AFRL-HE-WP-TR-2001-0125

UNITED STATES AIR FORCE RESEARCH LABORATORY

THE DEVELOPMENT OF A COMPUTER-AIDED COGNITIVE SYSTEMS ENGINEERING TOOL TO FACILITATE THE DESIGN OF ADVANCED DECISION SUPPORT SYSTEMS

Scott S. Potter
William C. Elm

Logica Carnegie Group
Five PPG Place
Pittsburgh, PA 15222

Emilie M. Roth

Roth Cognitive Engineering
89 Rawson Road
Brookline, MA 02445

David D. Woods

The Ohio State University
Institute for Ergonomics
1971 Neil Ave.
Columbus, OH 43210

JUNE 2001

FINAL REPORT FOR THE PERIOD APRIL 1998 TO APRIL 2000

Approved for public release; distribution is unlimited.

Human Effectiveness Directorate
Crew System Interface Division
2255 H Street
Wright-Patterson AFB OH 45433-7022

TECHNICAL REVIEW AND APPROVAL

AFRL-HE-WP-TR-2001-0125

This report has been reviewed by the Office of Public Affairs (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

MARIS M. VIKMANIS
Chief, Crew System Interface Division
Air Force Research Laboratory

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	June 2001	Final, April 1998 - April 2000

4. TITLE AND SUBTITLE
The Development of a Computer-Aided Cognitive Systems Engineering Tool to Facilitate the Design of Advanced Decision Support Systems

5. FUNDING NUMBERS
Contract: F41624-98-C-6008
PE: 62202F
PR: 3005
TA: HC
WU: 84

6. AUTHOR(S)
Scott S. Potter (Logica Carnegie Group)*, William C. Elm (Logica Carnegie Group)*, Emilie M. Roth (Roth Cognitive Engineering), and David D. Woods (The Ohio State University)

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Logica Carnegie Group, Inc.
Five PPG Place
Pittsburgh, PA 15222
USA

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory
Human Effectiveness Directorate
Crew System Interface Division
Air Force Materiel Command
Wright-Patterson AFB OH 45433-7022

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-HE-WP-TR-2001-0125

11. SUPPLEMENTARY NOTES
* Currently with Aegis Research Corporation, Cognitive Systems Engineering Center, 501 Grant Street; Suite 475, Pittsburgh, PA 15219

12a. DISTRIBUTION AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
The Computer-Assisted Cognitive Systems Engineering (CACSE) toolkit provides a methodology and associated software to facilitate the use of human factors/warfighter operational performance requirements data in the design of complex man-machine systems, such as those found in military C4I applications. Specifically, CACSE provides a Cognitive Task Analysis approach and associated documentation of the design thread from operator information processing and decision-making requirements to the information visualizations needed to support these functions. Thus, CACSE will help bridge the gap between the human factors community and the software engineering community for inserting decision support systems in a wide variety of operational domains.

14. SUBJECT TERMS
Cognitive Systems Engineering, Human Factors, Cognitive Task Analysis, System Design

15. NUMBER OF PAGES
192

16. PRICE CODE

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

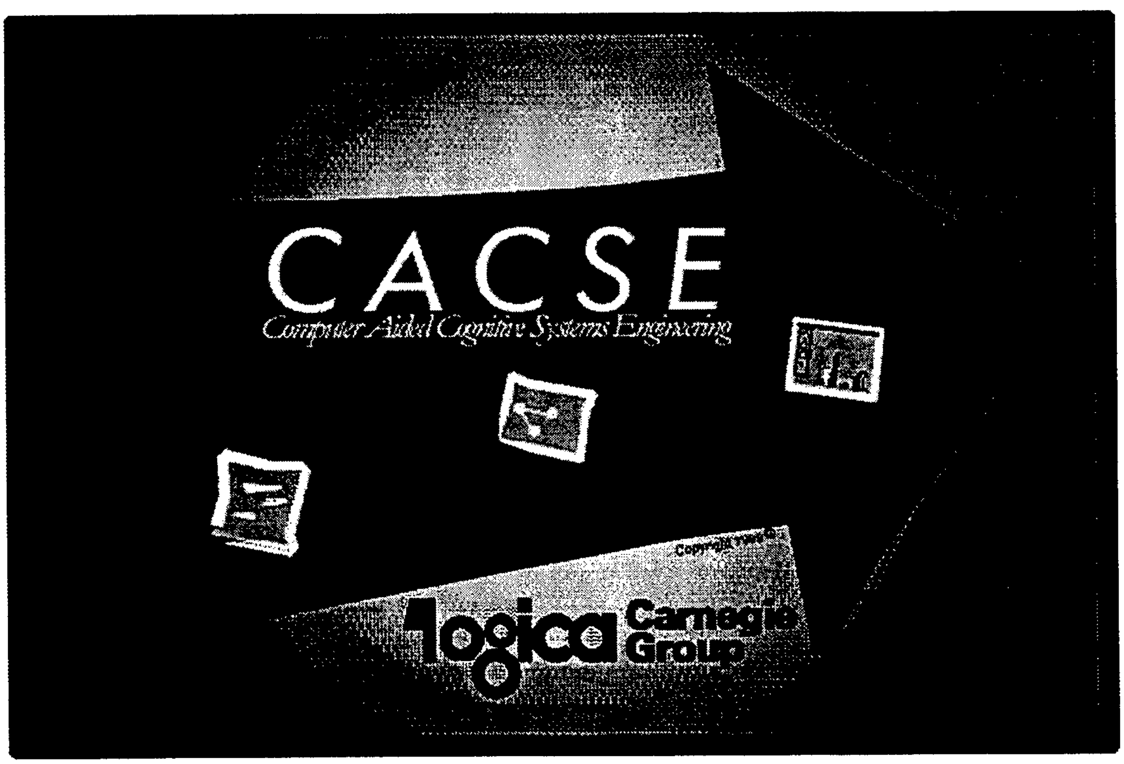
Standard Form 298 (Rev. 2-89) (EG)
Prescribed by ANSI Std. 239.18
Designed using Perform Pro, WHS/DIOR, Oct 94

THIS PAGE INTENTIONALLY LEFT BLANK

logica Carnegie Group

Project Final Report

The Development of a Computer-Aided Cognitive Systems Engineering Tool to Facilitate the Design of Advanced Decision Support Systems



Prepared for:

Air Force Research Laboratory (AFRL/HECI)
ATTN: Dr. Michael D. McNeese
2255 H Street; Building 248
Wright-Patterson AFB, OH 45433-7022

Prepared by:

Logica Carnegie Group
Five PPG Place
Pittsburgh, PA 15222 USA

CDRL-A004

Under Contract No. F41624-98-C-6008

Proposal Title: "Intelligent Cognitive Engineering Suite for Information Warfare Domains"

Document Identifier: SBIR PH II--FINAL REPORT--v3.0

Version Notice

All revisions made to this document are listed here in chronological order.

Document Release	Date	Revision Purpose
Draft Version 1.0	25 October, 1999	Initial Internal Release
Draft Version 2.0	30 November, 1999	Second Internal Release
Final	31 January, 2000	Final report delivered to Customer

The current release of this document is considered a complete replacement of previous versions unless otherwise stated.

Logica Carnegie Group has made every effort to ensure that this document is accurate at the time of printing. Obtain additional copies of this document, as well as updated releases, from Dr. Scott S. Potter, Principal Investigator, at the above address.

Table of Contents

I. Executive Summary

This report describes a line of research working toward an integrated, tool-supported approach to Cognitive Task / Work Analysis (CTA / CWA) as a means of studying cognitive systems in context with the goal of (1) designing support systems (e.g., training system, decision support system) or (2) evaluating human performance/cognition in complex domains. Specifically, it discusses the development of computer-based tool support for CTA with the goal being to have CTA as an integral part of an iterative software/system development process. Thus, this effort is not oriented toward automating knowledge acquisition; rather toward supporting the integration of results from CTA efforts into system development and/or evaluation efforts.

The goal of this report is to document the development of an integrated, Cognitive Systems Engineering (CSE) based software/system development process supported end-to-end by computer-aided tools (as depicted in Figure 1) to build highly effective decision support systems (DSSs) within information warfare, command and control, and other complex military (as well as commercial) applications.

Our tool will be referred to herein as the Computer-Aided Cognitive Systems Engineering (CACSE) tool and will serve as the integration point between insights gained from an apriori CTA and the design artifacts that support the resulting software engineering development activities. This is expected to result in a seamless design database, from fundamental cognitive demands through software design artifacts to the resulting implementation of the DSS. This has the potential to build more effective DSSs, designed specifically to critical cognitive demands.

This CACSE tool supports cognitive engineers in capturing and maintaining the essential cognitive issues and relationships developed through a CTA. It supports a robust CSE methodology adapted from Rasmussen (1986) yet is sufficiently flexible to incorporate results from multiple, complementary CTA approaches into the design data base. This is designed to increase the maturity level of current approaches to Cognitive Systems Engineering and Cognitive Task Analysis – especially the weak or non-existent coupling of CTA results to the software development process.

Moreover, this has begun to be a tool for software developers to maintain awareness of the "design basis" underlying the resulting system requirements and specifications by forming a maintainable, traceable component of the functional design. Thus, our objective was to develop a software-based tool that simultaneously and seamlessly aids the experienced CTA analyst in the modeling and documentation aspects of the CTA process and the equally experienced software developer during the construction of the resulting DSS.

The primary benefit of an integrated, tool-supported process is expected to be the radical advance in the impact of CTA results on the resulting DSS design. The resulting system design, from user interface presentation to underlying representations, all the way to sensor placement will be based on the apriori CSE analysis. To facilitate this, the development environment will be

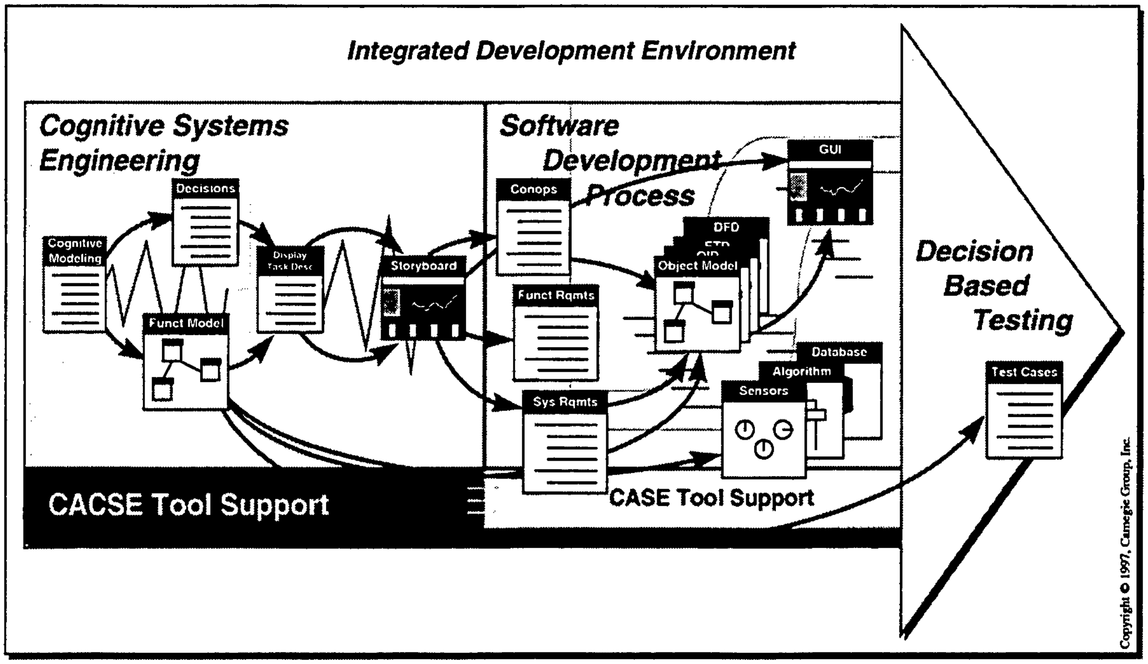


Figure 1. The SBIR vision – an integrated, CSE-based DSS development process where CTA products directly support software development artifacts resulting in an efficient process for effectively supporting cognitive demands identified by the CTA and maintaining these links throughout an iterative design process.

seamless, affordable to maintain, and dramatically improve a spiral, iterative design process. This will be in addition to productivity benefits from the introduction of a CASE-like power tool for improving the efficiency of the CTA process.

A second benefit of taking advantage of advances in computer technology (as has been done for CASE tools) is the potential for collaborative computing support communication and coordination of CTA results among design team members distributed within and across development organizations. System developers using the CACSE-supported process would deliver fully functional DSSs that embody solutions to the cognitive demands of the domain and provide dramatically improved joint (human + DSS) decision-making effectiveness – a critical component of information dominance.

As an approach to this vision, the following sections will include results from the major tasks in this development phase of the project, as they instantiate the requirements defined in the first phase. These include:

- A scenario of use of the CACSE tool, describing current functionality;
- The CSE development process oriented toward supporting software development;
- A case study of the use of CACSE to support visualization design for a Command and Control domain;
- A Users Manual to describe the features of the CACSE tool.

II. SCENARIO OF USE — CACSE IN ACTION

A. Introduction

It is difficult to describe a DSS in functional terms that effectively convey the intent. One way to provide some description of the "look and feel" of such a system is to describe a hypothetical user in front of the system carrying out his or her mission. The user's interactions with the system are representative of the operation of the system, and the usefulness and impact of the system can be visualized. This script is often called a "scenario of use."

In order to visualize the decision support provided to users of the CACSE tool, one sample session is described below. In order to make certain aspects of the system explicit, or to clarify certain points, commentary is added to the basic script in the following manner:

Any commentary will be presented in this style. It is intended to be read in a manner like stage directions in a play; i.e., to add clarity to the actual occurrences presented in the main script. This style will be used to clarify points about the system, or to point out differences between various options. This is particularly important where the "user" in the scenario takes one particular action when several alternatives will be possible in the actual system. The alternative options can be reviewed in this style, while the primary option will be the one chosen by our user.

Note: the following sections are based on current functionality of Operational Prototype (OP) 3.0 of CACSE (delivered 31 Jan 2000). This scenario is intended to convey the end result of this development phase, while at the same time point to potential enhancements. The performance of the system described here exists today, but the actual performance of any implemented system may differ considerably based on users' desires and other factors. This scenario is intended primarily to help the reader envision the kinds of support possible from the current system.

B. The Characters:

WP-CSE: The Wright Patterson Air Force Base Research Lab Cognitive Systems Engineers. This is a notional target user base, with an established stable of CSE literate, practiced engineers; they have become part of the USAF's program development process. They routinely get asked to perform the initial CSE for a new program. They are part of a larger system development team.

WP-SW: The software engineering team assigned to create the actual DSS.

WP-PM: The Program Management office charged with building the system.

C. The Starting Point

This scenario is designed around a fictitious Information Warfare development project to demonstrate potential use within this domain and to avoid any use of classified data required to accurately depict real IW applications.

The WP-CSE Engineers have just started a new project. As part of a larger USAF program to develop an "Information Warfare Counterstrike Command Center" (IW-CCC), the WP-CSE designers have been asked to define a remarkably challenging DSS design. They received a system "mission statement" from the PM:

> "Develop a suitable command center that provides situational awareness and supervisory decision-making for monitoring the progress of an enemy attack on computers, software, and networking, the automatic counterstrike deployments launched by the software protection systems built into the systems, and the high level supervisory decisions which must be made by the Shift OIC of the IW-CCC."

While there has been extensive work on C^2 centers, network operations centers, etc., the WP-CSE team is breaking some new ground – the IW fight unfolds extremely rapidly, placing human decision makers at an incredible disadvantage, under incredible time pressures, and highly reliant on automatic agents to make what seem to be instantaneous decisions that have a substantial impact on the friendly operations themselves. The supervisory decisions to isolate portions of networks can have very disruptive effects on friendly operations. The decisions are mission critical, expensive, and often have no clear "right answer"... acting too late increases enemy damage... acting too soon/or when not warranted creates damage of its own on the friendly information systems.

D. CACSE Comes Online

The WP-CSE team has spent considerable time learning the basics of the IW domain. From discussions with a couple of subject matter experts (SMEs) that not only understand the issues, but also have the ability to explain it, the WP team has some ideas for the beginnings of the functional structure.

The CACSE tool now facing them was built to support the underlying science arising from the work of Rasmussen (c.f., Rasmussen, 1986; Woods and Hollnagel, 1987; Rasmussen, Pejtersen, and Goodstein, 1994; Roth and Mumaw, 1995; Vicente, 1999). It does not replace their engineering judgment and skills, but it provides never before possible support features to improve the productivity and quality of their efforts. It is a DSS designed to aid the design of DSS's. Like any DSS it supports a knowledgeable professional, it does not replace the human in the loop.

It is important to emphasize that the goal of the CACSE tool is NOT to create a tool that automates and simplifies the application of CTA methods so that people with minimal expertise *can use them, but to develop software tools that aid the experienced CTA analysts in the modeling and documentation aspects of the CTA process to yield a more useful product that makes direct contact with the software development process and supports communication and coordination of CTA results among design team members distributed within and across development organizations. (Just as a good CAD tool does not make a novice into an architect.)*

E. The CACSE power tool pays off

The WP-CSE team comes together for an initial design meeting. One member has volunteered to fly the CACSE tool, now projected on the wall to help the discussion along. During this meeting, an initial concept map built by one of the team members is used as the starting point and a functional structure begins to take rough shape.

This functional abstraction hierarchy (FAH) is one of the initial design artifacts, forming the "skeletal structure" of the information organization to come. It will drive even the navigation between screens in the resulting GUI (Graphical User Interface) far downrange in the design process.

As the discussion progresses, the Functional Goals of the IW-CCC domain (and their inter-relationships) begin to be recorded and organized in the FAH. The functional structure grows and shifts on the wall as the discussion progresses. Over the next few days, the key CSE engineers come to a consensus on a reasonable initial design for the functional structure.

The CACSE tool provides the "CAD-like" features it needed to define the nodes in the functional structure. They are automatically numbered for easy reference by the CACSE tool. The editing features: to add names, annotations, the "is composed of" decomposition relations, and other types are key components of this initial CSE step. Similar in many ways to the features of a software engineering team's CASE environment's support for defining OMT-style objects, the CACSE features' heritage is evident. By reusing many of the same techniques, CACSE was able to exploit many of the lessons learned from the CASE tool developers... getting a jump start on a mature system.

Major Williams, a WP-CSE team member, leaves the meeting with a plotter printout of the Abstraction Hierarchy that has an area highlighted. This is his portion of the IW-CCC domain to complete the CSE design thread from here on out. As he returns to his workspace and logs back in to CACSE, he notices several other team members have already begun expanding the analysis. It's easy to see from the overall view provided by the tool. Maj. Williams gets started by tailoring a view to show just the functional goal nodes within his area.

As the work progresses, Maj. Williams adds additional depth and breadth to the structure. As he works near the 'edge' of his assigned piece of the structure, he opens a "read only" copy of a functional goal object from Captain Smith, another WP-CSE member. Since there was an explicit relation between the two goals, Maj. Williams needs to understand the two perspectives to make sure they come together into one seamless overall design. He gets a read only copy since Captain Smith has that object open as part of his design effort.

The current version of the tool has implemented "ownership" locking to serve as a starting point for a distributed / collaborative work environment. SW Engineering experiences have indicated that this is too restrictive, as it really impacts the productivity of the team to have to wait for an "owner" to make changes. The ideal solution (for future versions) will be to control the access administratively. Such collaboration features are critical to having multiple CSE engineers working in parallel, as well as working together with the multiple SW engineers. Collaborating across the disciplines depends even more heavily on the features of the design environment.

F. Integration/Tailoring of the Methods

Maj. Williams has done CSE on a couple other systems already. He was trained as part of the WP-CSE team to dramatically improve the effects of the massive amount of C2 software being purchased by the USAF. The CSE methodology being used for the IW-CCC is a 'second generation' method, adapted and improved from the original. The "first generation" CSE method was very effective at doing CSE, but wasn't exactly suited to an on-line, tool-supported, version. More importantly, the first generation method didn't couple well with the software engineering design products. When it came time to build the run time DSS the SW engineers had to rework the complete design, and then couldn't keep the two perspectives for very long. The second generation methodology actually changed both the software engineering and CSE methodologies to "meet in the middle".

Maj. Williams' piece of the design included one of the top-level decisions: "Monitor the automatic activation of Software Counterattack measures." Supervisory, high level displays like this are where CSE really adds value (the simple data displays at the bottom of the abstraction hierarchy are much more straightforward, even though they still profit from good HCI visualization practices). So far, Maj. Williams helped define the Functional Node: "Employ IW Counter Attack Measures." For that node he then added the following key decisions: "Monitor automatic actuations," "Detect failure to automatically initiate," "Manually disable automatic initiation," and "Manually initiate countermeasures." (This functional node is a supporting goal beneath the higher level goals related to "Monitoring the success of countermeasures.") The Functional Goal "Employ counterattack measures" is visible on the screen. It shows explicit links to each of the Goals listed above. Maj. Williams selects the goal node, allowing both functional processes and decisions to be visible.

Maj. Williams selects the "Monitor automatic actuations" decision object. There are some design rationale notes he added earlier, as well as a couple inserted by Captain Smith about how it relates to some similar 'monitor automatic...' things in another part of the system. All of these 'monitor automatics...' are specializations of the more general 'monitor automatic triggers" which in turn are specializations of 'monitor' decision object. These generalizations were reused from a library of CSE templates the WP-CSE team maintains. This library is routinely offered to other CSE practitioners across the USAF.

This component library is anticipated to be a part of future versions of CACSE, as a case-study library is built up. This is an example of the influence of object-oriented methodology on

CACSE design. Parent-child relationships between objects and inheritance is a major advantage of the OO approach. Exploiting similar relationships forms the basis of the CSE templates.

After selecting the "Monitor Automatic Actuations" decision, Maj. Williams adds an object for Information Requirements, giving it the object name of "CounterAttack Method X, Trigger Logic." Inside that object he completes the attributes:

- Description – additional descriptive text;
- Rationale – underlying reason for the identification of this information requirement;
- Supporting Data Elements – the data that comprise this information requirement.

There can be many of these information requirement objects for a particular decision, as it is a one to many relation.

G. Transition from Modeling to Visualization Design

After some time, Maj. Williams has been able to fairly well complete the front-end of the "Design Thread" for his piece of the design – defining goal / process nodes, annotating decisions onto their home within this FAH structure, and specifying information requirements for each of these decisions. There are still missing data elements for several of these information requirements, but he will fill those gaps after meetings with the WP-SW group next week. This does not hinder his ability to make progress in CACSE, however – nodes below "decisions" (information requirements and data elements) are not mandatory for completing the design thread.

Throughout this analysis-centered period, there have been (as is the case with any design project) visualization concepts proposed by different members of the WP-CSE team. Maj. Williams has collected and saved these design concepts (putting together quick mock-ups in his favorite drawing application). Now, in CACSE, he has created a representation of these design concepts and linked them to their respective files in his drawing application. Working in CACSE, he links the visualizations together to represent the workspace layout.

CACSE is not designed to replace COTS tools for drawing and mocking up prototypes. It can, however, in OP 3.0 provide direct links between the supporting design thread and the resulting instantiation of the visualization concept.

Now that the front-end of the design thread is fairly well complete, Maj. Williams starts a more disciplined design process. He starts thinking about reasonable boundaries for groups of decisions that will be supported by the visualization designs. He then "attaches" these decisions to elements in the display space. These connections between decisions and display elements define the context and information requirements for the display elements. Now, he has a specification (all the information requirements and supporting data elements for each display element) to use as his basis for visualization design. In a couple cases, the early sketches serve as an excellent starting point for the solution. Using the information requirements, Maj.

Williams easily identifies necessary modifications to existing visualizations. In other cases, none of the display concepts address specified information requirements, so Maj. Williams begins working from scratch to lay out ideas.

CACSE uses a portion of the display space model of Woods (visualizing function book) in which the display workspace is composed of integrated process views that are further decomposed into graphical forms. Decisions can be attached to a process view when it is the collection of graphical forms that, in parallel, address that particular decision requirement.

H. Closing the loop — Design Checking forward and back

The GUI Lead engineer from the WP-SW team is working on the software already. It's typical for the schedule compression to require greater than ideal parallelism. Based on Maj. Williams' work, CACSE has generated the basic GUI structure based on the unified CSE-SW Engineering methodology (exploiting the 1:1 relationship between DTD and GUI Window Object). The GUI SW Engineer picks up the design "on the fly" and begins to expand the software particular aspects of the design.

The critical issue here is that the products of the CTA "end" of CACSE directly support the software developers in their modeling, design, and development processes.

As the IW-CCC initial version is completing system testing, the WP-CSE team receives an alpha release to begin decision-centered testing. The CSE team had to wait until the software passed unit and system testing, when it was stable enough to be stressed with real users in realistic, decision-centered tests. Before the use of CSE in system development, the testing of a DSS ended when the software team completed system testing, verifying that the software performed the functional requirements. Now that's the starting point for a whole new level of testing: Does the DSS, captured in those functional requirements, help human decision-making effectiveness? Have we built the larger Human-DSS system well?

During the execution of the decision-centered test case for "Monitor Automatic Actuation," five out of six shift supervisor test subjects expressed some confusion with the display, and the sixth misinterpreted the information being presented. In reviewing the verbal protocols from these test subjects and in post-test review questioning, the WP-CSE team immediately recognized a classic cause. The team realized that the functional map was missing a component of the domain's CounterAttack options. Because it had been omitted from the model, the IW-CCC shift supervisors were not provided with a key piece of information about that particular option, and in searching for it, lost track of the decision they were working. They dedicated too much cognitive "work" to the search.

Maj. Williams goes back to the functional model and inserts the new object into the functional abstraction hierarchy. Each of the impacts on functional goal objects, and their associated decision objects, information needs, and GUI objects reflect the need to update these objects to restore the overall CSE-SW design database to a consistent condition. Maj. Williams has responsibility for all the CSE objects and begins to 'walk' the changes across the Decision objects, Information Requirements objects, and to the Display objects. He uses the top-down and bottom-up query feature to identify functional goals that are not supported by displays, decisions that have become "orphaned" (not attached to functional goals) or detached from displays. As the size of the model and supporting artifacts becomes large, these query features become invaluable.

He is sure his counterpart on the SW team will be working on the necessary changes to the software Object Model to reflect the changes. Already, the CSE testing team has put Maj. Williams' storyboard of the new display to a part-scale test with two of the test subjects... they got the key decision right this time. These results are entered into the CACSE design database on the appropriate storyboard object.

This kind of consistency / completeness checking is common in CASE environments. CACSE will broaden the capability to include the state of the CSE design. This will prove to be a major benefit of CACSE – supporting the maintenance of CTA models during an iterative design process.

I. Summary

This scenario has demonstrated many of the potential benefits of the development of an integrated, Cognitive Systems Engineering based software development process supported by the CACSE tool. As indicated herein, this tool will generate a seamless design data base, from fundamental cognitive demands through software design artifacts all the way to decision effectiveness test cases. This will result in a radical advance in the impact of CTA results on the resulting DSS design.

III. CSE DEVELOPMENT PROCESS ORIENTED TOWARD SUPPORTING SOFTWARE DEVELOPMENT

A. Introduction

The goal of this section is to lay out a CTA process that orchestrates different types of specific CTA techniques in order to provide design relevant CTA results and integrate CTA results into the software development process. Then, we can describe how our CACSE tool supports this CTA process, provides support for the CTA modeling activities, and provides results that can integrate into the software development process. A major insight that emerged from the development of this CSE framework is that in order to have maximum impact on the software engineering process, what is needed are not only software tools to automate aspects conducting a CTA or increase its speed or efficiency, but also coordinated methodologies and tools that would facilitate integration of the insights gained from conducting the CTA into the software development process. The results of this effort have provided the initial target for supporting the difficult and far from seamless transition from an isolated CTA effort to an integral and critical part of developing useful as well as usable computer systems.

To support development of computer based tools intended to aid cognition and collaboration we have found, first of all, that CTA is more than the application of any single CTA technique. Instead, developing a meaningful understanding of a field of practice relies on multiple converging techniques in a bootstrapping process. This bootstrapping process has been used to model cognition and collaboration and to develop new online support systems in time pressured tasks such as situation assessment, anomaly response, supervisory control, and dynamic replanning across domains such as military intelligence analysis (Potter et al., 1997), military aeromedical evacuation planning (Cook, et al., 1996; Potter et al., 1996; Walters, et al., 1996), military command and control (Shattuck and Woods, 1997), commercial aviation (Sarter and Woods, in press), cardiac anesthesia (Cook and Woods, 1996), space shuttle mission control (Watts, et al., 1996), and nuclear power plant emergencies (Roth et al., 1997).

Our approach to CTA is best depicted in Figure 3. The left side of this figure is intended to convey how CTA is an iterative, bootstrapping process focused on understanding both the domain (mapping the cognitive demands of the fields of practice) and practitioners (modeling expertise and cognitive strategies) through a series of complementary (empirical and analytical) techniques. As indicated by the right side of Figure 3, the CTA process continues into the design/prototype development process. The CTA model (the output of the left side) becomes the initial hypothesis for artifacts embodied in the design prototypes which in turn are used to discover additional requirements for useful support (Woods, in press).

Critical issues addressed by this framework include the need for:

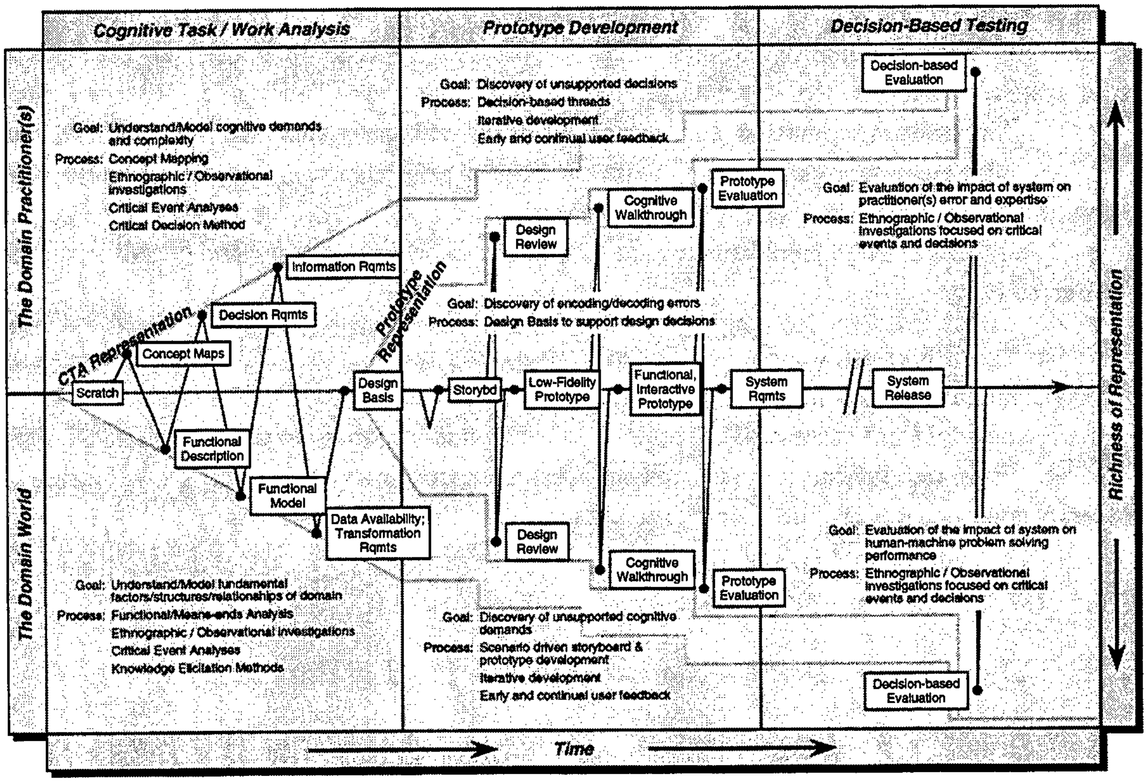


Figure 2. Detailed depiction of an integrated approach to Cognitive Task Analysis within an iterative system development process. Phases within the CTA process are represented by vertical columns and the domain world / practitioner distinction (within the field of practice) is represented by the horizontal rows. Time is on the abscissa and growth of understanding is on the ordinate. CTA products/artifacts are represented by the nodes along the activity trajectory.

- Multiple, coordinated approaches to CTA. No one approach can capture the richness required for a comprehensive, insightful CTA. However, in an iterative manner, a set of approaches can successively (and successfully) build the required understanding.
- Analytical and empirical evidence to support the CTA. Analytical models need to be refined and verified through empirical investigations. Only through bootstrapping complementary approaches can a robust CTA model be developed.
- Tangible products from CTA that clearly map onto artifacts used by system designers. CTA must work within a system development process and support critical system design issues. This point is expanded in the second phase of this effort.
- The use of prototypes as tools to discover additional CTA issues. CTA cannot be viewed as a standalone, apriori analysis. It needs to be an iterative process that learns from subsequent design activities. The initial CTA model as well as resulting support concepts are hypotheses that are at best incomplete and must be tested within the demands of the application domain.

The second critical issue is that, since CTA is a means to support the design of computer-based artifacts that enhance human and team performance, CTA must be integrated into the software and system development process. Thus, Section V presents two case studies of conducting a CTA within military command and control decision support system environment and then working with the system development team on designing the system around the insights gained from the CTA effort. This phase is from the perspective of a CTA-based system development model and is expected to help formalize the requirements for any CTA method or tool to support design activity.

As part this SBIR effort, we have developed a process that orchestrates different types of specific CTA techniques to provide design relevant CTA results and integrates CTA results into the software development process. The results of this effort is beginning to provide the initial target for supporting the difficult and far from seamless transition from an isolated CTA effort to an integral and critical part of developing useful as well as usable computer systems.

B. CTA as a Modeling Process

1. Uncovering Cognitive Activities in the Field of Practice

In trying to understand and evaluate CTA methods and when to use them, it is important to focus on the goals of CTA. CTA is fundamentally an *applied* activity. The goal of CTA is not to capture domain practitioner expertise as an end in itself. CTA is a means to a larger end of specifying ways to improve human and team performance in the domain (be it through new forms of training, user interfaces or decision-aids). From this perspective CTA is best thought of as a process for uncovering the cognitive activities entailed by the field of practice and identifying opportunities for more effective support.

In performing CTA, two mutually reinforcing perspectives need to be considered (as depicted by the two "dimensions" on the ordinate axis in Figure 3 and also presented in a more abstracted manner in Figure 3). One perspective focuses on the fundamental characteristics of the domain and the cognitive demands they impose. The focus is on understanding the way the world works today and what factors contribute to making practitioner performance challenging. Understanding domain characteristics is important both because it provides a framework for interpreting practitioner performance (why do experts utilize the strategies they do? What complexities in the domain are they responding to? Why less experienced practitioners perform less well? What constraints in the domain are they less sensitive to?) and because it helps define the requirements for effective support (what aspects of performance could use support, what are the hard cases where support could really be useful) as well as the bounds of feasible support (what technologies can be brought to bear to deal with the complexities inherent in the domain, which aspects of the domain tasks are amenable to support, and which are beyond the capabilities of current technologies).

The second perspective focuses on how today's practitioners respond to the demands of the domain. Understanding the knowledge and strategies that expert practitioners have developed in

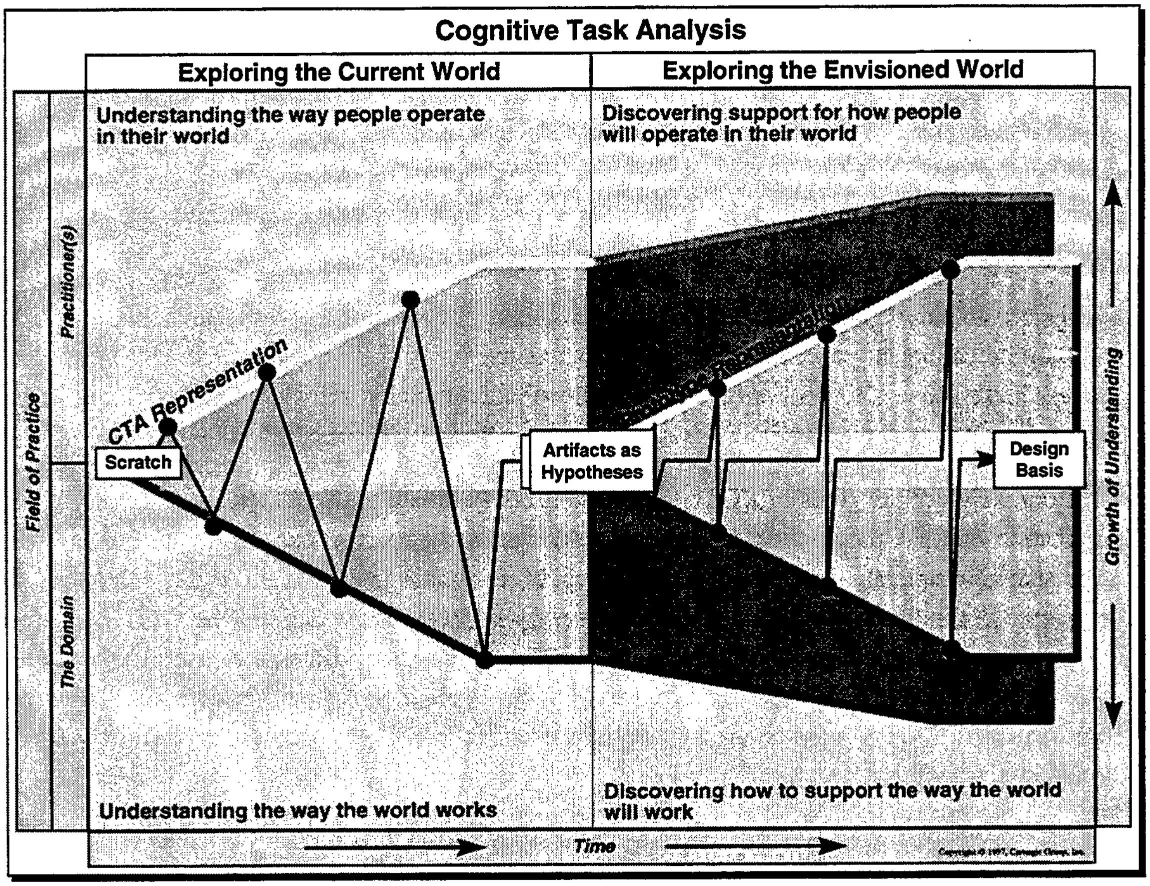


*Figure 3. Abstracted version of the integrated approach to Cognitive Task Analysis within an iterative system development process. The first phase – Exploring the Current World – focuses on understanding the way the world works and the way people operate in this world. The second phase, then – Exploring the Envisioned World – focuses on discovering how to support the way the world **will** work and for the way people **will** operate in this envisioned world.*

response to domain demands provides a second window for uncovering what makes today's world hard and what are effective strategies for dealing with domain demands. These strategies can be captured and transmitted directly to less experienced practitioners (e.g., through training systems) or they can provide ideas for more effective support systems that would eliminate the need for these compensating strategies. Examining the performance of average and less experienced practitioners is also important as it can reveal where the needs for support are.

