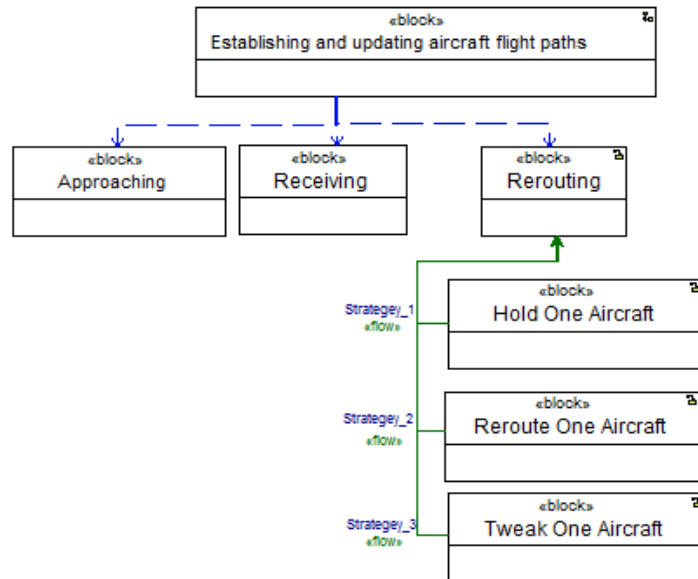
The Social Organization and Cooperation Analysis Structure provides information management of user tasks and strategies. Information management is essential for a larger, more complex analysis.

Double click any block and go to the “description” tab for more detail.

The icon in the upper right hand corner of the “Rerouting” task block is a state machine diagram icon. State machine diagram describes the actions that a operator will perform using the system in order to complete a control task.

The icons in the upper right hand corner of the “Establishing and updating aircraft flight paths” general function is an activity diagram. Activity diagrams will be used to show the three levels of conscious effort used by the Air Traffic Controller when completing an information processing activity.

The icon in the upper right hand corner of the Strategies (i.e. Hold one Aircraft, Reroute One Aircraft and Tweak One Aircraft) blocks is a state machine diagram icon. State machine diagram describes the actions that a operator will perform using the system in order to complete a control task.



[Previous](#) | [Next](#) | [Outline](#)

Figure 75: A Social Organization and Cooperation Analysis Structure for the TRACON System.

The second step in the SOCA phase is to construct a Use Case (UC) diagram. A UC is constructed with the resultant information of the SA phase. The UC diagrams consist of use cases, which are the information processing activities and knowledge states identified in the SA phase (i.e., strategies 1, 2, and 3). The UC also consists of actors, which are humans or things that have specific roles when interacting with the system. The actors in the TRACON example are the Air Traffic Controller (ATC) and the automation of the TRACON system. Finally, UC diagrams have a variety of relationships that could exist between actors, between actors and use cases, or between use cases. The relationships include: association, generalization, include, extend, and dependency. Figure 76, Figure 77, and Figure 78 are examples of UCs for the TRACON user and system.

The Use Case diagram allocates responsibilities of holding one aircraft between actors.

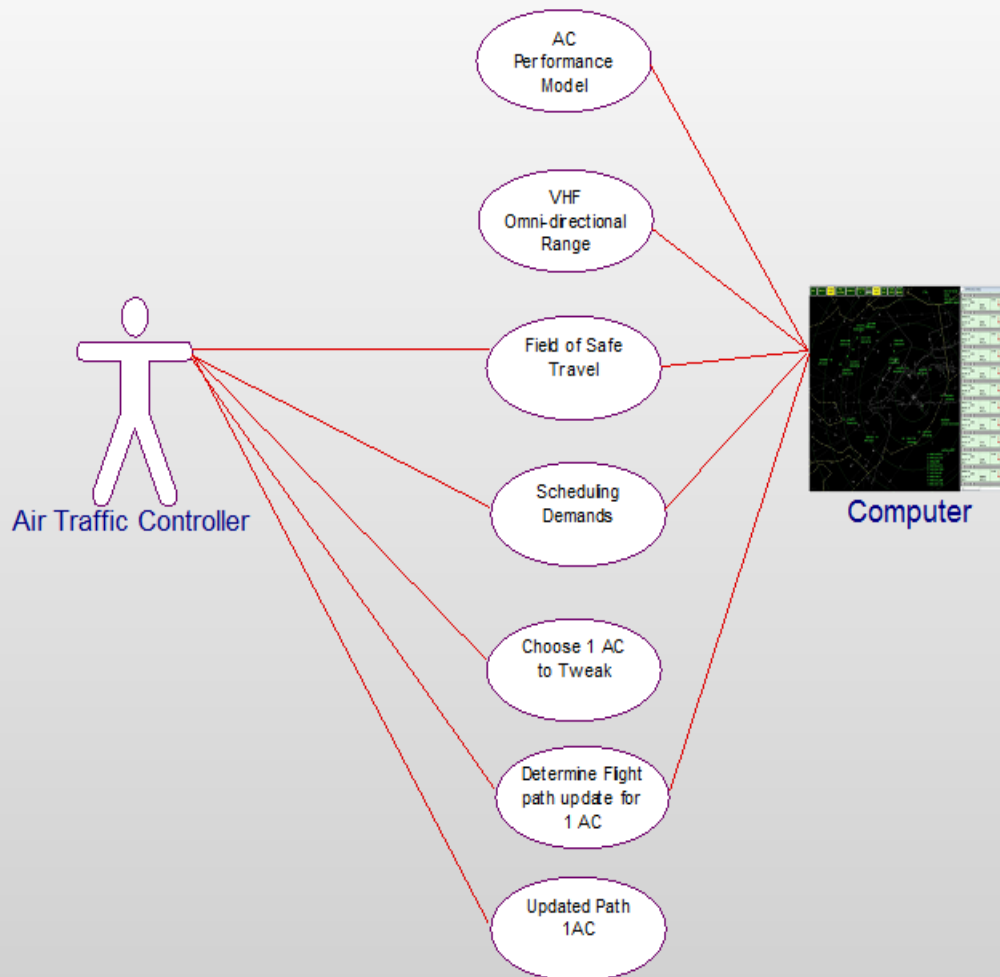


Figure 76: A Use Case Diagram Showing Allocation of Tasks Between TRACON User and the System for Holding One Aircraft.

The Use Case diagram allocates responsibilities of rerouting one aircraft between actors.

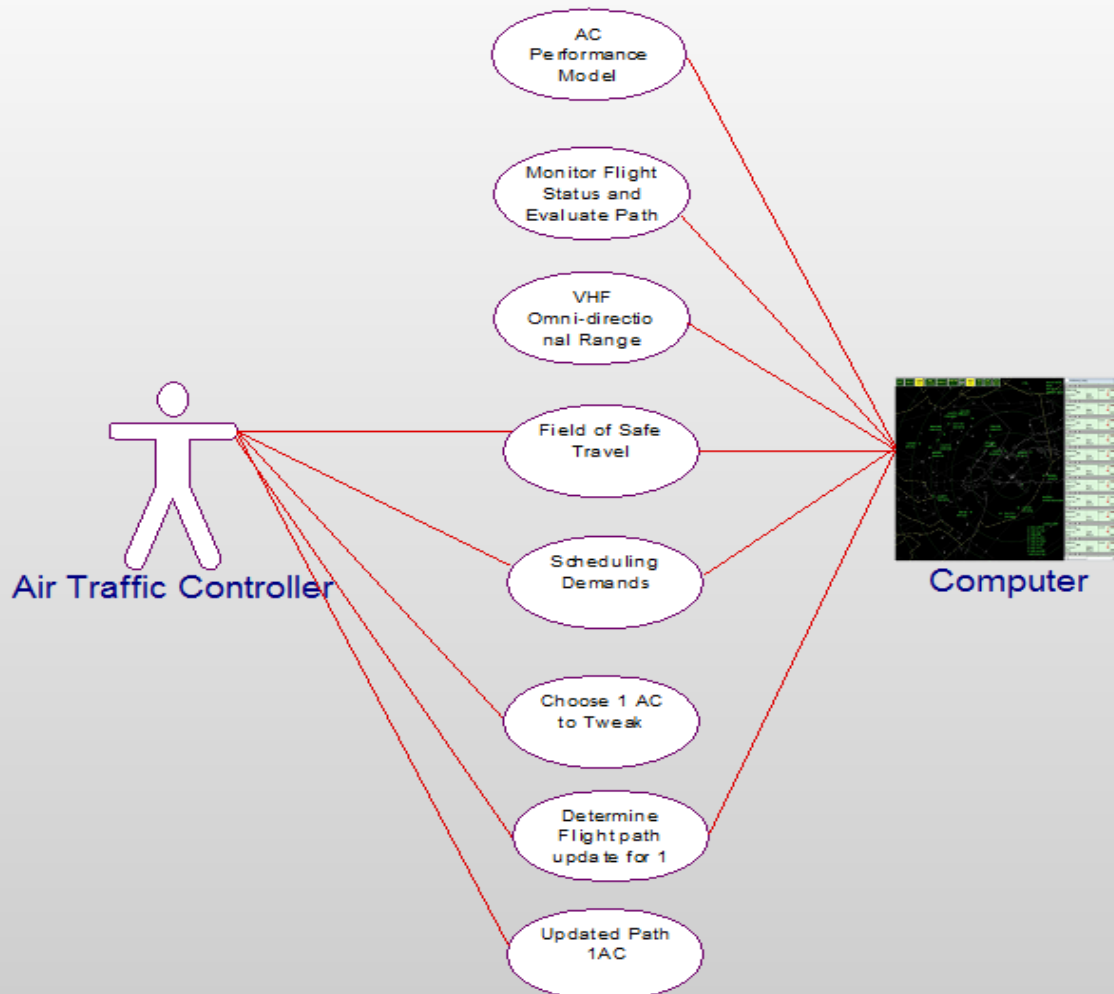


Figure 77: A Use Case Diagram Showing Allocation of Tasks Between TRACON User and the System for Rerouting One Aircraft

The Use Case diagram allocates responsibilities of tweaking one aircraft between actors.

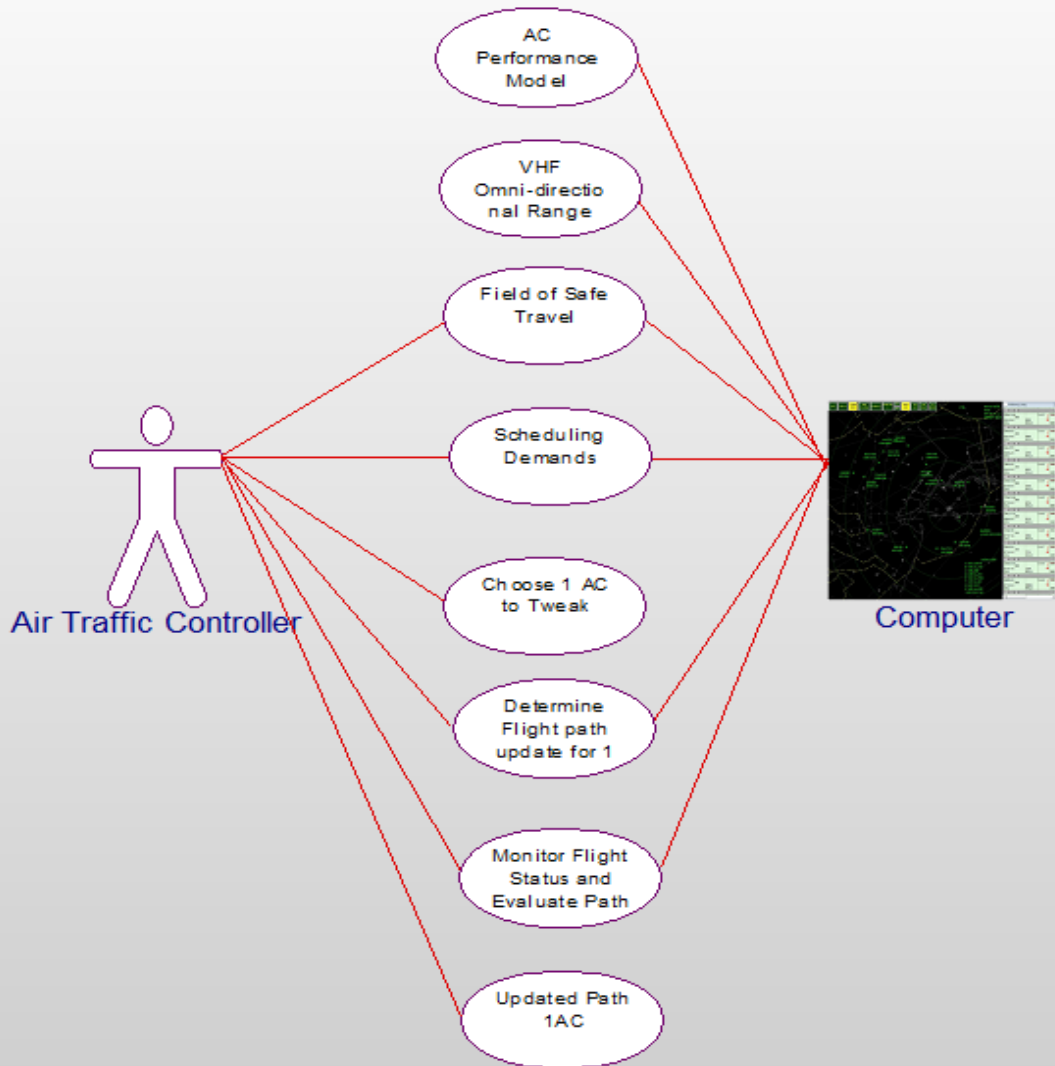


Figure 78: A Use Case Diagram Showing Allocation of Tasks Between TRACON User and the System for Tweaking One Aircraft

4.6.7 Example of a CWA Skill, Rule, and Knowledge Inventory

In the final CWA phase of the Worker Competencies Analysis (WCA) phase, the goal is to identify the knowledge and skills required by the end users to operate the system. The phase is critical to personnel selection and training. The first step in the WCA phase is to construct a WCA structure. This structure provides the capability to organize and manage all general functions, user control tasks, and user strategies. Figure 79 shows an example of a WCA structure for the TRACON system. The WCA structure is built in an IBD of the Worker Competencies Analysis block on the CWA structure. The input to the WCA phase is the information processing activities from the ConTA and SA phases. The WCA structure is built with the same diagrams as the ConTA, SA, and SOCA structures. Each general function will have a WCA structure created in an IBD of the Worker Competencies Analysis block on the CWA structure. For the purpose of the TRACON example, only one general function is spotlighted. The “establishing and updating aircraft flight paths” general function is further decomposed into three control tasks. The “rerouting” control task is further decomposed into three strategies. The strategies are represented by block diagrams. The strategies are linked to the control tasks by information flow relationships. The strategies provide the course of action required to complete a control task which satisfies the general function.