A comprehensive CTA needs to encompass analysis and modeling of both the domain and how domain practitioners respond to it. The focus is on building a model that captures the analysts' evolving understanding of demands of the domain, knowledge and strategies of domain practitioners, and how existing artifacts influence performance that can be used to guide specification of requirements for improved performance.

In selecting and applying CTA techniques the focus needs to be on the products to be generated from the techniques rather than on the details of the method. Some families of methods focus more on uncovering specific domain expertise, and other families of methods focus more on analyzing the demands of the domain. In performing a CTA it is important to utilize a balanced suite of methods that enable both the demands of the domain and the knowledge and strategies of domain experts to be captured in a way that enables clear identification of opportunities for improved support. The goal, however, must be to model the interconnections between the demands of the domain, the strategies and knowledge of practitioners, the cooperative interactions across human and machine agents, and how artifacts shape these strategies and coordinating activities, so as to generate concepts for more effective support.

The goal is to aid cognitive task analysts in developing CTA products that make contact with the software development process. The goal is to become integrated into the software development process and to have CTA viewed as on the software development critical path. Products of CTA should feed into system developments requirement and quality assurance testing criteria.

Thus our vision of the goals of a CTA tool suite is not to create tools that necessarily simplify the application of CTA methods so that people with minimal expertise can use them, nor to create tools that dramatically increase the efficiency of the CTA methods so that a CTA can be completed dramatically faster, but rather to develop software tools that aid the CTA analysts in the modeling and documentation aspects of the CTA process to yield a more useful product that makes direct contact with the software development process and supports communication and coordination of CTA results among design team members distributed within and across development organizations.

2. An Opportunistic Bootstrap Process

CTA is fundamentally a modeling process focused on understanding the domain (the cognitive demands of the fields of practice) and how practitioners cope with domain demands (modeling expertise, cognitive strategies, errors) with the objective of generating concepts for support. A question arises as to how to start the process and how to gather the knowledge necessary to evolve this model.

CTA is best conceptualized as a bootstrap process. While Figure 2 and Figure 3 provided an overview of this process, Figure 4 illustrates additional details of this idea. One starts from an initial base of knowledge regarding the domain and how practitioners function within it (often very limited). One then uses a number of CTA techniques to expand on and enrich the base understanding and evolve a CTA model from which ideas for improved support can be generated. The process is highly opportunistic. Which techniques are selected, whether one starts by focusing on understanding the domain or by focusing on the knowledge and skills of domain practitioners, depends on the specific local pragmatics. The key is to focus on evolving and enriching the model as you go to ultimately cover an understanding of both the characteristics of the domain and an understanding of the way practitioners operate in the domain. This means that techniques that explore both aspects will most likely need to be sampled, but where one starts, and the path one takes through the space will depend on what is likely to be most informative and meet the local constraints at any point in time.

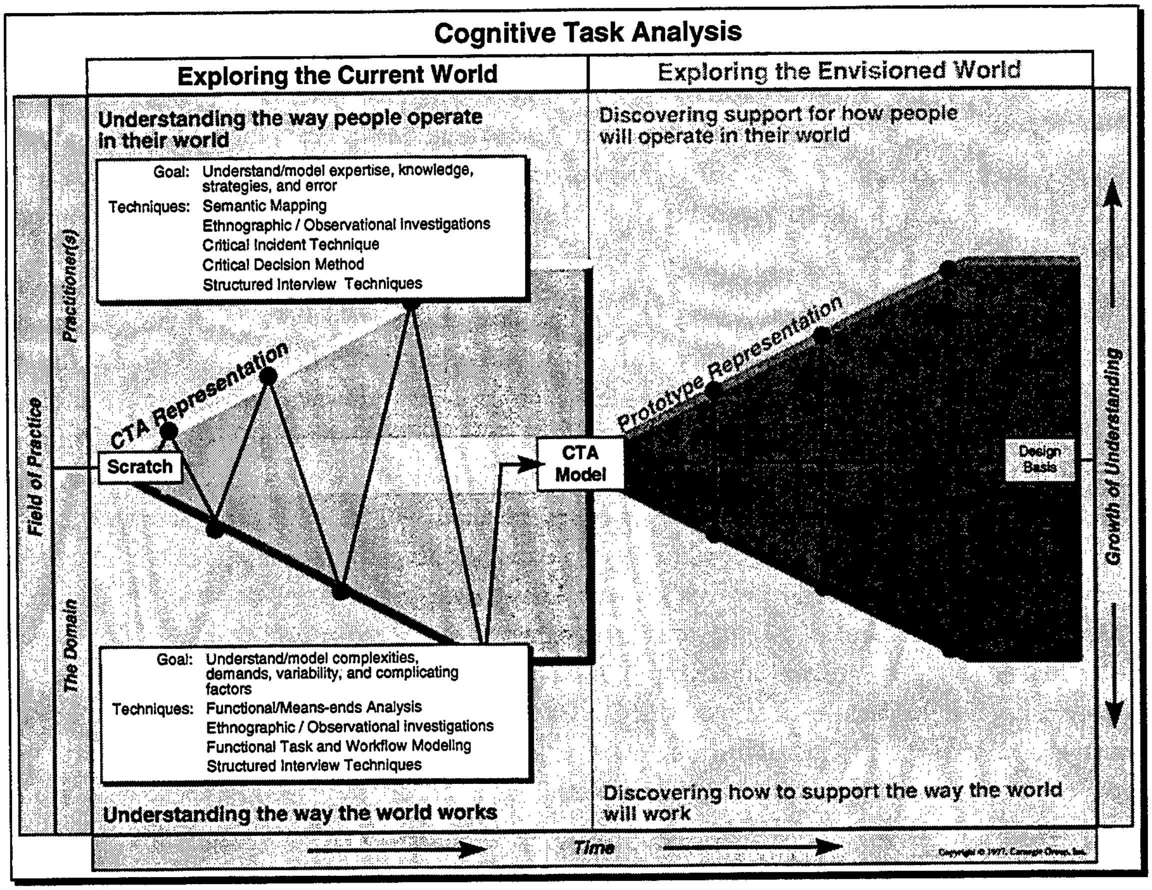


Figure 4. Detailed depiction of the first phase of an integrated approach to Cognitive Task Analysis within an iterative system development model. The critical issue is the mutually reinforcing analyses that, in combination, work toward an understanding of the practitioner(s) and the domain.

The phrase 'bootstrapping process' is used to emphasize the fact that the process builds on itself. Each step taken expands the base of knowledge providing opportunity to take the next step. Making progress on one line of inquiry (understanding one aspect of the field of practice) creates the room to make progress on another. For example, one might start by reading available documents that provide background on the field of practice (e.g., training manuals, procedures), the knowledge gained will raise new questions or hypotheses to pursue that can then be addressed in interviews with domain experts, it will also provide the background for interpreting what the experts say. In turn, the results of interviews may point to complicating factors in the domain that place heavy cognitive demands and opportunities for error. This information may provide the necessary background to create scenarios to be used to observe practitioner performance under simulated conditions or to look for confirming example cases or interpret observations in naturalistic field studies.

In Phase I of this effort, we reviewed a number of specific techniques that have been offered to support the CTA modeling process. The techniques vary on a number of dimensions. Some of the techniques focus primarily on eliciting knowledge from domain practitioners. Other techniques focus more on understanding the inherent characteristics of the domain. Some of the techniques are empirical, involving observations or interviews of domain experts. Others are analytic involving reviews of existing documents (training manuals, procedures, system drawings). Some techniques involve structured interviews outside the context of practice (e.g., in a conference room); others involve observations in realistic work contexts (e.g., ethnographic methods, simulator studies). We believe that all these techniques are valid, and can contribute to the CTA modeling process.

The selection of which technique(s) to use and how many techniques to employ should be motivated by the need to produce a model of the field of practice and how domain practitioners operate in that field. In practice the modeling process generally requires the use of multiple converging techniques that include techniques that focus on understanding the domain demands as well as techniques that focus on understanding the knowledge and strategies of domain practitioners. The particular set of techniques selected will be strongly determined by the pragmatics of the specific local conditions. For example, access to domain practitioners is often limited. In that case other sources domain knowledge (e.g. written documents) should be maximally exploited before turning to domain experts. In some cases observing domain experts in actual work practice (using ethnographic methods or simulator studies) may be impractical, in those cases using structured interview techniques (such as concept mapping) and critical incident analysis may be the most practical methods available. Still in other cases domain experts may not be accessible at all (e.g., in highly classified government applications), in those cases it may be necessary to look for surrogate experts (e.g., individuals who've performed the task in the past) or analogous domains to examine.

It should be stressed that studying the practitioner vs. the domain are merely different access points that provide complementary perspectives. We present them here as distinct to stress the importance of considering both perspectives, but in practice the lines are not so clearly drawn. It is possible to uncover characteristics of the domain through interviews with domain practitioners or field observations. It is also possible to gain perspective on expert strategies by understanding the affordances provided by structural characteristics of the domain.

A point to emphasize is the CTA process is necessarily opportunistic and involves the use of multiple converging techniques. As a heuristic, if resources are limited, it is likely to be more effective to utilize several techniques that sample from both portions of the space (analysis of the domain and analysis of practitioner) even if done cursorily, that to expend all resources utilizing one technique. Unexpected complexities and surprises are more likely to be uncovered when multiple techniques are employed than when the focus is on only on one technique. When the results using several techniques reinforce each other and converge, it increases confidence in the adequacy of understanding. If differences are found it signals the need for a deeper analysis.

A second point to emphasize is that the goal of the CTA is to develop a *productive* model that points to contributors to performance difficulty, opportunities for improved performance and concepts for aiding. The focus of the CTA throughout the process must be on developing concepts related to the goal of the project/system. If the goal is to develop support systems then the focus needs to be on disentangling inherent complexities in the domain that system needs to deal with, from more superficial aspects that result from characteristics/limitations of existing artifacts. This requires differentiating features of the existing environment and essential artifacts that need to be preserved from non-critical features that can be changed or eliminated.

C. Tool Support for the CTA Modeling Process

1. Understanding the Way the World Works

As a foundation for this tool-development effort, the Logica Carnegie Group team has extended the state-of-the-art in cognitive systems engineering by adapting the function-based cognitive task analysis (FB-CTA) methodology to the CPOF domain environment. The FB-CTA output artifacts serve as the unifying framework for the integration of decision-centered design methodologies. The FB-CTA approach has roots in the formal, analytic goal-means decomposition method pioneered by Rasmussen and his colleagues as a formalism for representing cognitive work domains as an abstraction hierarchy (Rasmussen, 1985, Lind, 1993). This approach centers on a formal analysis of the application domain to uncover the inherent cognitive demands together with an assessment of the range and complexity of decisions facing the crewmembers (e.g., Rasmussen, 1986; Rasmussen, Pejtersen & Goodstein, 1994; Roth & Mumaw, 1995; Vicente & Rasmussen, 1992; Vicente, 1995; Woods & Hollnagel, 1987). It has been shown that an explicit functional abstraction model closely parallels the mental models of some of the best human problem solvers, faced with high-stress, high-value, uncertain (naturalistic) decision making conditions. This representation is robust enough to support analysis of a variety of decision making strategies, from Recognition Primed Decision Making (RPD) (Klein, et.al. 1989) through knowledge based strategies for new situations (Vicente & Rasmussen, 1992).

This analysis begins with a function-based goal-means decomposition of the system. High-level goals, such as impacting a critical function, are decomposed into supporting lower-level subgoals. The objective of performing this analysis is to develop a structure that links the purpose(s) of individual controllable entities with the overall purpose of the system. This includes knowledge of the system's characteristics, and the purposes or functions of the specific entities. The result of the first phase of this process is a functional abstraction hierarchy – a multi-level representation of the structure of the work domain defined as a recursive means-ends relation between levels. For example in the case of engineered systems, such as a process control plant, functional representations are developed that characterize the purposes for which the engineered system has been designed, and the means structurally available for achieving those objectives. In the case of military command and control systems, the functional representations characterize the functional capabilities of individual weapon systems, maneuvers, or forces and the higher level goals relate to military objectives.

For process control applications, five levels of abstraction have been found to be useful: functional purpose – the purposes for which the system was designed; abstract function – the intended causal structure of the system; generalized function – a description of the system in terms of standard functions that instantiate the abstract functions above; physical function – the components and their interconnections that carry out the system functions; physical form – the appearance and physical location of components (Rasmussen, 1986). Current research efforts have begun to adapt these same techniques to representing the critical command functions.

Typically, this function-based representation is used to organize the critical domain knowledge and relationships to enable the person-machine interface system design team to identify and answer questions regarding the information and display requirements to support operator monitoring and control (an off-line, manual tailoring of the information presentation workspace). The FAH can be used to:

- serve as the framework for inferring decision context;
- identify critical information requirements for supporting the current decision context;
- identify critical collaboration regions;
- map and identify the assigned roles and responsibilities of crew members
- identify requirements for the human-system interfaces to facilitate effective performance;
- provide a framework for interpreting and evaluating individual as well as team operational performance data.

An example of such a work environment might be a commander of air combat power who has the objective of conducting a Successful Ground Attack with Kinetic Weapons (i.e., munitions). The commander has three types of air borne resources available, which are functionally different, with which to accomplish this objective. These are 1.) close air support resources, 2.) air interdiction resources, and 3.) strategic attack resources. The close air support resources are aircraft capable of low altitude flight equipped with munitions and using tactics which are effective against enemy ground forces that are in close proximity to friendly ground forces. The air interdiction resources are aircraft whose flight characteristics, weaponry, and tactics make them effective against enemy ground forces which are not in close proximity to friendly ground forces. The strategic attack resources are aircraft whose flight characteristics, weaponry, and tactics make them effective against enemy ground forces without the expectation or need of friendly ground forces ever occupying or being near the area of attack, e.g., the recent all-air campaign in Kosovo.

Each of these three types of air borne resources can use two functionally different types of munitions, those which are guided and those which are not guided. Therefore, supporting goals are Successful Ground Attack Engagements with Guided Kinetic Weapons and, similarly, Successful Ground Attack Engagements with Unguided Kinetic Weapons. Each of these supporting goals has its own resources which can be used individually or possibly in combination to achieve the goal. For the Guided Kinetic Weapons, examples of these resources include Electro-Optical Bombs, Electro-Optical Missiles (e.g., Hellfire, IR, Maverick), laser Guided Bombs, and Navigation Guided Bombs. Examples of the Unguided Kinetic Weapons goal's

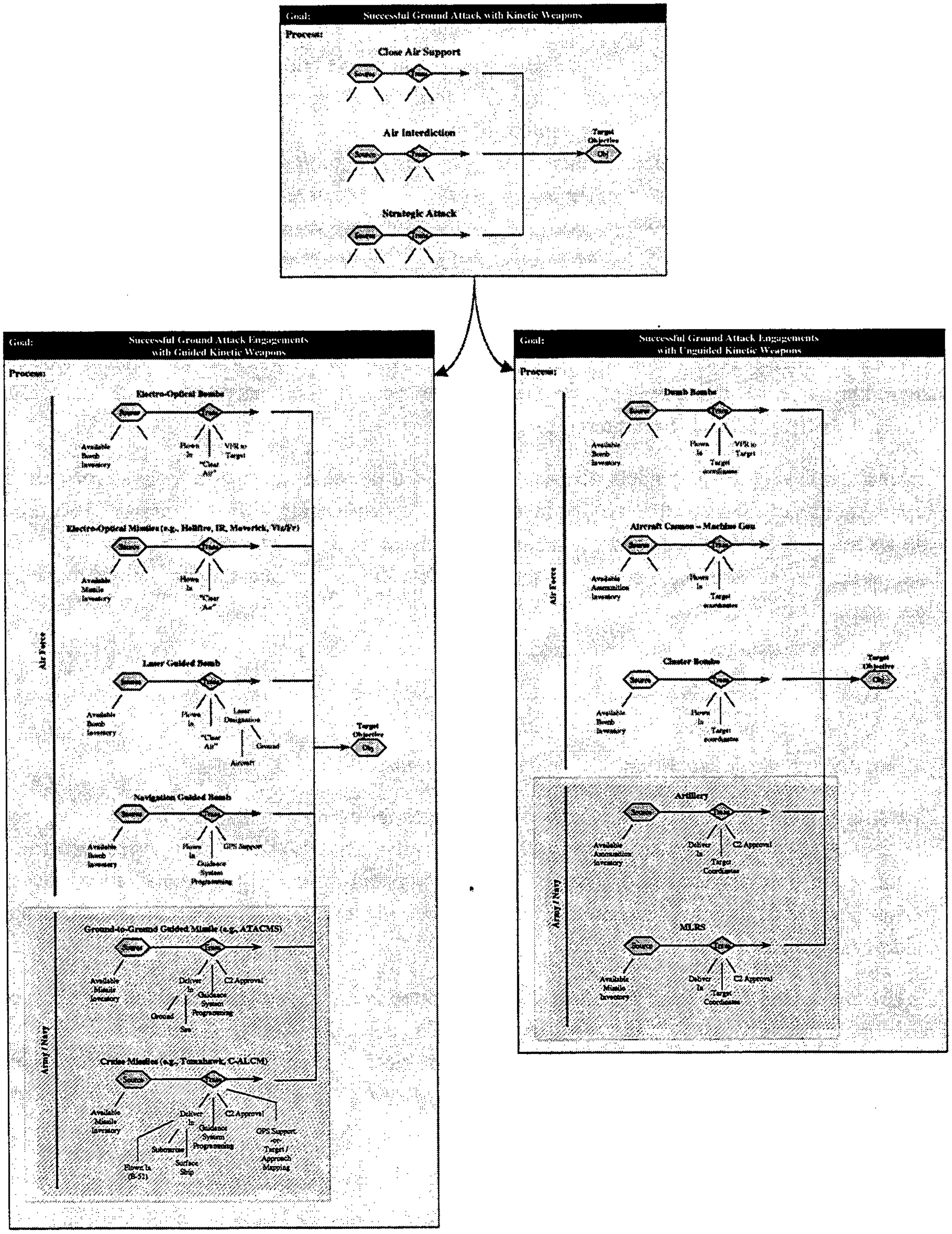


Figure 5. A portion of a Functional Abstraction Hierarchy for an Air Combat Power work domain. The functional relationships indicate the functionally different ways to accomplish the higher-order objective.

available resources are: Dumb Bombs, Aircraft Cannon – Machine Gun, and Cluster Bombs. Figure 5 depicts these functional relationships.

2. Understanding the Way People Operate in Their World

From this model of system functioning, it is possible to derive and preserve the critical decisions required for achieving goal success. These decisions center around goal-directed behavior, such as monitoring for goal satisfaction and resource availability, planning and selections among alternative means to achieve goals, and control of process initiation, tuning, and termination. All subsequent data sources and complementary methodologies can then add depth and substance to the model (e.g., critical decisions overlaid on the relevant goal/process relationship, strategies represented as paths through the hierarchy, collaborative work as represented by overlapping regions of responsibility within the hierarchical model). By organizing the specification of operator information, decision, and control requirements around nodes in the goal-means structure, rather than requirements for predefined task sequences (as in other approaches to cognitive analysis), the representation helps insure that the resulting person-machine systems explicitly reflect the decision centered perspective.

Rasmussen (1986, 1994) models decision-making as a decision ladder – a template or generic representation of the steps that may be involved in decision-making. It is a "template" because it allows the analyst to show how a particular step is instantiated in the specific case, and how, for that particular decision, some steps may not be needed. It is a non-linear, opportunistic modeling tool that contains:

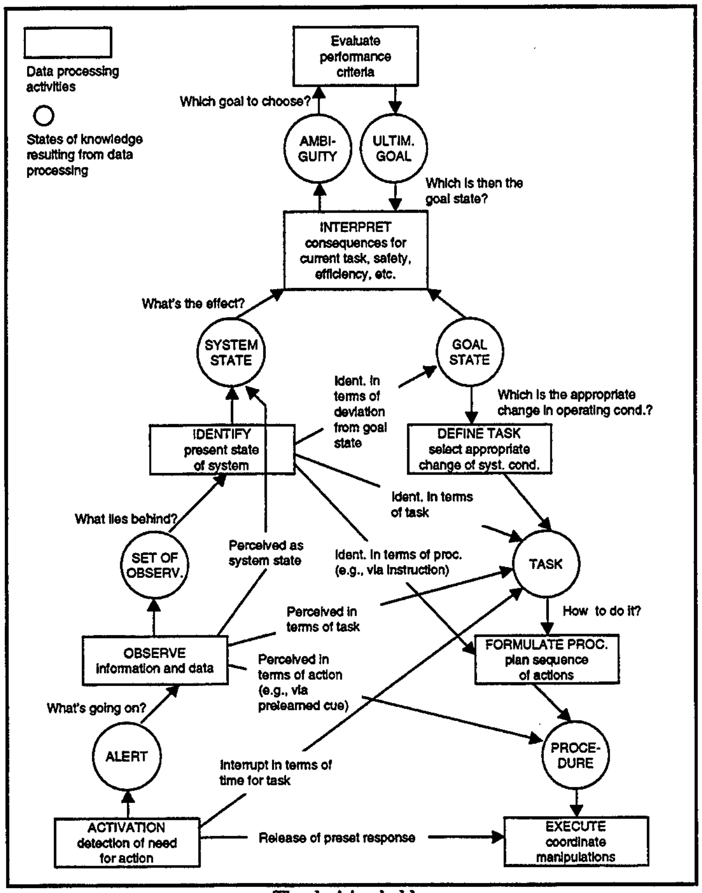
- Information processing activities;
- States of Knowledge;
- Multiple entry and exit points; and
- Opportunistic movement via "leaps" and "shunts".

Decision ladders can be used in either a descriptive or formative manner. As a descriptive tool, the template provides a systematic way of parsing verbal protocol data. As a formative tool, it provides a means of focusing on designing support systems that will foster and support expert performance (e.g., by providing for information processing leaps). Figure 6 presents the decision ladder framework.

As the underlying framework for the CACSE-supported analysis, we have modeled our decision modeling processes based on the work of David Woods, Emilie Roth, and Randy Mumaw (see Roth and Mumaw, 1995) in which the types of decisions in a goal-directed system are decomposed into the following categories:

Goal Monitoring:

- What are the Goals? (e.g., Win the Battle vs. Sustain Minimal Damage?)
- Goal Satisfaction: Are the function-related goals satisfied under current conditions?
- Margin to Dissatisfaction: Are goal limits/restrictions being approached?
- Consequences of goal dissatisfaction: What higher-level goals are supported by this function, and what are the consequences if this function is not achieved?



The decision ladder

Figure 6. Decision ladder framework for modeling problem solving behavior in operational contexts.

Process Monitoring:

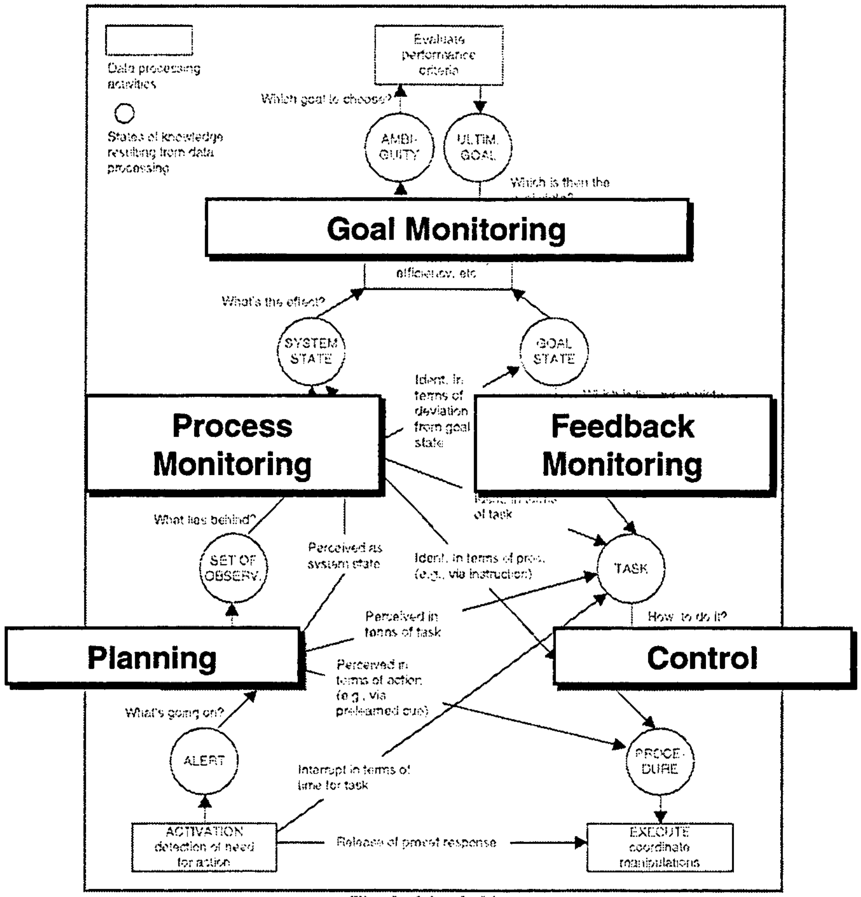
- Active processes: What resources are currently active? What is the relative contribution of each of the active resources to goal achievement? Are the resources performing correctly?
- Process element monitoring: Are the individual resources and their components working as they are supposed to?
- Automation monitoring: Are automated support systems functioning properly? What goals are the automated support systems attempting to achieve? Are these appropriate goals?

Feedback Monitoring:

- Procedure adequacy: Is the current procedure achieving the desired goals?
- Control action feedback: Are the CO's battle actions achieving their desired goals?

Planning:

- Battle plan goals priority: Which goal has the highest priority?



The decision ladder

Figure 7. Decision ladder framework for modeling problem solving behavior in operational contexts.

- Battle resource availability: What alternative resources are available for achieving the battle's goals?
- Choices among alternatives: Can an alternative resource be deployed?
- Consequences and side-effects of actions: What other battle resources and functions are affected by the current battle actions?

Control:

- Battle resource control: How is the battle resource controlled for resource deployment, tuning for optimum performance, terminated?

These decision categories are simplifications of the decision ladder, yet capture the critical aspects (or regions) of the ladder template. Figure 7 represents the mapping between decision categories and Rasmussen's decision ladder.

Within our example above of the "Air Combat Power" portion of the work domain, suppose that the commander's decision in question is the one of 'Choosing the Resource(s) necessary to accomplish the goal of Successful Ground Attack Engagements with Guided Kinetic Weapons'.

The set of surrounding or supporting decisions is as follows, based upon the partial Functional Abstraction Hierarchy (FAH) that was created above:

Primary Decision:

- D1 – "Choose the combined combat power (resources) necessary to accomplish the objectives of a Successful Ground Attack using Guided Kinetic Weapons.

Secondary Decisions:

- D1a – "Monitor the enemy ground forces to determine their survivability of air attack." (Goal Satisfaction Monitoring)
- D1b – "Determine which of the Kinetic Weapons resources can be available for delivery to enemy ground forces consistent with known time constraints (e.g., consider time schedule for delivery platform, etc.)." (Planning – process availability)
- D1c – "Determine which of the available Kinetic Weapons resources, either individually or in combination, would be most effective against enemy ground forces defensive positions and weapons." (Planning – choice among alternatives)
- D1d – "Determine how the selected Kinetic Weapons resources delivery are initiated and executed". (Control – process control)
- D1e – "After execution/delivery of the selected Kinetic Weapons on enemy ground forces positions, determine the effectiveness of the weapons had on achieving the objectives of a Successful Ground Attack using Guided Kinetic Weapons." (Feedback Monitoring)

Side Effects (other impacted goals):

- What will be the impact of using/diverting these kinetic weapons (munitions) and delivery platforms (aircraft) to the current objective? Are these weapons and platforms already engaged or scheduled to be engaged on other missions at the time they are needed to accomplish this objective?

D. Designing Support for the Envisioned World

1. Using Prototypes as Tools for Discovery

In the development of advanced joint cognitive systems (i.e., a system including human and machine agents where each work toward the same goals), one main caution with any CTA approach is the potential for focusing on and capturing only attributes of the "as-is" world. This is especially difficult in the development of an innovative decision support environment that enables revolutionary changes in the (decision) business processes. In designing an "envisioned world" type of system, it is critical to build on the strengths of the current world and focus technology development on supporting weaknesses in the current processes. Within this challenge, it becomes essential to be able to discriminate between aspects of task performance that result from "surface" characteristics of the current work environment and fundamental cognitive demands that will carry over to the new environment. The FB-CTA approach explicitly models the fundamental goals, processes, and relationships within the domain and the cognitive demands and strategies of the operators in response to these functional properties (Roth and Woods, 1989). It has been shown to be a powerful, generative methodology for identifying and understanding aspects of the current world that effectively support difficult decision-making while at the same time identifying needed decision support for the to-be world. The result is an innovative yet well grounded solution to the essential, underlying demands of the domain. The technique has proven robust enough to be used across a wide spectrum of domains: Nuclear Power, Military Airspace Control and Air Defense management, Logistics Distribution, Medical Evacuation C2, NASA Ground Control, and is currently being used to support the FAA transition to Free Flight Operations.

Based on this fundamental grounding, it is possible to:

- Analyze existing technologies to understand the positive aspects of current approaches;
- Explore a variety of implementation technologies, while evaluating them against a consistent fundamental set of decision effectiveness criteria.

As conveyed in Figure 8, the FB-CTA process continues into the design/prototype development phase. The CTA model (output of the first phase of effort) becomes the initial hypothesis for artifacts embodied in the design prototypes which in turn are used to discover additional requirements for useful support (Woods, in press). A variety of techniques exist to use design artifacts (prototypes) as increasingly robust hypotheses about cognition and collaboration within the target domain. This is based on one of the fundamental findings of Cognitive Science – how a problem is represented influences the cognitive work needed to solve that problem (e.g., Zhang and Norman, 1994). As indicated by the figure, each opportunity to assess the utility of the design artifacts provides additional understanding of requirements for effective support but also additional understanding of our initial CTA model. Thus, designs – among other levels of analysis – embody hypotheses about how technology change will shape cognition and collaboration.

One critical distinction between these two phases of CTA is the shift from 'exploring the current world' to 'exploring the envisioned world.' While some techniques for CTA provide natural links between the two phases (e.g., functional abstraction modeling is grounded in the fundamental goals, constraints, and complexities of the domain; those which will remain unchanged in the envisioned world), there is a distinct shift in thinking about how new tools will play a role in supporting practitioners operating in their world. In this perspective, the CTA model becomes a basis to express how these design artifacts would improve performance.

Based on this, the critical issue becomes how to link CTA research to design. This can be expressed in several different ways, including: (Woods, in press)

- How do we close the gap between user-centered intentions and technology-centered actual design practice?
- How do we get research that is relevant to design?

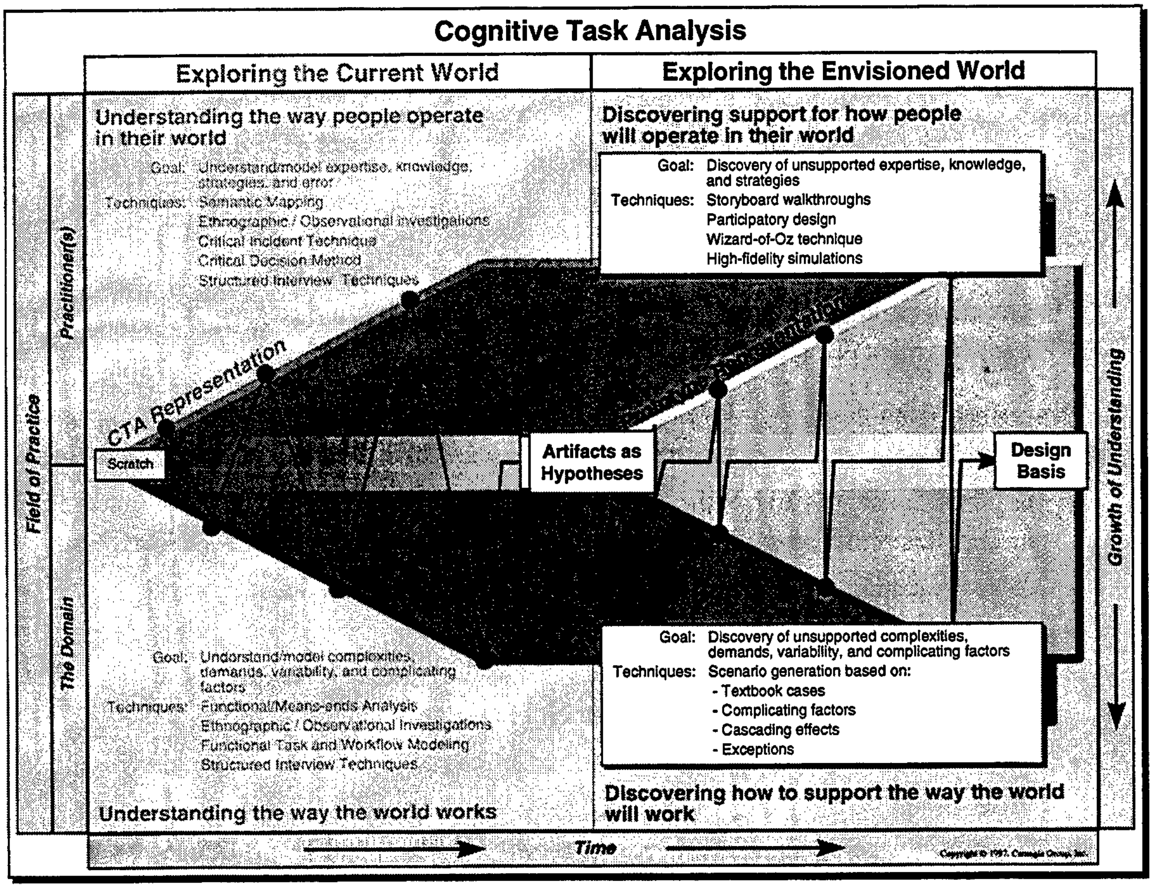


Figure 8. The transition to the second phase of CTA. This figure indicates how prototypes are used as tools for discovery and enrich the understanding achieved through the initial phase.

- How do we focus more on discovering requirements, what would be useful, and innovating new support concepts and less on polishing usability?
- How can one achieve a balance between short term pressures and long term learning, between the needs of particular projects and the broader goals of building up a research base to drive more useful designs and innovations?

An important issue from this perspective is that the vision of CTA as an initial, self-contained technique that is handed-off to system designers needs to evolve into a process of modeling not only the interconnections between the demands of the domain, the strategies and knowledge of practitioners, but also the cooperative interactions across human and machine agents, and how artifacts shape these strategies and coordinative activities.

A second equally important claim, then, is that it is only when we are able to design appropriate support that we truly understand the way the world works and the way people will operate in this world. This is the flip side of the claim by Winograd (1987; p. 10) that designing 'things that make us smart' depends on "...developing a theoretical base for creating meaningful artifacts and for understanding their use and effects."

One of the important roles that prototypes as tools for discovery can serve is that it allows one to discover design constraints and requirements for effective support that you might otherwise have overlooked. Examples of this include:

- information that should be available in parallel, yet contending for the same display space;
- tasks that should require minimal effort to achieve, yet are taking too much time or attentional resources (e.g., in nuclear power plant anything that requires more than one click to navigate to, becomes too resource intensive to employ under rapidly evolving plant disturbances).

Insights gained from designing, developing, and testing are critical in the iterative approach of this program. In this manner, ethnographic data can be useful at a variety of levels, including:

- as a mechanism to validate the cognitive models (how well did the model capture the critical functional properties and cognitive/decision demands of the domain);
- as an evaluation of operators' understanding of the critical relationships and functioning of the system;
- as an evaluation of operators' understanding of the roles, responsibilities, and information availability of the different crew members;
- as an evaluation of the operators' skills and strategies in response to the domain demands;
- as a mechanism to define the interactions between intelligent agents (human-human, human-machine, and machine-machine);
- as an evaluation of support concepts – current as well as envisioned (how well does the decision support system support these properties and demands).

Once the FAH has been augmented with critical decisions to form a "decision map" of the work domain, the process of building the "design thread" (i.e., the end-to-end connection from goal nodes in the FAH to supporting visualizations in the workspace design) can begin along two directions. First, information required to satisfy the critical decisions can be identified and documented. At this point in the analysis, the focus is on the "envisioned world" information, not "as-is" data-centered information. In some cases, the information requirements specified may not currently be able to be satisfied; this suggests data collection, transformation, or processing requirements. If the data sources for satisfying the identified information requirements are known, they can be captured in the CACSE tool. In this manner, a complete specification of data requirements can be handed off to the software developers.

Secondly, user interface objects (i.e., components of the user interface workspace) can be defined and linked to the associated decisions that they are designed to support. In this manner, decisions become the critical "lynchpin" between the functional analysis and the resulting visualization design.