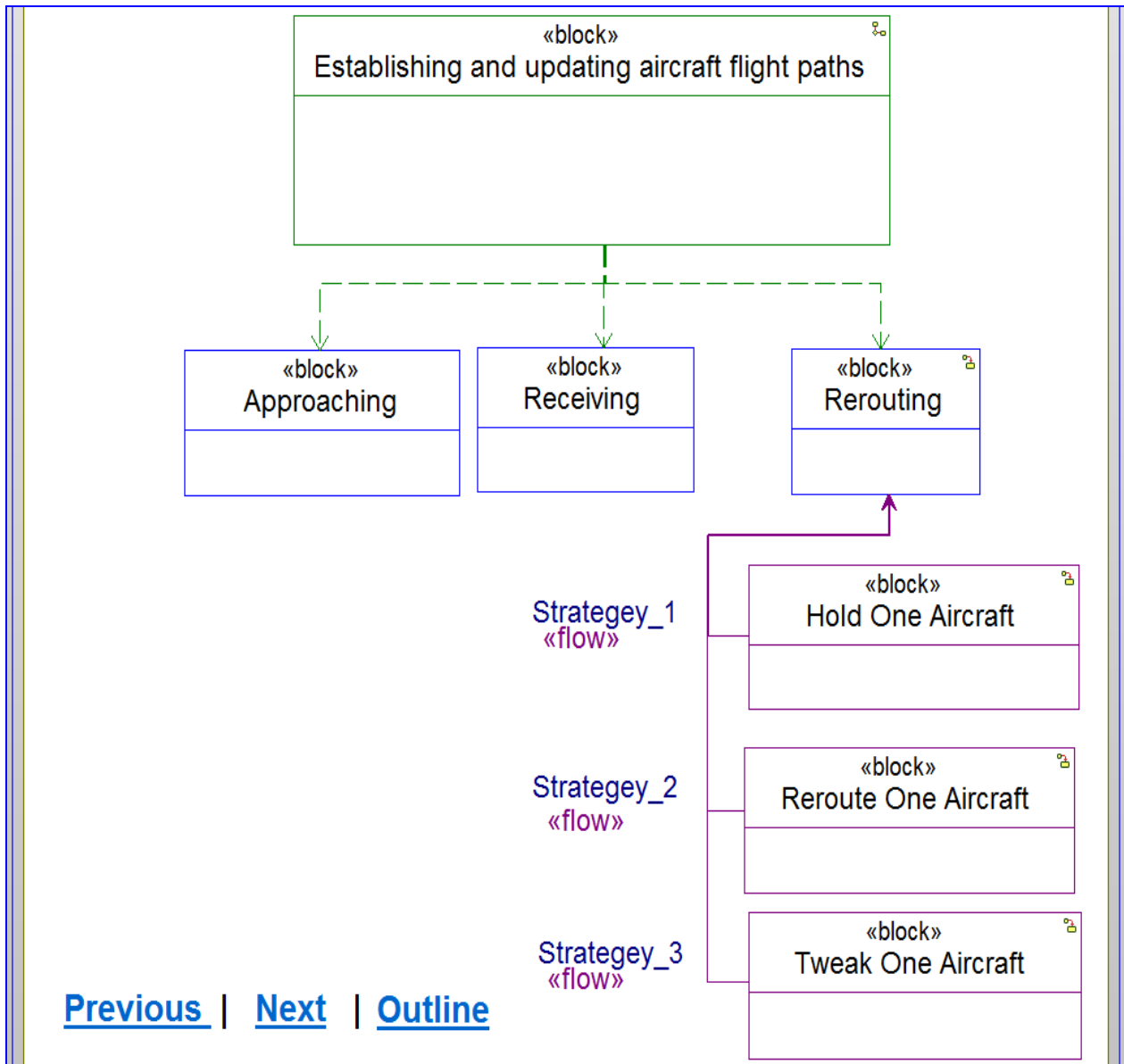


Figure 79: A Worker Competencies Analysis Structure for the TRACON System

The second step in the WCA phase is to construct an SRK inventory. An SRK inventory is constructed with the resultant information processing activities of the ConTA SA phases. The SRK inventory is created using the swimlane diagrams available in

SysML. The swimlanes are divided into information processing activities and skill-based, rule-based, and knowledge-based behavior lanes. Send Action diagrams are used to represent information processing activities, while Action diagrams are used to represent Skill-, Rule-, and Knowledge-Based Behaviors. The red lines are control flow lines. The control flow lines show the relationship between the information processing activity and the Skill-, Rule-, and Knowledge-Based Behaviors. The control flow lines show the transition from information processing activity to the three psychological processes. Each psychological process is a level of competency required by the system end-user to complete an information processing activity. Figure 80 shows two information processing activities, which are “Scan for aircraft presence in area of responsibility” and “Determine the criticality of a pending convergence” as well as the end-user’s level of competency required to complete the information processing activities.

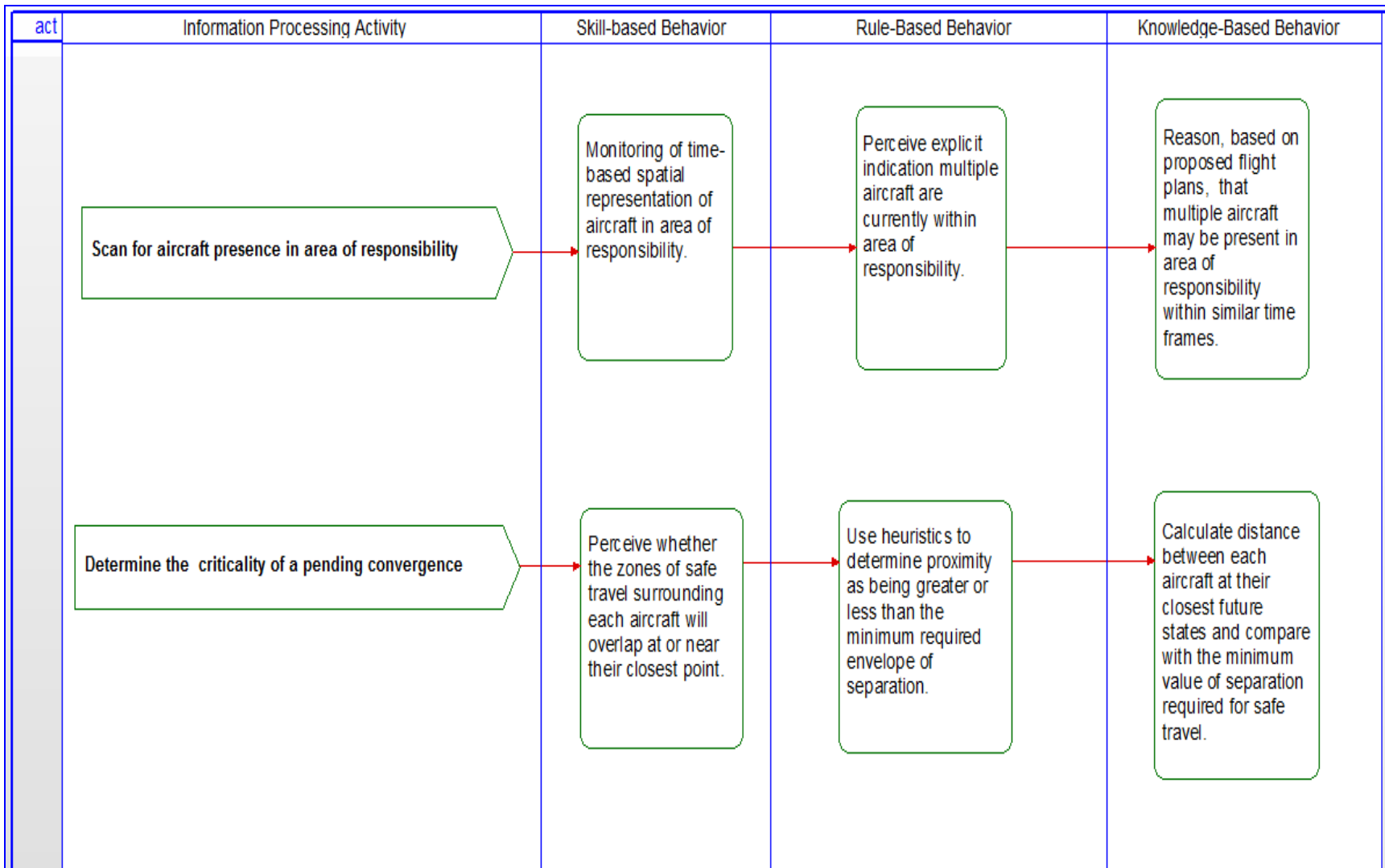


Figure 80: A Use Case Diagram Showing Allocation of Tasks

4.7 CWA Framework ATM Example

This section presents an example of how to apply the CWA framework to an Automated Teller Machine (ATM). Unlike the TRACON example, this example focuses on gathering the information to be modeled. The TRACON example focused on modeling the results of CWA and translating CWA terminology into SysML diagrams. This example focuses on performing the CWA method.

Most people in an industrialized country have used an ATM before. So most system designers using this tutorial can relate to the cognitive tasks that are performed when interacting with an ATM. The scope of this example is limited to showing how each phase of CWA is constructed in SysML. Therefore, the breadth of the analysis is limited, but the depth is thorough.

Figure 81 shows the introduction page in the tutorial for the ATM example. It describes the process of performing CWA. Before the systems engineer can start constructing CWA models with SysML, there has to be some data collection. More information about the system and the end-user is required. Knowledge elicitation methods should be implemented in order to acquire the necessary system and end-user information. There are many knowledge elicitation methods that can be used for acquiring system design requirements. The options include questionnaires, focus groups, group task analysis, case studies, etc. The CWAT only focuses on interviews, review of legacy system references, and observations of end-user performing tasks that the new system is required to carry out. Interviewing SMEs or end-users is the most

popular form of knowledge elicitation. How many interviews and who will be interviewed will vary across projects. Interviews will have to be performed for all CWA phases, and the focus of the questions will vary at each CWA phase. Subject Matter Experts (SMEs), system operators, or other system stakeholders should be actively involved in the interview process.

bdd [Project] SYSML_CWA_DRAFT_15JAN2011 [Cognitive Work Analysis Example]

Section 5

This section demonstrates the application of the SysML CWA framework methodology from begin to completion. This application of the methodology is taken from sections 2 and 3 of the SysML CWA framework tutorial.

The system we will apply the SysML CWA methodology to is an Automated Teller Machine (ATM). Most people in an industrialized country have used an ATM before. So most system designers using this tutorial can relate to the cognitive tasks that are performed when interacting with an ATM. The scope of this example is limited to showing how each phase of CWA is constructed in SysML. Therefore, the breadth of the analysis is limited, but the depth is thorough.

Before construction of the SysML CWA models can begin, we need to more information about the system we will design and the end user or users of the system. In order to acquire this information, we need to implement knowledge elicitation methods of customer, Subject Matter Experts(SMEs), system operators, or other system stakeholders is required. Interviews of SMEs or end-user are the most popular form of knowledge elicitation. Other knowledge elicitation method can be used for acquiring system design requirements. These options include questionnaires, focus groups, group task analysis, case studies, etc.

The tutorial will not provide more detail about knowledge elicitation method. There are many books that provide more complete explanation of knowledge elicitation. This tutorial focuses on interviews, review of legacy system references and observations of end user performing tasks that the new system is required to carry out. '

How many interviews and who will be interviewed will vary across projects. Interviews will have to be performed for all CWA phases and the focus of the questions will vary at each CWA phase. Examples of questions will be included in the following pages of the tutorial.

During this section of the tutorial we will switch between analysis and model building in SysML. Normally, a system designer will conduct an interview for all CWA phases before modeling the results.

[Previous](#) | [Next](#) | [Outline](#)




Figure 81: Introduction Page for the ATM Example

4.7.1 CWA Structure for ATM Example

The Cognitive Work Analysis (CWA) structure does not require any analysis to construction. The structure provides the capability to organize and manage the analysis of the different aspects of the system. This capability is critical for a larger, more complex analysis and is one of the major advantages of using SysML.

The CWA structure is created in a Block Definition Diagram (BDD) using blocks to represent the five phases of CWA. In addition, flow arrows between the block diagrams show the exchange of information between each phase. Also, each block includes a description of each CWA phase as it relates to an Automated Teller Machine (ATM) and the customers who use the system. Within each of the blocks is an Internal Block Diagram (IBD). More detail about each of the CWA phases is decomposed in the IBD. Figure 82 represents a CWA structure.

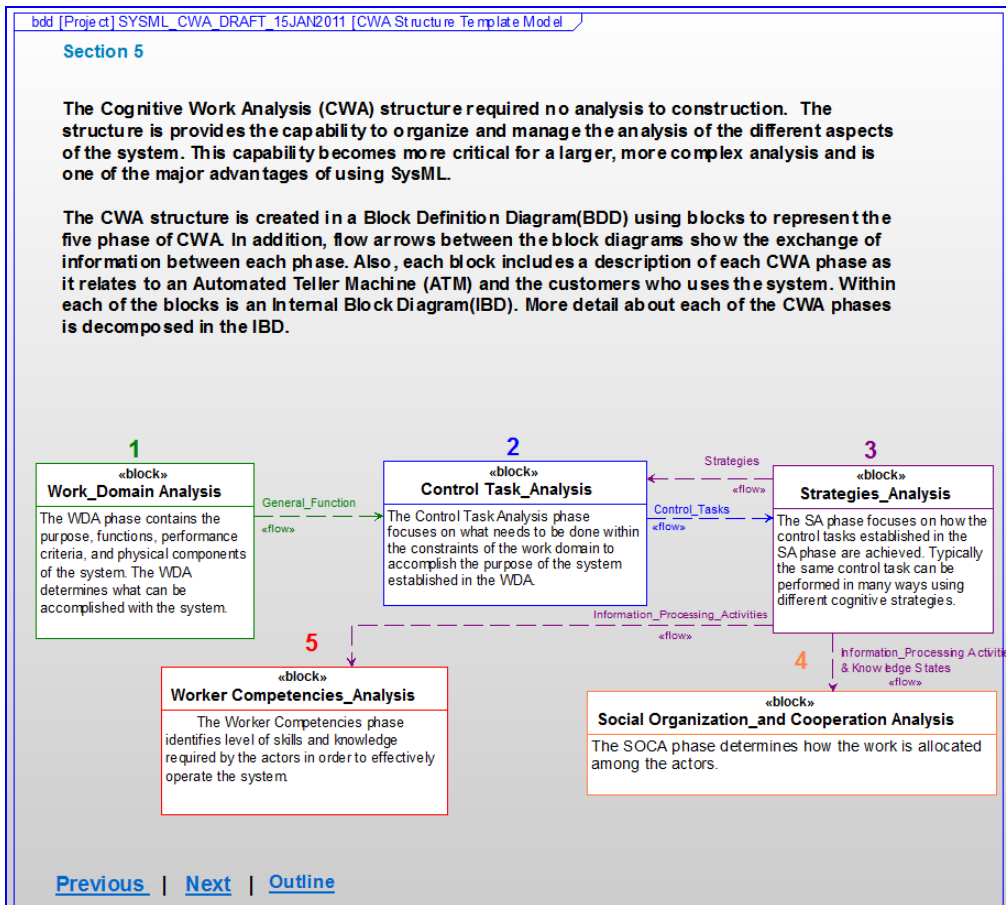


Figure 82: CWA Structure for ATM example

4.7.2 Work Domain Analysis

The first step in acquiring information about the system to be designed is to review similar legacy system documents. Getting background information and domain terminology prior to interviews is essential in establishing a knowledge base. Reviewing legacy system references creates a starting point for system designers. It helps to identify the physical equipment, the goals, the functions, and the policy constraints of the system. These documents include, but are not limited to, instructions, operating

manuals, and maintenance manuals for the system. Figure 83 shows a text-based description of the knowledge elicitation process for the Work Domain Analysis (WDA) of CWA.

The second step in acquiring information about the system and the end-user is to conduct interviews of SMEs, system operators, or other system stakeholders. The following questions are typically asked during the interview process:

- What are the main goals of the expected system?
- What might get in the way of achieving set goals?
- What do you have to do to obtain the goals?
- What resources are required to help reach the goals?
- What regulations/policies are necessary in the work domain?
- What resources or parameters are required to achieve upper level of AH?

Section 5

So far we have discussed how we elicit knowledge about our new system. Lets apply this to an example. The system we will design is an Automated Tell Machine (ATM). Most people in an industrialized country have used an ATM before. So most system designers using this tutorial can related to the cognitive tasks that are performed when interacting with an ATM. The scope of this example is limited to showing how each phase of CWA is constructed in SysML. Therefore, the breadth of the analysis is limited, but the depth is thorough. The following background information is based on the questions asked to the customer of the system we are designing.

The following is questions typically asked, but may include other questions:

- What are the main goals of the expected system?
- What might get in the way of achieving set goals?
- What do you have to do to obtain the goals?
- What resources are required to help reach goals?
- What regulations/policies are necessary in the work domain?
- What is resources or parameters are required to achieve upper level of AH?

Part of the questions asked in the WDA phase is the means/ends relationships that exist between the upper and lower levels of the Abstraction Hierarchy. Each level is connected by means-ends relationships. The means is a level below the ends. For example, the general function is the means for the abstract function. The lower levels describe the actions, components or parameters that are necessary for achieving the ends or upper levels of the AH.

The next step is to populate abstraction hierarchy. Once interviews and documentation reviews are completed, the Abstraction Hierarchy can be populate with the appropriate data for each level. SysML blocks are used to represent the data at the different levels of the Abstraction Hierarchy. Block diagrams should be use to represent the functional purpose and general function levels. Part diagrams should be used to represent the physical function and component levels. Constraint property diagrams should be used to represent the abstract function level. A detailed description should be added to each diagram. After the Abstraction Hierarchy block are filled with the appropriate data, each block will be connected by SysML dependency lines. Dependency lines are used to show the means-ends relationships.

The Abstraction Hierarchy model is based on the results of questions asked during the interviews of SMEs.

Figure 83: A Text-Based Description of the Knowledge Elicitation Process for the WDA Phase

Other questions may be discussed during the interview. The number of questions and the subject of each question will vary across projects. Part of the questions asked in the WDA phase are the means/ends relationships that exist between the upper and lower levels of the Abstraction Hierarchy. Each level is connected by means-ends relationships. The means is a level below the ends. For example, the general function is

the means for the abstract function. The lower levels describe the actions, components or parameters that are necessary for achieving the ends or upper levels of the AH. Another form of obtaining information about the new system is to observe domain experts performing tasks the new system will be required to carry out.

The third step is to populate abstraction hierarchy. Once interviews and documentation reviews have been completed, the Abstraction Hierarchy can be populated with the appropriate data for each level. SysML blocks are used to represent the data at the different levels of the Abstraction Hierarchy. Block diagrams should be used to represent the functional purpose and general function levels. Part diagrams should be used to represent the physical function and component levels. Constraint property diagrams should be used to represent the abstract function level. A detailed description should be added to each diagram. After the Abstraction Hierarchy blocks are filled with the appropriate data, each block will be connected by SysML dependency lines. Dependency lines are used to show the means-ends relationships. Figure 84 illustrates the results of questions asked and the construction of the Abstraction Hierarchy for the ATM.

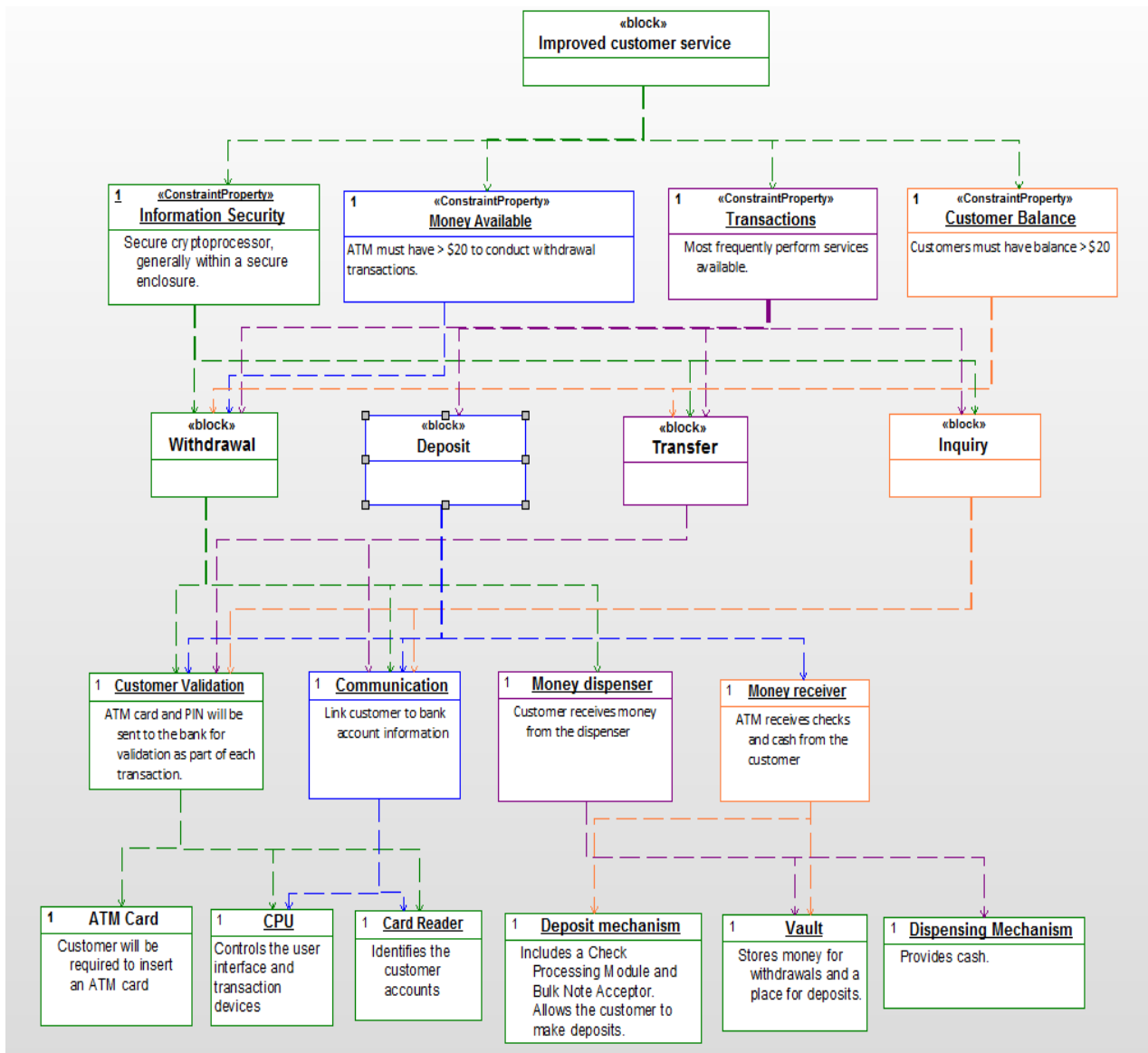


Figure 84: An Abstraction Hierarchy Model for the ATM Example

4.7.3 Control Task Analysis

The Control Task Analysis (ConTA) phase requires a similar interview of SMEs to identify user tasks. Questions that should be asked to domain experts should focus on the tasks necessary to successfully complete their jobs. Domain experts should

describe the cognitive activities required to complete a system task and identify shortcuts experts would use when completing a task. The questions may include, but are not limited to, the following:

- What are some of the steps taken to achieve a task?
- What kinds of events can act as alerts?
- What kinds of data or facts are available?
- What kinds of assessments about the system's condition or situation are possible with the information?
- What kinds of choices or alternatives are available for the system's desired or target state?
- What kinds of aims or objectives can be relevant or influence decisions?
- What kinds of target states are possible?
- What kinds of tasks are necessary and what kinds of resources are available?
- What kinds of procedures or sequences of steps are necessary?

Figure 85 shows a text-based description of the knowledge elicitation process for the ConTA phase of CWA and a ConTA structure. The ConTA structure provides the capability to organize and manage the general functions and control tasks of the system. This capability is critical for a larger, more complex analysis.

Section 5

The Control Task Analysis phase requires a similar interview of SME to identify user tasks. Questions that should be asked to domain experts should focus on the tasks necessary to successfully complete their jobs. Have domain experts describe the cognitive activities required to complete a system task and identify shortcuts experts would use when completing a task.

These questions may include, but limited to the following:

- What are some of the steps taken to achieve a task?
- What kinds of events can act as alerts?
- What kinds of data or facts is available?
- What kinds of assessments about the system's condition or situation is possible with the information?
- What kinds of choices or alternatives are available for the system's desired or target state?
- What kinds of aims or objectives can be relevant or influence decisions?
- What kinds of target states are possible?
- What kinds of tasks are necessary and what kinds of resources are available?
- What kinds of procedures or sequences of steps are necessary?

The diagram to the right is a Control Task Analysis (ConTA) structure. The ConTA is built in an Internal Block Diagram (IBD) of the Control Task Analysis block on the CWA structure. The input to the ConTA phase is the general functions from the AH. Each general function will have a ConTA structure created in an IBD of the Control Task Analysis block on the CWA structure. In this case there would be four IBDs for the ATM system (i.e. withdrawals, deposits, transfers and inquiries).

In this example, one general function will be represented by a block diagram in the each IBD. The example will focus on the "deposits" general function. The "deposits" general function is further decomposed into two control tasks. The control tasks are deposit cash or deposit checks. The definition of each is located in the description tab of each block. To access the definitions, double click the block diagram for the control task. The control tasks are represent by block diagrams.