Back to our example of the "Air Combat Power" portion of the work domain, the following are sample information requirements to satisfy the decisions related to 'Choosing the Resource(s) necessary to accomplish the goal of Successful Ground Attack Engagements with Guided Kinetic Weapons'. For the most part, this information is based on "sensed" data or access to databases:

Supporting Information Needs

- IR 1a.1 Physical location of enemy ground forces.
- IR 1a.2 Characteristics of enemy ground forces' defenses, e.g., heavily fortified bunkers vs. hastily dug trenches, anti-aircraft missiles vs. rifle or artillery fire, etc.
- IR 1b.1 Determine inventory of guided kinetic weapons, i.e., how many of each type.
- IR 1b.2 Determine which weapons in the inventory are not assigned to other objectives.
- IR 1b.3 Determine the availability (i.e., schedule) of appropriate guided kinetic weapons' delivery platforms.
- IR 1c.1 Determine the characteristics (e.g., armament penetration capability, explosive power, guidance accuracy, etc.) of the guided kinetic weapons that have been determined to be available (see items D1b, above).
- IR 1d.1 Determine which delivery platform(s) can be used with which guided kinetic weapons.
- IR 1d.2 Determine weapons' guidance needs/requirements.
- IR 1d.3 Determine weapons' firing sequence and timing requirements, i.e., arming sequence, necessary permissions for weapon actuation, etc.
- IR 1e.1 Enemy defenses destruction data, e.g., damage to fortifications, enemy killed and wounded, damage to enemy weapons systems, etc.

Once the FAH has been augmented with critical decisions and information requirements, a target region of the FAH can be identified and appropriate visualization(s) designed to reflect these information requirements. The focus of this task is to develop the mapping between information on the state and behavior of the domain (i.e., critical decision and information requirements uncovered) and the syntax and dynamics of the support system being developed (i.e., the form and behavior of the visualizations). The goal is to reveal the critical information requirements and constraints of the decision task through the user interface in such a way as to capitalize on the characteristics of human perception and cognition.

Along these lines, the following products can be a critical part of the design process:

- ***Display task description*** – a definition of the ideal information (not data) needed to fulfill the decision requirement(s) and an explicit description of the goal(s) of a particular display to meet these requirements. While this is partially a result of the CTA, the emphasis in this second stage turns to display units instead of function and decision units. This provides the link between the CTA and the graphical depiction.
- ***Information space map*** – within the context the display task description, this is a definition of the data requirements to satisfy the information requirements. This includes a description of the context that the information must be represented in (time, geographical, organizational, etc.) as well as any necessary informational

relationships that must be conveyed. The important issue here is the impact that the information requirements will have on the underlying data processing portion of the system in terms of critical computations, transformations, and integrations to be performed.

- *Graphical depiction* – based on the content of the display task descriptions and information space maps, decisions can be made on how to most efficiently depict information within the necessary context. Initially this usually takes the form of off-line sketches, grouped together to form a storyboard representing various states corresponding to events in the process or user tasks. To characterize or design a graphic form one must consider how the form behaves or changes as the state of the referent changes. A successful mapping from domain semantics to visual representation must help the practitioner extract relevant information under conditions of actual task performance.
- *Workspace model* – As the graphical depiction expands to address the decision and information requirements, a workspace model is also constructed. This defines the relationships between displays and identifies how the entire suite of displays will be used as an orchestrated system to satisfy the information needs of the user. For example, this may include orienting cues that indicate in mentally economical ways whether something interesting may be going on in another part of the process as well as combinations of displays that, in concert, provide a total picture of the situation.

The Decision Requirements inform the interface designer as to the critical user issues that must be reflected in the interface design, The Information Requirements and Supporting Data Elements provides the interface designer with the raw data that must be available upon which the dynamics of the display can be built. A comparison between the two sets of requirements enables the interface designer to begin to understand what synthetic variables need to be created to enable the interface to show the user the impact on his/her critical issues as the raw data changes. For example, one can see from looking at these sets of requirements that there is a need to match guided kinetic weapons destructive characteristics to the characteristics of the enemy ground forces' defenses characteristics. The requirement sets along with this type of analysis of them is the bases for the Cognitive Systems Engineer to formulate the interface design requirements. Such design requirements are then used by interface designers to create one or more suggestions as to an appropriate interface design. The interface design requirements are also used as the basis for evaluating the success of the proposed interface designs (i.e., do the designs meet the requirements?).

Typical approaches to CTA/CWA struggle with the transition from analytical modeling to system design, where insights gained from the cognitive analysis effort must be crystallized into design requirements and specifications in order to impact the characteristics of the resulting system. An integrated, tool-supported FB-CTA methodology explicitly links object-oriented software engineering practices with the decision centered, innovative cognitive engineering 'preamble'. The CACSE development effort is aimed at extending this current manual design practice into an on-line, runtime, embedded FB-CTA forming the heart of the decision support solution.

An important side note – the functional abstraction hierarchy and associated decision map has been shown to be a very effective tool for integrating disparate elicitations from multiple Subject Matter Experts (SMEs). Insights from different perspectives actually serve to refine and deepen the model into a more robust representation of the domain.

In order to demonstrate one approach to this issue of linking CTA to design, the following section outlines a case study in which a CTA was performed on an 'envisioned world' problem with the critical need of providing design concepts for developing requirements for a decision support system. This case illustrates how prototypes serve two roles. First, they are instantiations of hypotheses about aiding concepts; second, they exist as partially refined final product. The focus, however, was on discovering requirements – what would be useful to enhance decision-making in this domain and reduce the paths to ineffective decisions. While considerable detail and scope has been omitted, it is hoped that this will be useful in demonstrating the value of the type of tool support CACSE provides for this process.

IV. REFERENCES

Bonar et al., Guide to Cognitive Task Analysis, Learning Research and Development Center, University of Pittsburgh, October, 1985.

Cook, R. I., Woods, D. D., Walters, M. and Christoffersen, K. (Aug., 1996). Coping with the complexity of aeromedical evacuation planning: Implications for the development of decision support systems. In *Proceedings of the 3rd Annual Symposium on Human Interaction with Complex Systems*. Dayton, OH: IEEE.

Cook, R. I. and Woods, D. D. (1996). Adapting to new technology in the operating room. *Human Factors*, 38(4), 593-613.

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

Klein, G. A., Calderwood, R., and MacGregor, D. (1989). Critical decision method for eliciting knowledge. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-19, No. 3, May/June, pp. 462-472.

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

Potter, S. S., Ball, R. W., Jr., and Elm, W. C. (Aug., 1996). Supporting aeromedical evacuation planning through information visualization. In *Proceedings of the 3rd Annual Symposium on Human Interaction with Complex Systems*. Dayton, OH: IEEE. pp. 208-215.

Potter, S. S., Elm, W. C., Roth, E. M., and Woods, D. D. (1997a). Intelligent cognitive engineering suite for information warfare domains. Unpublished technical report. Pittsburgh, PA: Carnegie Group, Inc.

Potter, S. S., Elm, W. C., Roth, E. M., and Woods, D. D. (1997b). Cognitive task analysis as bootstrapping multiple converging techniques. Unpublished technical report. Pittsburgh, PA: Carnegie Group, Inc.

Potter, S. S., McKee, J. E., and Elm, W. C. (1997). Decision centered visualization for the military capability spectrum project. Unpublished technical report. Pittsburgh, PA: Carnegie Group, Inc.

Ranson, D. S. and Woods, D. D. (1996). Animating Computer Agents. In *Proceedings of the 3rd Annual Symposium on Human Interaction with Complex Systems*. Dayton, OH: IEEE. pp. 268-275.

Rasmussen, J. (1985). The role of hierarchical knowledge representation in decision-making and system management. *IEEE Transactions on Systems, Man, and Cybernetics*. Vol. SMC-15, pp. 234-243.

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

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

Roth, E. M., Brockhoff, C. S., Rusnica, L. A., Kenney, S., Kerch, S. P., Thomas, V. M. & Sugibayachi, N. (1997). Rapid prototyping and simulator evaluation of a wall panel overview display. Proceedings of the 1997 IEEE Sixth Conference on Human Factors and Power Plants, June 8-13, Orlando, Florida, 1997. (pg. 18-14 - 18-19).

Roth, E. M. & Mumaw, R. J. (1994). Cognitive engineering: Issues in user-centered system design. In J. J. Marciniak (Ed.), *Encyclopedia of Software Engineering*. New York: Wiley-Interscience, John Wiley & Sons, 110-123.

Roth, E. M. & Mumaw, R. J. (1995). Using cognitive task analysis to define human interface requirements for first of a kind systems. *In Proceedings of the Human Factors and Ergonomics Society 39th Annual Meeting*, October 9-13, San Diego, CA. (pp. 520 -524).

Roth, E. M., Mumaw, R.J. , Vicente, K. J. & Burns, C. M. (1997). Operator monitoring during normal operations: Vigilance or problem-solving? In *Proceedings of the Human Factors and Ergonomics Society 41st Annual Meeting*. September, 1997. Albuquerque, NM.

Roth, E. M. & Woods, D. D. (1988) Aiding human performance: I. Cognitive analysis. *Le Travail Humain, 51* (1), 39-64.

Roth, E. M. and Woods, D. D. (1989). Cognitive task analysis: An approach to knowledge acquisition for intelligent system design . In G. Guida and C. Tasso (Eds.), *Topics in Expert Systems Design.* Elsevier Science Publishers B. V. (North Holland).

Salas, E., Prince, C., Baker, D. P., and Shrestha, L. (1995). Situation awareness in team performance: Implications for measurement and training. *Human Factors*, 37 (1), pp. 123-136.

Sarter, N. and Woods, D. D. (in press). Teamplay with a powerful and independent agent: A corpus of operational experiences and automation surprises on the Airbus A-320. *Human Factors*, in press.

Shattuck, L. and Woods, D. D. (1994). Critical Incident Technique 40 Years Later. In *Proceedings of the 38th Annual Meeting of the Human Factors and Ergonomics Society*, October, 1997. Nashville TN.

Shattuck, L. and Woods, D. D. (1997). Communication Of Intent In Distributed Supervisory Control Systems. In *Proceedings of the 41st Annual Meeting of the Human Factors and Ergonomics Society*, September, 1997. Albuquerque, NM.

Smith, P., Woods, D., Billings, C., Sarter, N., McCoy, E., Denning, R., and Dekker, S. (1997). *Human-Centered Technologies And Procedures For Future Air Traffic Management.* Cognitive Systems Engineering Laboratory Report. Columbus OH: The Ohio State University, March 1997. Prepared for NASA Ames Research Center.

Vicente, K. J. and Rasmussen, J. (1990). The ecology of human-machine systems II: Mediating direct perception in complex work domains. *Ecological Psychology*, vol. 2, pp. 207-250.

Vicente, K. J. and Rasmussen, J. (1992). Ecological interface design: Theoretical foundations. *IEEE Transactions on Systems, Man, and Cybernetics*. Vol. SMC-22, pp. 589-606.

Vicente, K. J., Christoffersen, K., and Hunter, C. N. Response to Maddox critique. *Human Factors*, 38 (3), pp. 546-549.

Walters, M., Woods, D. D., and Christoffersen, K. (Aug., 1996). Reactive replanning in aeromedical evacuation: A case study. Presentation at the *3rd Annual Symposium on Human Interaction with Complex Systems*. Dayton, OH: IEEE.

Watts, J. C., Woods, D. D., Corban, J. M., Patterson, E. S., Kerr, R. and Hicks, L. (1996). Voice loops as cooperative aids in space shuttle mission control. In *Proceedings of Computer-Supported Cooperative Work*. Boston MA, 1996.

Winograd, T. (1987). Three responses to situation theory. Technical Report CSLI-87-106, Center for the Study of Language and Information, Stanford University.

Woods, D. D. (in press). Designs are hypotheses about how artifacts shape cognition and collaboration. *Ergonomics.*

Woods, D. D., Elm, W. C., and Easter, J. R. (1986). The disturbance board concept for intelligent support of fault management tasks. In *Proceedings of the International Topical Meeting on Advances in Human Factors in Nuclear Power.* American Nuclear Society/European Nuclear Society.

Woods, D. D. and Hollnagel, E. (1987). Mapping cognitive demands in complex problem-solving worlds. *International Journal of Man-Machine Studies*, 26, pp. 257-275.

Woods, D. D. and Roth, E. M. (1988). Cognitive systems engineering. In M. Helander (Ed.), *Handbook of Human-Computer Interaction.* Elsevier Science Publishers B. V. (North-Holland).

Woods, D. D. and Roth, E. M. (1988). Cognitive engineering: Human problem solving with tools. *Human Factors*, 30 (4), pp. 415-430.

Zhang, J. and Norman, D. A. (1994). Representations in distributed cognitive tasks. *Cognitive Science,* 18. pp. 87-122.

V. CASE STUDY: NAVAL COMMAND AND CONTROL

A. Introduction

This section is designed to walk through a case study in which we, through a FB-CTA, provided an analytical foundation and the development of design concepts for the next generation decision support system (DSS) to a Naval Command and Control Center. The primary motivation for this project is the need for decision-centered technologies to support information superiority within a mid-life upgrade to a Naval war ship.

This FB-CTA effort has focused on establishing the initial, innovative concepts required to transform a data-centered approach into powerful decision-centered visualizations of the target domain. The primary focus of the resulting analyses and design concepts is on the critical information necessary for rapid assessment and prioritization of threats from the various contacts identified by the ship's sensor equipment.

This target domain of building a Threat Assessment Decision Support System (TA-DSS) is a tangible example of C^4I decision support applications comprising complex human-machine systems wherein success depends heavily on the human operator's ability to efficiently cut through large amounts of data, visualize the state of the world and courses of action, and collaborate in a fast-paced, multi-person environment. In these types of domains, the critical need focuses on understanding and providing for the information needs of the operator, supporting the collaboration needs of the environment, and providing *effective* decision support in order to transform the environment from an inefficient, data-intense, high cognitive demand situation to an efficient, information-rich, high-performance human-machine system.

1. Function-Based Cognitive Task Analysis Overview

The Logica Carnegie Group team constructed a function-based Cognitive Task Analysis of the threat assessment portion of the Command and Control. This decision-centered analysis consisted of:

- building a Functional Abstraction Hierarchy to identify and model critical domain relationships;
- identifying decision requirements and the resulting information requirements;
- defining the relationships between information requirements and user interface design concepts;
- exploring techniques to implement these design concepts into powerful, flexible, visualizations of domain semantics.

This effort was accomplished by a goal-directed analysis in which the domain was structured in terms of goals to be accomplished, relationships between goals, and the means to achieve these goals. This approach consists of an analysis of system objectives that determine higher-order functional properties that need to be conveyed to the human operator. The result is a functional decomposition of system behavior, a description of decision requirements to achieve these system goals, and information requirements for assessing goal state, and determining courses of action.

2. Visualization Design

We then created a set of display designs to illustrate the manifestation of the FB-CTA analysis into support concepts. The product of this effort was a storyboard consisting of screen mock-ups and supporting design rationale.

The focus of this task was to develop the mapping between information on the state and behavior of the domain (i.e., critical decision and information requirements uncovered in the first phase) and the syntax and dynamics of the support system being developed (i.e., the form and behavior of the graphical user interface) in order to establish the "breakthrough" concepts to move to decision-centered visualizations.

Along these lines, the following products are a critical part of the design process:

- Display task description – a definition of the ideal information needed to fulfill the decision requirement(s) and an explicit description of the goal(s) of a particular display to meet these requirements.
- Graphical depiction – based on the content of the display task descriptions and information space maps, decisions can then be made on how to most efficiently depict information within the necessary context.

3. Scenario Development

As a parallel effort, we developed a realistic, operational scenario to provide a basis for design decisions. This operational scenario contains hypothetical input data, potential starting points into the TA-DSS, and strategies and paths through the information and data space as well as through the control structure of the user interface. The value of this scenario-based approach is that scenarios are used not only to provide data for future prototype development, but also to create incidents to address the predefined cognitive demands of the domain (identified in the FB-CTA) and to identify potential modifications to the user interface mechanisms which are used to interact with the domain.

B. Functional Abstraction Hierarchy Development

1. Approach

In this FB-CTA, the first step was to build an initial functional model of the domain. This was based on the benefits of this approach as a means to represent the underlying, unchanging system functions and relationships that form the basis for system complexity and goal achievement as well as the limited opportunity for interviews and observations of practitioners. This underlying

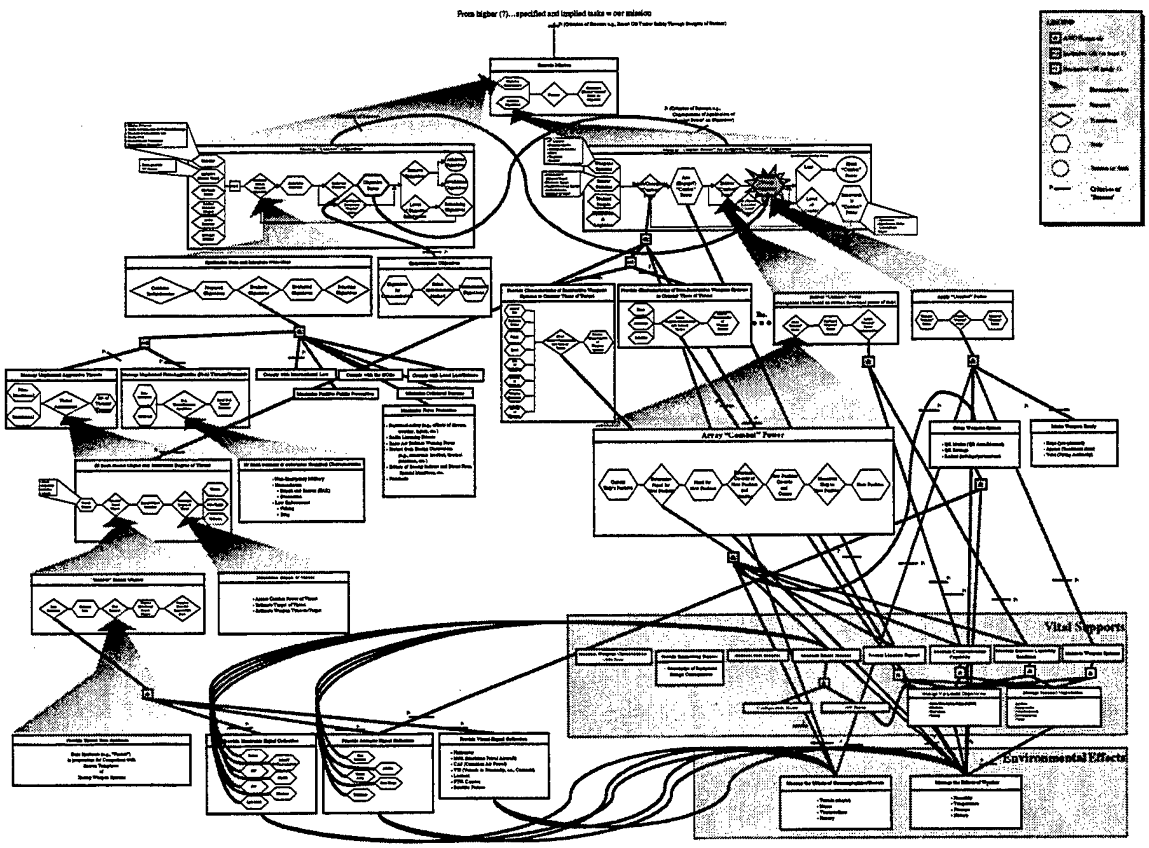


Figure 9. Functional Abstraction Hierarchy for Naval Command and control, focusing on threat assessment and response management.

model of system function is critical for understanding the context (in terms of goals to be achieved, strategies for achieving these goals, etc.) in which practitioner(s) must perform.

In addition, this approach allows one to identify, *a priori*, the information needed to cope with events which are unfamiliar to operators and which may not have been anticipated by designers. Thus, it is intended to define person-machine and inter-person information requirements to support operator decision-making and problem solving in unanticipated situations (as opposed to approaches based on predefined task sequences or incidents). Thus, it is a means of constructing a robust model for a complex environment independent of specific events, tasks, and strategies.

2. Results

Figure 9 presents the FAH after several opportunities to interview with domain experts and observe training simulations. At the top of this Functional Abstraction Hierarchy (FAH) is the primary goal, which is to "Execute the Mission." This goal is decomposable into two objectives, namely "Manage Combat Power" and "Manage Combat Objectives". The first element of the decomposition has to do with assembling and applying one's available resources so as to accomplish the mission, using the commander's decision-making abilities to drive the process of assembling and applying the resources. The second goal, related to managing combat objectives,

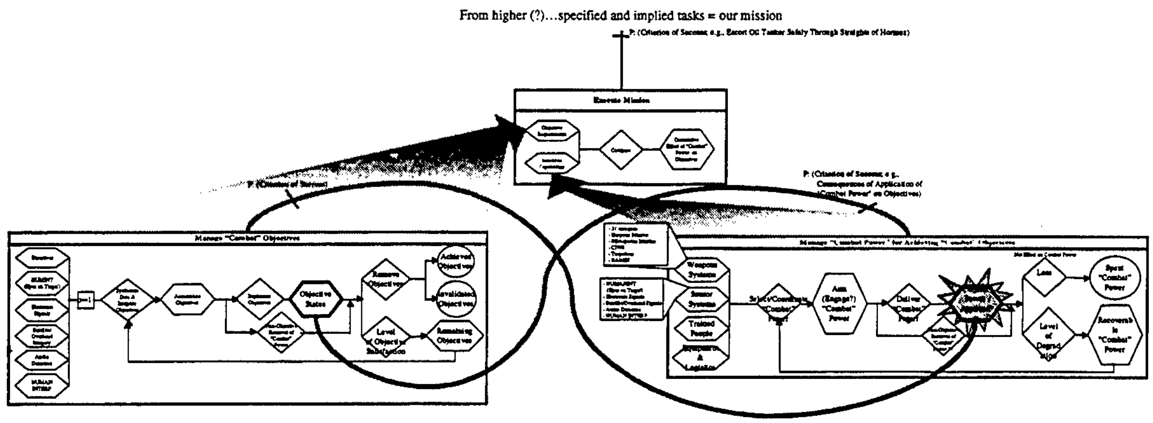


Figure 10. Top three nodes of the Functional Abstraction Hierarchy for Naval Command and control.

allows the commander to monitor the situation as his decision(s) is implemented and to adjust the moment-to-moment objectives as the situation dictates. These three goals, executing a mission by extracting one or more objectives from it and assembling and applying available resources in an attempt to accomplish the mission, serve as the core of the FAH. They form a continuously on-going process to which all other goals present themselves as interrupts as the situational context demands. These three nodes are shown in more detail in Figure 10.

Immediately under the "Manage Combat Objectives" is a set of goals that are required to synthesize objectives. These goals are presented in Figure 11. The primary goal in this section relates to the "Synthesis of Data" from various intelligence sources, and organizing that data such that a commander is able to distill desired objectives from it. Implicit in the distillation activity is the notion of prioritization of the possible objectives. Such a model depicts the one-to-many mapping of datum to objectives, i.e., a single piece of data may well relate to or provide additional information about more than one possible objective. The “Synthesis of Data” goal is a decomposition of "Manage Combat Objectives", while the second goal in this area, "Communicate Objectives", is a supporting function of "Manage Combat Objectives". “Communicate Objectives” relates to the transmission of the commander’s intent and other objective issues to subordinates, rather than to the formulation of the objective(s) itself, hence the notion of “supporting” rather than of “decomposition”. Beneath "Synthesize Data" there are three supporting goals "Manage Unplanned Aggressive Threats", "Manage Unplanned Non-Aggressive Demands", and "Assess Risks." Each of these goals has the potential of dynamically changing the priority and relationship of the objectives that need to be managed, and of changing the effectiveness of the assembled combat power. The first two goals are self-descriptive. The third, "Assess Risks", involves the comparison of factors, which, depending upon the context, may compete with each other. Under such circumstances, the commander is expected to weigh the competition between these goals and to decide or select those goals that are to be satisfied and those that are to be denied. Such potentially competing goals include, but are not limited to, “minimizing collateral damage” and “maximizing self-protection”.

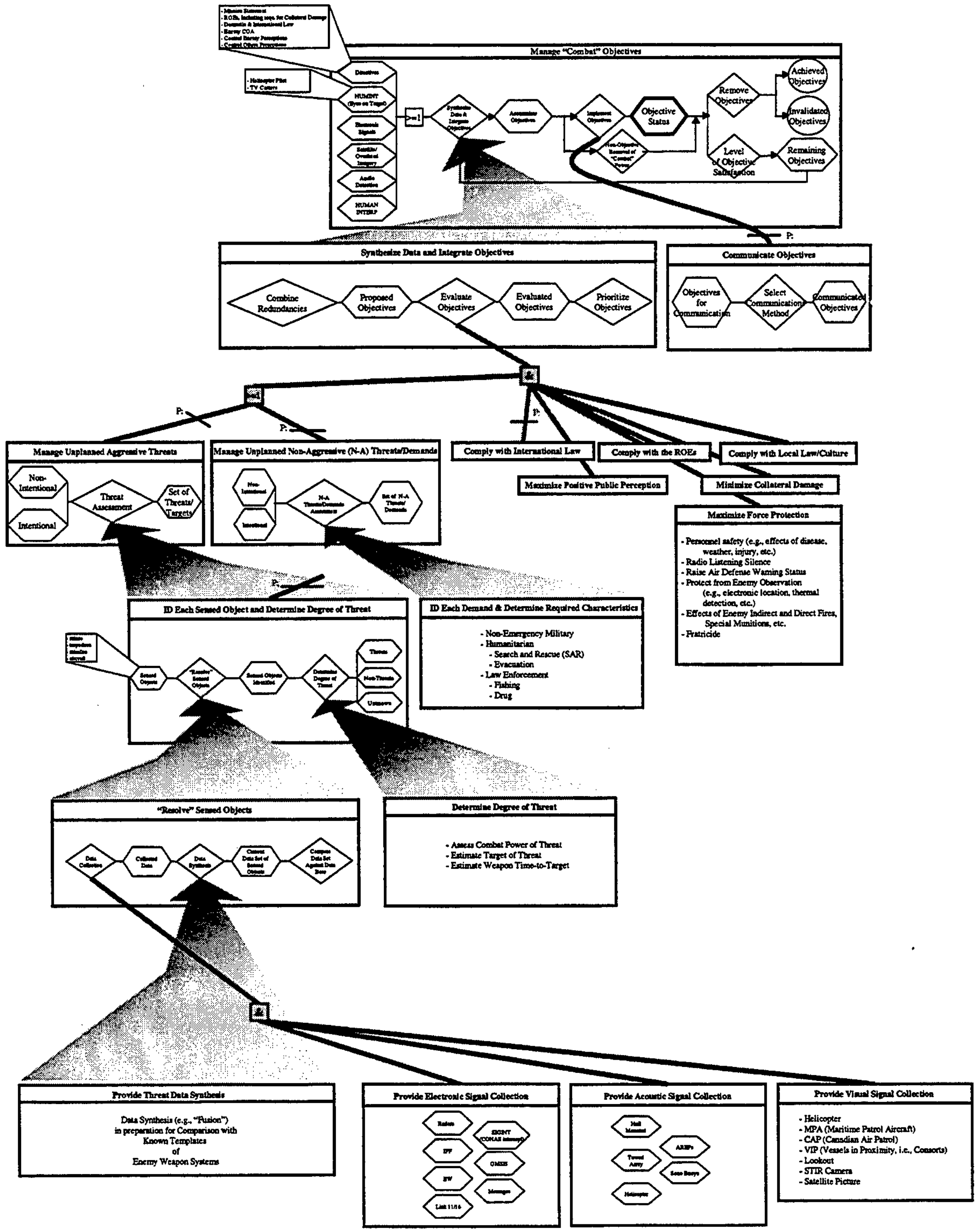


Figure 11. Decomposition of the "Manage 'Combat' Objectives" node.

Below the "Manage Unplanned Aggressive Threats" area is a series of four increasingly decomposed goals that relate to a commander gaining a better understanding of the threat, "Identify Each Sensed Object and Determine Degree of Threat", "Resolve Sensed Object", “Determine Degree of Threat” and "Provide Threat Data Synthesis." The first decomposition attempts to determine what a sensed object is and whether it represents a threat or not. Specific characteristics of the sensed object inform that goal which, in turn, is informed by data collected and synthesized about the threat. Note that the goal of "Identify Each Sensed Object and Determine Degree of Threat" affects not only the “Manage Combat Objectives” goals but also many of those associated with the “Manage “Combat” Power for Achieving ‘Combat’ Objectives”.

C. Identifying Decision Requirements and Supporting Information Needs

Just as the insights gained from user interviews provided input for refining the functional model, they also helped to identify critical decisions within the work domain. These decisions must then be supported by the resulting visualization design. In this step, the critical decisions for each node in the “threat assessment” region of the FAH were defined and attached to that particular region of the model.

1. Critical Decisions:

From this FAH framework, four critical decisions were identified (decision number in parentheses):

- *Process Monitoring:*
  Monitor all contacts (D1);
- *Goal Monitoring:*
  Determine degree of threat for all contacts (D2);
  Determine available response time for all aggressive threats (D3);
- *Control:*
  Determine appropriate response, including priority and order of response, to aggressive threats (D4).

2. Supporting Information Requirements:

In order to support the critical decisions listed above, the following information requirements were identified (associated decision in parentheses):

- Availability and functionality of contact sensing equipment; (D1)
- Contact characteristics (e.g., type, location, capability); (D2)
- Contact behavior (e.g., course, speed, response to warnings); (D2, D4)
- For incoming aggressive weapons; time until impact; (D3)
- For incoming aggressive platforms; time until likely weapon release (e.g., CPA); (D2, D3)
- Contact characteristics vs. Degree of Threat vs. Characteristics of On-board weapons database; (D2, D4)

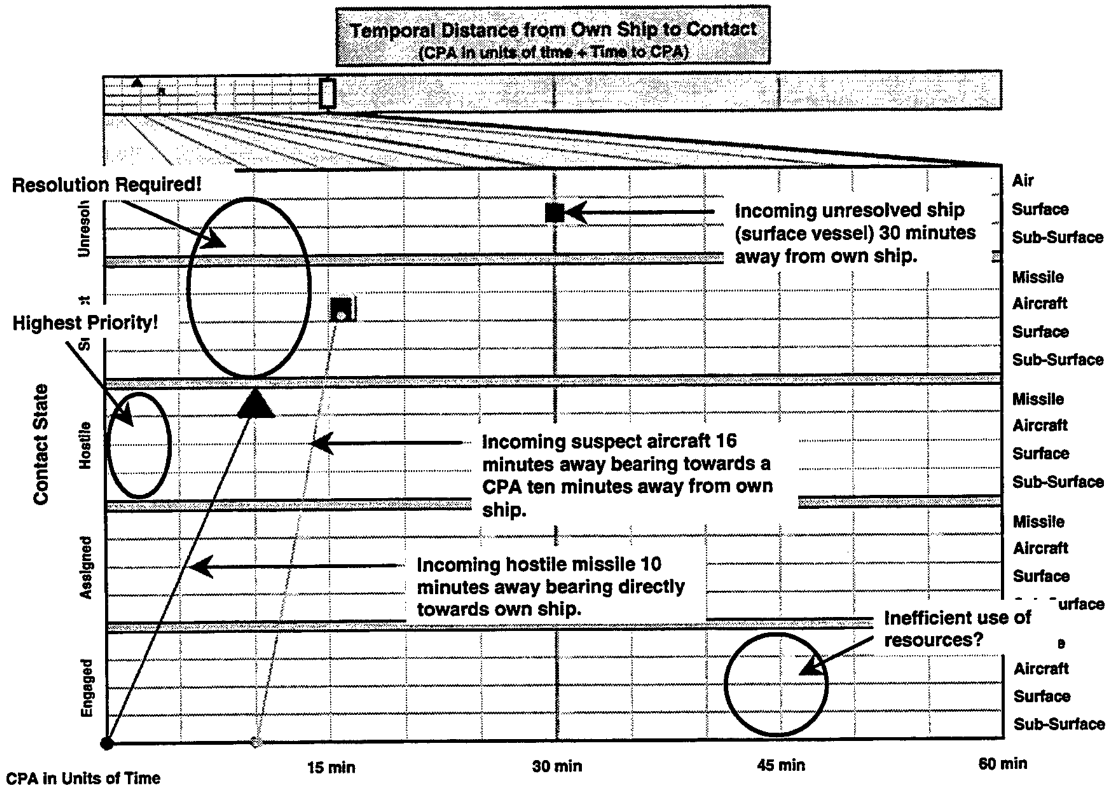


Figure 12. Resulting visualization design for the "Threat Assessment" node within the functional model.

D. Visualization Design

Figure 12 and Figure 13 presents the visualization design to support the information needs for each of the critical decisions identified in the previous step. Based on the identified decision requirements, the goal is to have a functionally-organized visualization (organized around time) of the state of those contacts identified as potential aggressive threats to provide an integrated means to achieve situation and threat assessment and response management. Critical issues include:

- Functional distance to threat defined as a function of:
  - time to closest point of approach,
  - CPA transformed into units of time;
- Threat state defined as a function of:
  - contact classification (i.e., unknown / suspect / hostile);
  - threat identification (i.e., missile / aircraft / torpedo);
  - threat ranking from automated system (DREV work);
  - planned response (i.e., weapon of choice);
  - engagement status (i.e., assigned / engaged / fired);

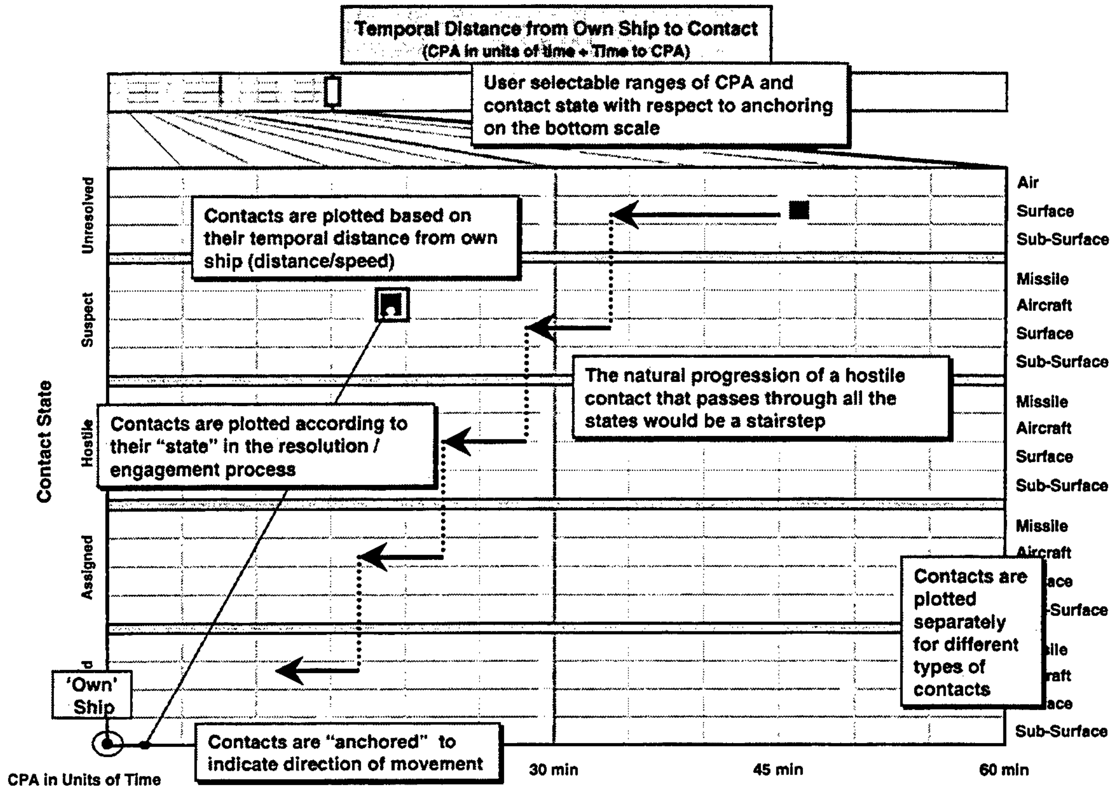


Figure 13. Annotations describing behavior of threat assessment visualization.

  - availability of response (i.e., time until threat is in arc of fire of weapon of choice);
- Based on these parameters, critical contacts to focus on include:
  - unresolved contact functionally close to ship;
  - hostile / unengaged contact close to ship;
  - hostile contact outside ship's arc of fire;
- At the highest level, need a clear indication of the presence of threat(s)!

E. Summary — Critical Needs for Design Support

One of the primary results of this case study is an identification of the critical issues in the transition from CTA to system design. The next section contains presentation material related to this case study indicating the impact of tool support for this process. The issues addressed by this case study include the need for:

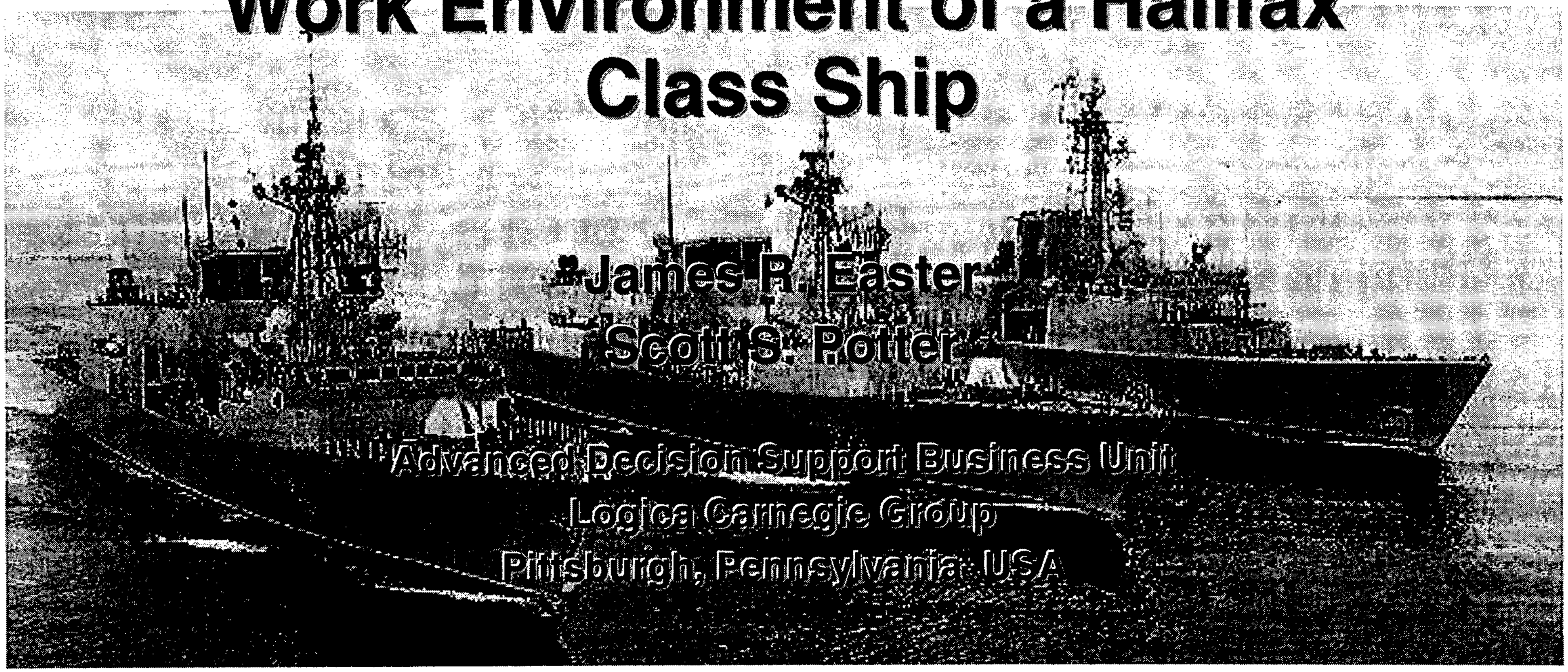
- CTA to go well beyond an initial CTA model. A CTA needs to provide concrete, decision-centered design concepts (e.g., information requirements, proof-of-concept storyboards) to provide sufficient support for system design. Initial CTA artifacts such as semantic maps, functional models, decision requirements are inadequate by themselves for software developers.