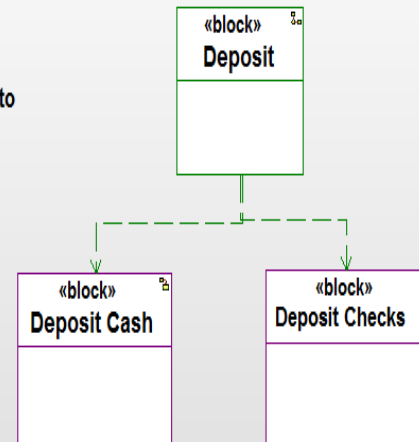


Figure 85: A Text-Based Description of the Knowledge Elicitation Process for the ConTA Phase

Once interviews have been completed, the Decision Ladder (DL) can be populated with the appropriate data for each information processing activity and knowledge state. The ConTA is built in an Internal Block Diagram (IBD) of the Control Task Analysis block on the CWA structure (i.e., Figure 82) The input to the ConTA phase is the general functions from the AH. Each general function will have a ConTA

structure created in an IBD of the Control Task Analysis block on the CWA structure (i.e. Figure 82). In this case, there would be four IBDs for the ATM system (i.e., withdrawals, deposits, transfers, and inquiries). In this example, one general function will be represented by a block diagram in each IBD. The example will focus on the “deposits” general function. The “deposits” general function is further decomposed into two control tasks. The control tasks are “deposit cash” or “deposit checks.” The definition of each is located in the description tab of each block. To access the definitions, double-click the block diagram for the control task. The control tasks and the general function are represented by block diagrams. A state machine diagram is attached to the “deposit cash” block diagram. The notation is an icon located in the upper right hand corner of the “deposit cash” block diagram. The relationship that exists between the general function and the control tasks is a dependency relationship. The general function needs the control tasks to be completed by the user. Once the user completes the control task, the general function is satisfied. When the general functions are satisfied, the purpose of the system can be fulfilled.

The state machine diagrams are used to construct the Decision Ladder. The Send Action and State elements are used to represent the information processing activities and knowledge states at the different steps of the DL. A detailed description should be added to each element. Transition lines are used to show the transition from information processing activities to knowledge states. Figure 86 shows the answers to some of the questions asked and the construction of the DL.

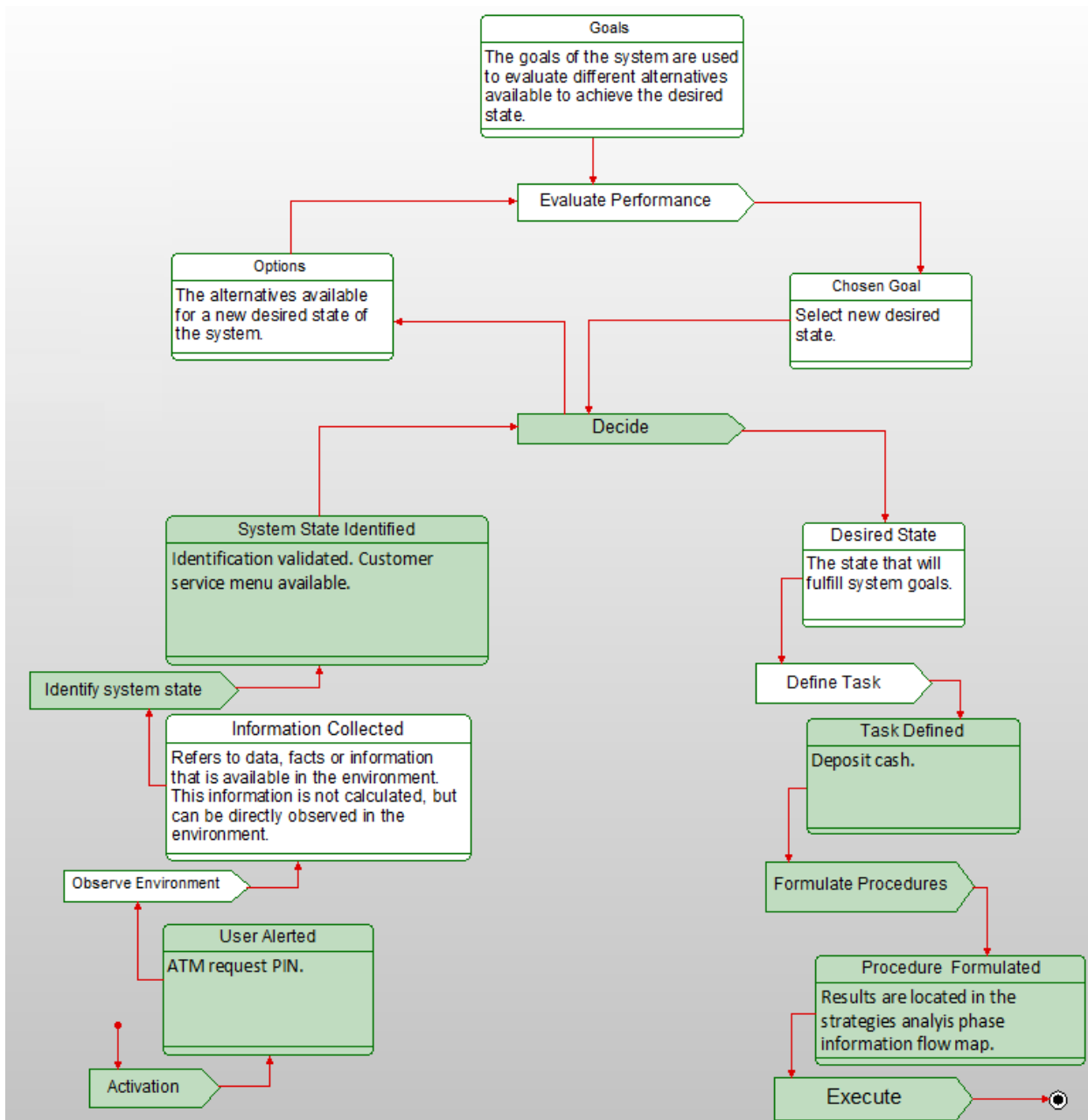


Figure 86: A Decsion Ladder Model for the ATM Example

4.7.4 Strategies Analysis

The Strategies Analysis (SA) phase also requires an interview of SMEs to identify user task strategies. The questions should focus on the course of action used and the options available to complete a task. Questions may include, but are not limited to, the following:

- What are some of the possible strategies that can be used to complete a task?
- Which of the strategies mentioned before would most system operators use to complete a task?
- What steps would a system novice use to complete a task?
- What steps would a system expert use to complete a task?

Figure 87 shows a text-based description of the knowledge elicitation process for the SA phase of CWA and an SA structure. The ConTA structure provides the capability to organize and manage the general functions, the control tasks, and the user strategies. The SA phase is built in an Internal Block Diagram (IBD) of the Strategies Analysis block on the CWA structure (i.e., Figure 82). The input to the SA phase is the general functions from the AH and the control tasks of each DL. Each general function will have a SA structure created in an IBD of the Strategies Analysis block on the CWA structure (i.e., Figure 82). In this case, there would be four IBDs for the ATM system (i.e., withdrawals, deposits, transfers, and inquiries). In this example, one general function will be represented by a block diagram in each IBD. The example will focus on the

“deposits” general function. The “deposits” general function is further decomposed into two control tasks. The control tasks are “deposit cash” or “deposit checks.” The control tasks are further decomposed into user strategies, which are “deposit individual bills” or “all cash at once.” The example focuses on the “all cash at once” user strategy.

The definition of each block is located in the description tab of each block. To access the definitions, double-click the block diagram for the control task.

«block»
Strategies_Analysis

Section 5

The Strategies Analysis phase also requires an interview of SMEs to identify user tasks. The questions should focus on the course of action used and the options available to complete a task.

Questions may include, but not limited to the following:

- What are some of the possible strategies that can use to complete a task?
- Which of the strategies mentioned before would most system operators use to complete a task?
- What steps would a system novice use to complete a task?
- What steps would a system expert use to complete a task?

The diagram to the right is a Strategy Analysis (SA) structure. This structure provides the capability to organize and manage all general functions, user control tasks and user strategies. This capability becomes more critical for a larger, more complex analysis and is one of the major advantages of using SysML.

The SA structure is built in an Internal Block Diagram (IBD) of the Strategies Analysis block on the CWA structure. The input to the SA phase is the control tasks from the ConTA phase. The SA structure is built with the same diagrams as the ConTA structure with the exception of the strategies that are added to this phase.

Each general function will have an SA structure created in an IBD of the Strategies Analysis block. For the purpose of the ATM example, only one general function will be spotlighted. The “Deposit” general function is further decomposed into two control task. The “Deposit Cash” control task is further decomposed into two courses of action.

The strategies include: deposit all cash together and deposit cash separately. Definitions of each strategies is located in the description tab of each block. To access the definitions, double click the block diagram for the strategy. The strategies are represent by block diagrams. The strategies is linked to the control tasks by an information flow relationship. The strategies provide the course of action required to complete a control task which satisfies the general function.

```

graph TD
    Deposit["«block» Deposit"]
    DepositCash["«block» Deposit Cash"]
    DepositChecks["«block» Deposit Checks"]
    AllCash["«block» All Cash at Once"]
    Individually["«block» Individually Deposit Cash"]

    Deposit -.-> DepositCash
    Deposit -.-> DepositChecks
    AllCash -.->|«flow» Strateg| DepositCash
    Individually -.->|«flow»| DepositCash
    
```

[Previous](#) | [Next](#) | [Outline](#)

Figure 87: A Text-Based Description of the Knowledge Elicitation Process for the SA Phase

A state machine diagram is attached to the “all cash at once” block diagram. The notation is an icon located in the upper right hand corner of the “all cash at once” block diagram. The relationship that exists between the general function and the control tasks is a dependency relationship. The relationship that exist between the user strategies and the control tasks is a flow relationship. The control tasks and user strategies exchange information between them. The control task identifies the task that the user needs to complete and the strategy block states how the task will be completed.

Once interviews have been completed, the Information Flow Map (IFM) can be populated with the appropriate data for completing a control task. Use data collected during interviews to construct the IFM. Send Action and State diagrams are used to represent the information processing activities and knowledge states, respectively. Transition lines are used to shows the transition from information processing activities to knowledge states. Finally, a detailed description to each diagram should be added. Figure 88 shows an IFM strategy for depositing all cash bills at once for the ATM example.

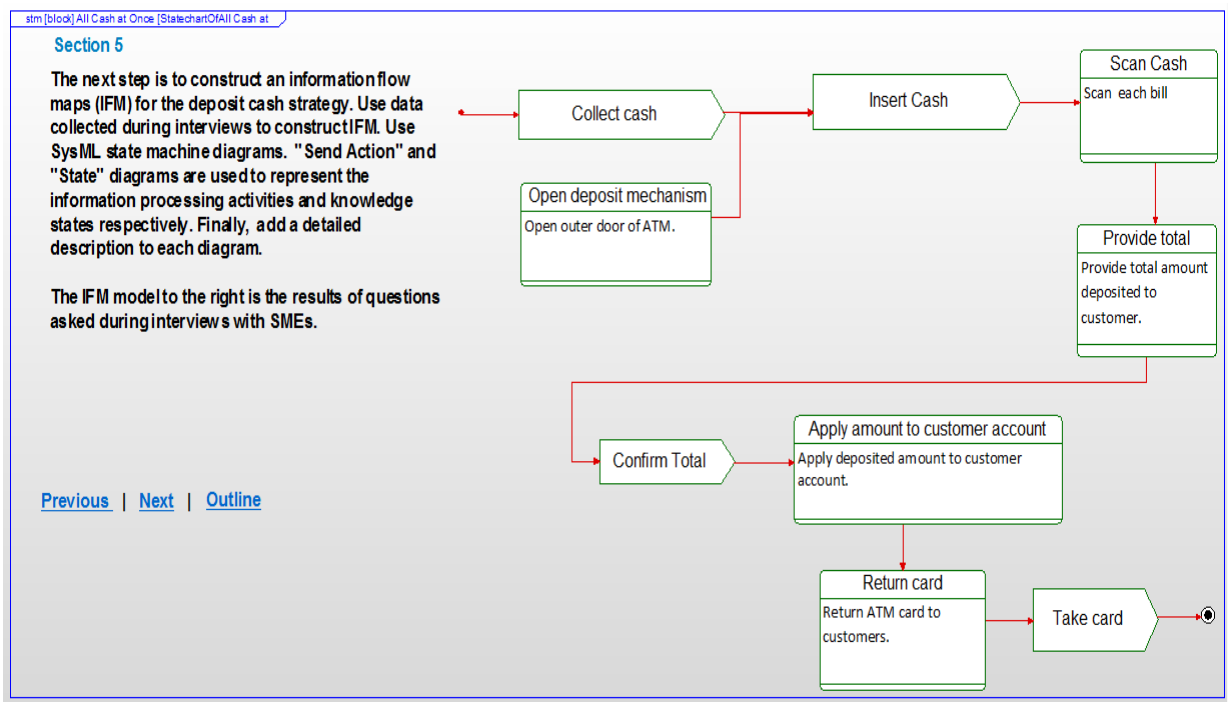


Figure 88: An Information Processing Activity Model for the ATM Example

4.7.5 Social Organization and Cooperation Analysis

The Social Organization and Cooperation Analysis (SOCA) phase considers the actors' strengths and weaknesses to determine allocation of task responsibility. The ATM example is simplistic and very intuitive. So an in-depth analysis is not required. An interview of SMEs should be carried out if a more in-depth analysis is required. The interview should identify actors and assign task responsibilities among the actors. Questions may include, but are not limited to, the following:

- Can you describe the various teams using the system?
- How do you allocate responsibilities for each person?

- Who depends on whom for help to complete a task?
- What is the specific role of each team member?
- How are decisions usually made?

Figure 89 shows a text-based description of the knowledge elicitation process for the SOCA phase of CWA and a SOCA structure. The SOCA structure is used to keep track of each general function and all associated control tasks and user strategies. The SOCA structure is created in the same format as the ConTA and SA structures. The SOCA phase is built in an Internal Block Diagram (IBD) of the Social Organization and Cooperation Analysis block on the CWA structure (i.e., Figure 82). The input to the SOCA phase is the information processing activities and knowledge states generated in the SA phase of CWA.

The ATM example focuses on the “all cash at once” user strategy. A use case diagram will be constructed for the “all cash at once” user strategy. Data collected during the interview process and the information processing activities and knowledge states from the Strategies Analysis phase are utilized in constructing the use case diagrams. Actors and use case elements are used to represent the system users, information processing activities, and knowledge states. Figure 90 shows a use case diagram for depositing all cash bills at once for the ATM example. In the description section of each use case element is a explanation of its meaning.

ibid [block] Social Organization and Cooperation Analysis [SOCA Level1]

«block»
Social Organization and Cooperation Analysis

Section 5

In the Social Organization and Cooperation Analysis (SOCA) phase we will evaluate the actors' strengths and weaknesses to determine allocation of tasks responsibility. The ATM example is simplistic and very intuitive. So an in-depth analysis is not required. If more analysis was needed, we would conduct an interview of subject matter experts to identify actors and assign task responsibilities.

Questions may include, but not limited to the following :

- Describe the various teams using the system?
- How do you allocate responsibilities for each person?
- Who depends on who for help to complete a task?
- What is the specific role of each team member?
- How are decisions usually made?

The SOCA structure is built in an IBD of the Social Organization and Cooperation Analysis block on the CWA structure. The input to the SOCA phase is the strategies from the SA phase. The SOCA structure is built with the same diagrams as the SA structure. Each general function will have an SOCA structure created in an IBD of the Social Organization and Cooperation Analysis block on the CWA structure.

For the purpose of the ATM example, only one general function is highlighted. The "Deposit" general function is further decomposed into two control tasks. The "Deposit Cash" control task is further decomposed into two strategies. The strategies are represented by block diagrams. The strategies is linked to the control tasks by an information flow relationship. The strategies provide the course of action required to complete a control task which satisfies the general function.

[Previous](#) | [Next](#) | [Outline](#)

```

graph TD
    Deposit["«block» Deposit"]
    DepositCash["«block» Deposit Cash"]
    DepositChecks["«block» Deposit Checks"]
    AllCash["«block» All Cash at Once"]
    Individually["«block» Individually Deposit Cash"]

    Deposit -.-> DepositCash
    Deposit -.-> DepositChecks
    DepositCash -.-> AllCash
    DepositCash -.-> Individually
  
```

Figure 89: A Text-Based Description of the Knowledge Elicitation Process for the SOCA Phase

Section 5

The next is to construct a use case(UC) diagram. We will use data collected during interview process and the information processing activities and knowledge states from the Strategies Analysis phase to construct use case diagrams. Use SysML use case diagrams. "Actors" and "Use case" diagrams are used to represent the system users, information processing activities, and knowledge states.

[Previous](#) | [Next](#) | [Outline](#)

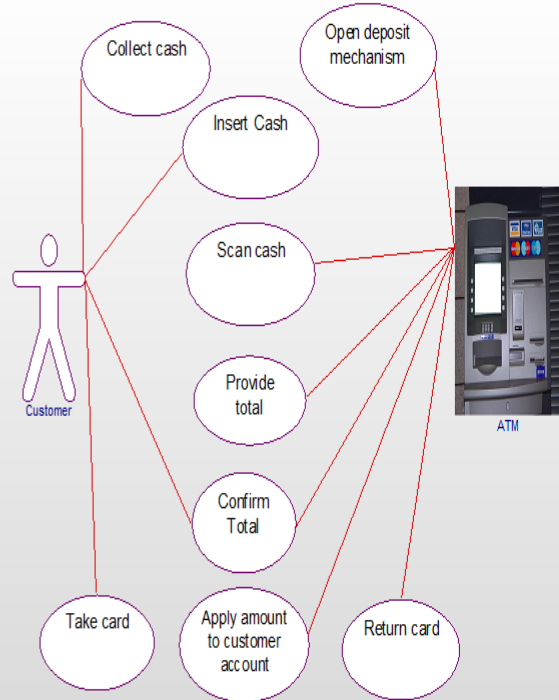


Figure 90: A Use Case Diagram for the ATM Example

4.7.6 Worker Competencies Analysis

The Worker Competencies Analysis (WCA) phase is the final phase of CWA. It outlines the competencies that the system users must have or acquire in order to effectively perform control tasks. As with all the other phases, knowledge about the system and the end-user is critical to modeling a Skill, Rule, and Knowledge (SRK) inventory. SMEs will be queried about the level of knowledge required by the user to complete information processing activities. The information processing activities are

outputs of the Control Task Analysis and Strategies Analysis phases. Questions may include, but are not limited to, the following:

- What information does the user have to know in order to complete the information processing activities?
- What rules, regulations, or policies does the user need to know?
- What problem solving procedures will the user have to be familiar with?

Figure 91 shows a text-based description of the knowledge elicitation process for the WCA phase of CWA and a WCA structure. The WCA structure is used to keep track of each general function and all associated control tasks and user strategies. The WCA structure is created in the same format as the SA and SOCA structures. The WCA phase is built in an IBD of the Worker Competencies Analysis block on the CWA structure (i.e., Figure 82). The input to the WCA phase is the information processing activities generated in the ConTA and SA phase of CWA. The definition of each block is located in the description tab of each block. To access the definitions, double-click the block diagram for the control task. An activity diagram is attached to the “deposit” block diagram. The notation is an icon located in the upper right hand corner of the “deposit” block diagram. The activity diagram will contain the SRK inventory modeling tool for the WCA phase. Each general function should have an activity diagram attached to it.

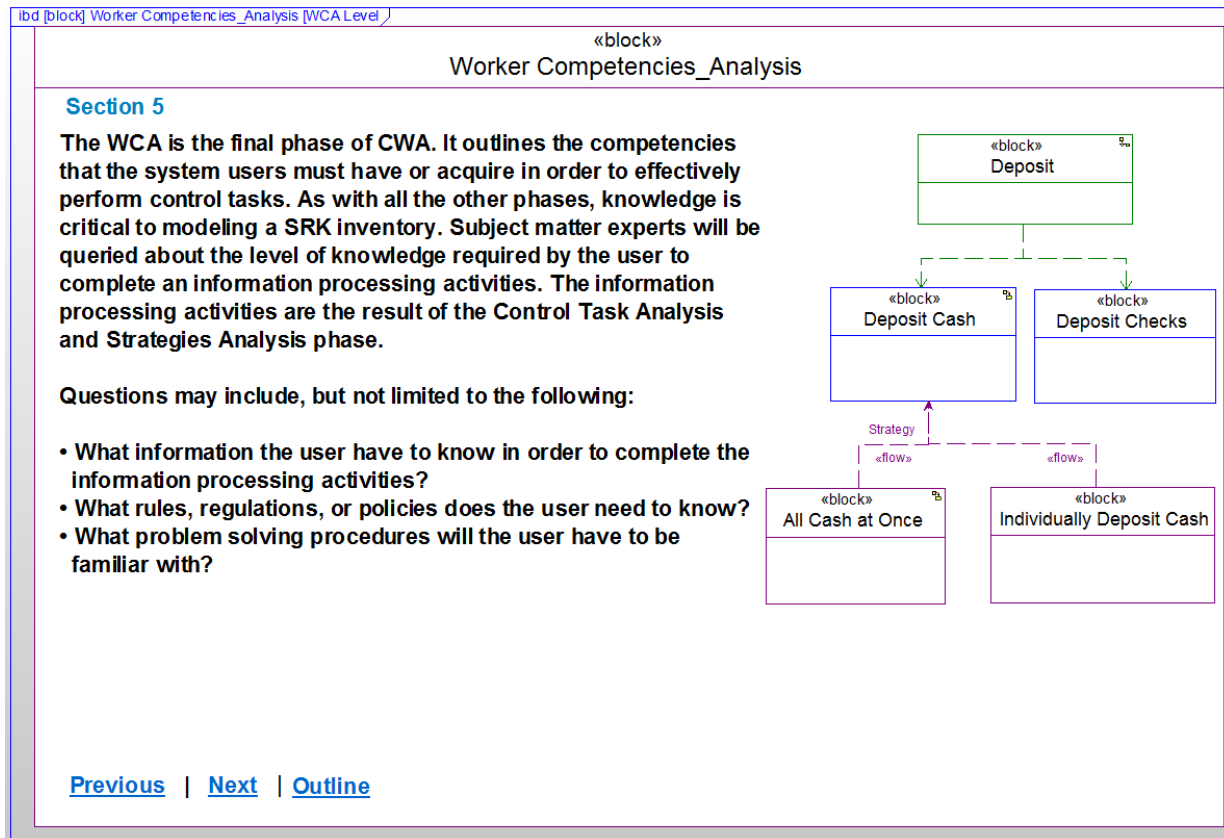


Figure 91: A Text-Based Description of the Knowledge Elicitation Process for the WDA Phase

Data collected during interviews and information processing activities from the Control Task Analysis and Strategies Analysis phases will be used to construct an SRK inventory model. Swimlanes elements are used to create a table which separates the level of cognitive behaviors (i.e., Skill-Based Behavior (SBB), Rule-Based Behavior (RBB), Knowledge-Based Behavior (KBB)). Send action elements represent information processing activities from the ConTA and SA phases. Action elements represent the level of knowledge required by the user. Transition lines are used to show

the transition from information processing activities to higher levels of cognitive behavior.

Finally, a detailed description of each level of cognitive behavior should be added.

Figure 92 shows a SRK inventory for the “collect cash” and “confirm total” information processing activities in the ATM example.

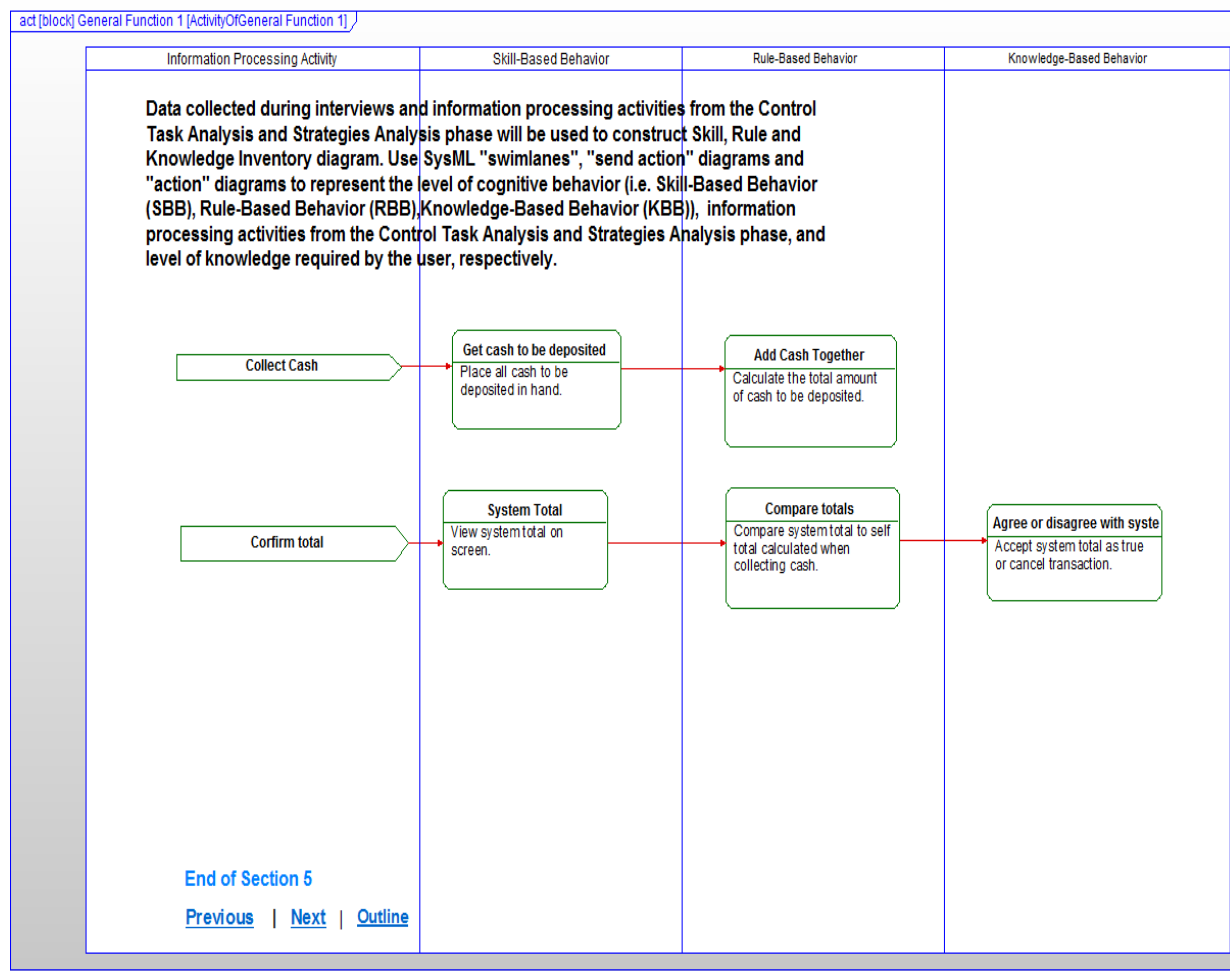


Figure 92: A Skill, Rule, and Knowledge Inventory Model for the ATM Example

4.8 Preliminary Usability Evaluation

A usability assessment of the CWAT was conducted by four college graduate students. They reported any navigational errors, uncertainty in understanding, usability issues, or confusion using the tutorial. Table 16 through Table 19 show the feedback received from the graduate students. Each graduate student had no experience using SysML or the Rational Rhapsody software. Each graduate had no experience using the Cognitive Work Analysis (CWA) methodology. Two out of the four graduate students have knowledge of ergonomic and human factors. The students provided their feedback on a discrepancy sheet that was given to them. Most of the recommendations have been incorporated into the CWAT.