- an understanding of the artifacts used by software engineers (e.g., system requirements, object model) and how results from a CTA can be integrated into these artifacts (and effectively support system design activity). Given these artifacts form the underlying specification for system development, they are the critical targets if CTA is to effectively impact design.
- a mechanism for capturing design rationale in order to provide underlying basis for design concepts resulting from CTA effort (in order to separate the design concept from the instantiation). This is important from several dimensions. First, to separate the information from the presentation (in order to isolate the source of the problem in an ineffective design). Second, given the inevitable tradeoffs within implementation, to identify the critical aspects of the design concepts.
- scenario development to be a central part of CTA. Scenarios become a critical part of system development (e.g., concept of operations documents, event trace diagrams, test case generation) and need to be designed around complexities, variability, and complicating factors of the domain.

VI. CASE STUDY PRESENTATION MATERIAL

Logica Carnegie Group

Decision-Centered Visualizations for the Command and Control Work Environment of a Halifax Class Ship



James R. Easter
Scott S. Potter

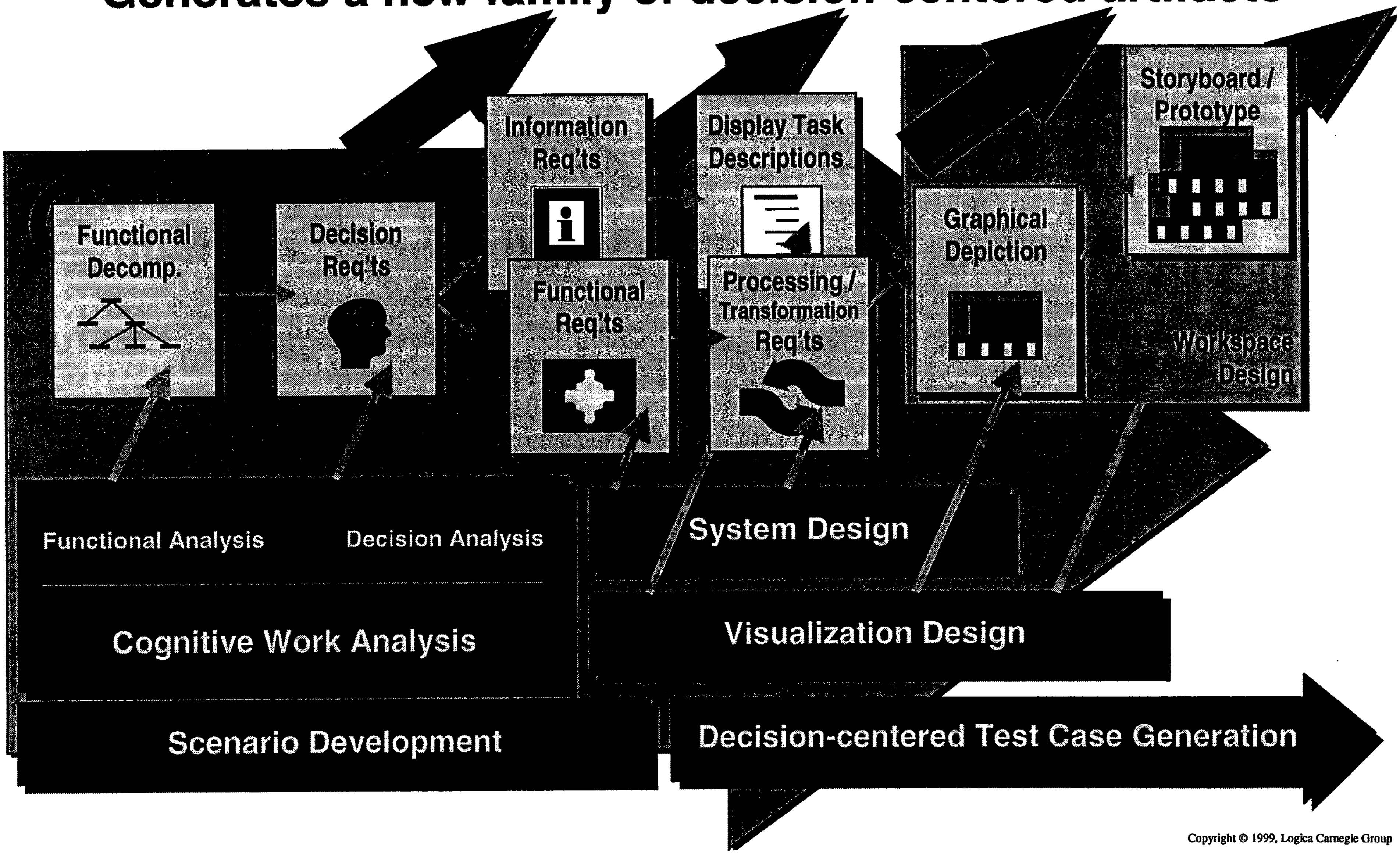
Advanced Decision Support Business Unit
Logica Carnegie Group
Pittsburgh, Pennsylvania USA

Project Goals:

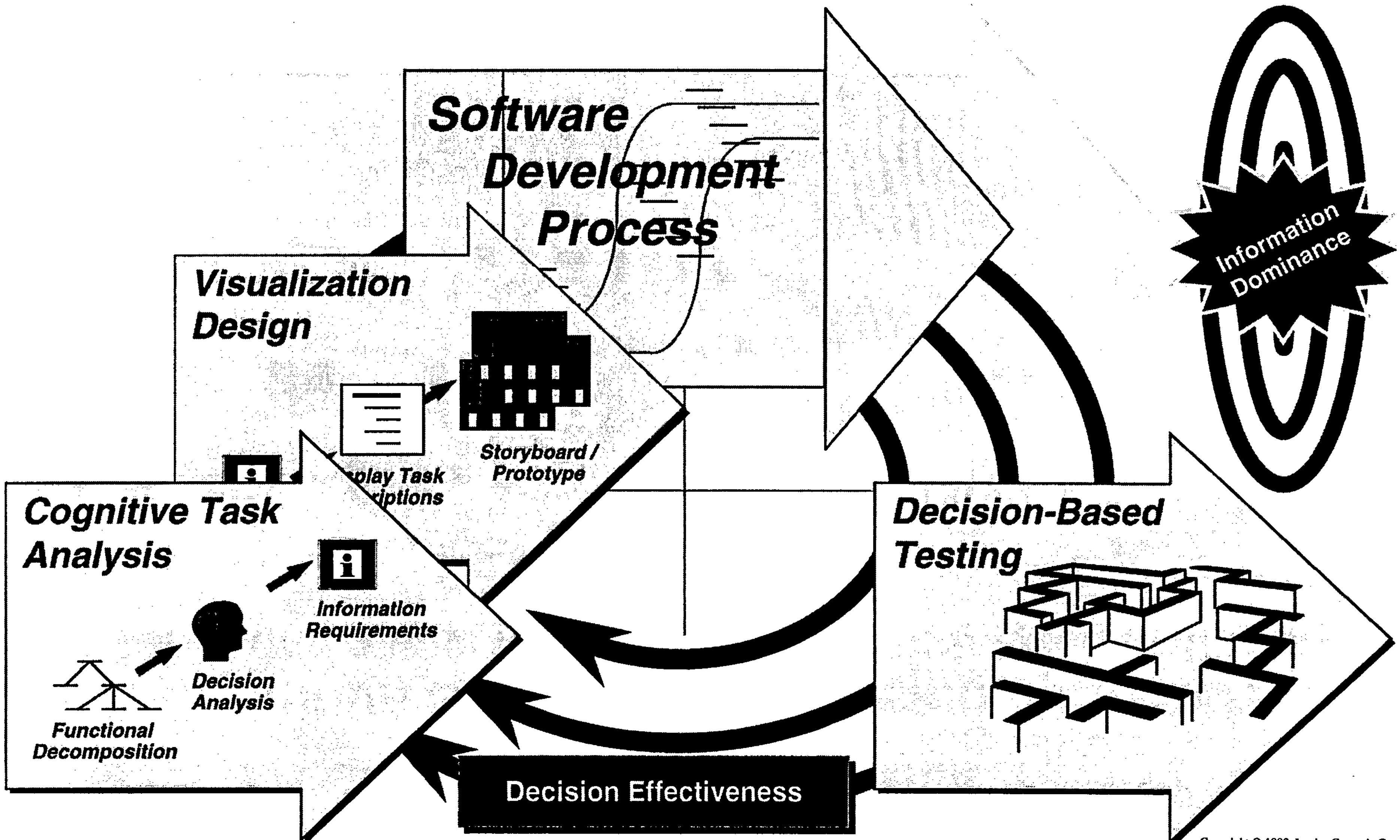
Demonstrate the utility of a multi-layered Cognitive Work Analysis (CWA) methodology in generating novel computer-based decision support for the cognitive demands and operator strategies in a complex, military CCS – specifically the tactical C2 work environment of a HALIFAX Class ship.

- Assess current support tools and technologies for the Operations Room Officer (ORO) within the context of a CWA.
- Assess ongoing technology development (e.g., data fusion, resource/response management) to support ORO decision requirements.
- Apply the results of the CWA to the definition of visualization and processing requirements for advanced human-system interface design technologies.
- Demonstrate the pragmatics of performing a multi-layered CWA methodology as an integral part of decision support software development.

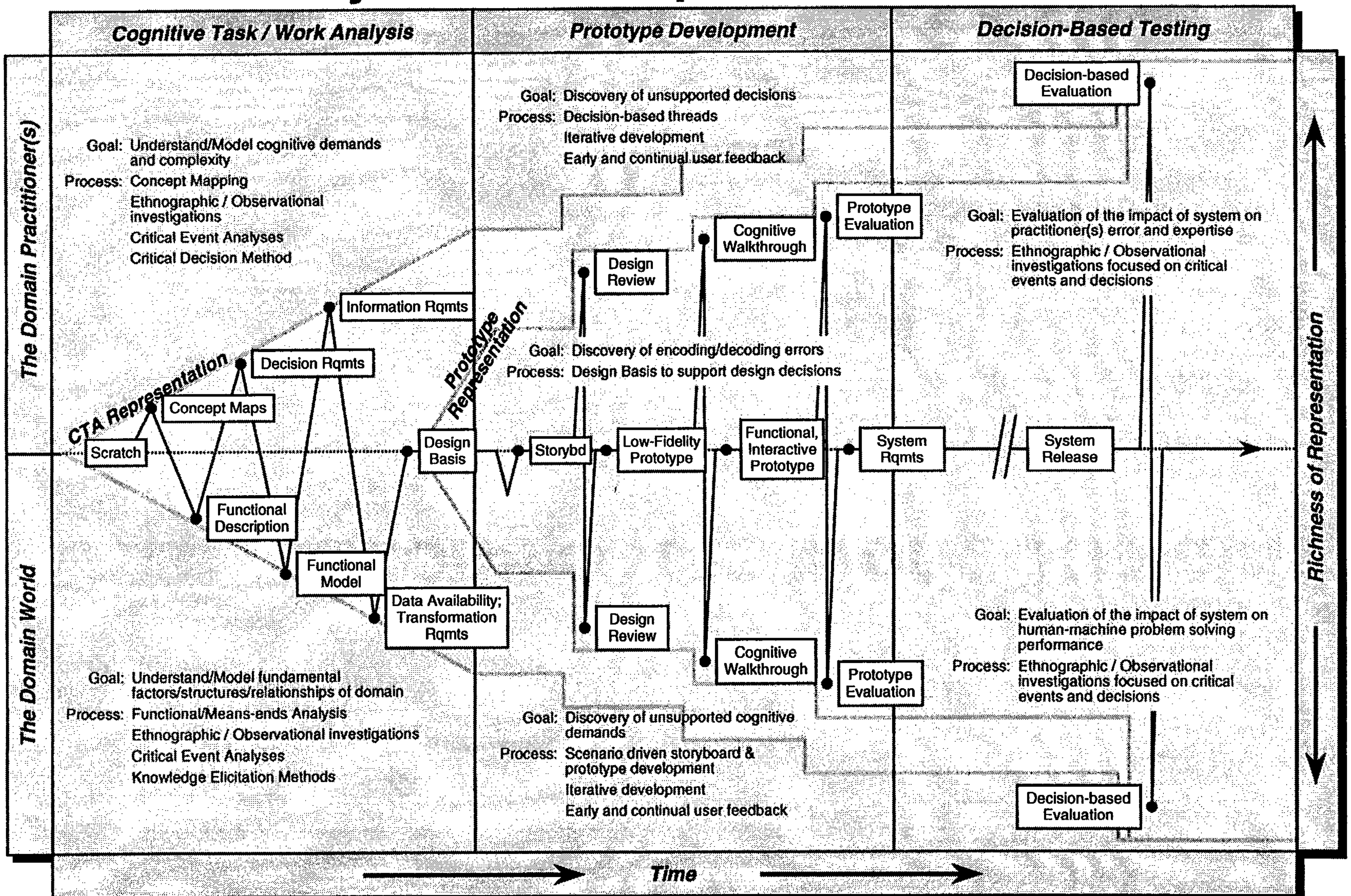
A Cognitive Systems Engineering approach...
Generates a new family of decision-centered artifacts



First Step Towards the Solution…
CSE-Based System Development Process



A Closer Examination...
CSE-Based System Development Process



Observations of the Work Domain

Key Aspects of the Work Domain:

- Highly structured collaborative work environment with explicit role definition;
- Explicit duties to individual operators; directed by ORO;
- Time-critical performance in a complex, dynamic, uncertain environment;
- Much attention directed at making inferences out of the raw data;
- Tremendous “knowledge in the head” to help in this interpretation;
- Even at the ORO level, information presentation is at the same level as subordinates;
- In summary – unbelievable / untenable data overload!

Key Aspects of the Work Domain...

Observed "Problem Areas" with the current CCS:

Current field of view vs. presence of contacts...

- ➤ OROs zoomed in on a region relevant to a specific threat when a second threat needed attention, but was outside the current zoom range of the radar.

Monitoring the auto-engage process...

- ➤ OROs incorrectly "expected" auto-engage events to occur but in fact did not due to the specific threat being outside of prescribed parameter envelope.

Responding to multiple threats...

- ➤ OROs failed to respond appropriately to multiple threats due to incorrect prioritization / temporal sequencing.

Missing the "information" for the "noise"...

- ➤ OROs experienced difficulty in focusing on the threats within a display that was cluttered with information on all contacts.

Logica Carnegie Group

Key Aspects of the Work Domain...

Critical Cognitive Demands:

Integrate the three worlds:

- The ORO serves as the "integration point" of the three worlds (sub-surface, surface, and air as well as both internal and external events relative to his/her own ship) and must keep one foot in each world to coordinate actions and events.

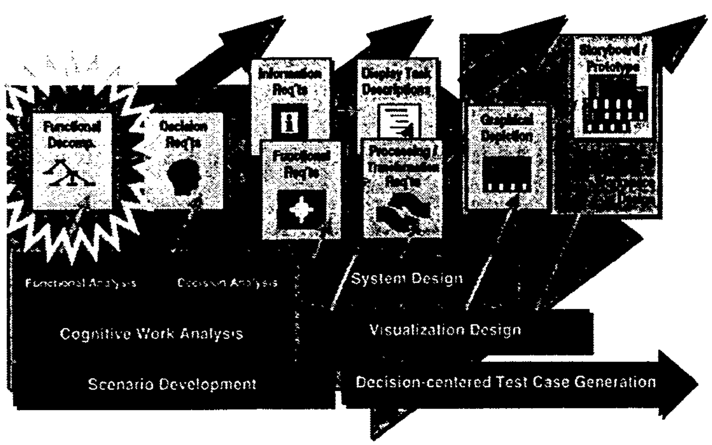
Keep the "big picture":

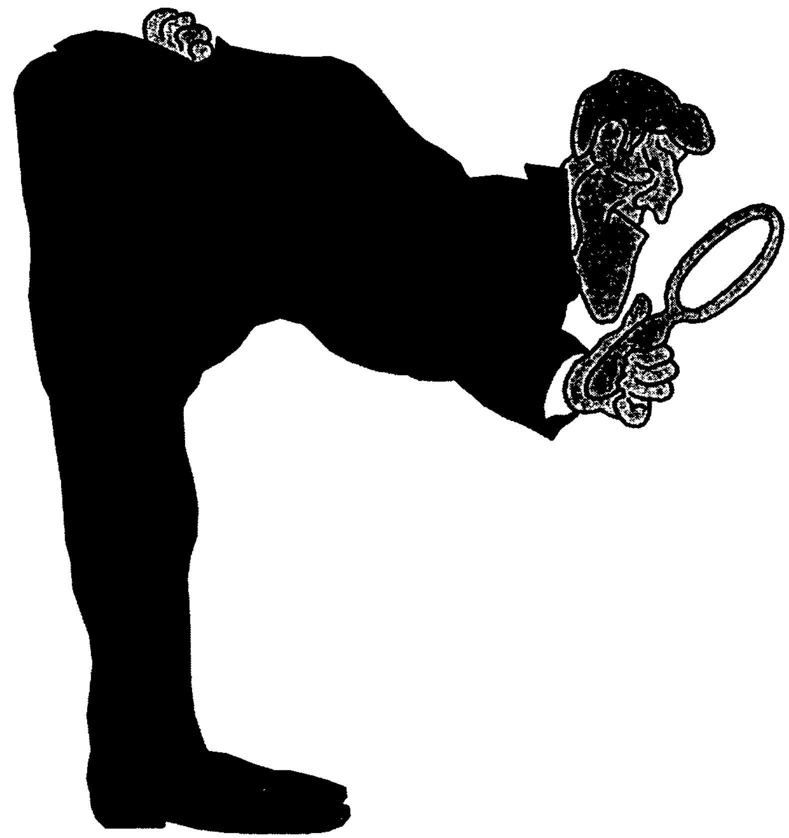
- The ORO must keep an overview of the entire situation and avoid becoming locked onto a particular problem at the expense of others. The ORO may need to alert or request the assistance of others on significant events outside of their current field of view.

Maintain an understanding of "functional impact" of events:

- The ORO must synthesize information about the different events and activities in the world into a coherent representation of the state of affairs. In other words, he/she must maintain a high-level situation awareness.

Initial Functional Analysis





Levels of Cognitive Work Analysis:

Work Domain Analysis:

- Describes the ***functional structure*** of the system being acted upon;
- Identifies constraints that are independent of any particular task, event, or goal;
- Provides the framework for subsequent analysis.

Control Task Analysis:

- Focuses on the goals to be satisfied in specific situations;
- Identifies levels of ***decisions / cognitive processing*** required for a specific goal context;
- Goes beyond the "OODA Loop".

Strategies Analysis:

- Focuses on understanding the different ways of accomplishing the activities identified in the above analysis;
- Identifies ***information requirements*** for goal achievement;
- Serves as the transition to visualization design.

Cognitive Work Analysis...
Functional "Commodities":

Functional Role:

- What's my ship's role within the TG (usually well defined)? (e.g., ASW screen);
- What are my ship's mission objectives (that may changes within a given context)? (e.g., tanker escort);
- What are the pre-defined ROEs? (that impact acceptable actions / responses).

Functional Capabilities:

- What's the range and quality of sensor systems?
- What's the range and kill probability of the weapon systems?
- What's the range (weapon and sensor) of the identified threats?

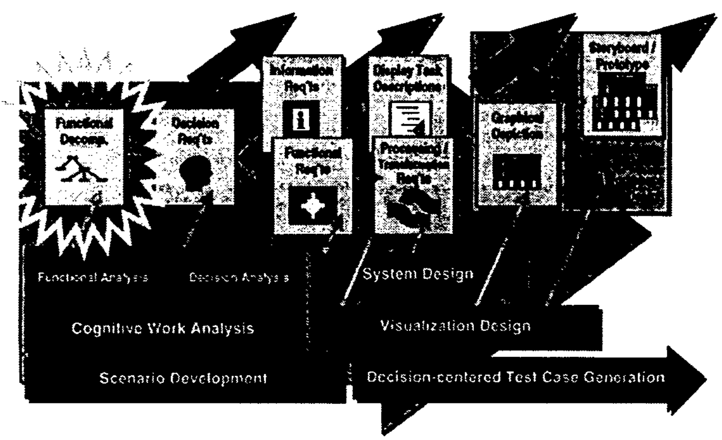
Functional Predictions:

- What are my known threats?
- Where can the enemy go? Where can't they go?
- What threats can I respond to? When can I respond to them?
- What's the impact of current conditions (onboard and environmental) on resource capability (own ship as well as threat)?

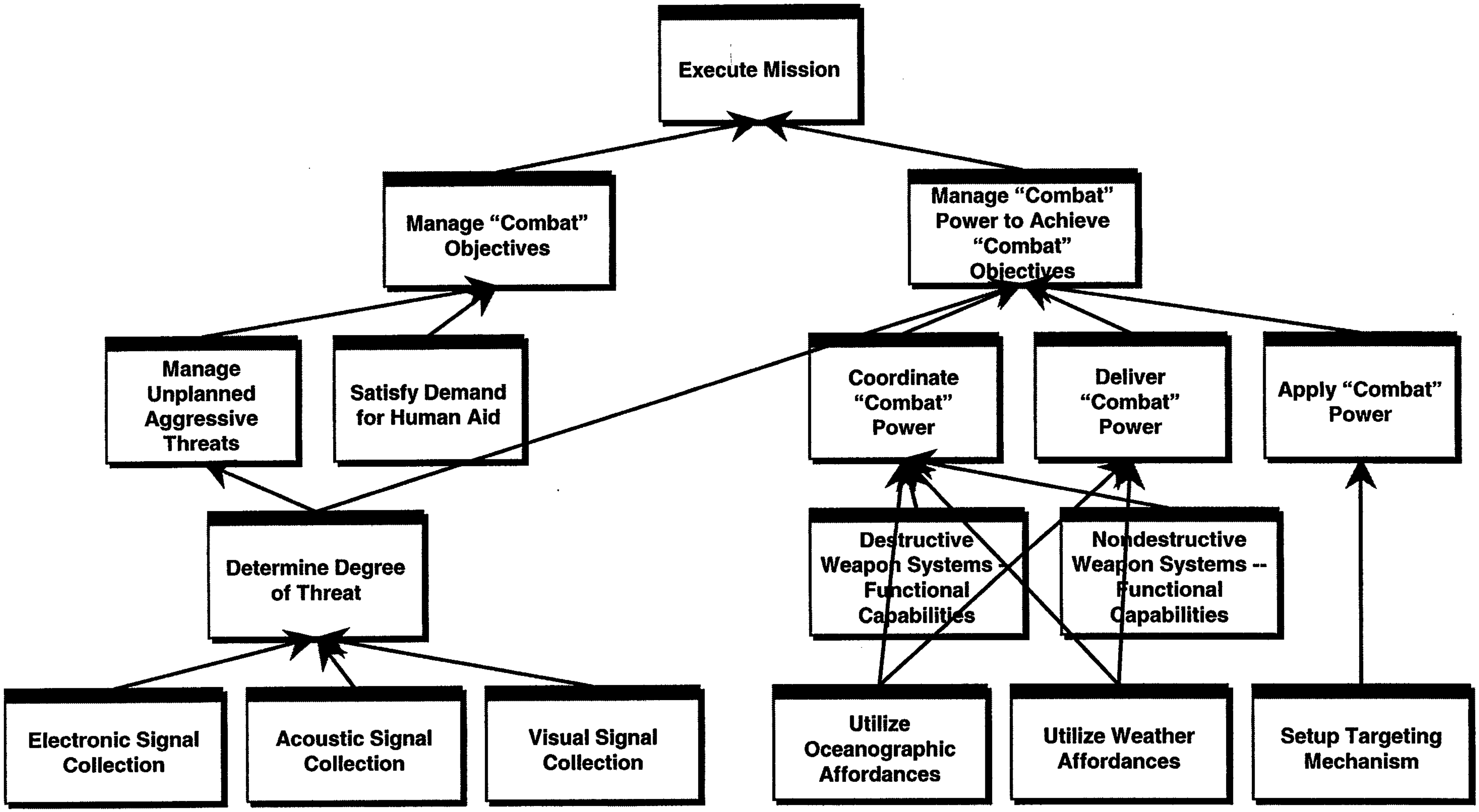
Levels of Abstraction

Functional Purpose	➤ Mission objectives; impact of disturbances on these objectives ➤ Enemy objectives;
Abstract Function	➤ Prioritization of disturbances; ➤ Projection of threats and own ship weapon and sensor systems;
Generalized Function	➤ Degree of threat of disturbances; ➤ Functional status of weapon systems and sensors;
Physical Function	➤ Functional capabilities of threats ➤ Functional capabilities of weapon systems and sensors
Physical Form	➤ Identification of tracks ➤ Current weapon systems and sensors

Functional Abstraction Hierarchy Modeling

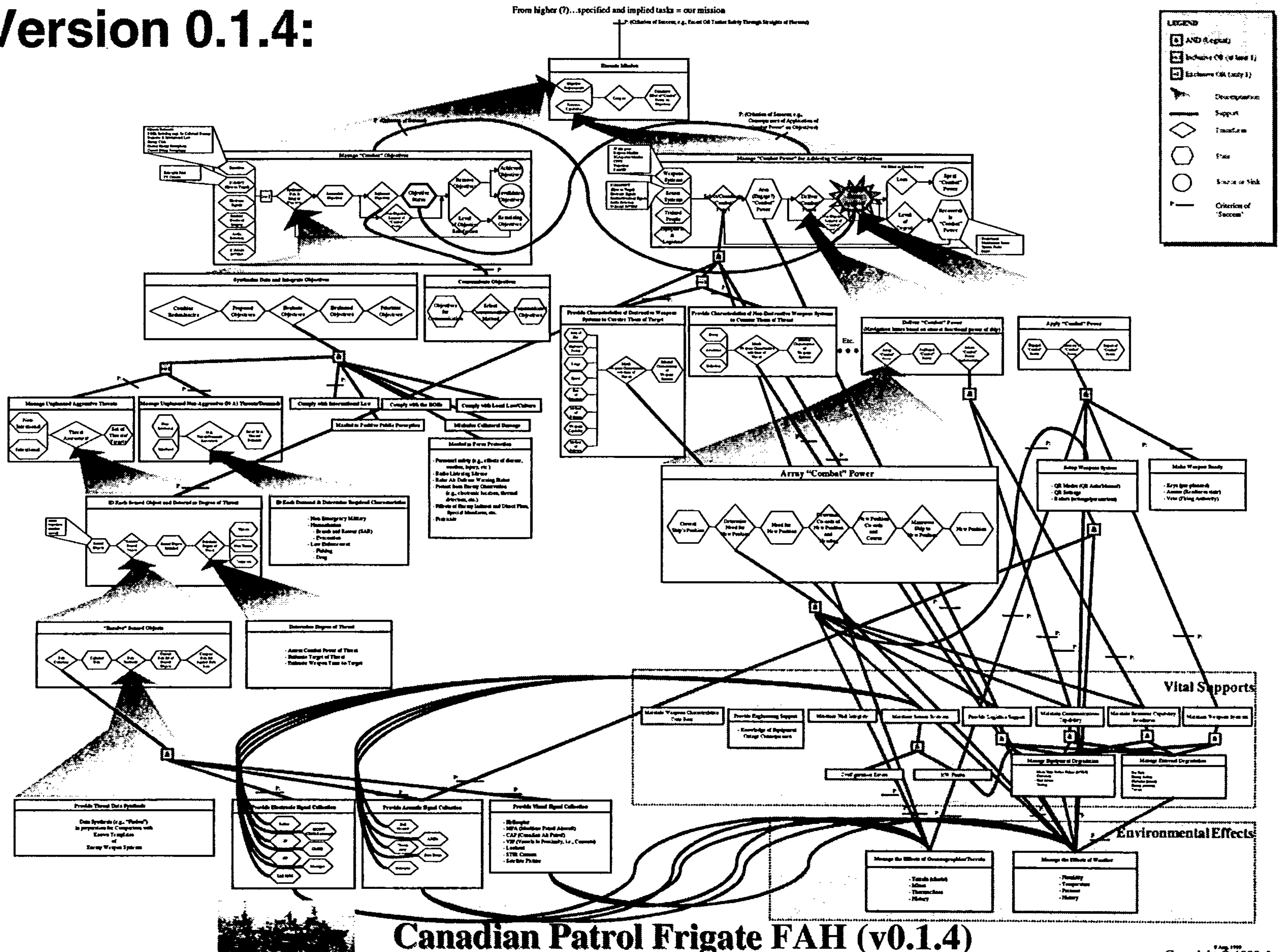


Functional Abstraction Hierarchy...
High-Level Overview:



Functional Abstraction Hierarchy...

Version 0.1.4:



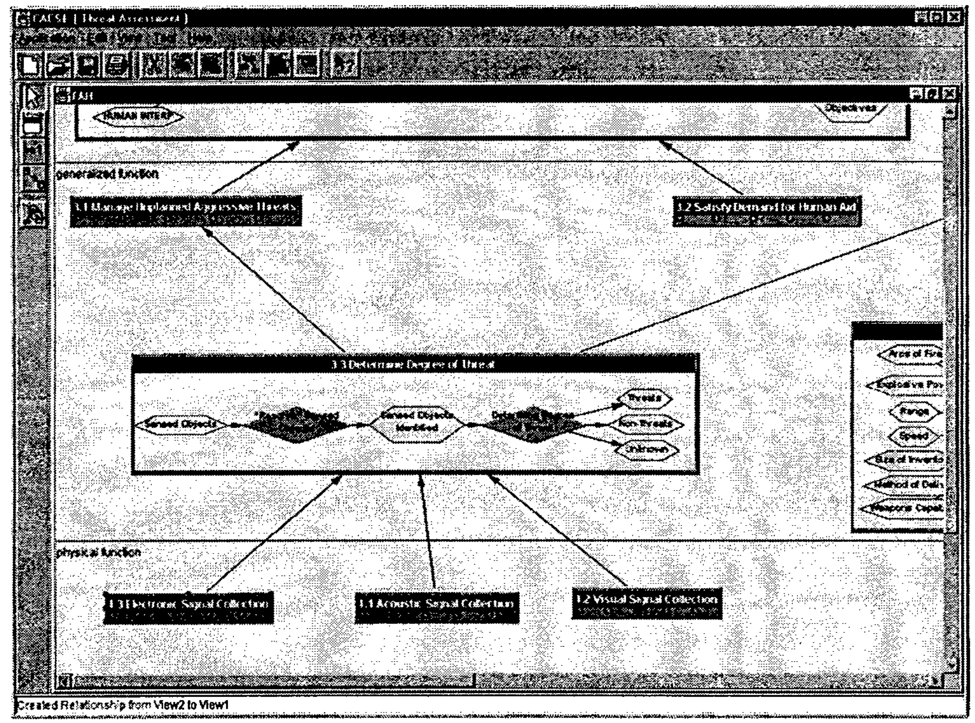
Canadian Patrol Frigate FAH (v0.1.4)

Tool Support for the CSE Process...

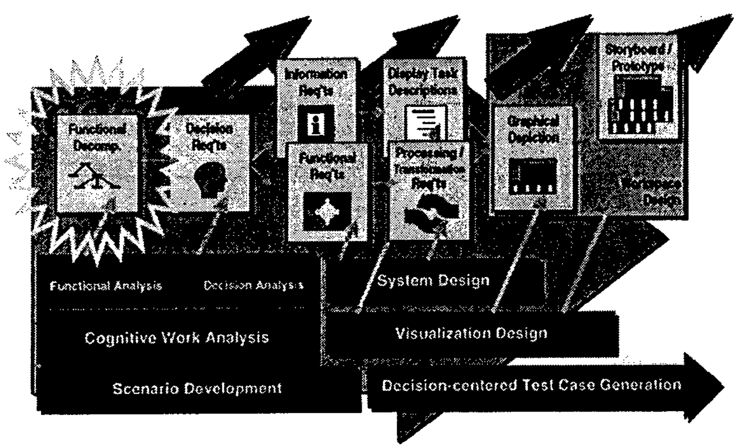
Building the Functional Abstraction Hierarchy

Graphical Modeling Tool for:

- ➤ Capturing goal/process relationships – supporting goals, constraints, side-effects – within an abstraction hierarchy;
- ➤ Iterating and refining the model;

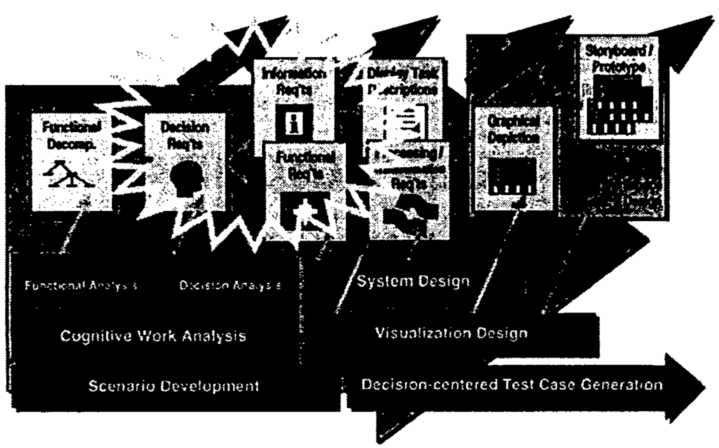


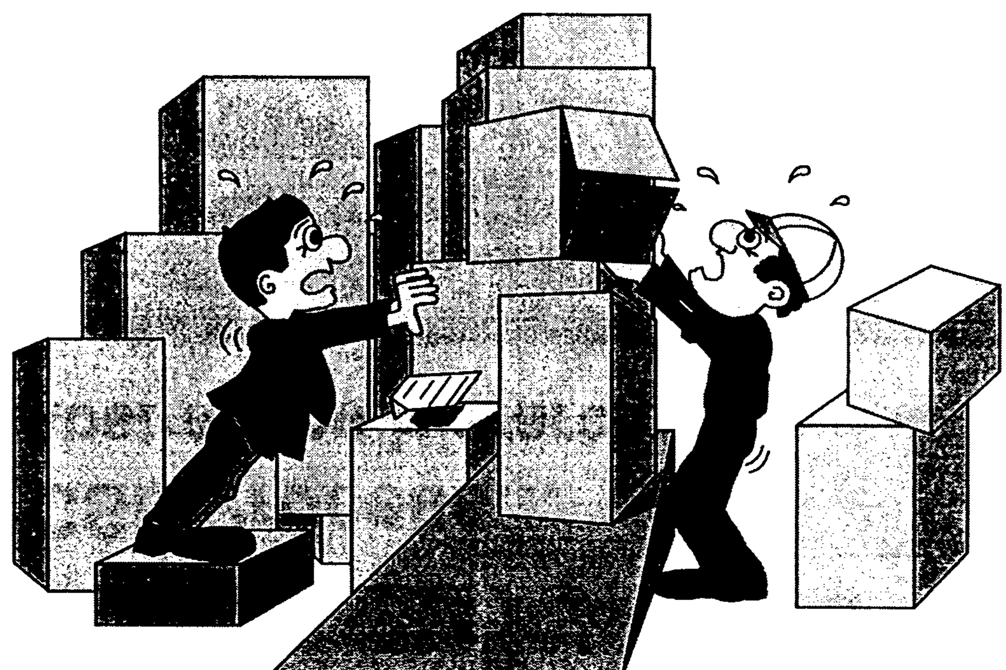
Serving as the starting point to the analysis!