Table 16: Usability Evaluation for User 1

Section #	Diagram Title	Usability Issue
2	CWA Structure	Add "Icon" word for click-on attributes and operations
2	CWA Structure	Picture of icons in word instructions
2	SRK Inventory description	WCA --> SRK Inventory title change to skill, rule, and knowledge
3	Process Flow CWA Structure	Adjust flowlines of SA and adjust blocks and lines
3	CWA Structure	Lines instead of arrows from CWA structure
3	All	Likes colors representing each phase
3	Abstraction Hierarchy	Change AH (clean it up)
3	All	AH process flow remove ----> transition lines
4	TRACON Example	Adjust flow lines

Table 17: Usability Evaluation for User 2

Section #	Diagram Title	Usability Issue
0	Title Page	Needs to be more explanatory
0	Title Page	Reword “construction of a CWA framework”
0	Title Page	Reword “within their designs”
0	Section Outline	When clicked, some word content disappears
0	Section Outline	Font needs to be bigger
1	CWA Introduction	Objective of tutorial should be moved to the beginning of tutorial
1	High Level Overview	Include icons for picture
1	CWA sequence and information	Enlarge State Machine Diagram icon
All	All	Acronyms must be written out on each page
2	Abstraction Hierarchy	Need an acronym key per page
2 & 3	All	“State Machine Diagram” needs to have “Back” button
2	All	Add SysML translation, model-based and text-based description
3	Process flow chart	No description in each block

Table 18: Usability Evaluation for User 3

Section #	Diagram Title	Usability Issue
2	CWA sequence and information	The zoom pan should be added before
2	State chart and strategies	Slides key should be more obvious
2	CWA Structure	Enlarge font of Attributes and operation
	CWA Structure	Change structural view to
3	AH Process flow	Decide diagram letters are mistake
3	AH Process flow	Bold (Step 1, Step 2, etc.)
3	AH Process flow	Place Figures Next to Process Flow
3	UC process flow	"Previous" Button not working
All	All	Verify all "Previous" buttons
All	All	Move "End of Section X" to middle and make large
All	All	Add key user instruction/summary of
4	AH	Change aircraft responsibility to
4	DL	Explain pathway through DL with more
4	IFM	Add Strategy 1, 2, and 3 to each
4	IFM TRACON Example	Explain order of IFM and add sequence
4	UC	Add Strategy 1, 2, and 3 to each UC
5	Generic Models	Use template instead of generic
5	All	Space out words

Table 19: Usability Evaluation for User 4

Section #	Diagram Title	Usability Issue
0	Section Outline	CWA definition
1	CWA Overview	3rd line typo “incorporated”
1	CWA Overview	Bottom of page blue ribbon text. Not sure what is the instruction or suggestion
1	CWA High Level view	Diagram fonts are small in the upper boxes
1	CWA High Level view	Use icons to help users to learn the different CWA components
1	CWA Sequence and Info	Paragraph: “the icons in the upper...” What icons?? Do you mean the attributes in the upper part of the box?
1	CWA Sequence and Info	Diagram: Some text on boxes is not showing
1	CWA Sequence and Info	Blue text should mention that you are going to section 2 model description.
2	Abstraction Hierarchy	Bullets for the five levels of abstraction
2	Information flow Map	End of first sentence needs to be reviewed (first paragraph)
3	Process Flow Chart	When double-clicking the boxes, there is no description
3	All diagrams	Numbers in flow diagrams should start with the word “step”, if they match the examples (some steps are repeated steps)

4.9 Results Summary

Chapter 4 illustrated how this methodology can be applied by systems engineers. The tutorial informs users about the purpose of CWA, knowledge elicitation methods, SysML diagrams used to construct the CWA framework, and CWA construction process flow. The tutorial is constructed in SysML using appropriate diagrams to translate CWA terminology into SysML terminology for systems engineers to comprehend. The scope of this study was limited to the development of the cognitive work analysis framework tutorial using system modeling language. A formal evaluation of the cognitive work analysis framework tutorial using system modeling language should be done in the future.

CHAPTER FIVE: CONCLUSION

5.1 Summary of Study

The purpose of this study was to develop a Cognitive Work Analysis Tutorial (CWAT) using Systems Modeling Language. The study had two phases. The first was to align the CWA terminology with the SysML to produce a CWA framework using SysML. The second was to create an instruction using SysML to inform systems engineers of the process of incorporating cognitive requirements into their system designs.

The initial focus of the systems engineering process involves developing a set of complete, consistent, and achievable requirements. Cognitive work requirements are critical for defining effective systems. The systems engineering process lacks cognitive factors in system design (Stoner et al., 2006). This deficiency can be attributed to time constraints, budget limits, a lack of access to the information or training, or underestimation of the value of cognitive requirements methods. Additionally, cognitive engineers have difficulty in translating their findings into a format that systems engineers can understand and use. Most cognitive engineers have an excellent understanding of the human user, but have a poor grasp of how to incorporate their understanding of the human user into the development of a set of design requirements (Rasmussen, 1986).

At the present time, most systems engineers do not have access to cognitive work analysis information or training in terms they can understand. This lack of access leads to a disregard of the cognitive aspect of system design. The result of this issue is system requirements that do not account for the cognitive strengths and limitations of users. Systems engineers cannot design effective decision support systems without defining cognitive work requirements. In order to improve system requirements, integration of cognitive work requirements into the systems engineering process has to be improved. One option to address this gap is to develop a CWA Framework using SysML. The CWAT developed in this study translated CWA terminology into a system design language (i.e., SysML) that systems engineers currently use.

5.2 Using SysML to Model a CWA Framework

Complex sociotechnical systems developed using the CWA methodology require large amounts of documentation to assess the design parameters of the system. The SysML Model-Based Systems Engineering (MBSE) approach can reduce the amount of documentation required, which contributes to better communication, understanding, and coordination of cognitive work requirements. In addition, relationships among different CWA modeling techniques can be easily established with a model-based approach. This capability is not available with the paper-based and Microsoft software currently used to document cognitive work requirements. Studies have shown that the graphics and text modeling frameworks help alleviate cognitive loads (Dori, 2008). Combining

graphics and text to represent cognitive work requirements contributes to an easier understand of complex sociotechnical systems.

Another benefit of using SysML to model CWA phases is the ability to pass changes to one modeling element forward to association model elements in other CWA phases. This means that minor changes to text in a diagram will feed through from the initial stages to the subsequent phases. This has particular benefits in the CWA framework because each phase builds upon the preceding phase. This capability increase the speed and accuracy at which CWA framework models can be developed, edited, and reviewed.

5.3 Challenge to the Three Evils of Systems engineering

The three evils of systems engineering are “complexity,” “a lack of understanding,” and “communication issues”(Holt & Perry, 2008). Complexity, lack of understanding, and communication are interrelated. Any deficiencies in one will lead to deficiencies in the others evils of systems engineering. Likewise, any improvements to one will lead to improvements in the other factors.

The complexity of a system is based on the number of relationships that exist between system elements. The higher the number of relationships, the more complex a system will be. This method confronts the complexity evil by providing systems engineers with a structured framework to define, manage, organize, and model cognitive work requirements.

Lack of understanding can occur in any phase of the systems engineering process. A lack of understanding can lead to the needs of the user not being addressed, problems that are not clearly defined, improper application of systems engineering principles, inaccurate requirements, or incorrect component interactions. CWA frameworks provide a tool for systems engineers to incorporate the cognitive strengths and limitations of a system user in system design. Additionally, it contributes to defining more accurate use cases. This increases all stakeholders' understanding of the system and the user.

Communication problems can exist between all system stakeholders. A set of system requirements can be interpreted differently by different system stakeholders. The CWA framework developed in SysML provides systems engineers with a structured and standard way to define cognitive work requirements when designing a system.

5.4 Future Research

Future research may include a formal usability evaluation of the CWAT, an expanded use of other SysML capabilities, or a mapping of the CWA framework to Human System Integration(HSI) domains.

A formal usability evaluation could assess the user's ability to learn and apply CWA framework method. There are several methods that can be implemented to evaluate the CWAT users. These methods include cognitive walkthroughs, interviews, Goal-Operators-Method-Selection Rules (GOMS), Function Mechanism Hierarchy, and pluralistic walkthroughs. A usability evaluation of the CWAT could assess the following:

- How easily the user can navigate the tutorial.
- Verify information organization and format.
- Verify the ease of use and understanding of CWA and SysML diagrams.

5.4.1 Trade-off Analysis Capability

One capability that deserves further exploration is trade-off analysis of cognitive work requirements. SysML provides a model-based form of documentation. This form of documentation allows for meaningful tradeoffs to be considered. Models are critical to trade-off analysis and the evaluation of alternatives (Karwowski & Ahram, 2009). SysML requirement and parametric diagrams can be utilized to support cognitive work requirements tradeoff analysis. Parametric diagrams can represent the relationship between measures of effectiveness and system properties to evaluate the effectiveness of a particular system model (Friedenthal et al., 2006). A parametric relation can be defined to represent an evaluation function to evaluate alternative solutions. The evaluation function produces one or more outputs that represent a measure of effectiveness (Weilkiens, 2008). This evaluation function may include a weighting of the functions associated with various criteria used to evaluate the alternatives. The criteria may be associated with system performance, cost, or schedule. For example, the values of the criteria are X and Y. The weights of importance would be w_tX and w_tY . The criteria, weights and resulting scores must be combined in order to select the preferred alternative. The combining function in SysML is called an objective function.

The corresponding properties from each alternative are put into the evaluation function to determine the overall measure of effectiveness.

5.4.2 Human Systems Integration

Another area for future research is mapping the CWA framework to the Human System Integration (HSI) domains. There are many HSI tools available to systems engineers. The abundance of options can make it difficult for systems engineers to locate the right tool to address the appropriate HSI domains. One option would be to minimize the number of tools needed, while maximizing the number of HSI domains that one tool would address. CWA can potentially increase the number of HSI domains evaluated by one tool to approximately five HSI domains. SysML can provide the capability to explicitly link five domains of HSI.

Currently, most HSI tools tend to be domain-specific. Most tools used for HSI do not provide an explicit means of mapping cognitive work requirements to HSI requirements. HSI domains consists of manpower, personnel, training, Human Factors Engineering (HFE), system safety, health hazards, and personnel survivability. Most HSI tools cover four of the seven HSI domains. These four are HFE, training, manpower, and personnel (Hale, Ching, Brett, & Rothblum, 2009). This coverage is not uniform among HSI tools. A USCG survey of HSI tools discovered that about two-thirds of the tools in the survey are software applications. One-third of the tools are based on specific techniques and methodologies. About half of the tools are general-purpose tools, while the others are more specialized. The Improved Performance Research Integration Tool (IMPRINT), Information and Functional Flow Analysis (IFFA), Jack, and Job Assessment Software System (JASS) can each be used to conduct analyses

related to three or four of the seven HSI domains. CWA can potentially increase the number of HSI domain evaluated by one tool to approximately five HSI domains. None of the HSI tools reviewed provide an explicit means of mapping cognitive work requirements to HSI requirements.

5.5 Concluding Statement

In conclusion, the three evils cannot be eliminated in systems engineering, but they can be minimized using model-based systems engineering. CWAT guides systems engineers with integrating cognitive work requirements in system design to support users' cognitive functions, including situational awareness, problem solving, and decision making.

APPENDIX: COPYRIGHT PERMISSION



Human
Factors
and
Ergonomics
Society

President
Anthony D. Andre
President-Elect
Mica R. Endsley
Immediate Past President
Kathleen L. Mosier
Secretary-Treasurer
Ronald G. Shapiro
Secretary-Treasurer-Elect
Valerie J. Rice
Immediate Past
Secretary-Treasurer
Scott A. Shappell
Executive Council
at-Large Members
Nancy J. Cooke
Francis T. Durso
Paul A. Green
M. Susan Halbeck
John D. Lee
Marc L. Resnick

Executive Director
Irene Souders, CAE

P.O. Box 1369
Santa Monica, CA
90406-1369 USA
TEL 310 / 394-1811
FAX 310 / 394-2410
info@hfes.org
http://hfes.org

May 12, 2011

Wilfred Wells
1427 Bella Coola Dr.
Orlando, FL 32828

Dear Wilfred,

Thank you for your request for permission to reproduce the following figures ("the Figures") from our publication, *Proceedings of the 52nd Human Factors and Ergonomics Society Meeting*:

Figures 1-5, St-Cyr, O., & Kilgore, R. M., (2008). A five-phase CWA for air traffic control: Example from a TRACON microworld. *Proceedings of the 52nd Human Factors and Ergonomics Society Meeting*, 272-276.

We are pleased to grant you a nonexclusive, nontransferable license to reproduce the Figures for your dissertation on the following terms and conditions:

1. Acknowledgment of HFES copyright must appear in the work. The following wording is recommended:

Reproduced with permission from *Proceedings of the 52nd Human Factors and Ergonomics Society Meeting*. Copyright 2008 by the Human Factors and Ergonomics Society. All rights reserved.

2. This permission extends to this request only. To reproduce the Figures in other derivative works, (in either print or electronic format) etc, will require separate permission.

Thank you for your interest in the Human Factors and Ergonomics Society.

Sincerely,
HUMAN FACTORS AND ERGONOMICS SOCIETY


Lois Smith
Communications Director
lois@hfes.org

REFERENCES

- Aarts, E. & Roovers, R. (2003). IC design challenges for ambient intelligence. Paper presented at the *Proceedings of the Conference on Design, Automation and Test in Europe Conference*, Munich, Germany. 2–7.
- Ahmad, S. Z. (2007). *Analyzing suitability of SysML for system engineering applications*. (Master's Thesis, Blekinge Institute of Technology).
- Ahram, T., & Karwowski, W. (2009). Human systems integration modeling using systems modeling language. Paper presented at the *Human Factors and Ergonomics Society Annual Meeting Proceedings*, San Antonio, TX, 53(24) 1849-1853.
- Andersson, H., Herzog, E., Johansson, G., & Johansson, O. (2009). Experience from introducing unified modeling Language/Systems modeling language at Saab Aerosystems. *Wiley Periodicals, Inc.*, 13(4), 369–380. doi:10.1002/sys.20156
- Armour, F., & Miller, G. (2001). *Advanced use case modeling: Software systems*. Boston, MA: Addison-Wesley Longman Publishing Co., Inc.
- Balmelli, L. (2007). An overview of the systems modeling language for products and systems development. *Journal of Object Technology*, 6(6), 149–177.
- Bijker, W. E. (1997). *Of bicycles, bakelites, and bulbs: Toward a theory of sociotechnical change*. Cambridge, MA: MIT Press.
- Bisantz, A. M., Roth, E., Brickman, B., Gosbee, L. L., Hettinger, L., & McKinney, J. (2003). Integrating cognitive analyses in a large-scale system design process. *International Journal of Human-Computer Studies*, 58(2), 177–206.
- Blanchard, B. S., & Fabrycky, W. J. (1990). In Fabrycky W. J., Mize J. H. (eds.), *Systems engineering and analysis* (4th ed.). Englewood Cliffs, NJ: Prentice Hall.
- Bock, C. (2005). Systems engineering in the product lifecycle. *International Journal of Product Development*, 2(1), 123–137.

- Boehm, B., Valerdi, R., & Honour, E. (2008). The ROI of systems engineering: Some quantitative results for software-intensive systems. *Systems Engineering*, 11(3), 221–234. doi:10.1002/sys.20096
- Boehm, B. W. (1986). A spiral model of software development and enhancement. *SIGSOFT Software Engineering Notes*, 11(4), 14–24. doi:10.1145/12944.12948
- Booton, R. C., & Ramo, S. (1984). The development of systems engineering. *Aerospace and Electronic Systems, IEEE Transactions on*, (4), 306–310.
- Briand, L., Falessi, D., Nejati, S., Sabetzadeh, M., & Yue, T. (2010). *Traceability and SysML design slices to support safety inspections: A controlled experiment*. (Technical Report No. 08). Lysaker, Norway: Simula Research Laboratory.
- Buede, D. M. (2009). *The engineering design of systems* (2nd ed.). New York, NY: John Wiley & Sons.
- Burns, C. M., Bryant, D. J., & Chalmers, B. A. (2000). A work domain model to support shipboard command and control. Paper presented at the *2000 IEEE International Conference on Systems, Man, and Cybernetics*, Nashville, TN. 2228–2233. doi:10.1109/ICSMC.2000.886447
- Burns, C. M., Momtahan, K., & Enomoto, Y. (2006). Supporting the strategies of cardiac nurse coordinators using cognitive work analysis. Paper presented at the *Human Factors and Ergonomics Society Annual Meeting Proceedings*, San Francisco, CA, 50(3) 442–446.
- Calvano, C. N., & John, P. (2004). Systems engineering in an age of complexity. *Systems Engineering*, 7(1), 25–34.
- Cancila, D., Terrier, F., Belmonte, F., Dubois, H., Espinoza, H., Gérard, S., & Cuccuru, A. (2009). Sophia: A modeling language for model-based safety engineering. Paper presented at the *ACES-MB Workshop Proceedings*, Denver, CO. 11–25.
- Chin, M., Sanderson, P., & Watson, M. (1999). Cognitive work analysis of the command and control work domain. Paper presented at the *Proceedings of the 1999 Command and Control Research and Technology Symposium*, Newport, RI. 233–224.
- Cockburn, A. (1997). Structuring use cases with goals. *Journal of Object-Oriented Programming*, 10(5), 56–62.

- Constantine, L. L., & Lockwood, L. A. (2001). Structure and style in use cases for user interface design. In M. Van Harmelen (Ed.), *Object modeling and user interface design: Designing interactive systems* (pp. 245–280). Boston, MA: Addison-Wesley Longman Publishing Co.
- Cooper, D. E. (2002). *Agencies need to better define and track the training of their Employees*. (Government Report No. GAO-02-737). Washington, D.C.: United States General Accounting Office.
- Crone, D. J., Sanderson, P. M., & Naikar, N. (2003). Using cognitive work analysis to develop a capability for the evaluation of future systems. Paper presented at the *Human Factors and Ergonomics Society Annual Meeting Proceedings*, Santa Monica, CA, 47(17), 1938–1942.
- Cummings, M., & Guerlain, S. (2003). The tactical tomahawk conundrum: Designing decision support systems for revolutionary domains. Paper presented at the *IEEE International Conference on Systems Man and Cybernetics*, Washington, D.C., 2, 1583–1588.
- Cummings, M. L. (2006). Can CWA inform the design of networked intelligent systems? Paper presented at the *Moving Autonomy Forward Conference*, Lincoln, UK.
- Curtis, B., Krasner, H., & Iscoe, N. (1988). A field study of the software design process for large systems. *Communications of the ACM*, 31(11), 1268–1287.
- Darses, F. (2001). Providing practitioners with techniques for cognitive work analysis. *Theoretical Issues in Ergonomics Science*, 2(3), 268–277.
- Defense Acquisition University. (2009). *Systems engineering process*. Retrieved June 15, 2011, from <https://acc.dau.mil/CommunityBrowser.aspx?id=250180>
- Department of the Navy. (2001). *Surface warfare program manager's guide to human systems integration*. Washington, DC: Department of the Navy.
- Dobre, D., Morel, G., Pétin, J. F., & Bajic, E. (2008). Improving digital interaction for operator-driven process-plant operation. Paper presented at the *9th IFAC Workshop on Intelligent Manufacturing Systems*, Szczecin, Pologne.
- Dori, D. (2008). Words from pictures for dual-channel processing. *Communications of the ACM*, 51(5), 47–52.

- Doyle, L., & Pennotti, M. (2005). Systems engineering experience with UML on a complex system. Paper presented at the *Conference on Systems Engineering Research*, Hoboken, NJ.
- Drath, R., Luder, A., Peschke, J., & Hundt, L. (2008). AutomationML-the glue for seamless automation engineering. Paper presented at the *13th IEEE International Conference on Emerging Technologies and Factory Automation*, Hamburg, Germany. 616–623. doi:10.1109/ETFA.2008.4638461
- Elm, J. P., Goldenson, D. R., El Emam, K., Donatelli, N., & Neisa, A. (2007). *A survey of systems engineering effectiveness - initial results* (Technical Report No. 2008-SR-034). Pittsburgh, PA: Carnegie Mellon Software Engineering Institute. Retrieved from <http://www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA475195&Location=U2&doc=GetTRDoc.pdf>
- Erguner, A. (2008). *A novel project management theory and its applicability*. (Master's Thesis, Naval Postgraduate School).
- Feiler, P. H., & De Niz, D. (2008). *ASSIP study of real-time safety-critical embedded software-intensive systems engineering practices*. (Technical Report No. CMU/SEI-2008-SR-001). Carnegie Mellon University Software Engineering Institute.
- Fidel, R., & Pejtersen, A. M. (2004). From information behaviour research to the design of information systems: The cognitive work analysis framework. *Information Research*, 10(1), 10–11.
- Forsberg, K., & Mooz, H. (1992). The relationship of systems engineering to the project cycle. *Engineering Management Journal*, 4(3), 36–43.
- Friedenthal, S., Moore, A., & Steiner, A. (2006). OMG systems modeling language (OMG SysML) tutorial. Paper presented at the *INCOSE 16th Annual International Symposium*, Orlando, FL.
- Friedenthal, S., Moore, A., & Steiner, R. (2008). *A practical guide to SysML: Systems model language*. San Francisco, CA: Morgan Kaufmann.
- Gates, D. (2008, Nov 20). Boeing 787 fastener problems caused by boeing engineers. *The Seattle Times*, Retrieved from http://seattletimes.nwsourc.com/html/business/technology/2008413389_dreamliner20.html

- Goknil, A., Kurtev, I., & Van den Berg, K. (2008). A metamodeling approach for reasoning about requirements. Paper presented at the *Model Driven Architecture—Foundations and Applications*, Berlin, Germany. 5095 310-325. doi:10.1007/978-3-540-69100-6_21
- Gonzalez-Horta, F. A., Enriquez-Caldera, R. A., Ramirez-Cortes, J. M., Martinez-Carballido, J., & Buenfil-Alpuche, E. (2010). Towards a cognitive handoff for the future Internet: Model-driven methodology and taxonomy of scenarios. Paper presented at the *2nd International Conference on Advanced Cognitive Technologies and Applications*, Lisbon, Portugal.
- Graves, H. (2009). Integrating SysML and OWL. Paper presented at the *OWL Experiences and Directions Workshop*, Chantilly, VA.
- Hale, C., Ching, C., Brett, B., & Rothblum, A. (2009). *Survey of human systems integration (HSI) tools for USCG acquisitions*. (Technical Report No. CG-D-04-09). Springfield, VA: National Technical Information Service.
- Hause, M. (2006). The SysML modeling language. Paper presented at the *5th European Systems Engineering Conference*, Edinburgh, Scotland.
- Hause, M. C., & Thom, F. (2007). An integrated safety strategy to model driven development with SysML. Paper presented at the *2nd Institution of Engineering and Technology International Conference on System Safety*, London. 124–129.
- Helms, B., Shea, K., & Hoisl, F. (2009). A framework for computational design synthesis based on graph-grammars and function-behavior-structure. Paper presented at the *International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, San Diego, CA. 53–54.
- Herzog, E., Pandikow, A., & Syntell, A. B. (2005). SysML an assessment. Paper presented at the *15th INCOSE International Symposium*, Rochester, NY.
- Hirtz, J., Stone, R. B., McAdams, D. A., Szykman, S., & Wood, K. L. (2001). A functional basis for engineering design: Reconciling and evolving previous efforts. *Research in Engineering Design*, 13(2), 65–82.
- Hofmann, H. F., & Lehner, F. (2001). Requirements engineering as a success factor in software projects. *IEEE Software*, 18(4), 58–66. doi:10.1109/MS.2001.936219
- Holt, J., & Perry, S. (2008). *SysML for systems engineering*. Stevenage, United Kingdom: Institution of Engineering & Technology (IET).

- Honour, E. C. (2004). Understanding the value of systems engineering. Paper presented at the *Proceedings of the INCOSE International Symposium*, Toulouse, France. 1–16.
- Honour, E. C. (2006). A practical program of research to measure systems engineering return on investment (SE-ROI). Paper presented at the *Proceedings of the 16th Annual Symposium of the International Council on Systems Engineering*, Orlando, FL.
- Honour, E. C., & Valerdi, R. (2006). Advancing an ontology for systems engineering to allow consistent measurement. Paper presented at the *Conference on Systems Engineering Research*, Los Angeles, CA.
- Huang, E., Ramamurthy, R., & McGinnis, L. F. (2007). System and simulation modeling using SysML. Paper presented at the *Proceedings of the 39th Conference on Winter Simulation: 40 Years! the Best is Yet to Come*, Washington, DC. 796–803.
- INCOSE. (2006). In Haskins C. (Ed.), *Systems engineering handbook: A guide for system life cycle processes and activities*. San Diego, CA: International Council on Systems Engineering.
- Insfrán, E., Pastor, O., & Wieringa, R. (2002). Requirements engineering-based conceptual modeling. *Requirements Engineering*, 7(2), 61–72.
- Jarraya, Y., Debbabi, M., & Bentahar, J. (2009). On the meaning of SysML activity diagrams. Paper presented at the *16th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems*, San Francisco, CA. 95–105. doi:10.1109/ECBS.2009.25
- Jenkins, D. P., Stanton, N. A., Salmon, P. M., Walker, G. H., Young, M. S., Whitworth, I., Hone, G. (2007). The development of a cognitive work analysis tool. In D. Harris (Ed.), *Engineering psychology and cognitive ergonomics* (pp. 504–511). Berlin, Germany: Springer. doi:10.1007/978-3-540-73331-7_55
- Jenkins, D. P., Stanton, N. A., Walker, G. H., Salmon, P. M., & Young, M. S. (2008). Applying cognitive work analysis to the design of rapidly reconfigurable interfaces in complex networks. *Theoretical Issues in Ergonomics Science*, 9(4), 273–295.
- Jenkins, D., Stanton, N., Salmon, P., Walker, G., & Young, M. (2008). Using cognitive work analysis to explore activity allocation within military domains. *Ergonomics*, 51(6), 798–815.

- Jenkins, D. P., Stanton, N. A., Salmon, P. M., & Walker, G. H. (2009). Using work domain analysis to evaluate the impact of technological change on the performance of complex socio-technical systems. *Theoretical Issues in Ergonomics Science*, 12(1), 1–14. doi:10.1080/14639220903353401
- Jenkins, D. P., Stanton, N. A., Salmon, P. M., Walker, G. H., & Rafferty, L. (2010). Using the decision-ladder to add a formative element to naturalistic decision-making research. *International Journal of Human-Computer Interaction*, 26(2-3), 132–146. doi:10.1080/10447310903498700
- Johnson, T. A., Paredis, C., Burkhart, R., & Jobe, J. (2007). Modeling continuous system dynamics in SysML. Paper presented at the *ASME International Mechanical Engineering Congress and Exposition*, Seattle, WA.
- Johnson, T., Paredis, C., & Burkhart, R. (2008). Integrating models and simulations of continuous dynamics into SysML. Paper presented at the *6th International Modelica Conference*, Bielefeld, Germany.
- Kaiser, B., Klaas, V., Schulz, S., Herbst, C., & Lascych, P. (2011). Integrating system modelling with safety activities. *Computer Safety, Reliability, and Security*, 6351, 452–465.
- Karwowski, W., & Ahram, T. Z. (2009). Interactive management of human factors knowledge for human systems integration using systems modeling language. *Information Systems Management*, 26(3), 262–274.
- Khaleefa, O. (1999). Who is the founder of psychophysics and experimental psychology? *American Journal of Islamic Social Sciences*, 16, 1–26.
- Kilgore, R., & St-Cyr, O. (2006). The srk inventory: A tool for structuring and capturing a worker competencies analysis. Paper presented at the *Human Factors and Ergonomics Society Annual Meeting Proceedings*, San Francisco, CA. 50(3) 506–509.
- Kossiakoff, A., & Sweet, W. N. (2003). *Systems engineering: Principles and practice*. Hoboken, New Jersey: John Wiley & Sons, Inc.
- Law, A. M., & Kelton, W. D. (2000). *Simulation modeling and analysis* (3rd ed.). New York, NY: McGraw-Hill.
- Leonard, J. (1999). *Systems engineering fundamentals*. Fort Belvoir, VA: Defense Systems Management College Press.