Logica Carnegie Group

Activity Analysis





Control Task Analysis...
The Decision Ladder:

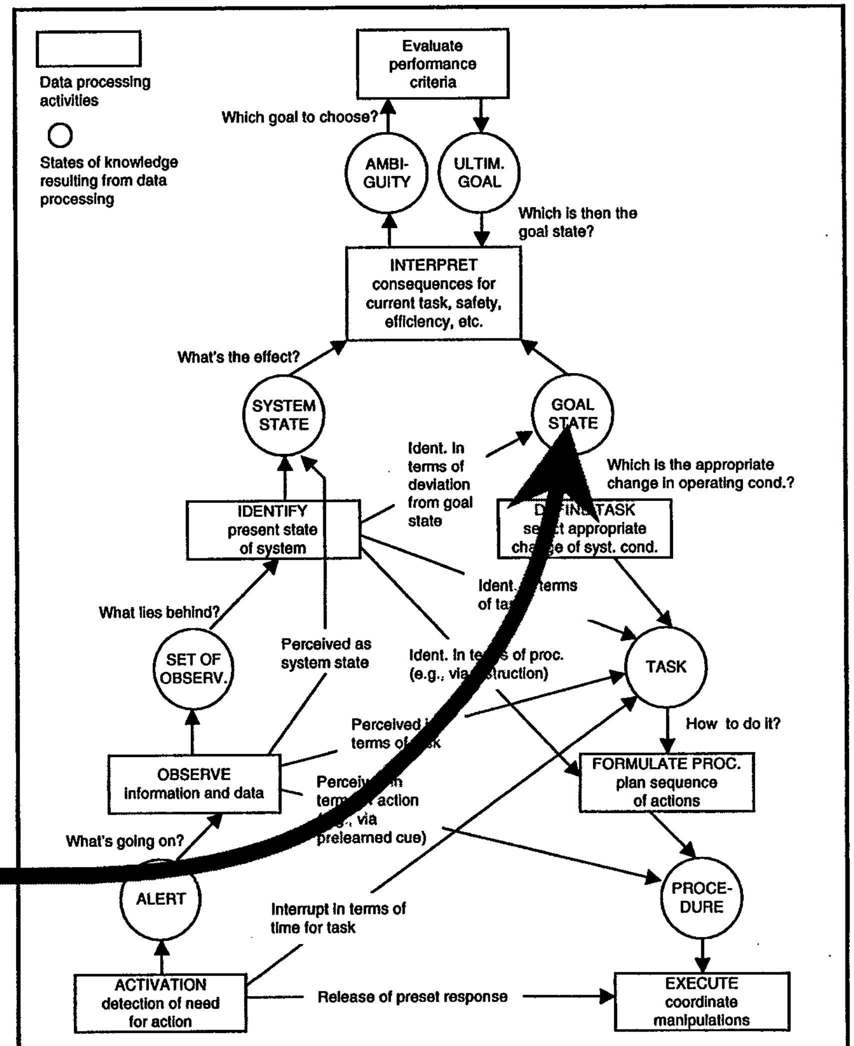
A non-linear, opportunistic modeling tool (template) containing:

- ➤ Information processing activities;
- ➤ States of Knowledge;
- ➤ Multiple entry and exit points;
- ➤ Opportunistic movement via “leaps” and “shunts”;

Can be descriptive or formative;

Provides a systematic way of parsing verbal protocol data;

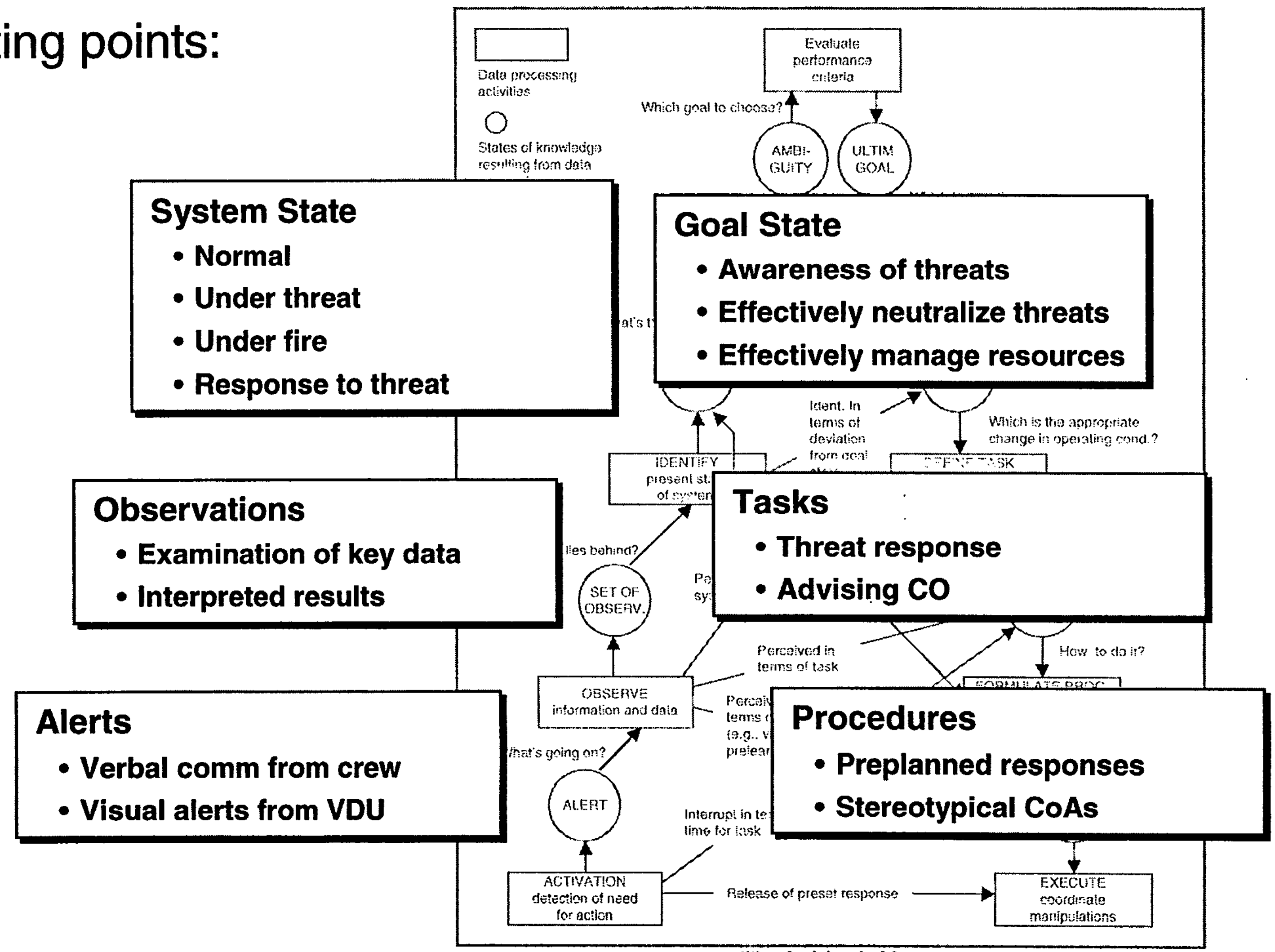
Goal State can be decomposed via the Abstraction Hierarchy;



The decision ladder

Control Task Analysis...
The Decision Ladder:

Multiple starting points:



The decision ladder

Control Task Analysis...
Decision Ladder - 1:

Goal State:

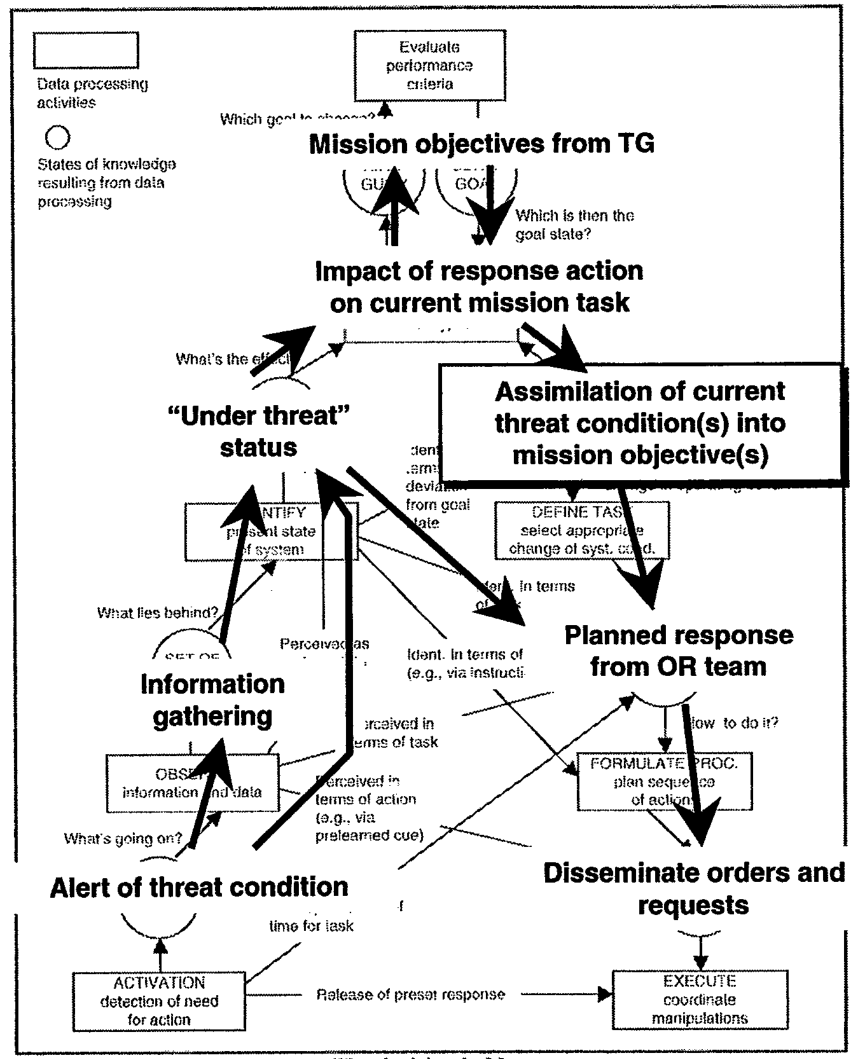
- ➤ Assimilation of current threat conditions into mission objectives

Critical Decisions:

- ➤ Determine impact of response options on mission objectives

Information Requirements:

- ➤ Contact status (e.g., threat classification)
- ➤ Threat characteristics (e.g., type, location, capability)
- ➤ Planned response actions (e.g., maneuvers, hardkill / softkill actions)



The decision ladder

Control Task Analysis...
Decision Ladder - 2:

Goal State:

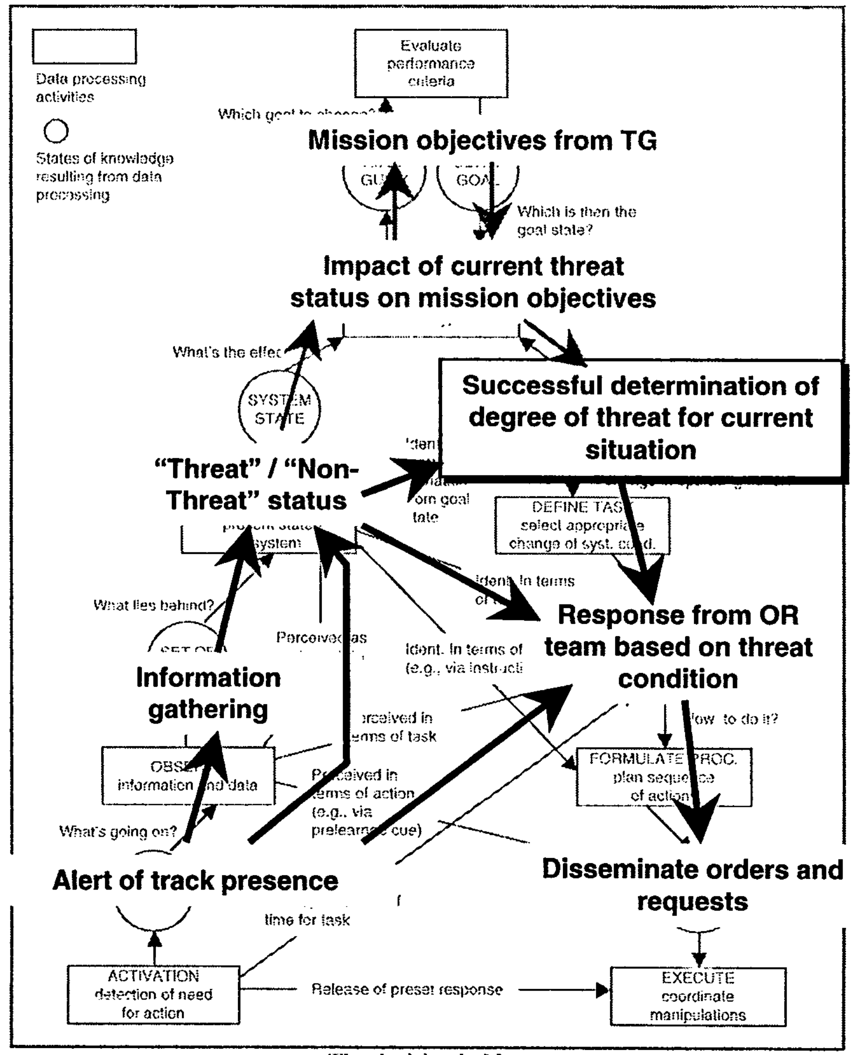
- ➤ Successful determination of degree of threat for current situation.

Critical Decisions:

- ➤ Determine friendly / enemy status of given contact;
- ➤ Determine intent of given contact.

Information Requirements:

- ➤ Contact characteristics (e.g., type, location, capability);
- ➤ Individual contact behavior (e.g., COA, CPA);
- ➤ Group contact behavior (e.g., coordinated actions.



The decision ladder

Control Task Analysis...
Decision Ladder - 3:

Goal State:

- ➤ Successfully respond to pre-planned ("Zippo") conditions.

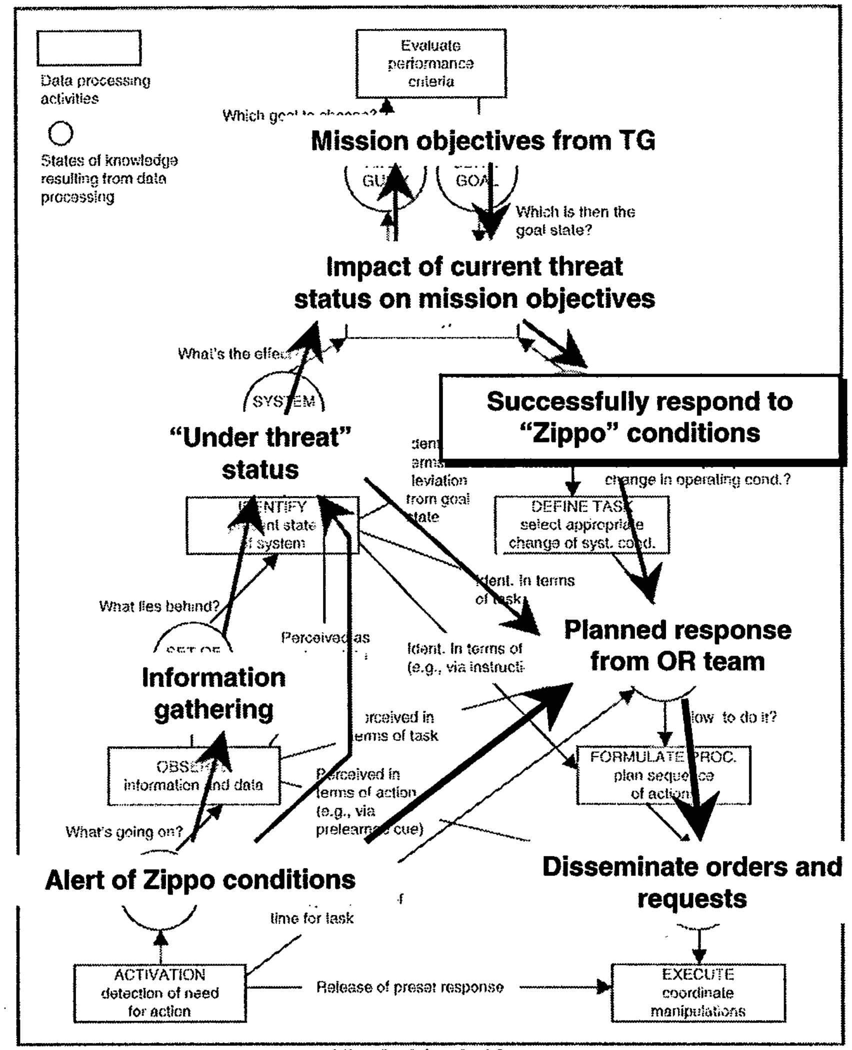
Critical Decisions:

- ➤ Verify determination of Zippo conditions (i.e., match characteristics of threat with known database).
- ➤ Execute appropriate response to particular threat.

Information Requirements:

- ➤ Threat characteristics (e, g., location, speed, profile, etc).
- ➤ Zippo pre-planned database.

An example of preset response to alert.

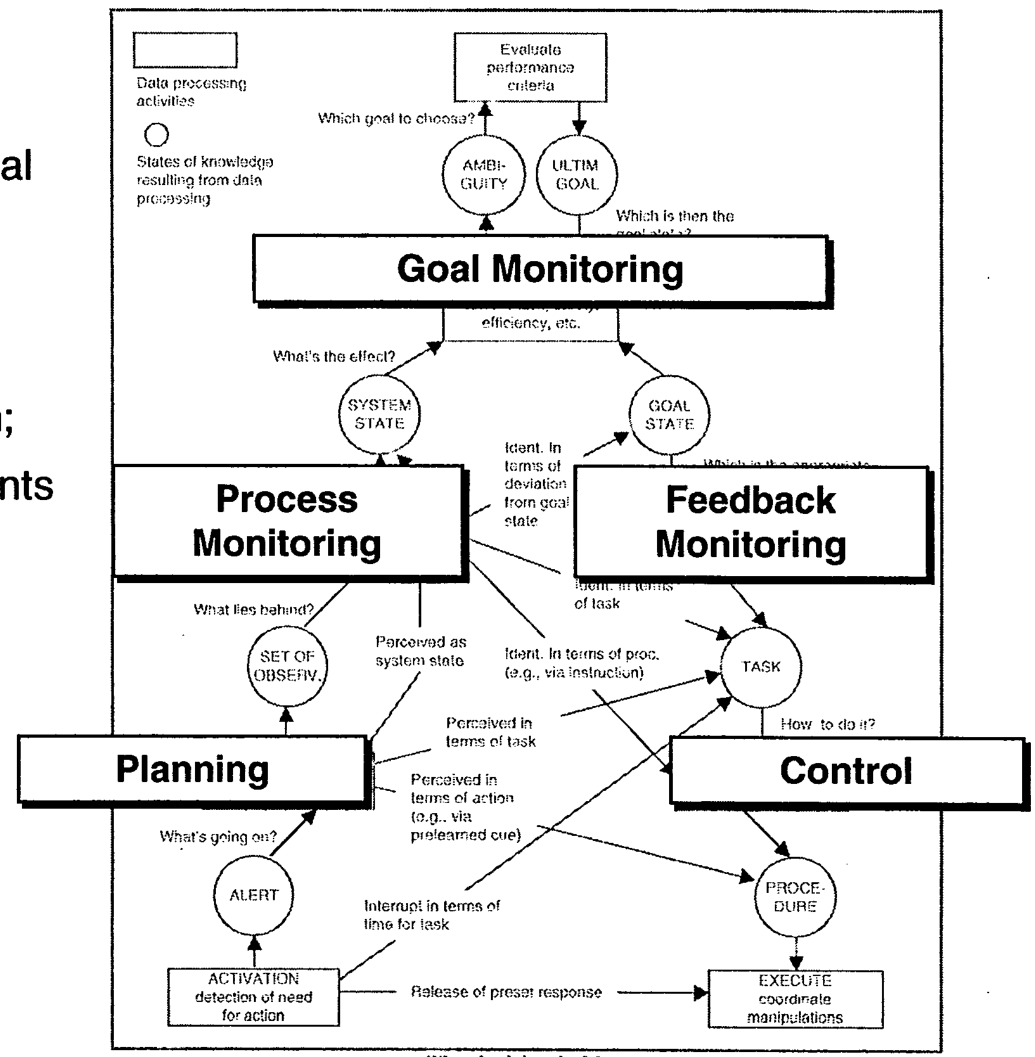


The decision ladder

Tool Support for the CSE Process...
Support for the multiple levels of analysis

Tool support for:

- ➤ Augmenting the Functional Abstraction Hierarchy with Critical Cognitive Demands (decisions) (based on the work of Roth and Woods);
- ➤ Documenting Information Requirements for each Decision;
- ➤ Defining supporting Data Elements comprising the Information Requirements.

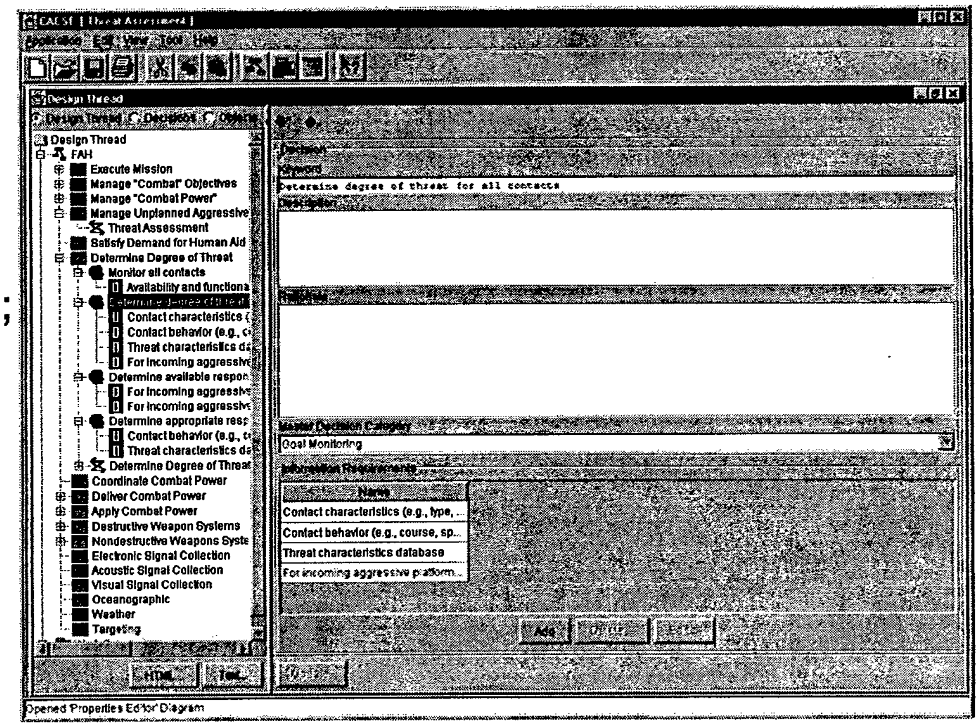


The decision ladder

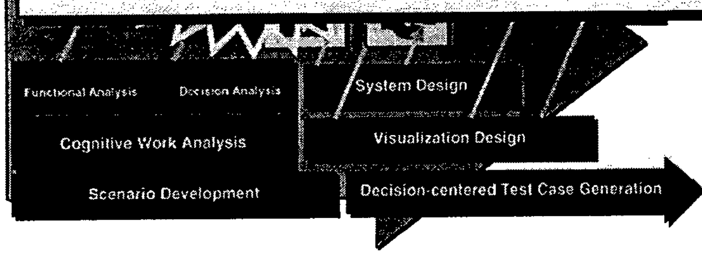
Tool Support for the CSE Process...
Support for the multiple levels of analysis

Tool support for:

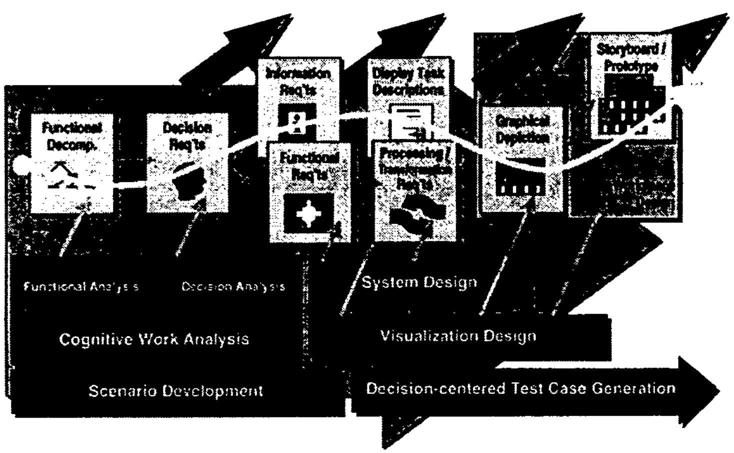
- Documenting the supporting downstream artifacts:
  - Goals;
    - Processes;
      - Cognitive Demands (decisions);
        - Information Requirements;
          - Data Elements;
- Hierarchical tree becomes the organizational structure.

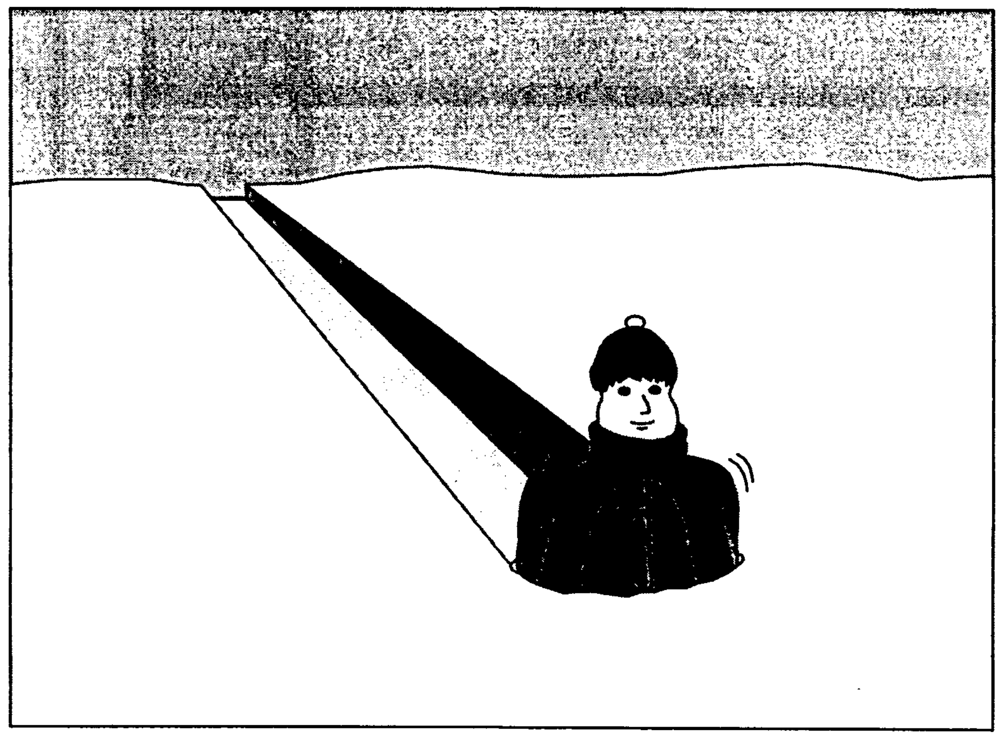


Builds on the underlying framework provided by the FAH!



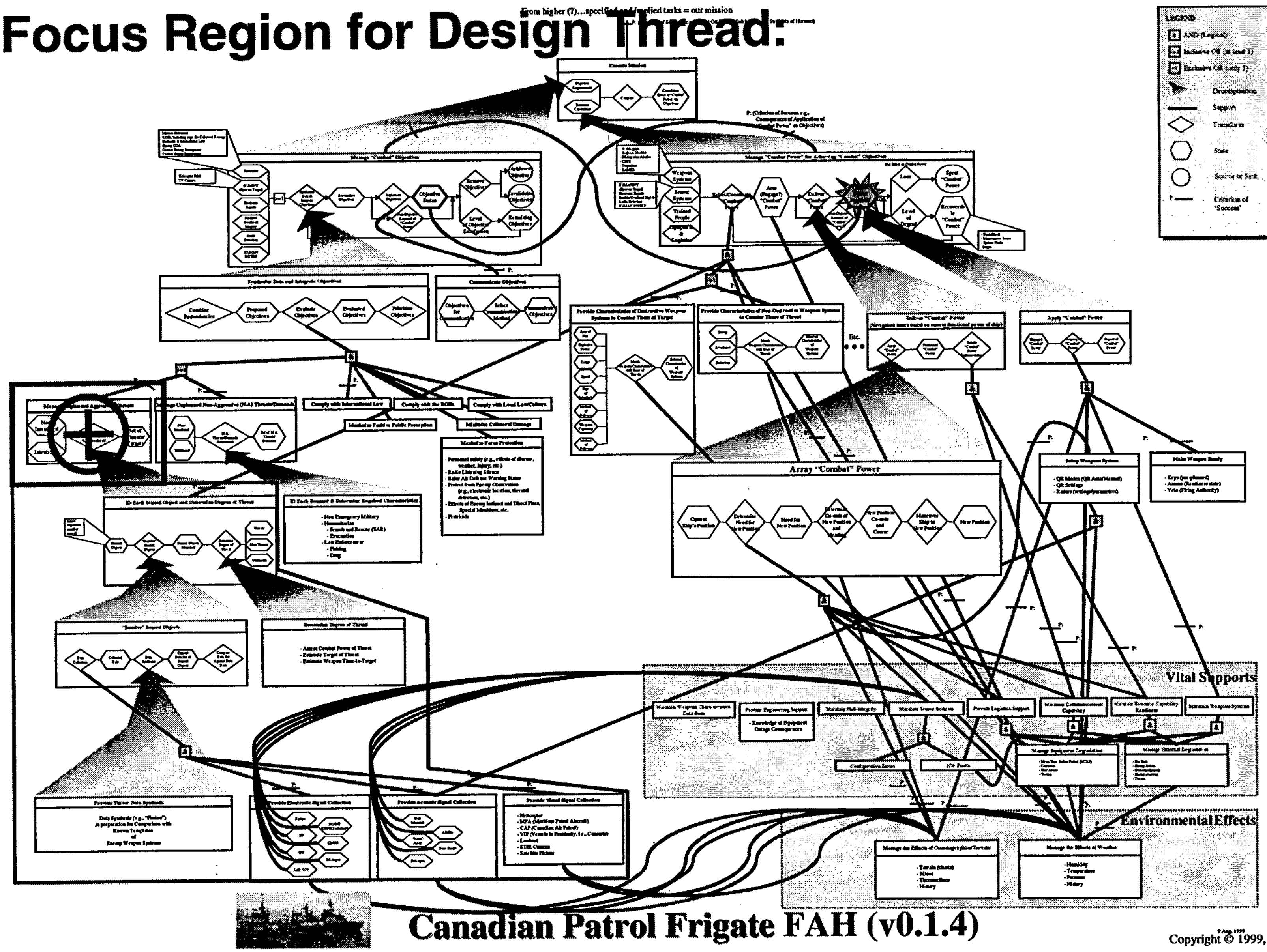
Design Thread





"Manage Unplanned Aggressive Threats"

Focus Region for Design Thread:



Canadian Patrol Frigate FAH (v0.1.4)

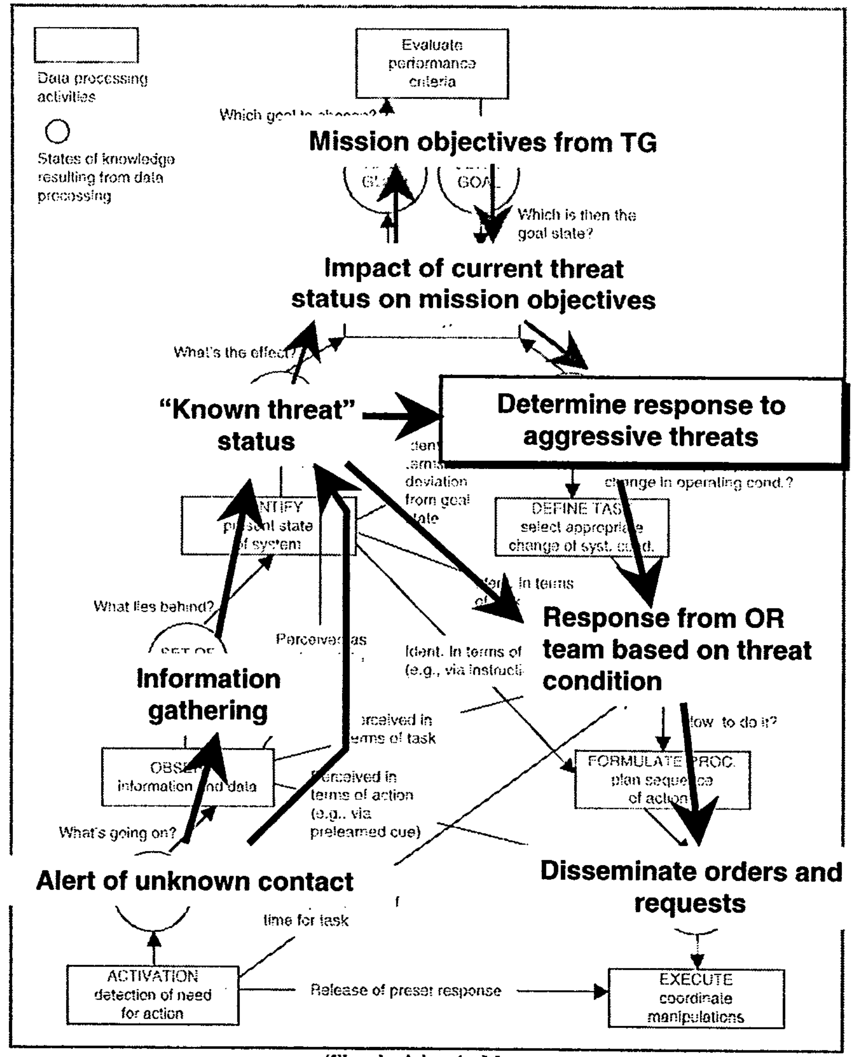
"Manage Unplanned Aggressive Threats" Decision Ladder:

Goal State:

- ➤ Successfully respond to all unplanned aggressive threats.

Critical Decisions:

- ➤ Monitor all contacts;
- ➤ Determine degree of threat for all contacts;
- ➤ Determine available response time for all aggressive threats;
- ➤ Determine appropriate response to aggressive threats;



The decision ladder

"Manage Unplanned Aggressive Threats"
Supporting Information Requirements:

Goal State:

- Successfully respond to all unplanned aggressive threats.

Critical Decisions:

- *Process Monitoring:*
  Monitor all contacts (D1);
- *Goal Monitoring:*
  Determine degree of threat for all contacts (D2);
- Determine available response time for all aggressive threats (D3);
- *Control:*
  Determine appropriate response, including priority and order of response, to aggressive threats (D4).

Information Requirements:

- Availability and functionality of contact sensing equipment; (D1)
- Contact characteristics (e.g., type, location, capability); (D2)
- Contact behavior (e.g., course, speed, response to warnings); (D2, D4)
- For incoming aggressive weapons; time until impact; (D3)
- For incoming aggressive platforms; time until likely weapon release (e.g., CPA); (D2, D3)
- Contact characteristics vs. Degree of Threat vs. Characteristics of On-board weapons database; (D2, D4)

Note: While this list of information requirements is necessary, it is recognized to be incomplete.

Decision-Centered Visualization...

Design Concept:

A functionally-organized visualization (i.e., organized around *time*) of the *engagement state* of those contacts identified as potential aggressive threats to provide a means to achieve Situation and Threat Assessment and Response Management approach across all of the warfare worlds.

- Functional distance to threat defined as a function of:
  - time to closest point of approach, and
  - CPA transformed into units of time;
- Threat state defined as a function of:
  - contact classification (i.e., unknown / suspect / hostile);
  - threat identification (i.e., missile / aircraft / torpedo);
  - threat ranking from automated system (DREV work);
  - planned response (i.e., weapon of choice);
    - engagement status (i.e., assigned / engaged / fired);
    - availability of response (i.e., time until threat is in arc-of-fire of weapon of choice);

Decision-Centered Visualization...
Design Concept (cont):

- Based on these parameters, critical contacts to focus on include:
  - unresolved contact temporally close to ship;
  - hostile / unengaged temporally contact close to ship;
  - hostile contact outside ship's arc of fire;
- At the highest level, need a clear indication of the presence of threat(s)!

Additionally, this design should:

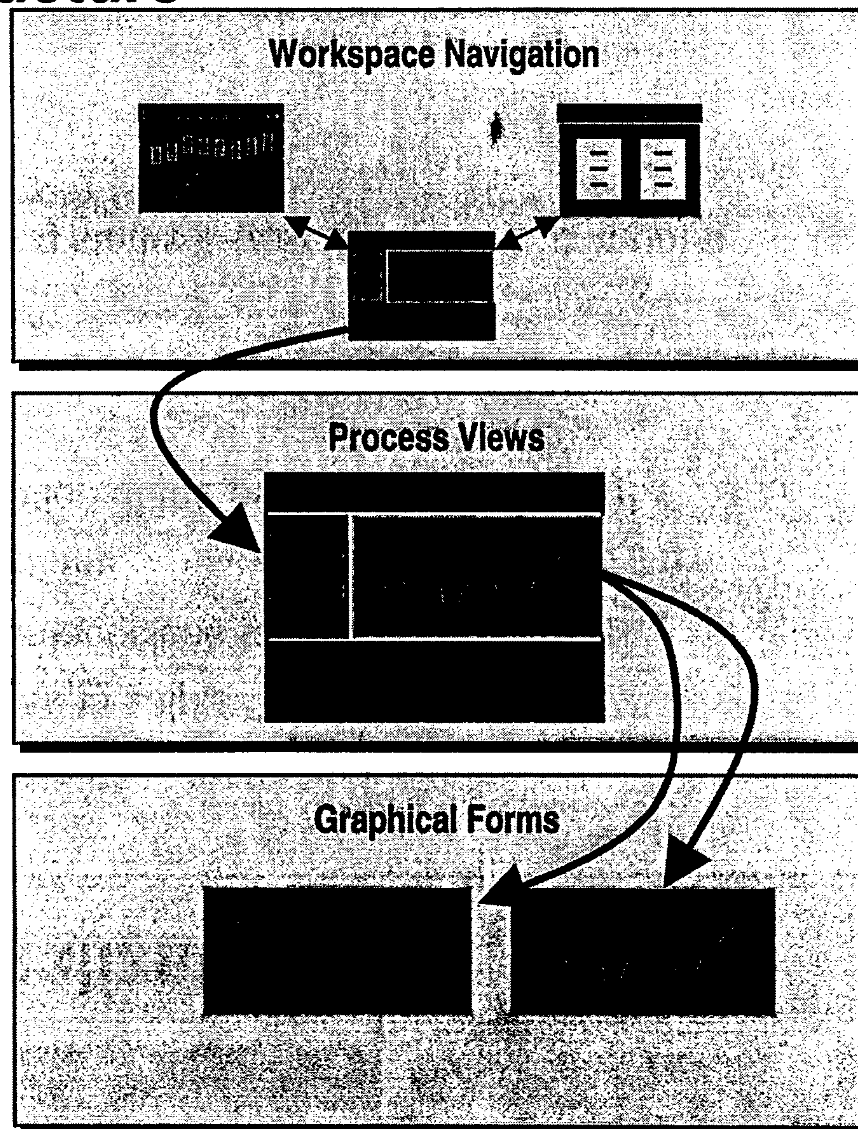
- Convey Data Acquisition and Processing status — are the sensor, computer systems, and OR staff functioning correctly?
  (outside of scope of current project)
- Coordinate with Geo-plot display:
  - Weapon range projections (both friendly and enemy)
  - Sensor system ranges
  - Waterspace management -- where can I / should I go vs. where can / probably will the enemy go?

Tool Support for the CSE Process...
Defining User Interface Structure

Graphical Tool for:

- Defining workspace navigation (total replacement of one coordinated view with another);
- Defining coordinated views (information that will be seen in parallel);
- Defining graphical forms (multi-purpose/reusable components) that comprise the coordinated views.

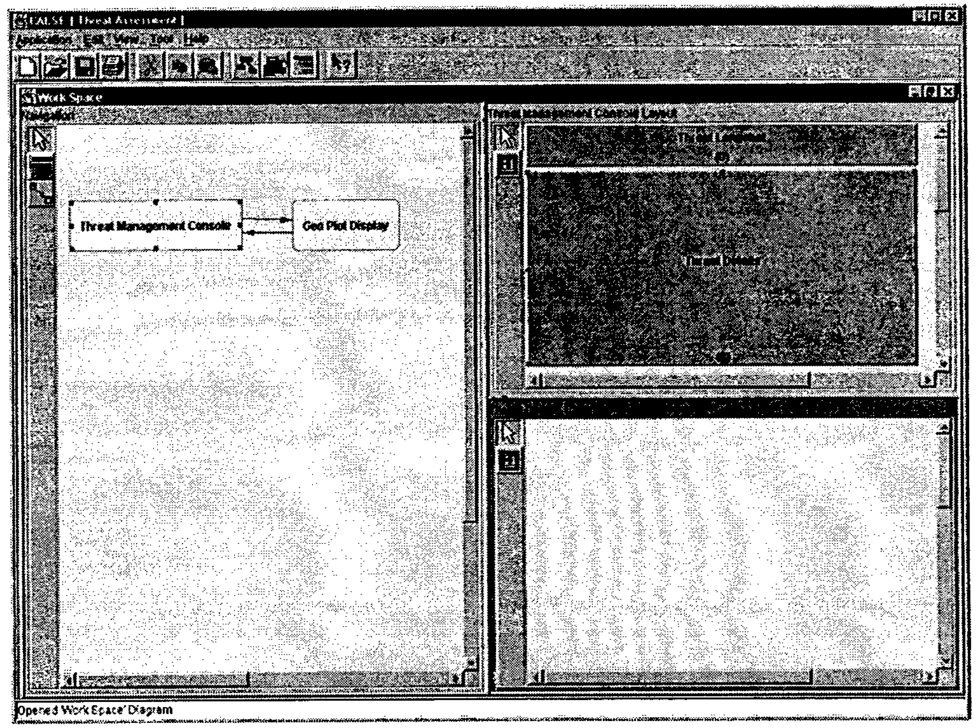
Supports multiple levels of visualization design.



Tool Support for the CSE Process...
Defining User Interface Structure

Graphical Tool for:

- ➤ Defining workspace navigation (total replacement of one coordinated view with another);
- ➤ Defining coordinated views (information that will be seen in parallel);
- ➤ Defining graphical forms (multi-purpose/reusable components) that comprise the coordinated views.



Provides a Basis for Systematic Construction of Effective Visual Workspaces.

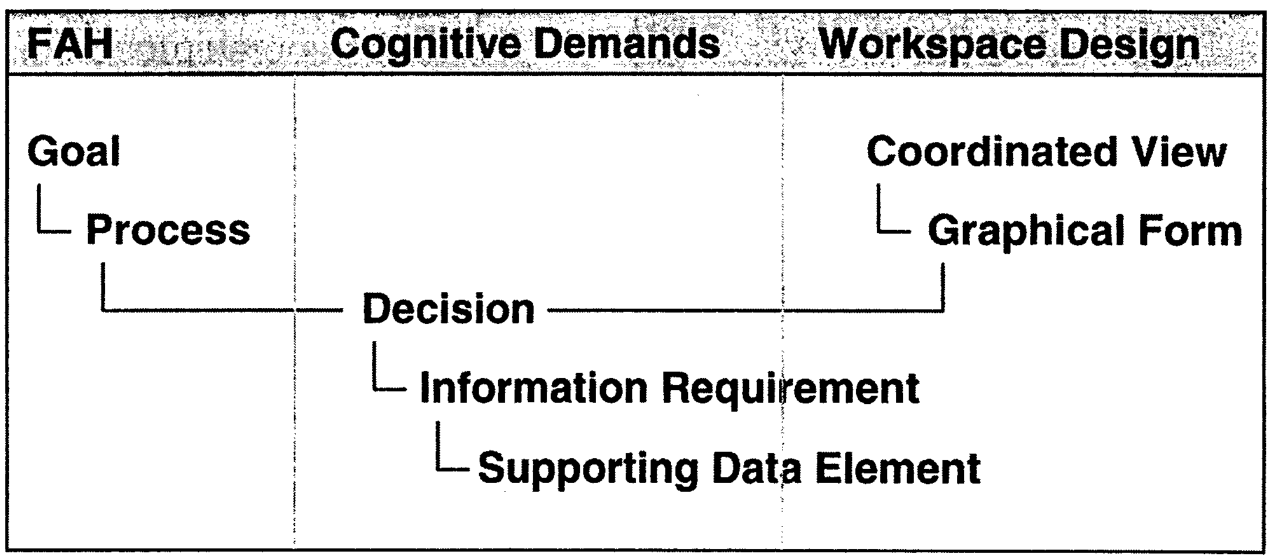
Logica Carnegie Group

Tool Support for the CSE Process...

End-to-end support for the entire "Design Thread"

A tool for:

- Defining functional "coverage" for each display by assigning decisions to displays;
- Decisions become the "linchpin" connecting goals to displays;

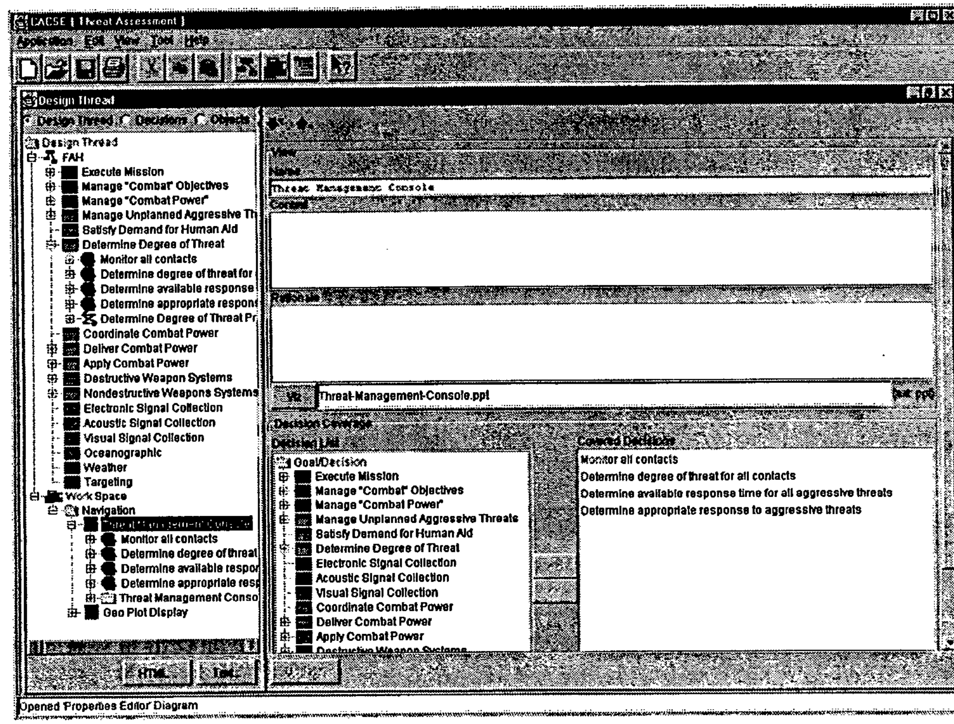


Tool Support for the CSE Process...

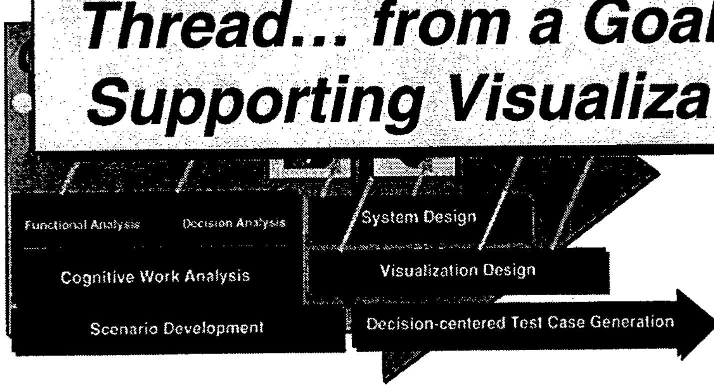
End-to-end support for the entire "Design Thread"

A tool for:

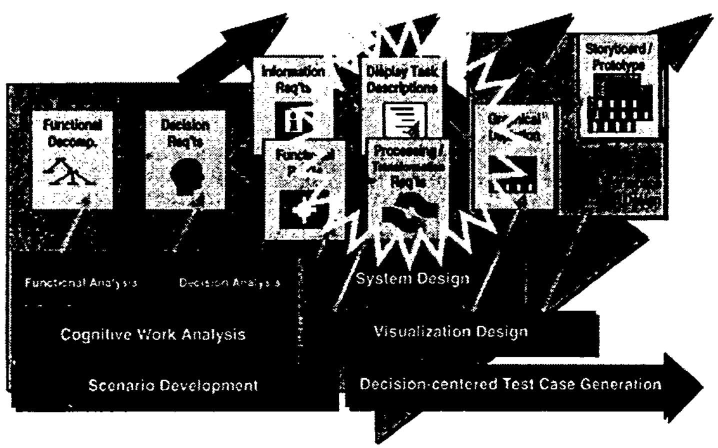
- ➤ Defining functional "coverage" for each display by assigning decisions to displays;
- ➤ Decisions become the "linchpin" connecting goals to displays;



Provides a Complete Thread... from a Goal to a Supporting Visualization.



Underlying Design Issues



Decision-Centered Visualizations...

Design Goals:

Maintain the "big picture":

- Provide the ORO with an overview of the situation to avoid cognitive tunnel vision on a particular aspect of the problem. This may include alerts with respect to significant events outside of the ORO's current field of view.

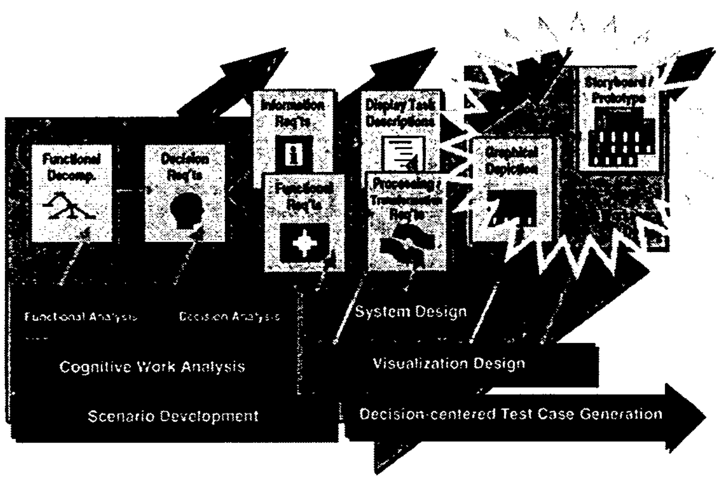
Provided a coordinated functional - physical display suite:

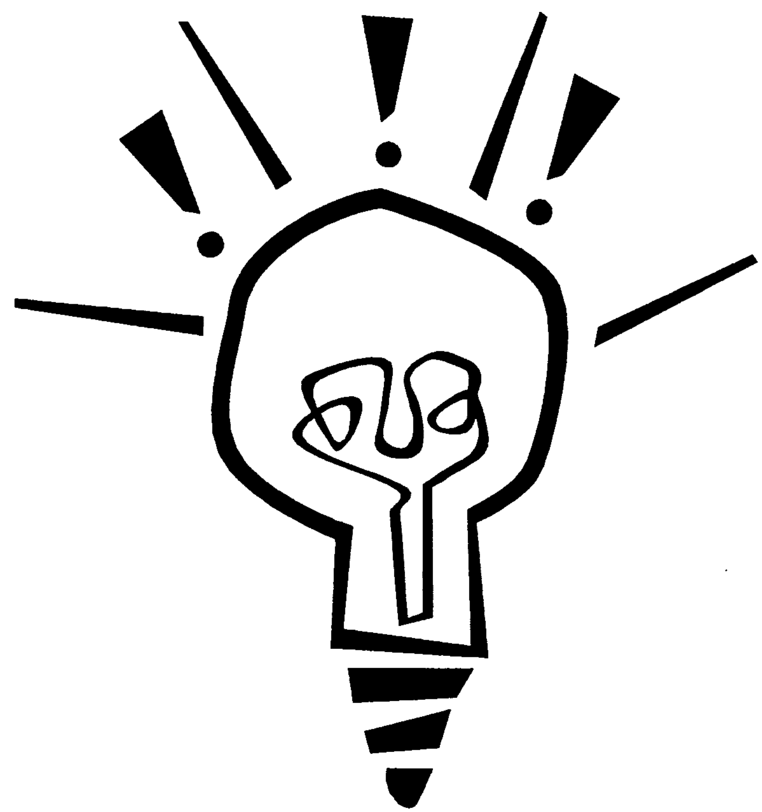
- Provide the ORO with a combination of functional and physical (geographical) information that will, together, provide a coherent representation of the state of affairs. One key is to allow the ORO to effortlessly transition from the physical to the functional information.
- In addition, the MMI design goal is to ***not*** add additional information onto an already crowded main display for the ORO.

Integrate the three worlds:

- Provide the ORO with a mechanism to integrate sub-surface, surface, and air worlds with respect to threats and resources. This will support the ORO as the "integration point" of these three worlds.

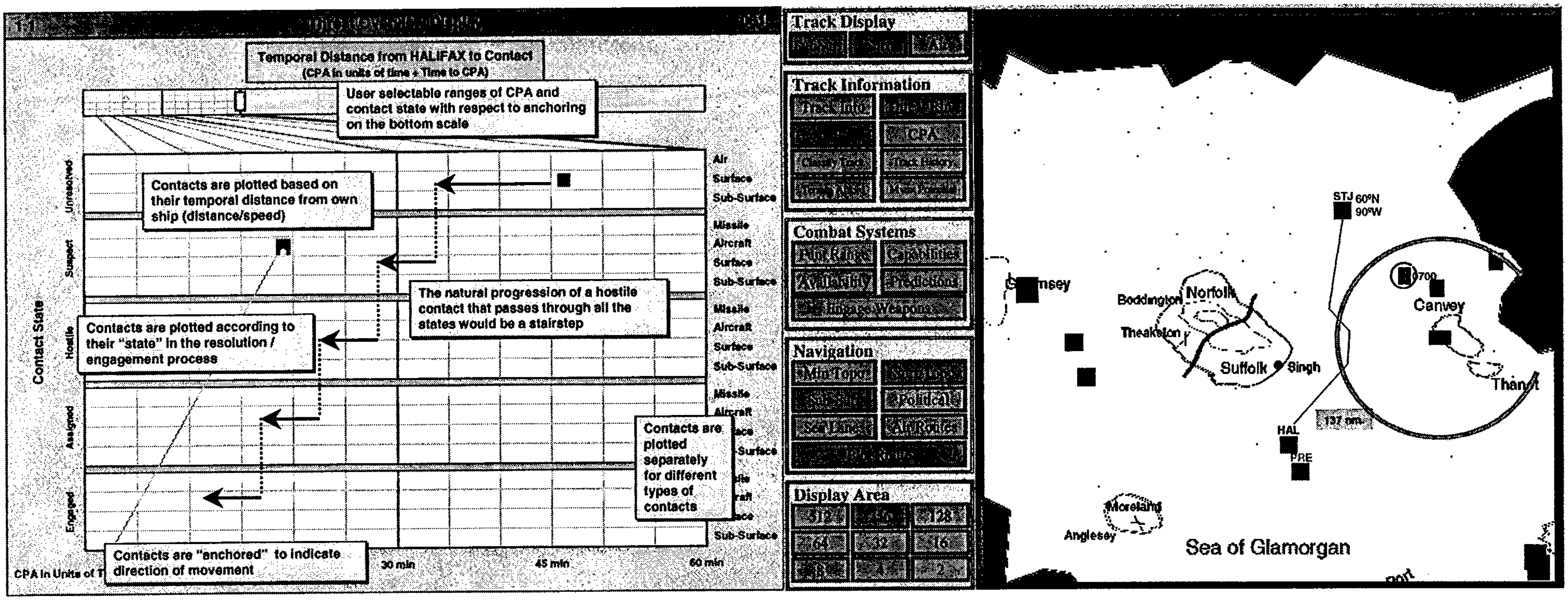
Visualization Concept





Decision-Centered Visualizations...

Display Interaction:

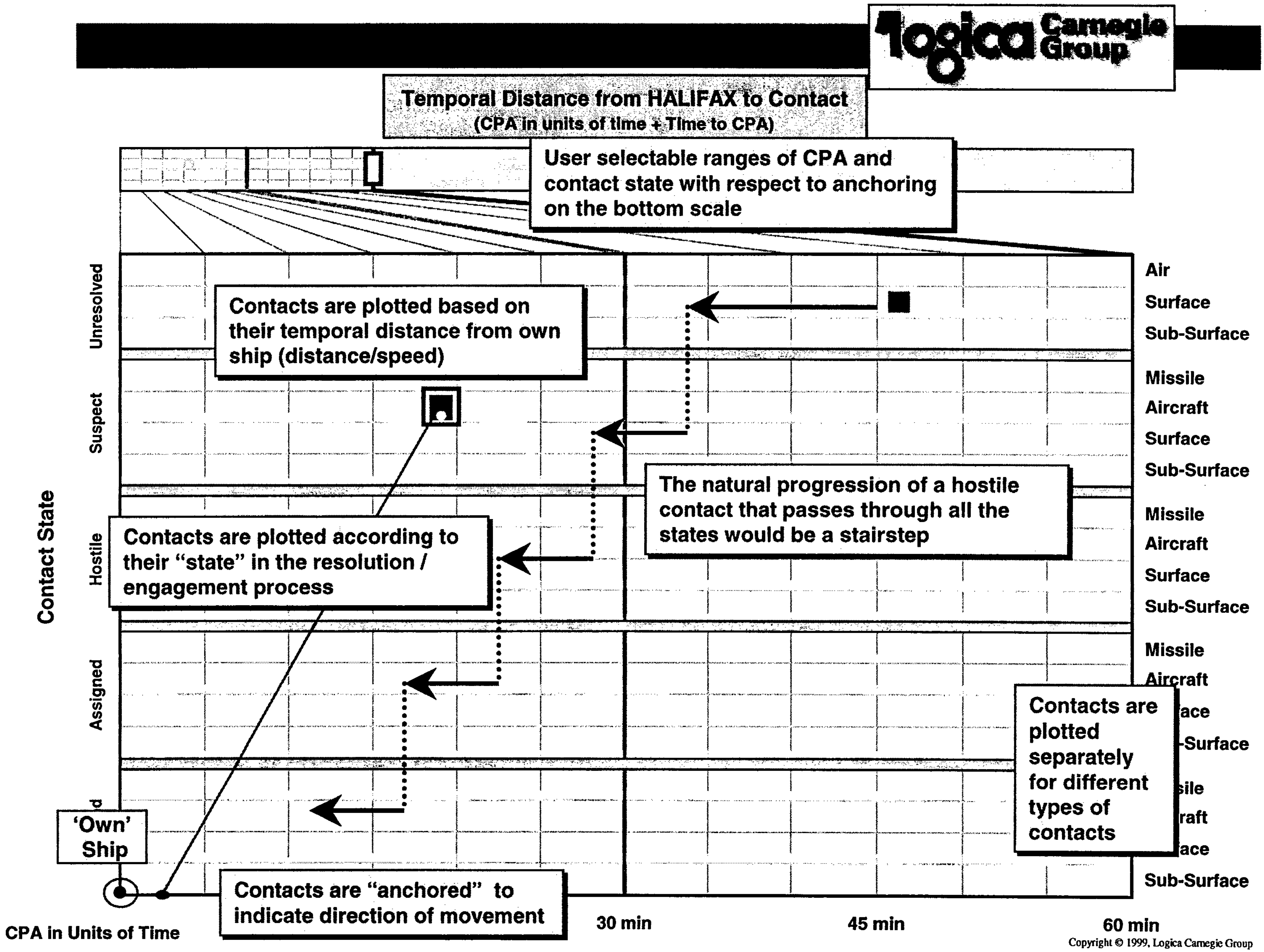


Selecting contact highlights its representation in threat overview display

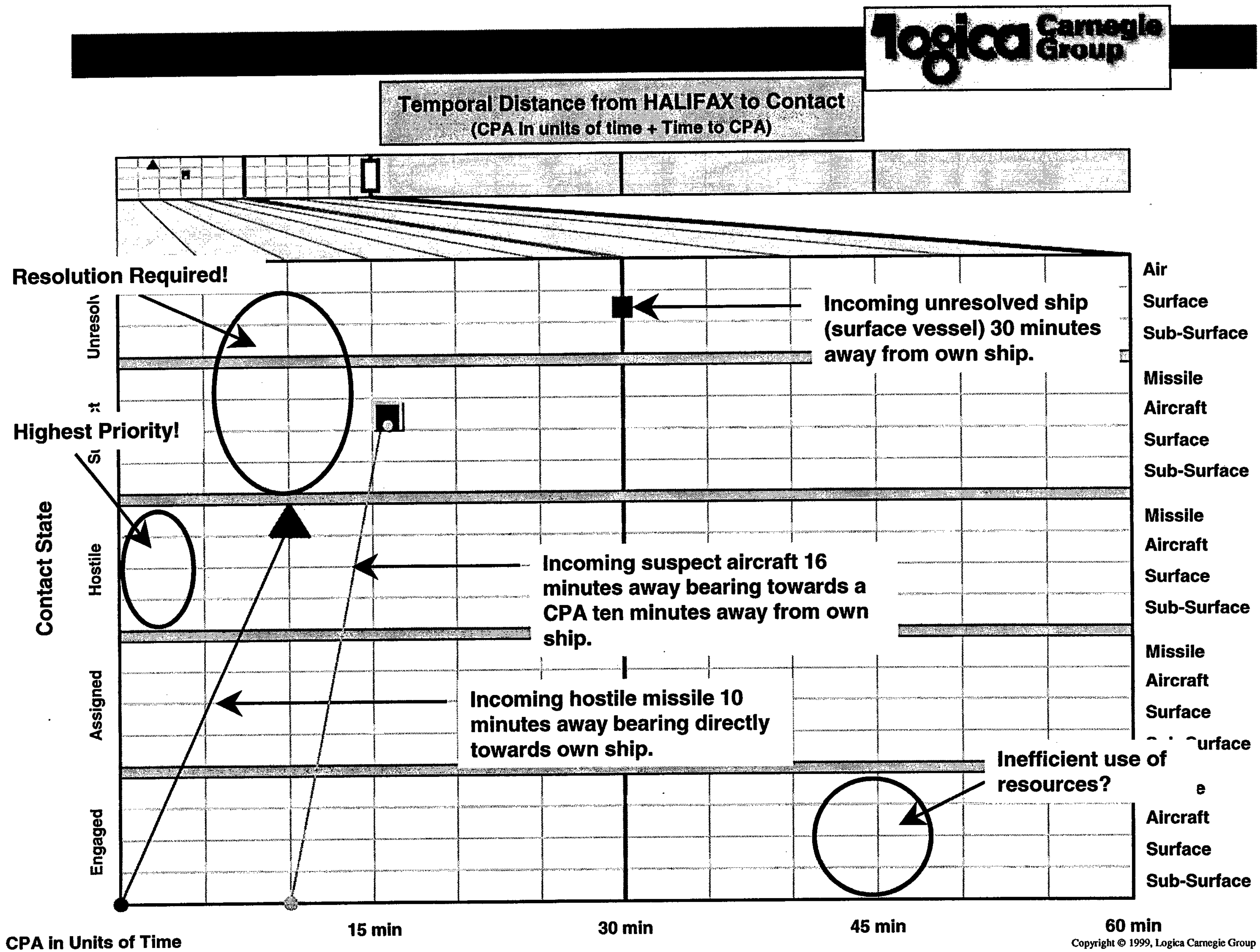
Selecting contact changes zoom in geo-plot display to include contact

Changing zoom level is reflected in the threat overview display

Can hook a contact via either display



Logica Carnegie Group
Temporal Distance from HALIFAX to Contact
(CPA in units of time + Time to CPA)
User selectable ranges of CPA and contact state with respect to anchoring on the bottom scale
Contact State
Unresolved
Suspect
Hostile
Assigned
'Own' Ship
Air
Surface
Sub-Surface
Missile
Aircraft
Surface
Sub-Surface
Missile
Aircraft
Surface
Sub-Surface
Missile
Aircraft
Sub-Surface
Contacts are plotted based on their temporal distance from own ship (distance/speed)
The natural progression of a hostile contact that passes through all the states would be a stairstep
Contacts are plotted according to their "state" in the resolution / engagement process
Contacts are plotted separately for different types of contacts
Contacts are "anchored" to indicate direction of movement
CPA in Units of Time
30 min
45 min
60 min

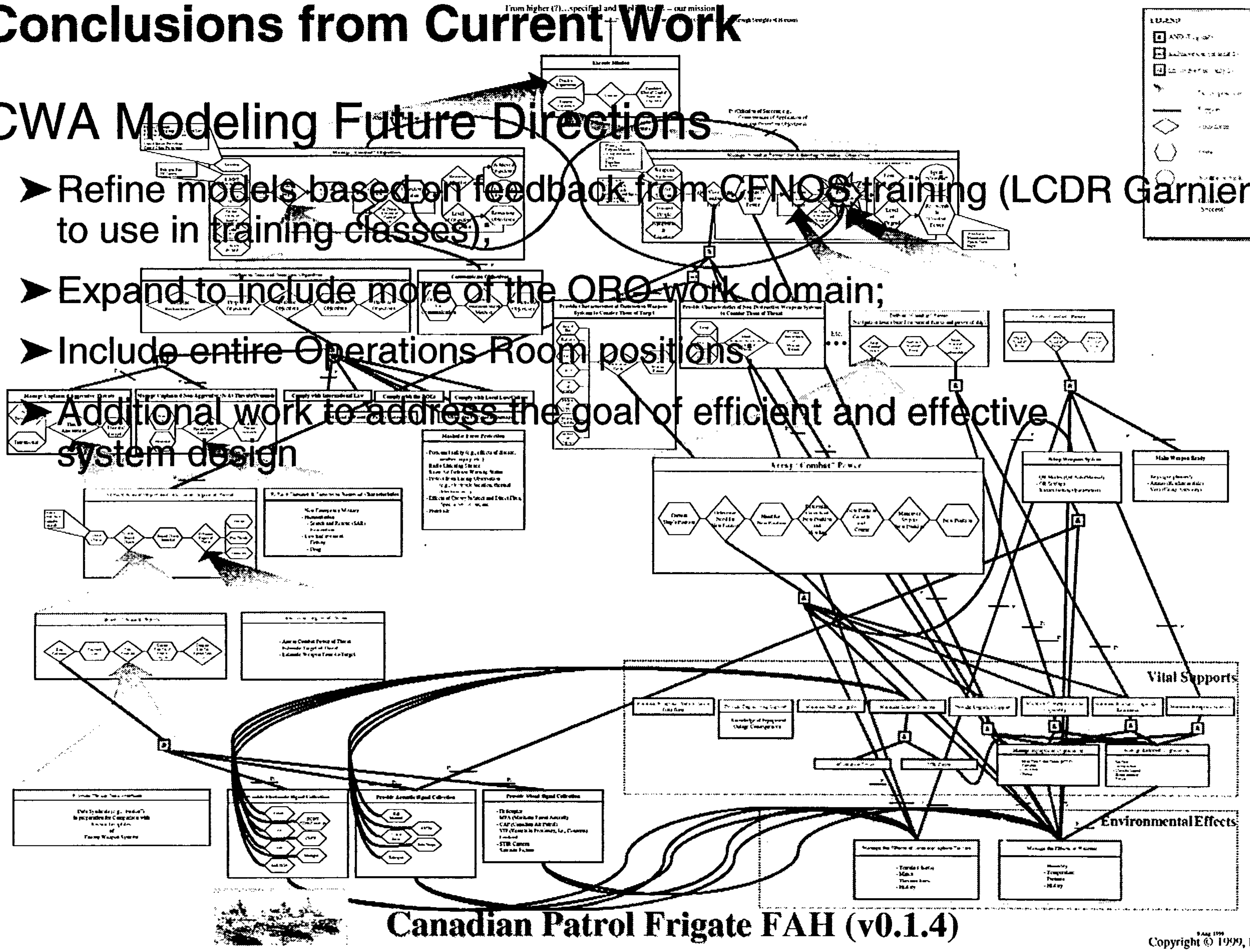

Temporal Distance from HALIFAX to Contact
(CPA in units of time + Time to CPA)
Resolution Required!
Highest Priority!
Contact State
Unresol
Hostile
Assigned
Engaged
Incoming unresolved ship (surface vessel) 30 minutes away from own ship.
Incoming suspect aircraft 16 minutes away bearing towards a CPA ten minutes away from own ship.
Incoming hostile missile 10 minutes away bearing directly towards own ship.
Inefficient use of resources?
Air
Surface
Sub-Surface
Missile
Aircraft
Surface
Sub-Surface
Missile
Aircraft
Surface
Sub-Surface
Missile
Aircraft
Surface
Aircraft
Surface
Sub-Surface
CPA in Units of Time
15 min
30 min
45 min
60 min

Expansion of the Cognitive Work Analysis...

Conclusions from Current Work

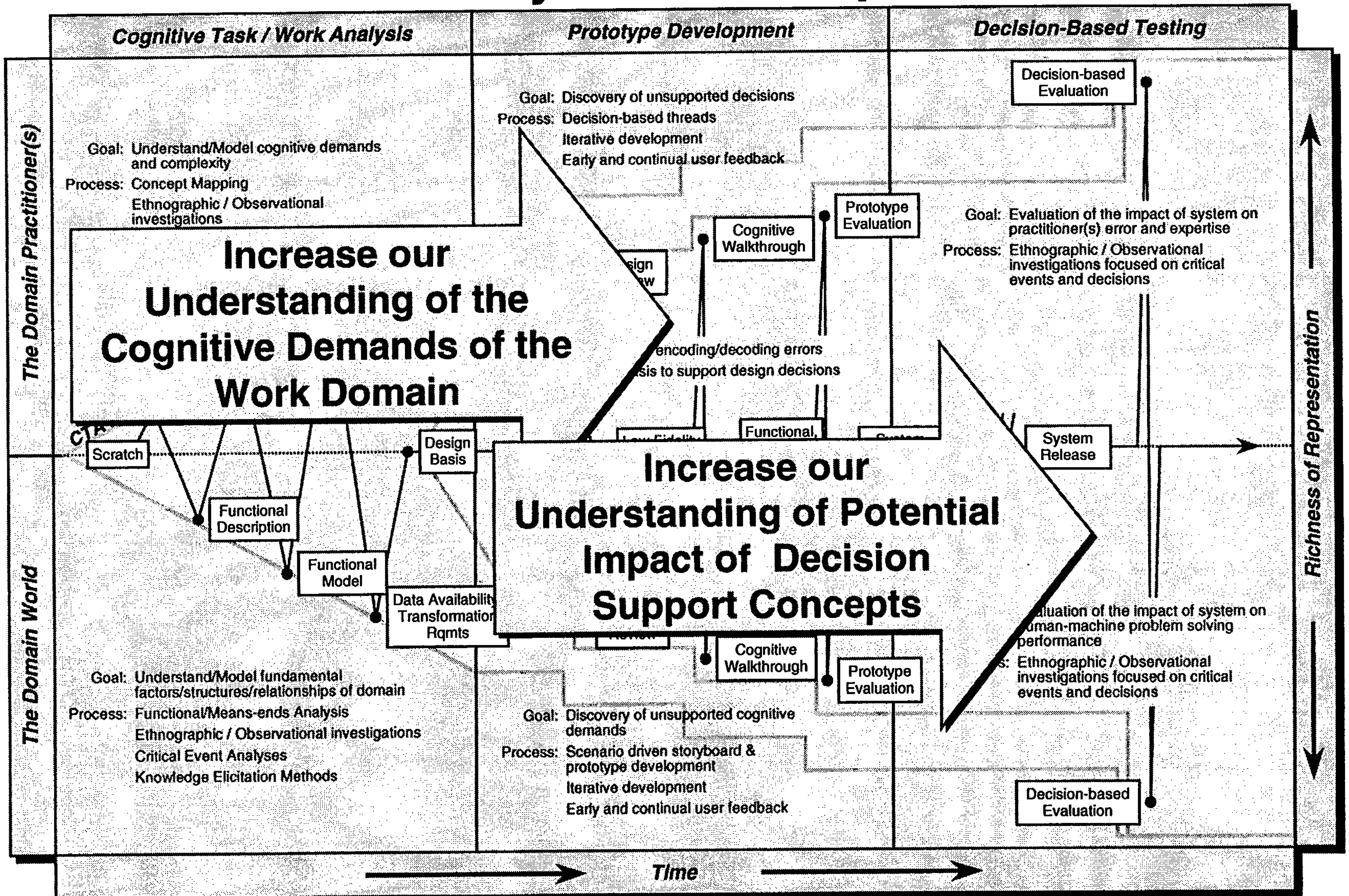
CWA Modeling Future Directions

- Refine models based on feedback from CFNOS training (LCDR Garnier to use in training classes);
- Expand to include more of the ORO work domain;
- Include entire Operations Room positions;
- Additional work to address the goal of efficient and effective system design



Canadian Patrol Frigate FAH (v0.1.4)

Directions for Future Work...
From a CSE-Based System Development Process



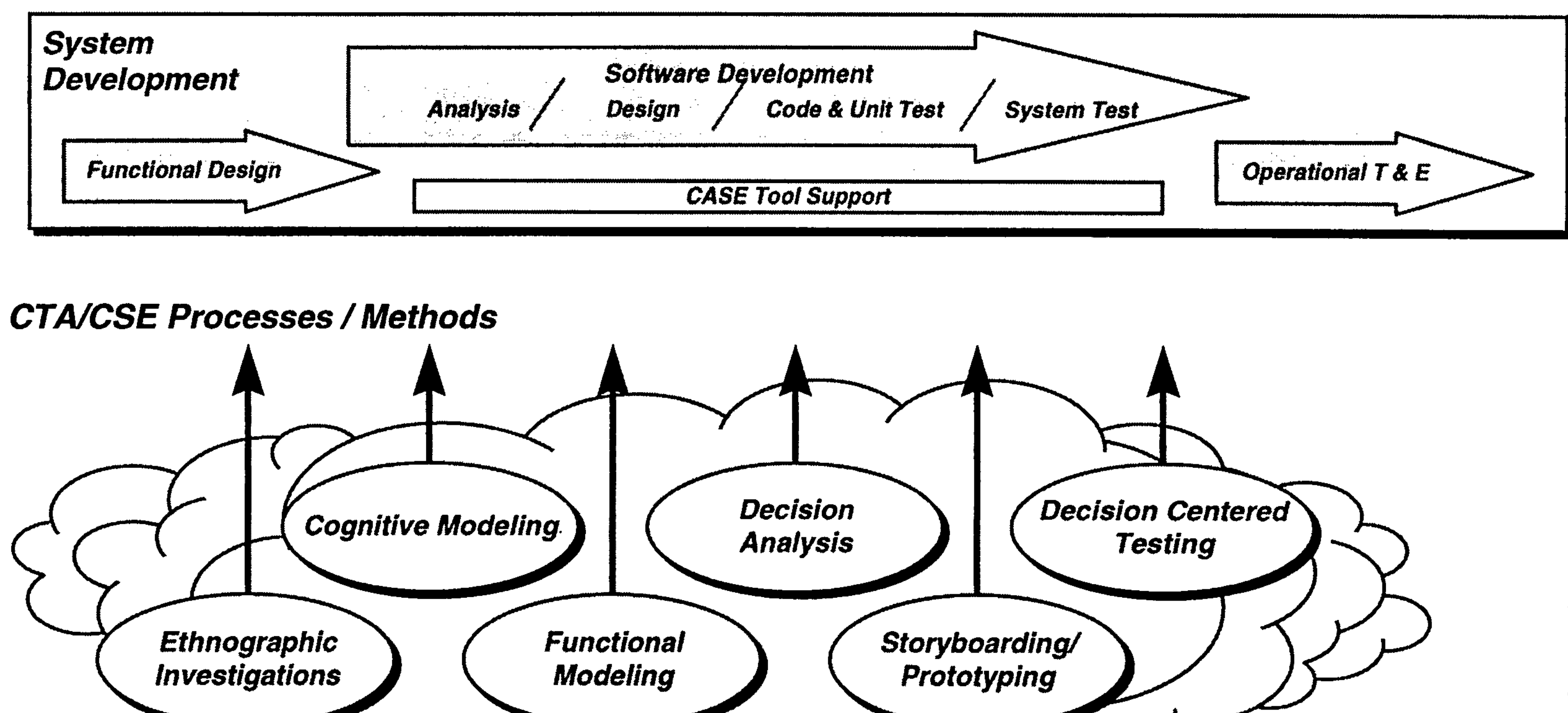
VII. CACSE FINAL PRESENTATION MATERIAL

CACSE

A Computer-Aided Cognitive Systems Engineering Tool to Facilitate the Development of Effective Decision Support Systems

Logica Carnegie Group
Advanced Decision Support Business Unit

The Problem



A loose collection of Cognitive Task Analysis (CTA) and Cognitive Systems Engineering (CSE) methods and approaches that are failing to live up to full potential due to lack of integration with system development process.