- Lilly, S. (2000, Jan). How to avoid use-case pitfalls. *Software Development*, 8, 40–45. Retrieved from www.sdmagazine.com/print/documentID=11235
- Lintern, G., & Naikar, N. (1998). Cognitive work analysis for training system design. Paper presented at the *Proceedings of the Australasian Computer Human Interaction Conference*, Adelaide, Australia. 252–259. doi:10.1109/OZCHI.1998.732222
- Luder, A., Hundt, L., & Biffel, S. (2009). On the suitability of modeling approaches for engineering distributed control systems. Paper presented at the *7th IEEE International Conference on Industrial Informatics*, Cardiff, Wales. 440–445. doi:10.1109/INDIN.2009.5195844
- Lui, F., Watson, M., & Queensland, A. (2002). Mapping cognitive work analysis (CWA) to an intelligent agents software architecture: Command agents. Paper presented at the *Proceedings of the Defence Human Factors Special Interest Group (DHFSIG)*, Melbourne, Australia.
- Luis de la Mata, J., & Rodríguez, M. (2009). Accident prevention by control system reconfiguration. *Computer Aided Chemical Engineering*, 26, 1117–1122.
- Ma, L., & Sun, Y. (2008). Object-oriented system-on-network-on-chip template and implementation: H. 263 case study*. *Tsinghua Science & Technology*, 13(1), 98–105.
- McGinnis, L., & Ustun, V. (2009). A simple example of SysML-driven simulation. Paper presented at the *2009 Winter Simulation Conference*, Austin, TX. 1703–1710.
- Naikar, N. (2006a). Beyond interface design: Further applications of cognitive work analysis. *International Journal of Industrial Ergonomics*, 36(5), 423–438.
- Naikar, N. (2006b). An examination of the key concepts of the five phases of cognitive work analysis with examples from a familiar system. Paper presented at the *Human Factors and Ergonomics Society Annual Meeting Proceedings*, San Francisco, CA., 50(3) 447–451.
- Naikar, N., Lintern, G., & Sanderson, P. (2002). Cognitive work analysis for air defense applications in Australia. In M. D. McNeese, & M. Vidulich (Eds.), *Cognitive systems engineering in military aviation environments: Avoiding cogminutia fragmentosa* (pp. 169–200). Dayton, OH: Human Systems Information Analysis Center.

- Naikar, N., Moylan, A., & Pearce, B. (2006). Analysing activity in complex systems with cognitive work analysis: Concepts, guidelines and case study for control task analysis. *Theoretical Issues in Ergonomics Science*, 7(4), 371–394.
- Naikar, N., Pearce, B., Drumm, D., & Sanderson, P. M. (2003). Designing teams for first-of-a-kind, complex systems using the initial phases of cognitive work analysis: Case study. *Human Factors*, 45(2), 202.
- Naikar, N., Sanderson, P., & Lintern, G. (1999). Work domain analysis: Identification of training needs for simulator acquisition. Paper presented at the *Proceedings of the Human Factors and Ergonomics Society 43rd Annual Meeting*, Houston, TX. 65–68.
- Naikar, N., & Sanderson, P. M. (1999). Work domain analysis for training-system definition and acquisition. *The International Journal of Aviation Psychology*, 9(3), 271–290.
- Naikar, N., & Sanderson, P. M. (2001). Evaluating design proposals for complex systems with work domain analysis. *Human Factors*, 43(4), 529.
- Naikar, N., & Saunders, A. (2003). Crossing the boundaries of safe operation: An approach for training technical skills in error management. *Cognition, Technology & Work*, 5(3), 171–180.
- Nejati, S., Sabetzadeh, M., Falessi, D., Briand, L., & Coq, T. (2011). *A SysML-based approach to traceability management and design slicing in support of safety certification: Framework, tool support, and case studies*. Unpublished manuscript. Retrieved Jun 10, 2011, from <http://modelme.simula.no/assets/ist-shiva.pdf>
- Object Management Group. (2008). *OMG systems modeling language*. Retrieved June 11, 2011, from <http://www.omg.org/spec/SysML/1.1>
- Ones, D. S., & Viswesvaran, C. (2003). Industrial-organizational (I/O) psychology to organizational behavior management (OBM). *Journal of Organizational Behavior Management*, 22(2), 41–57.
- Paradis, S., Breton, R., Elm, W. C., & Potter, S. S. (2002). A pragmatic cognitive systems engineering approach to model dynamic human decision-making activities in intelligent and automated systems. Paper presented at the *Proceedings of RTO Human Factors and Medicine Panel (HFM) Symposium*, Warsaw, Poland.

- Peak, R., Burkhart, R. M., Friedenthal, S. A., Wilson, M. W., Bajaj, M., & Kim, I. (2007). Simulation-based design using SysML part 2: Celebrating diversity by example. Paper presented at the *INCOSE International Symposium*, San Diego, CA.
- Preece, A. (2001). Evaluating verification and validation methods in knowledge engineering. In R. Roy (Ed.), *Industrial knowledge management: A micro-level approach* (1st ed., pp. 91–104). United Kingdom: Springer.
- Rasmussen, J. (1986). *Information processing and human-machine interaction: An approach to cognitive engineering*. New York, NY: Elsevier Science Inc.
- Rasmussen, J., Pejtersen, A. M., & Goodstein, L. P. (1994). *Cognitive systems engineering*. New York, NY: John Wiley & Sons, Inc.
- Rasmussen, J., Pejtersen, A. M., & Schmidt, K. (1990). *Taxonomy for cognitive work analysis*. (Technical Report No. 2871). Roskilde, Denmark: Riso National Laboratory.
- Rehak, L. A., Lamoureux, T. M., & Bos, J. C. (2006). Communication, coordination, and integration of cognitive work analysis outputs. Paper presented at the *Human Factors and Ergonomics Society Annual Meeting Proceedings*, San Francisco, CA, 50(3) 515–519.
- Resp Group. (2010). *Genesis of INCOSE*. Retrieved June 15, 2011, from <http://www.incose.org/about/genesis.aspx>
- Rittenbach, T., Kovarik, V. J., Krause-Aiguier, R., & Stewart, C. (2010). Complex terminal systems design. Paper presented at the *Military Communication Conference*, San Jose, CA. 2098–2103.
- Roth, E. M. (2008). Uncovering the requirements of cognitive work. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 50(3), 475.
- Ruegger, K. L. (2008). *Architecting a net-centric operations system of systems for multi-domain awareness*. (Master's Thesis, Naval Postgraduate School).
- Sage, A. P., & Armstrong, J. E. (2000). *Introduction to systems engineering*. New York, NY: Wiley.
- Sanderson, P., Naikar, N., Lintern, G., & Goss, S. (1999). Use of cognitive work analysis across the system life cycle: From requirements to decommissioning. Paper presented at the *Human Factors and Ergonomics Society Annual Meeting Proceedings*, Houston, TX, 43(3) 318-322.

- Sanderson, P. M., & Naikar, N. (2000). Temporal coordination control task analysis for analysing human-system integration. Paper presented at the *Human Factors and Ergonomics Society Annual Meeting Proceedings*, San Diego, CA, 44(1) 206–209.
- Schindel, W. D. (2005). Requirements statements are transfer functions: An insight from model-based systems engineering. Paper presented at the *Annual International Symposium of the International Council on Systems Engineering*, Rochester, NY.
- Schlager, K. J. (1956). Systems engineering-key to modern development. *IRE Transactions on Engineering Management*, EM-3(3), 64–66. doi:10.1109/IRET-EM.1956.5007383
- Schönherr, O., & Rose, O. (2009). First steps towards a general SysML model for discrete processes in production systems. Paper presented at the *Proceedings of the 2009 Winter Simulation Conference (WSC)*, Piscataway, New Jersey. 1711–1718. doi:10.1109/WSC.2009.5429164
- Shah, A., Kerzhner, A., Schaefer, D., & Paredis, C. (2010). Multi-view modeling to support embedded systems engineering in SysML. In G. Engels, C. Lewerentz, W. Schäfer, A. Schürr & B. Westfechtel (Eds.), *Graph transformations and model-driven engineering* (pp. 580–601). Berlin, Germany: Springer. doi:10.1007/978-3-642-17322-6_25
- Sheard, S. A. (1996). Twelve systems engineering roles. Paper presented at the *Proceedings of INCOSE*, Boston, MA.
- Skinner, B. F. (1984). The operational analysis of psychological terms. *Behavioral and Brain Sciences*, 7(04), 547–553.
- Soares, M. S., & Vrancken, J. (2008). Model-driven user requirements specification using SysML. *Journal of Software*, 3(6), 57–68.
- Stanton, N., Hedge, A., Hendrick, H. W., Salas, E., & Brookhuis, K. (2004). *Handbook of human factors and ergonomics methods* (1st ed.). Boca Raton, FL: CRC Press.
- Stanton, N. A., Salmon, P. M., Walker, G. H., Baber, C., & Jenkins, D. P. (2005). *Human factors methods: A practical guide for engineering and design*. Aldershot, England: Ashgate Publishing Company.
- St-Cyr, O., & Kilgore, R. M. (2008). A five-phase CWA for air traffic control: Example from a TRACON microworld. Paper presented at the *Human Factors and Ergonomics Society Annual Meeting Proceedings*, New York, NY, 52(4) 272–276.

- Stoner, H. A., Stelzer, E. M., Wiese, E. E., Paley, M., Darowski, A., Mizrahi, G., & Martin, E. A. (2006). The resource for applied cognitive and systems engineering (trace-se): An approach to understanding the gap. Paper presented at the *Human Factors and Ergonomics Society Annual Meeting Proceedings*, San Francisco, CA, 50(17) 1872–1876.
- Thagard, P. (2004). *Cognitive science*. Retrieved May 20, 2011, from <http://plato.stanford.edu/archives/win2004/entries/cognitive-science/>
- Valerdi, R., Miller, C., & Thomas, G. (2004). Systems engineering cost estimation by consensus. Paper presented at the *International Conference on Systems Engineering*, Las Vegas, NV.
- Vanderperren, Y., & Dehaene, W. (2005a). SysML and systems engineering applied to UML-based SoC design. Paper presented at the *2nd UML-SoC Workshop at 42nd DAC*, Anaheim, CA, 1–8.
- Vanderperren, Y., & Dehaene, W. (2005b). UML 2 and SysML: An approach to deal with complexity in SoC/NoC design. Paper presented at the *Proceedings of the Conference on Design, Automation and Test*, Munich, Germany, (2)716–717. doi:10.1109/DATE.2005.319
- Vicente, K. J. (1995). Task analysis, cognitive task analysis, cognitive work analysis: What's the difference. Paper presented at the *Human Factors and Ergonomics Society Annual Meeting Proceedings*, San Diego, CA, 39(9) 534–537.
- Vicente, K. J. (1999). *Cognitive work analysis : Toward safe, productive, and healthy computer-based work*. Mahwah, N.J.: Lawrence Erlbaum Associates.
- Vicente, K. J. (2000). HCI in the global knowledge-based economy: Designing to support worker adaptation. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 7(2), 263–280.
- Viehl, A., Schönwald, T., Bringmann, O., & Rosenstiel, W. (2006). Formal performance analysis and simulation of UML/SysML models for ESL design. Paper presented at the *Proceedings of the Conference on Design, Automation and Test*, Munich, Germany, 242–247.
- Wang, R., & Dagli, C. H. (2008). An executable system architecture approach to discrete events system modeling using SysML in conjunction with colored petri net.

Paper presented at the *2nd Annual IEEE Systems Conference*, Montreal, Canada, 1–8.

Weilkiens, T. (2008). *Systems engineering with SysML/UML: Modeling, analysis, design*. Burlington, MA: Morgan Kaufmann.

Wickens, C. D., Gordon, S. E., & Liu, Y. (2004). *An introduction to human factors engineering*. New York, NY: Pearson Prentice Hall.

Wölkl, S., & Shea, K. (2009). A computational product model for conceptual design using SysML. Paper presented at the *Proceedings of the ASME 2009 International Design Engineering Technical Conference & Computers and Information in Engineering Conference*. San Diego, 2009.

Woods, D. D., & Roth, E. M. (2005). Cognitive systems engineering. In N. Moray (Ed.), *Ergonomics: The history and scope of human factors* (pp. 59–125). New York, NY: Taylor & Francis.

Wray, R. B., McKinney, D., & Whalen, J. (2000). *Systems engineering handbook: A "how to" guide for all engineers*. Vancouver, Canada: International Council on Systems Engineering.

Zowghi, D., & Coulin, C. (2005). Requirements elicitation: A survey of techniques, approaches, and tools. In A. Aurum, & C. Wohlin (Eds.), *Engineering and managing software requirements* (1st ed., pp. 19–46). Berlin, Germany: Springer.
doi:10.1007/3-540-28244-0_2