CSE-based system design…
Key Obstacles

Lack of tool support for the CSE process

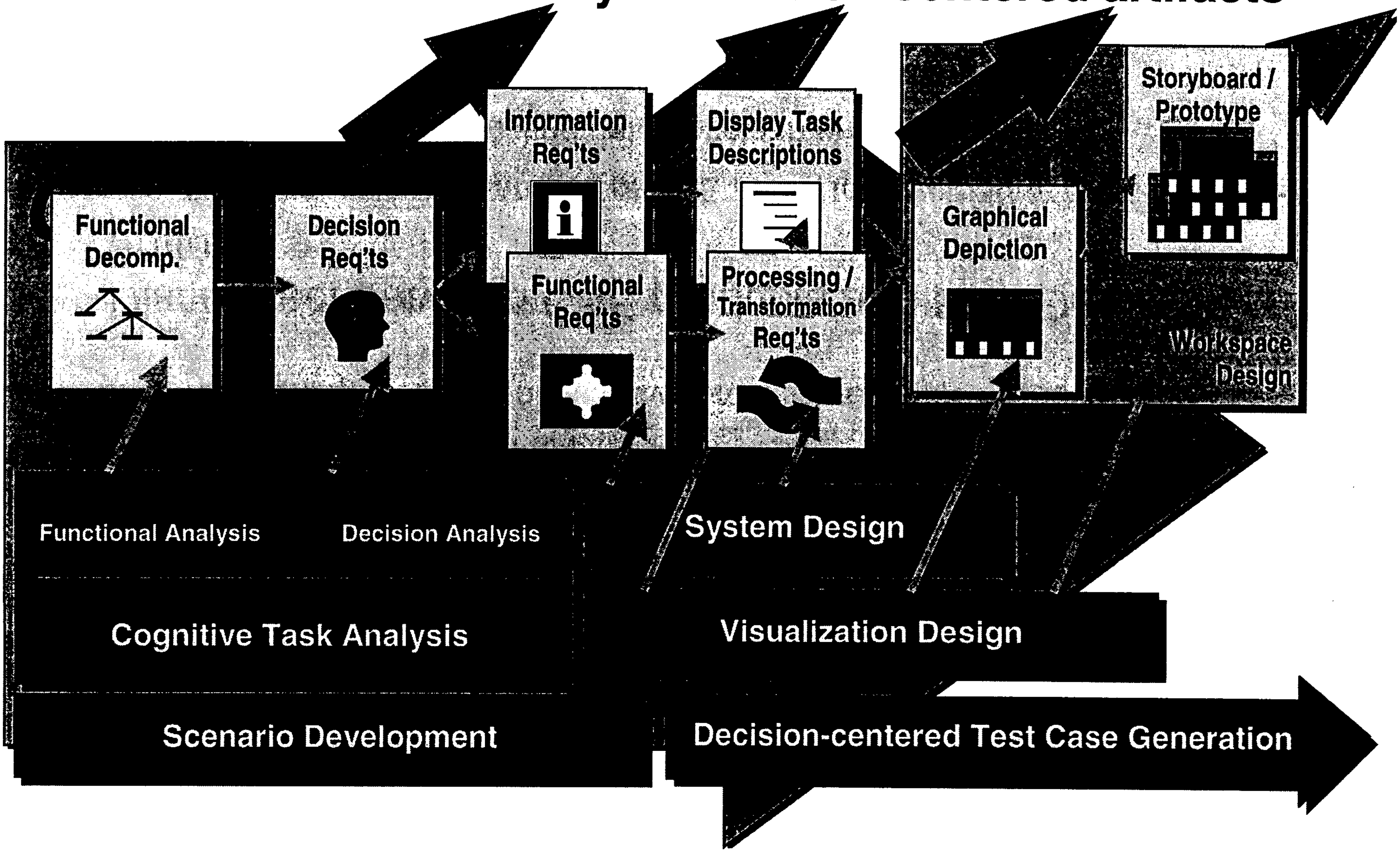
- Documenting critical insights gained in the CTA phase typically done in standard Office/COTS applications.
- Maintaining results in an iterative process becomes burdensome.

Disconnect between CSE products and S/W Dev. artifacts

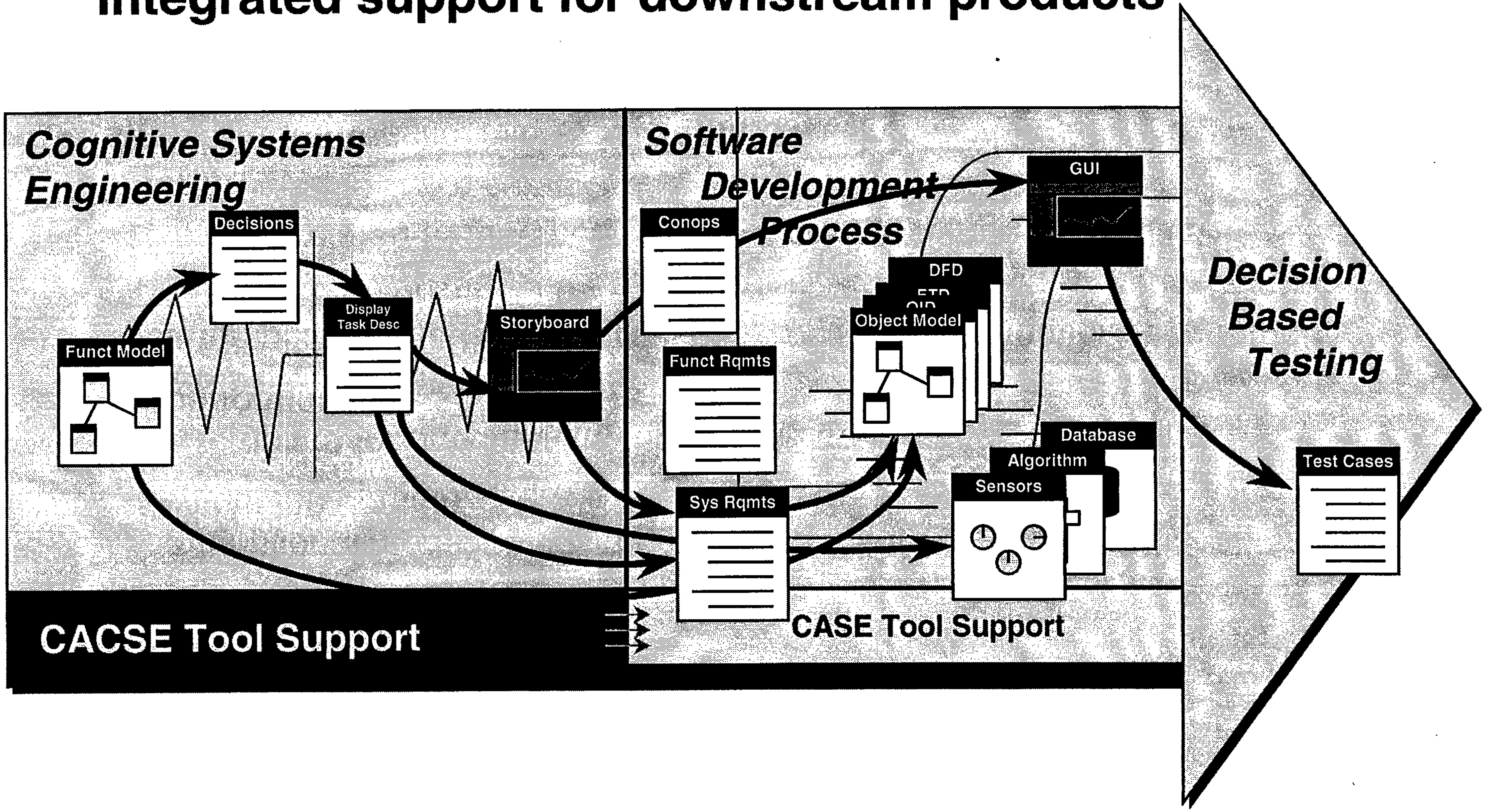
- Typical approach: hand-off of CSE report to developers.
- Much "cutting and pasting" into development artifacts.
- Hard to maintain – labor intensive on each iteration.

In order to provide effective and timely support to system developers, CSE needs to be in "mainstream" S/W development activities!

CSE Processes and Products...
Generates a new family of decision-centered artifacts



Technical Vision...
Integrated support for downstream products



CACSE — Critical Components (1 of 2)

As a tool for the CTA Researcher:

- a functional modeling environment for capturing critical relationships within the domain.
- a template for documenting underlying cognitive demands and performance requirements.
- a template for capturing critical decisions and information requirements in relation to functional relationships.
- links to related requirement documents, object models, and GUI components.
- CASE-like functionality – e.g., error checking, coverage, style compliance.

CACSE — Critical Components (2 of 2)

As a tool for the System Developer:

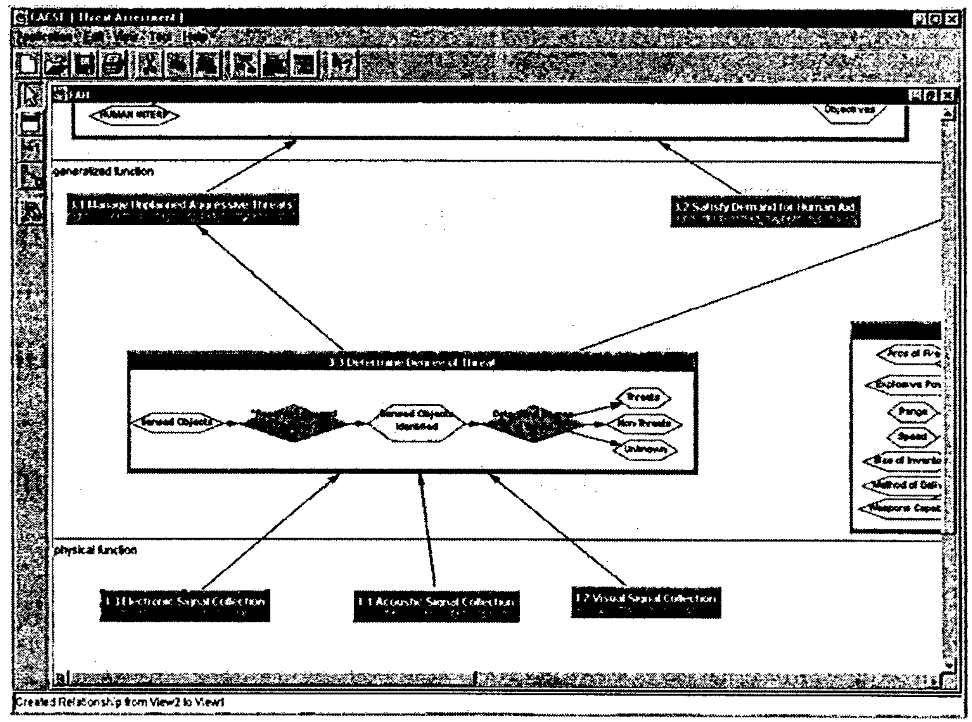
- links to related functional and system requirements supporting specific design activities.
- links to information requirements and information coding worksheet for each display.
- a navigation design tool to maintain awareness of workspace design while working on local displays.
- eventually, as an "add-on" to the developer's CASE development environment to form one integrated tool.

Tool Support for the CSE Process...

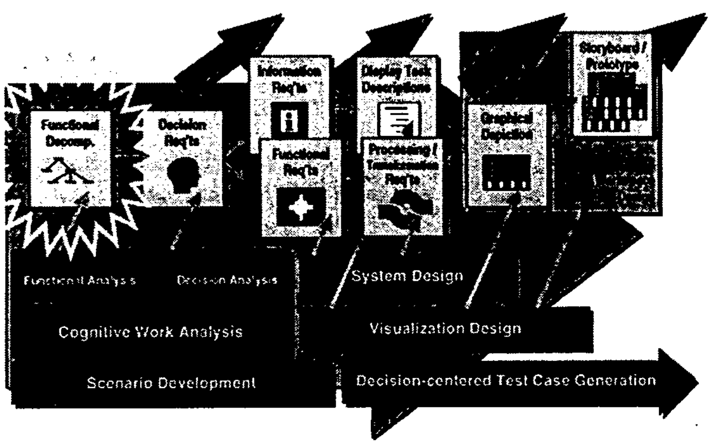
1. Building the Functional Abstraction Hierarchy

Graphical Modeling Tool for:

- Capturing goal/process relationships – supporting goals, constraints, side-effects – within an abstraction hierarchy;
- Iterating and refining the model;



Serving as the starting point to the analysis!



CACSE Key Features...
1. Building the Functional Abstraction Hierarchy

Current Functionality:

- Creating and modifying nodes in the hierarchy (via toolbar on right);
- Defining “properties” of a goal node (by double-clicking on node);
- Placing nodes at the desired level of abstraction (by dragging the node; goal numbering is automatically updated);
- Expanding or collapsing the abstraction levels (by dragging the separators);
- Changing the “view” of the goal node. Currently, three different views are supported (by right-clicking on node):
  - Icon view,
  - Detail view,
  - Process view.

CACSE Key Features...

1. Building the Functional Abstraction Hierarchy

Current Functionality (continued):

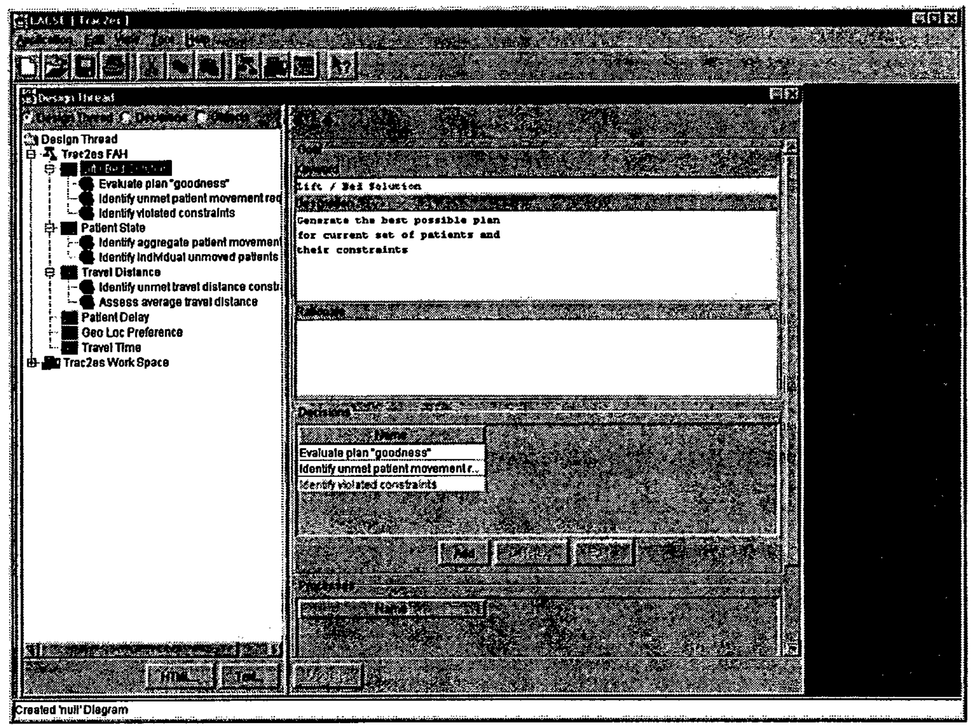
- Defining relationships between nodes in the hierarchy (via toolbar);
- Defining relationships as goal-goal (default) or goal-process (by expanding goal node to "process view" and then defining relationship to a process node);
- Defining type of relationship. Currently, three types are supported (by right-clicking on relationship):
  - Supporting,
  - Competing,
  - Decomposition;
- Defining multiplicity of relationships as AND (default) or OR (via the toolbar);

CACSE Key Features...

2. Support for the multiple levels of analysis

Tool support for:

- Documenting the supporting downstream artifacts:
  - Goals;
    - Processes;
      - Cognitive Demands (decisions);
        - Information Requirements;
          - Data Elements;
- Hierarchical tree becomes the organizational structure.



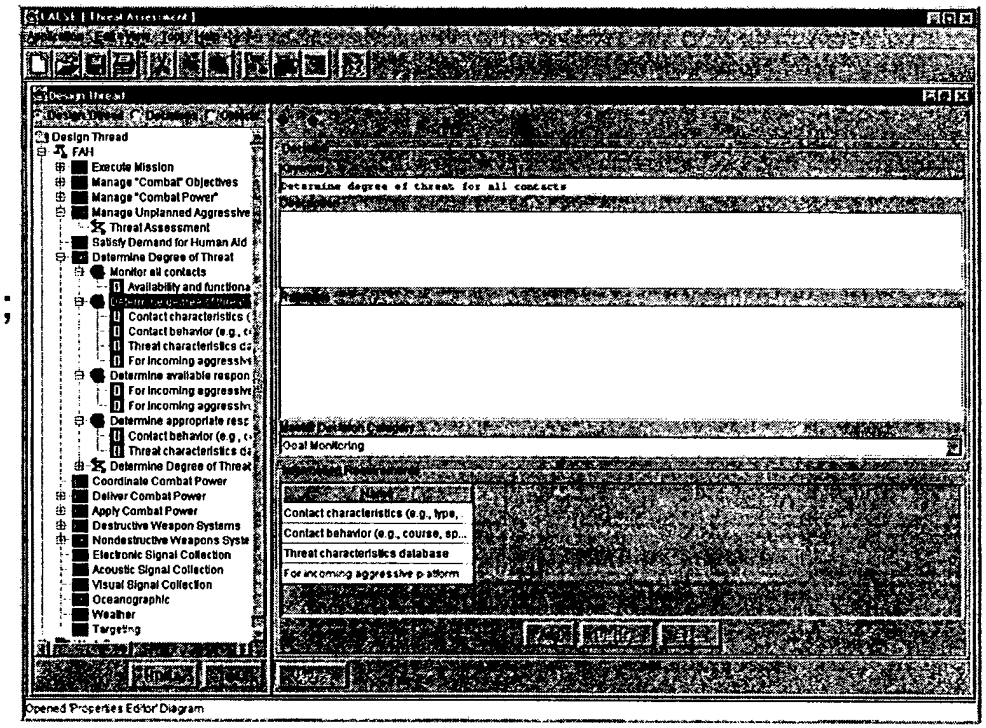
Builds on the underlying framework provided by the FAH!

Tool Support for the CSE Process...

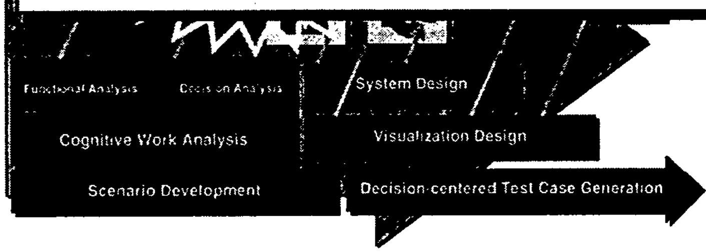
2. Support for the multiple levels of analysis

Tool support for:

- Documenting the supporting downstream artifacts:
  - Goals;
    - Processes;
      - Cognitive Demands (decisions);
        - Information Requirements;
          - Data Elements;
- Hierarchical tree becomes the organizational structure.



Builds on the underlying framework provided by the FAH!



logica Carnegie Group

CACSE Key Features…

2. Support for the multiple levels of analysis

Current Functionality:

- Documenting the supporting downstream artifacts (via the “Properties” region within the “Design Thread” window);
  - Processes (unique) can be attached to Goals;
  - Decisions (unique) can be attached to Processes or Goals;
  - Information Requirements (shareable) can be attached to Decisions;
  - Data Elements (shareable) can be attached to Information Requirements;
- Each artifact can have a description and rationale -- free form text fields;

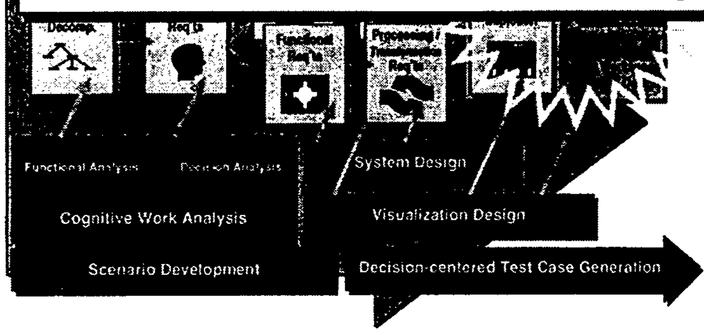
Tool Support for the CSE Process...

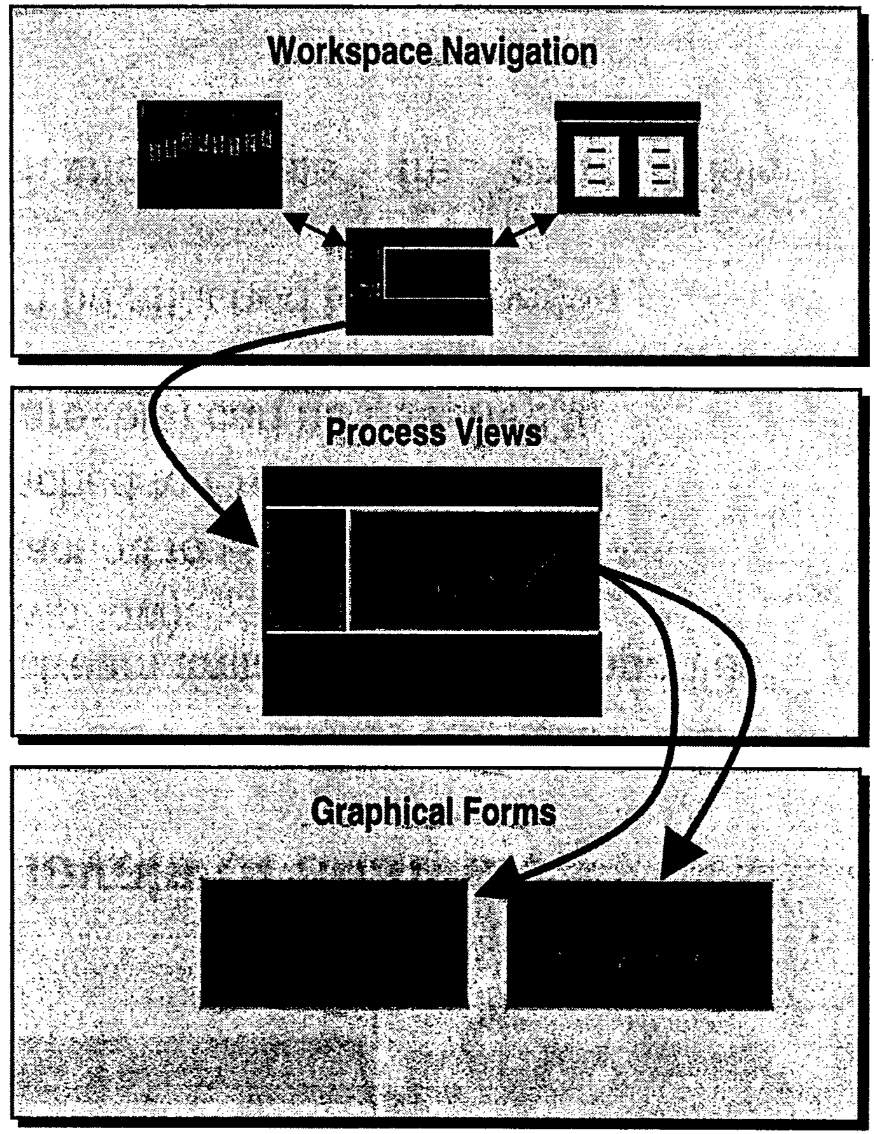
3. Defining User Interface Structure

Graphical Tool for:

- Defining workspace navigation (total replacement of one coordinated view with another);
- Defining coordinated views (information that will be seen in parallel);
- Defining graphical forms (multi-purpose/reusable components) that comprise the coordinated views.

Supports multiple levels of visualization design.



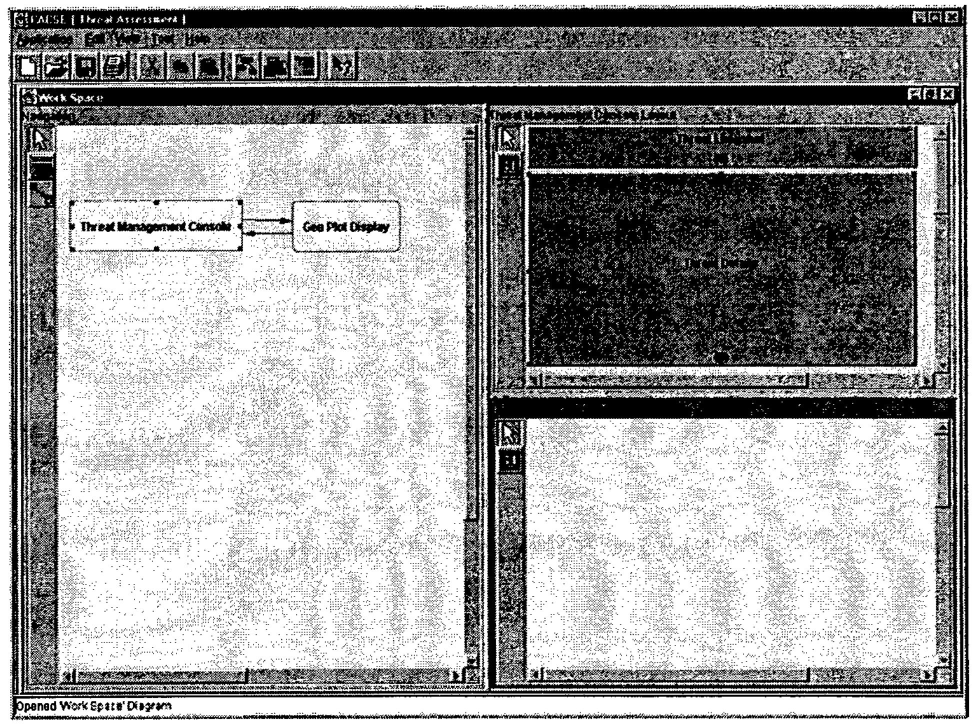


Tool Support for the CSE Process...

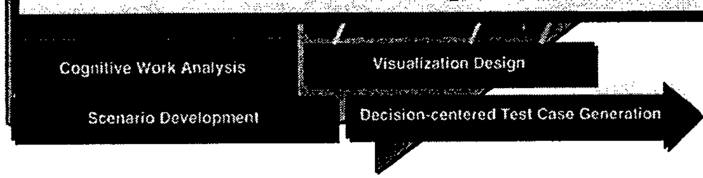
3. Defining User Interface Structure

Graphical Tool for:

- Defining workspace navigation (total replacement of one coordinated view with another);
- Defining coordinated views (information that will be seen in parallel);
- Defining graphical forms (multi-purpose/reusable components) that comprise the coordinated views.



Provides a Basis for Systematic Construction of Effective Visual Workspaces.



CACSE Key Features…
3. Defining User Interface Structure

Current Functionality:

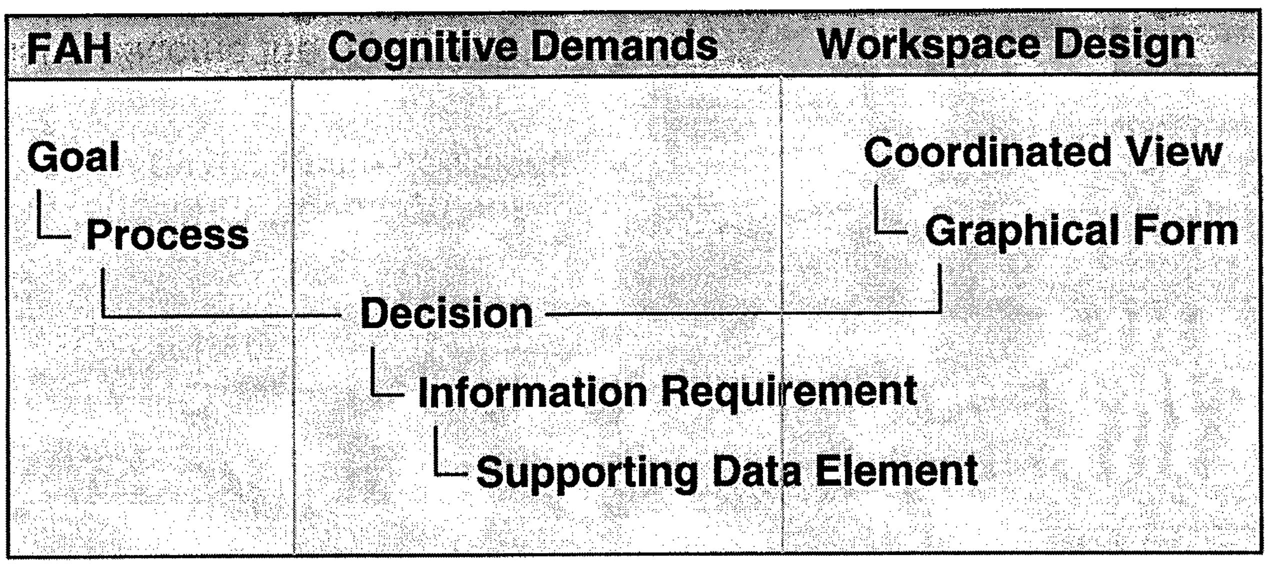
- Workspace Navigation:
  - Creating and modifying views in the workspace (via toolbar on right);
  - Defining navigation between views in the workspace (via toolbar on the right);
  - Displaying layout of a view (by right-clicking on view in workspace);
- Coordinated View layout:
  - Creating and modifying regions in the view (via toolbar on right);
  - Displaying interchangeable components of a view (by right-clicking on region in view);
- Graphical Form definition:
  - Creating and modifying graphical forms in the view (via toolbar on right);

Tool Support for the CSE Process...

4. End-to-end support for the entire "Design Thread"

A tool for:

- Defining functional "coverage" for each display by assigning decisions to displays;
- Decisions become the "linchpin" connecting goals to displays;

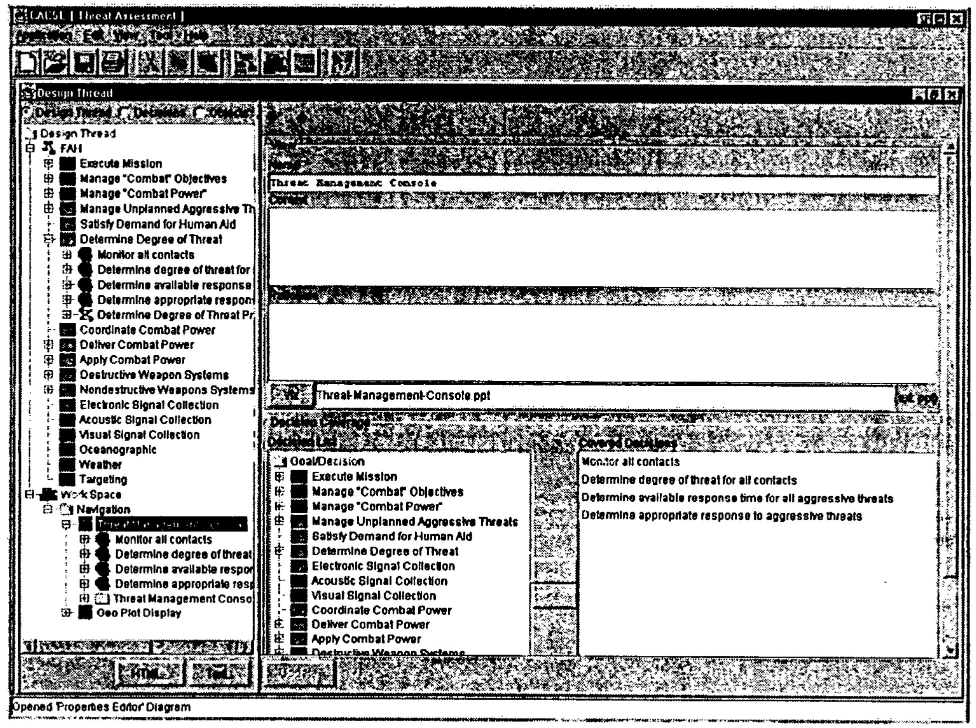


Tool Support for the CSE Process…

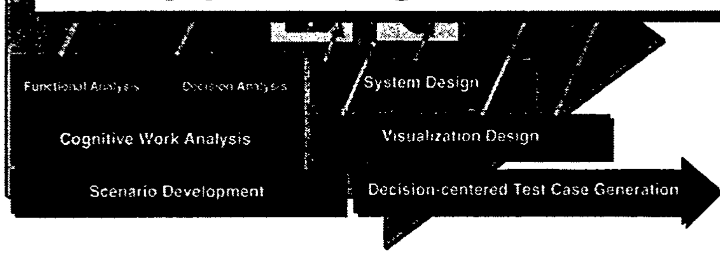
4. End-to-end support for the entire "Design Thread"

A tool for:

- Defining functional "coverage" for each display by assigning decisions to displays;
- Decisions become the "linchpin" connecting goals to displays;



Provides a Complete Thread… from a Goal to a Supporting Visualization.



CACSE Key Features…

4. End-to-end support for the entire "Design Thread"

Current Functionality:

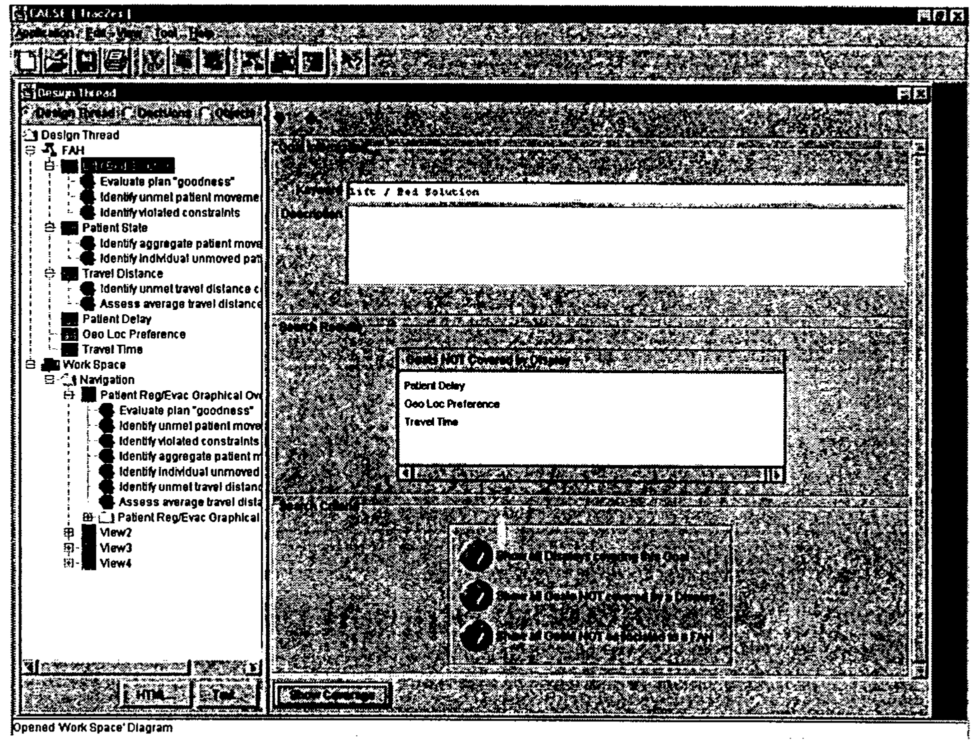
- Tracking the completeness of the design thread (via the indicators on the goal nodes);
- Query the Design Thread for gaps in the analysis (via the "bottom up" view in the "Design Thread" window;

CACSE Key Features…

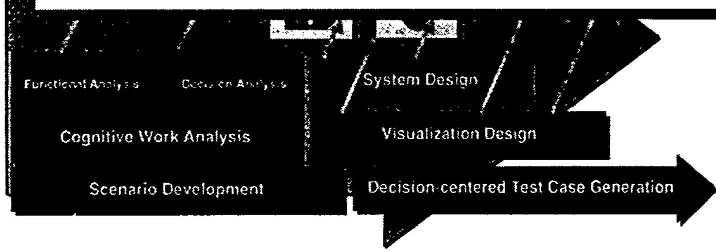
5. Completeness analysis

Graphical Tool for identifying:

- Decisions not associated with a Goal;
- Goals and Decisions not covered by Displays;
- Displays not associated to Decisions;



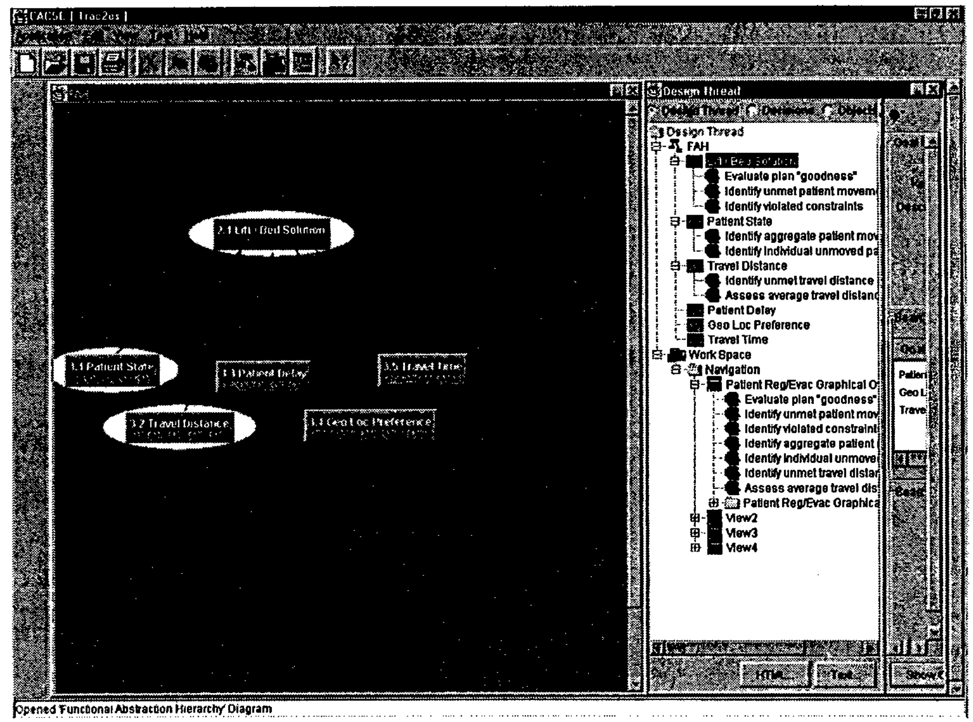
Provides checks for identifying gaps in the Design Thread.



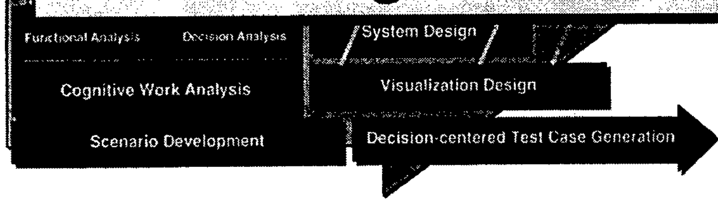
CACSE Key Features…
5. Completeness analysis

Graphical Tool for identifying:

- Decisions not associated with a Goal;
- Goals and Decisions not covered by Displays;
- Displays not associated to Decisions;
- Assessing "functional coverage" of Displays defined in the Design Thread;



Provides checks for identifying gaps in the Design Thread.



CACSE Key Features...

5. Completeness analysis

Current Functionality:

- Tracking the completeness of the design thread (via the indicators on the goal nodes);
- Query the Design Thread for gaps in the analysis (via the “bottom up” view in the “Design Thread” window;
- Tracking the completeness of the visualization design (via the toolbar on the Functional Abstraction Hierarchy window);

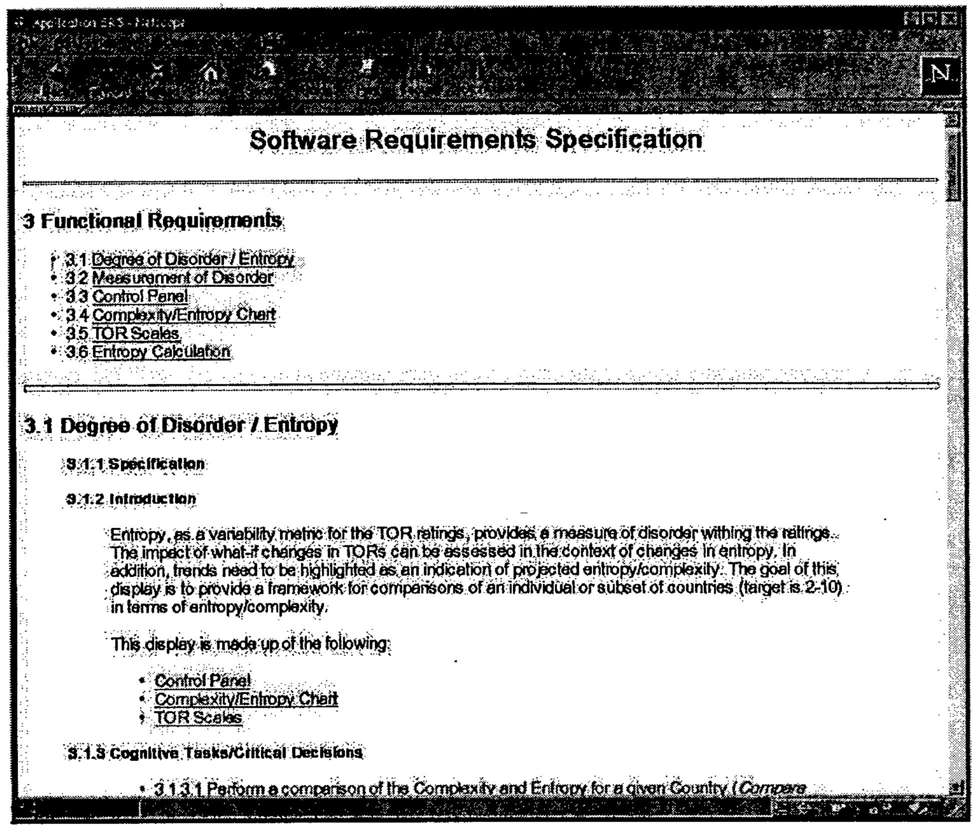
CACSE Key Features…

6. Seamless Generation of initial SRS

SRS generated with one click:

- Goal-based requirements;
- Display-based requirements;

Updates to the CSE design thread automatically included in SRS



Provides initial integration point with software development effort.

CACSE Key Features... (future version)

7. Generation of Initial Class Diagram Model

Initial object model generated within CACSE:

- classes based on display components (control panel, graph x, legend);

Updates to the CSE design thread automatically included in subsequent Object Models

Eventually, refinement of the object model triggers update in the CSE design thread

No more starting from scratch in Paradigm or Rose !!

CACSE Key Features... (future version)

8. Integrated GUI Support

Generate necessary GUI information:

- Data flow between displays and components;
- Support for code reuse between displays;

Eventually — integrated with a GUI builder tool

- Capture the entire design thread – from goal to display
- Support for GUI design issues
  - Salience control
  - Display design details

CSE-based system design...
Key Breakthroughs

Tool support for the CSE process

- A means of documenting critical insights gained in the CTA phase.
- Easy to expand and refine results in an iterative process.

Connection between CSE products and S/W Dev. artifacts

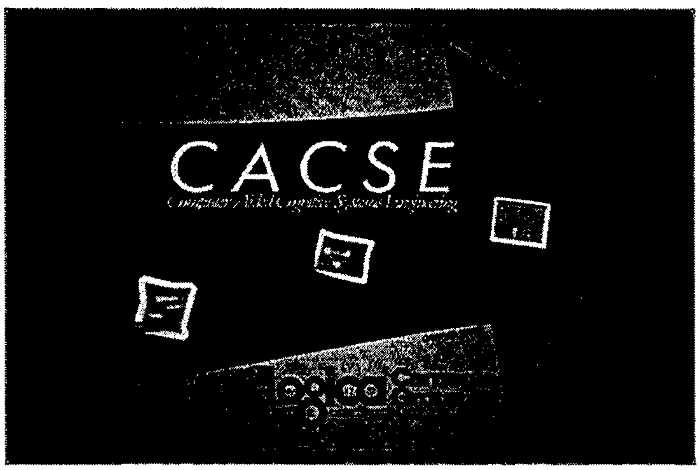
- Design thread establishes the links between functional relationships in the domain and visualization support.
- Design thread forms the "design basis" for each display.
- Decision-centered software requirements serves as critical joint artifact.

In order to provide effective and timely support to system developers, CACSE is in the "mainstream" of S/W development activities!

VIII. CACSE USERS MANUAL

CACSE

Computer-Aided Cognitive Systems Engineering



CACSE Tool User's Manual

for use with Version 3.0

TABLE OF CONTENTS

List of Illustrations

Icons & Menus

Displays, Explanations & Lists

Windows

Tables

INTRODUCING THE CACSE TOOL

The Computer-Aided Cognitive Systems Engineering (CACSE) Tool was created by Logica Carnegie Group to facilitate the development of effective decision support systems. Its purpose is to integrate cognitive work analysis (CWA) and cognitive systems engineering (CSE) methods and approaches with the system development process.

Functional and decision analyses created using multi-layered CWA methodology are generated during the modeling and analysis phase. System design requirements are generated during the creation of information requirements, which lead to display task descriptions, and functional requirements, which lead to processing-transformation requirements. System design requirements lead to the visualization design, where graphical depictions and storyboard/prototypes can be developed using the workspace design. This combination of these system and workspace design leads to decision-centered test case generation.

Successful application of the CACSE Tool allows you to generate human-centered systems requirements by capturing all of the output from a multi-layered CWA methodology. CACSE provides the ability to assess current support tools and technologies for specific users within the context of a CWA while assessing ongoing technology development to support decision requirements. The results of the CWA may be applied to the definition of visualization and processing requirements for advanced human-system interface design technologies.

At the completion of an analysis, CACSE Tool users will feel more completeness in reaching the goal of "Have I covered everything?" The analysis results then can be given directly to the system developers in hopes of building the ideal program. By generating CSE results in mainstream software development activities, analysts can provide effective and timely support to system developers.

Getting to Know CACSE's Components

The CACSE Tool is composed of three critical components: 1) CWA modeling tool; 2) system design tool; and 3) workspace design tool. The CWA modeling tool provides a functional modeling environment for capturing critical relationships within the domain. A template also is provided for documenting underlying cognitive demands and performance requirements, as well as for capturing critical decisions in relation to functional relationships. The system design tool provides a mechanism for identifying information requirements and supporting data elements linked to the critical decisions. In addition, the system design tool provides a mechanism to define the workspace design and link individual displays to the underlying decisions that the displays are designed to support.

Getting to Know CACSE's Key Features

The Functional Abstraction Hierarchy (FAH), the Design Thread and Workspace Design comprise the three key features of the CACSE Tool. The FAH is a graphical modeling tool which stores goal-process relationships, supporting goals, constraints and side-effects, and allows you to iterate and refine the model. The Design Thread supports documenting, including goals, processes, cognitive demands (decisions), information requirements and data elements for downstream artifacts. The Workspace Design is a tool for modeling the organization of the user interface, including navigation between different views and hierarchical relationships between user interface components.

Understanding the FAH

The FAH allows you to create and modify nodes in the hierarchy, define "properties" of a goal node, place nodes at the desired level of abstraction, expanding or collapsing the abstraction levels, all the while automatically updating goal numbering.

The FAH window can be viewed three different ways, as icons, showing detail and viewing process. Three types of relationships, supporting, competing, and decomposition may be defined between nodes in the hierarchy; these may be defined as goal-goal or as goal-process. Relationships may be ANDed or ORed.

Understanding the Design Thread

Multiple levels of analysis are supported by CACSE which build on the underlying framework provided by the FAH. Using the "Properties" region within the Design Thread window, a hierarchical tree becomes the organizational structure.

Unique processes can be attached to goals, and unique decisions can be attached to processes or goals. Shareable information requirements may be attached to decisions, while shareable data elements may be attached to information requirements; each artifact can have a description and rationale using free form text fields.

CACSE's graphical tool provides checks for identifying gaps in the design thread, allowing you to assess "functional coverage" of displays defined in the design thread. Finally, you may output the analysis to a text file in systems requirements specification (SRS) format.

To support multiple levels of visualization design, the CACSE Tool provides you with a graphical tool for defining workspace design. This graphical tool has the capability to model parallel (information presented together within a coordinated view) or serial (total replacement of one coordinated view with another) display of data. It allows you to define graphical forms that comprise coordinated views.

Meeting Equipment Requirements

At a minimum, the CACSE Tool runs on any personal computer equipped to run JAVA and Microsoft Windows, with an ability to connect to a database. In order to run the CACSE Tool the following hardware and software are suggested:

- Win32 Release for Windows 95, Windows 98 and Windows NT 4.0 on Intel hardware: 486/DX or faster processor minimum, Pentium processor recommended
- 32 MB of RAM
- 1024x768 color monitor (minimum 16 bit color)
- Double-speed CD ROM drive (or higher)

Understanding Terms Used for Instructions

The following terms are used throughout this manual to give instructions.

Click/Click on means to position the mouse over an item, press down the left mouse button, then. Instructions will indicate those times when you should press down the right mouse button, then release.

Double-click means to click the left mouse button, twice, quickly.

Drag means to position the mouse on or over an item, press down the left mouse button, keeping the button pressed, move the cursor to another location, then release.

Highlight means to change the shade of the background in a data entry field (by clicking on it or dragging).

Installing the CACSE Tool

Installation instructions are provided on the inside cover of the CACSE CD case and are as follows.

If Microsoft® Windows is not already running:

- Start **Windows**.

- Insert the **CACSE CD** disc into the CD ROM drive.

  The installer should run automatically.

  If the installer does not run automatically, assuming the D drive is the CD ROM drive:

  - Run **D:/setup.exe**.

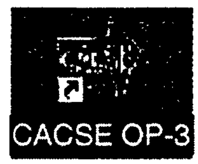
RUNNING THE CACSE TOOL

The CACSE Tool is a stand alone system. To run the CACSE Tool locate the CACSE OP-3 icon on the desktop.

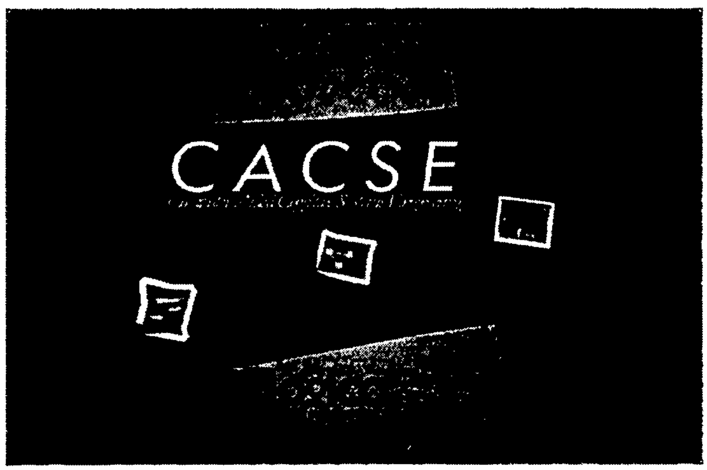
Starting CACSE

To start the CACSE Tool:

- Double-click on the **CACSE OP-3** icon.



A JAVA window appears and then the CACSE open



CACSE Opening Display

You may now choose to create a new application or open a previously saved application.

Creating CACSE Applications

Applications, or analyses, may be created, saved and queried. Saved applications may be modified, and application data can be output to an HTML file in SRS format.

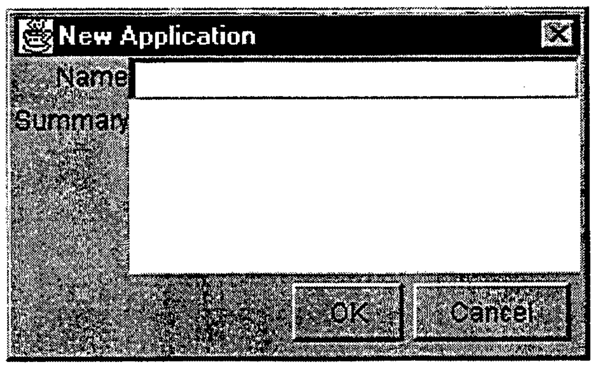
To create a new CACSE application:

- Click on the radio button preceding **"Create a new application"** in the CACSE window.

- Click on the **OK** button.

If you click on the Cancel button, the CACSE window closes and the desktop is empty.

The New Application window appears.



New Application Window

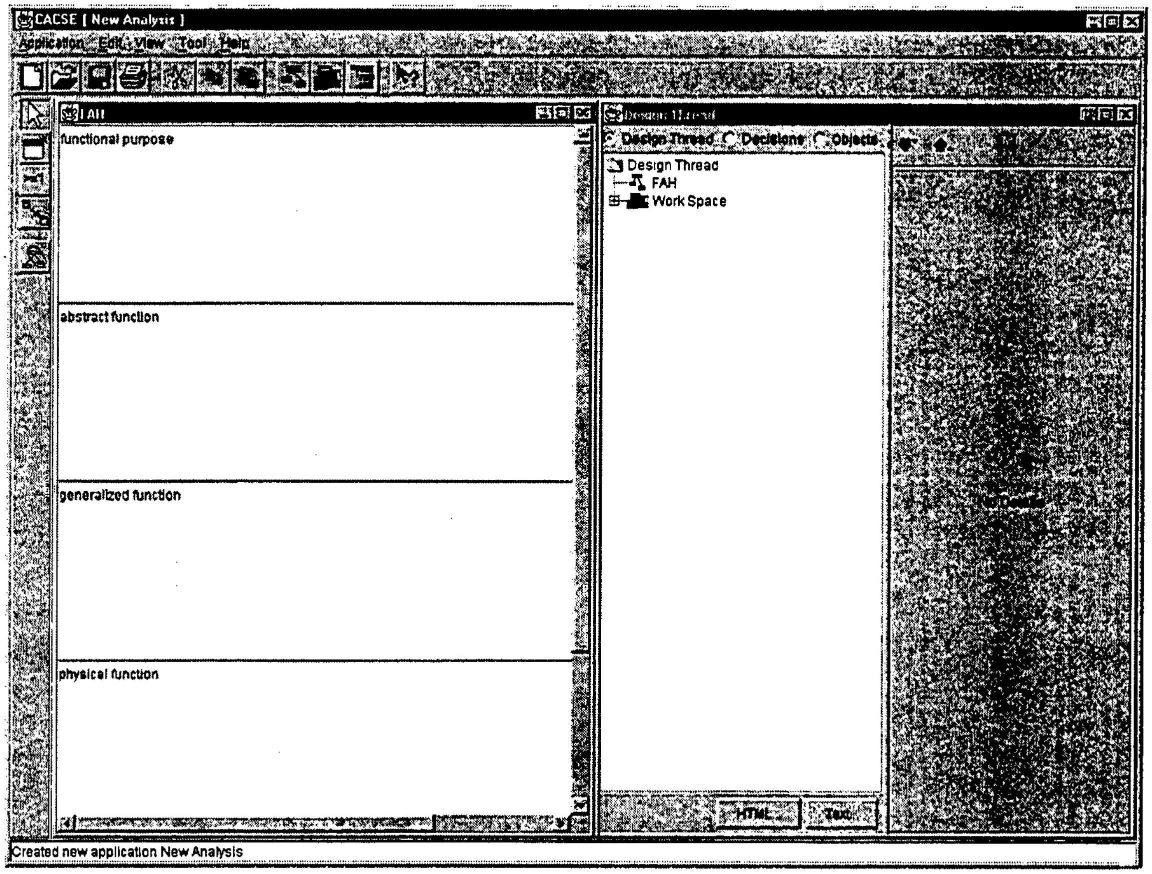
- Type a **Name** for the application you wish to create in the field provided.

- Type a **Summary** description, if you wish.

- Click on the **OK** button.

If you click on the Cancel button, the CACSE window closes and the desktop is empty.

The FAH and Design Thread windows appear. (See the illustration on the following page.)



FAH and Design Thread Windows

Opening a Saved Application

To open an existing CACSE application:

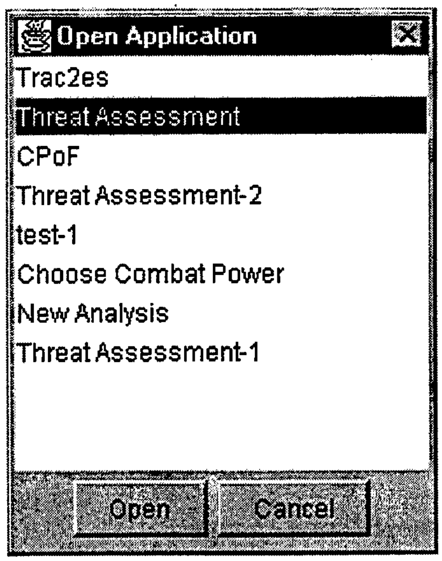
- Click on the radio button preceding "**Open an existing application**" in the CACSE window.

- Click on the **OK** button.

  If you click on the Cancel button, the CACSE window closes and the desktop is empty.

  The Open Application window appears. (See the example on the following page.)

- Click on the name of the **application** you wish to open to highlight it.



Open Application Window

- Click on the **Open** button.

If you click on the Cancel button, the Open Application window closes and the desktop is empty.

If you receive an error message stating that you cannot open the application because it is in use, you may need to remove the folder with the name of your data set and the extension ".ODX".

To do so:

- ***Locate the c:\CACSE\bin directory using your file manager.***
- ***Delete the respective ".ODX" file.***

The FAH and Design Thread windows appear. (Refer to the illustration on the previous page.)

You are now ready to begin building the FAH.

Using the Application menu, you may move from an application you are working on to one you have saved previously.

To do so:

- Click on the **Application** menu.
- Click on **Close**.

  If you have made changes to the application and have not updated it, the Application window appears. The Application window prompts you to save the application.

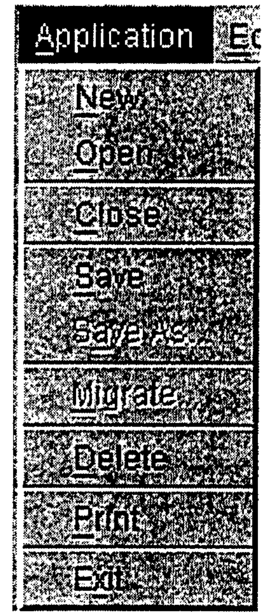
  To save application changes:

  - Click on the **Yes** button.

  Otherwise, to close the application without saving the changes:

  - Click on either the **No** or **Cancel** button in the Application window.

    The application closes. You may now open another application.



Application Menu

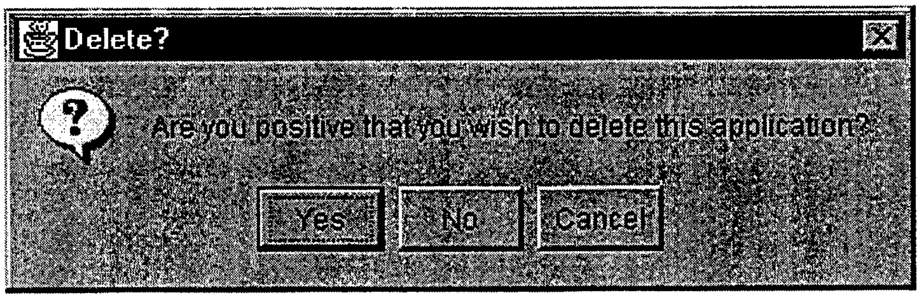
To open another application:

- Click on the **Application** menu.
- Click on **Open**.
- Follow the instructions on page 7.

To delete a CACSE application:

- Click on the **Application** menu.
- Click on **Delete**.

  The Delete window appears, prompting you with the question, "Are you positive that you wish to delete this application?"



Delete Window

To delete the application:

- Click on the **Yes** button.

Otherwise, to cancel and return to application:

- Click on either the **No** or **Cancel** button in the Delete window.

Quitting the CACSE Tool

To quit the CACSE Tool:

- Click on the **Application** menu.
- Click on **Exit**.

  If you have made changes to the application, you will be prompted by a dialog asking whether or not you wish to save the changes.

To save changes:

- Click on the **Yes** button.

Otherwise:

- Click on the **No** or **Cancel** button.

The program closes and the Finished CACSE window appears.

- Click on the **close box** on the Finished CACSE window.

BUILDING THE FAH

Building the Functional Abstraction Hierarchy (FAH) serves as the starting point to the analysis. The FAH is a graphical modeling tool designed to allow the analyst to capture goal/process relationships, supporting goals, constraints and side-effects within an abstraction hierarchy, and iterate and refine the model.

Using the FAH, you may create and modify nodes in the hierarchy using the tools in the toolbar to the left of the FAH window. You define "properties" of a goal node, e.g., decisions, processes etc. in the right side of the Design Thread window.

Understanding the Levels of Abstraction

The five sections that divide the FAH window horizontally are levels of abstraction; these abstraction levels provide the framework for the means-ends relationships. The levels of abstraction are explained in the table below. They help explain a path taken by a user through a workspace by depicting the complexity of the workspace taken together with the goals and resources of the user.

Functional Purpose	The Functional Purpose level should contain objectives; concepts of purpose and value necessary to establish relations between system performance and reasons for design.
Abstract Function	The Abstract Function level should contain those concepts necessary for setting priorities and allocating resources to various general functions and activities necessary to establish priorities.
Generalized Function	The Generalized Function level should contain general work activities and functions; activities or functions at this level are independent of the underlying processes involved, including their physical implementation.
Physical Function	The Physical Function level should contain specific work processes and physical Processes needed to create and maintain generalized functions.
Physical Form	The Physical Form level should contain appearance, location and configuration of objects for navigation of the system.

Abstraction Levels Explanation Table

Changing the Size of the Abstraction Level

You may expand or collapse the abstraction levels to vary the size of the section (vertically).

To change the size of a section of the FAH window:

- Click on the **Selection** tool.
- Position the Selection tool (mouse) over the **separator** (line at the bottom of the section) you wish to expand or collapse until the cursor takes the form of a crosshair.
- **Drag** the separator to the desired location.

Placing Nodes in the Abstraction Level

You may place goal nodes at any level of abstraction.

To do so:

- Click on the **goal node** to highlight it.

  Highlighting is indicated by solid red squares surrounding the perimeter of the goal.

- **Drag** the node to the desired location.

  Goal numbering is updated automatically.

Using the FAH Tools

Along the upper left side of the FAH window is a toolbar containing five tools used to build the FAH. These tools are explained in the table below.

Selection Tool	Click on the Selection tool to highlight goals in the FAH and Design Thread windows.
New Goal Tool	Click on the New Goal tool to create goals in the FAH window.
New Inclusive OR Tool	Click on the New Inclusive OR tool to establish that one of any number supporting goals must be satisfied to achieve a supported goal.
New Relationship Tool	Click on the New Relationship Tool to link one goal to another in order to show support: supporting, one requires the other; competing, one or the other; decomposition, expanding upon what's there in order to reduce the process (multiple steps or greater level of detail).
Graphic Overlay Tool	Click on the Graphic overlay tool to depict decision/goals used for a specific visual display (defined in the Workspace Design window).

FAH Toolbar Explanation Table

Generating a Goal

To create a new goal:

- Click on the **New Goal** tool to the left of the FAH window.
- Click inside the **FAH window** where you wish to place the goal.

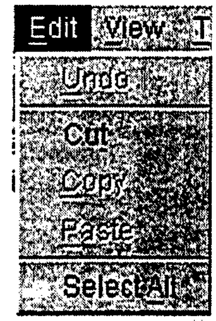
A goal node appears in the FAH window, and is automatically given a number in the hierarchy of the display.

You may either continue creating new goals or define the properties of the goal you have just placed. (Refer to the instructions on page 20.)

You <u>must</u> click on the Selection tool or another tool in the toolbar to discontinue placing goals, (i.e., turn off the New Goal tool).

Deleting a Goal

To delete a goal from the FAH window:



Edit Menu

- Using the **Selection** tool, click on the **goal** node you wish to delete.

  The goal will become highlighted (solid red squares).

- Click on the **Edit** menu.

- Click on **Cut**.
  The goal node is removed from the FAH window, goal numbering is automatically updated, and any property information for that goal is removed from the Design Thread window.

You also may use the Cut option on the Goal pop-up menu. ***See page 55.***

Selecting & Moving Nodes

To select all the nodes in the FAH window, perhaps to move them as a group:

- Click on the **Edit** menu.

- Click on **Select All.**

  All of the nodes and their relationships in the FAH window become highlighted (solid red squares).

To move the nodes:

- Using the **Selection** tool, click on one of the highlighted nodes and **drag** them.

  All of the nodes are moved.

  To deselect the nodes:

  - Click **anywhere** in the FAH window, <u>except</u> on a node.

Defining Relationships

You define relationships to make one goal support another by connecting the goals using the New Relationship tool. Relationship may be defined between goals (goal-goal (default)), or between goals and processes (goal-process), by expanding goal node to "process view" and then defining the relationship to a process. CACSE supports relationship multiplicity; the default relationship is "and", or you choose to "or" the relationship using the New Inclusive OR tool. You also may define relationships between goal nodes and inclusive OR nodes.

To define relationships between nodes in the hierarchy:

- Click on the **New Relationship** tool to the left of the FAH window.
- Click on the **supporting goal** node.
- Click on the **supported goal** node to complete.

You must <u>not</u> move the cursor AT ALL when you click.

A relationship <u>cannot</u> be drawn to itself.

You <u>must</u> click on the Selection tool or another tool in the toolbar to discontinue placing relationships, (i.e., turn off the New Relationship tool).

To define relationships between goal nodes and processes node:

- Click on the **New Relationship** node in the toolbar to the left of the FAH window.
- Click on the **supporting goal** or **process** node.
- Click on the **supported goal** or **process** node to complete.

(See notes above.)

Defining a New Inclusive OR

CACSE supports relationship multiplicity. When defining a relationship using the New Relationship tool, the default relationship is "and". You may choose to "or" the relationship using the New Inclusive OR tool with the New Relationship tool. The inclusive OR establishes that at least <u>one</u> of any number of supporting goals must be satisfied to achieve a supported goal. Use the New Relationship tool to link supporting goals to the inclusive OR and then link the inclusive OR to the supported goal.

To create a new inclusive OR:

- Click on the **New Inclusive OR** tool to the left of the FAH window.

- Click inside the **FAH window** where you wish to place the OR.

  A node appears in the FAH window. You may either continue creating ORs, or connect them to goals using the New Relationship tool.

  ✍ ***You must click on the Selection tool or another tool in the toolbar to discontinue placing "or"s, (i.e., turn off the New Inclusive OR tool).***

To define relationships between goal nodes and the inclusive or:

- Click on the **New Relationship** node in the toolbar to the left of the FAH window.

- Click on the **supporting goal** or **process** node.

- Click on the **inclusive OR** node.

  This defines the first of the supporting goals in the OR relationship.

- **Repeat** this process for additional goal nodes that are to be included in the OR relationship.

- Click on the **inclusive OR** node.

- Click on the **supported goal** or **process** node to complete.

  (See notes on the previous page.)

Deleting an Inclusive OR

To delete a relationship from the FAH window:

- Using the Selection tool, click on the **relationship arrow** you wish to delete.

  The arrow becomes highlighted (solid red squares).

- Click on the **Edit** menu. (See page 15.)

- Click on **Cut**.

  The relationship arrow is removed from the FAH window.

  You also may use the Cut option on the Relationship pop-up menu. (See below.)

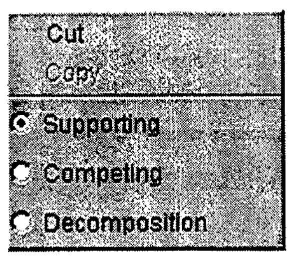
Changing the Relationship Type

The CACSE tool allows you to set three types of relationships: supporting, competing and decomposition. A supporting relationship means that one node provides a supporting function to the higher-order goal node; it appears solid black. A competing relationship means that one node works against the function of the higher-order goal node; it appears dashed-red. Finally, a decomposition relationship expands upon what's there in order to reduce the process (multiple steps or greater level of detail); it appears solid gray.

To change a relationship type:

- Using the **Selection** tool, position the mouse on the **relationship** (line with arrow) you wish to choose and click the **right mouse button.**

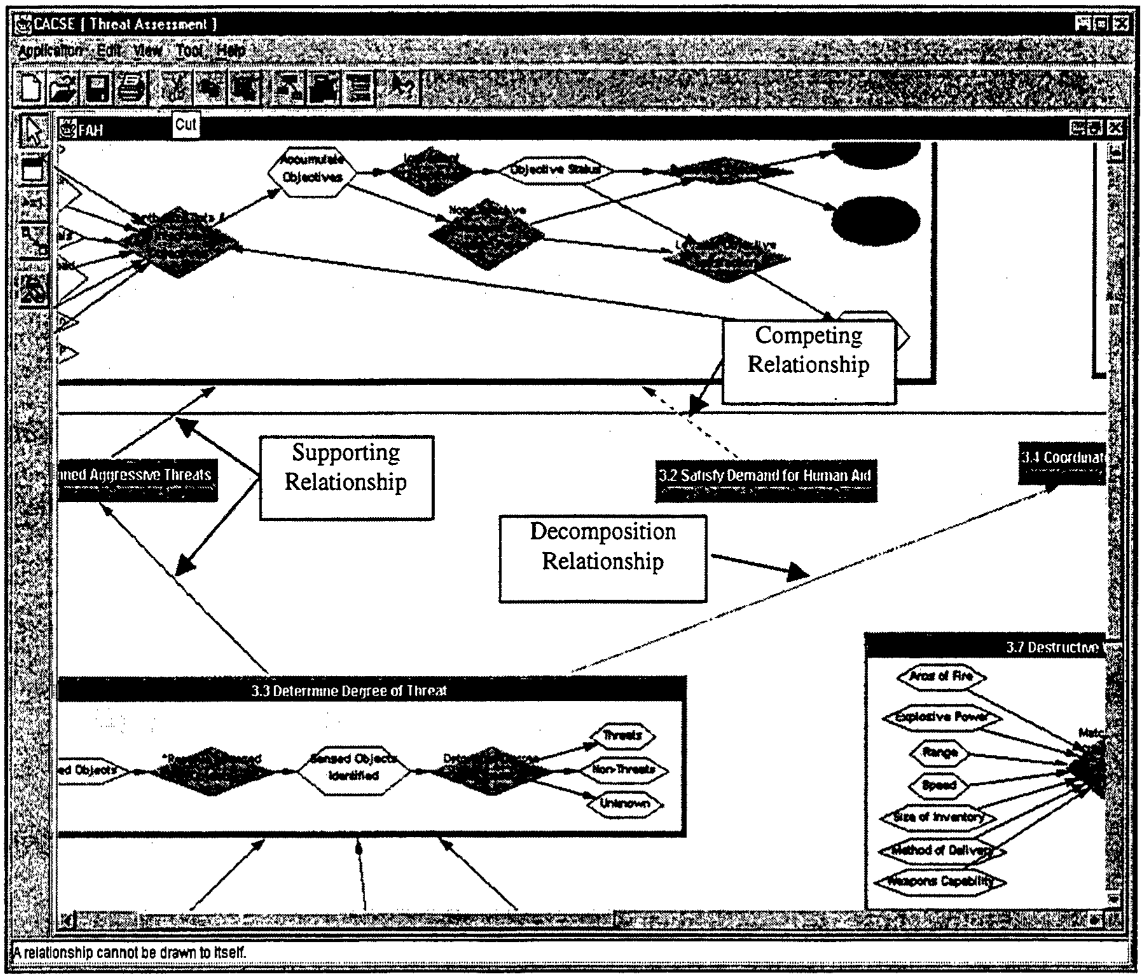
  The Relationship pop-up menu appears.



Relationship Pop-up Menu

- Click in the **radio button** preceding the type of relationship you wish to select.

  The selected relationship indicator appears. (See the example on the following page.)



FAH Window Showing Relationship Types

DEVELOPING THE DESIGN THREAD

Using the Design Thread window, unique processes can be attached to goals, and unique decisions can be attached to processes or goals created in the FAH window. Shareable information requirements may be attached to decisions, while shareable data elements may be attached to information requirements.

Each object may have a description and rationale using free form text fields.

Defining Properties of a Goal Node

To define the properties of a goal node:

- Using the **Selection** tool, double-click on the **goal node.**

  The properties of that goal node appear in the right side of the display in the Design Thread window. (See the example on the following page.)

  You also may select the Properties option from the Goal pop-up menu (See page 55).

- Click in the text field provided and type a **Keyword** for the goal.

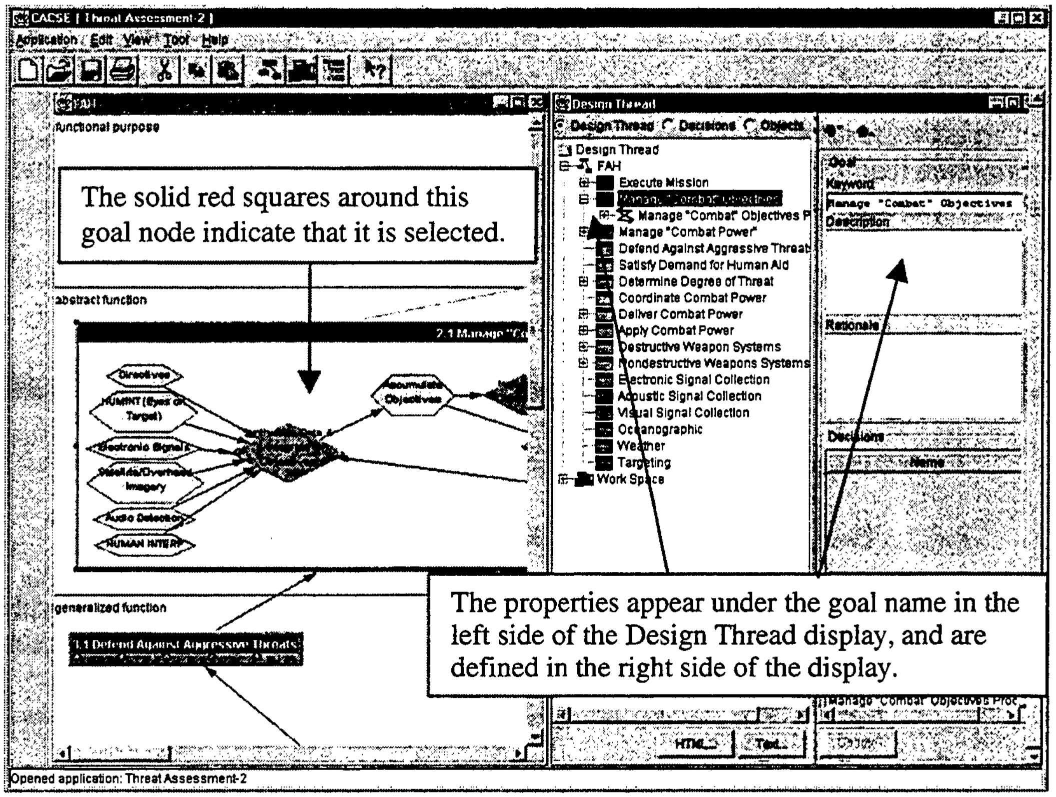
- Click in the text field provided and type a **Description.**

- Click in the text field provided and type a **Rationale.**

  You also may enter decisions and processes in the sections and fields provided.

  If you do not wish to enter decisions or processes:

  - Click on the **Update** button at the bottom of the Design Thread window.



Properties of a Goal Node Shown in the Design Thread Window

Entering Decisions

To enter a decision:

- Click on the **Add** button in the Decision section of the Design Thread window.

  The text field area below the field Name appears white.

  ☞ ***Ensure that the Design Thread radio button is selected in the Design Thread window; otherwise, you will view decision or object lists.***

- Click in the text field provided and type the **Decision**.

- Press the **Enter** key on the keyboard.

  Once you have added a decision, you may categorize the decision in the master decision list, as well as enter information requirements for the decision.

  If you do not wish to enter additional detail:

  - Click on the **Update** button at the bottom of the Design Thread window.

Entering Decision Details

Once you have entered a decision, you may select it and enter details concerning that decision.

To enter decision details:

- Click on the name of the **Decision** you wish to edit.

  If more than one decision is in the list and you do not select one, details are provided for the last item in the list. The pull down arrow at the end of the field provides you with the full list.

  That selected item in the list appears highlighted in the Decision text field.

- Click on the **Edit** button under the Decision section.

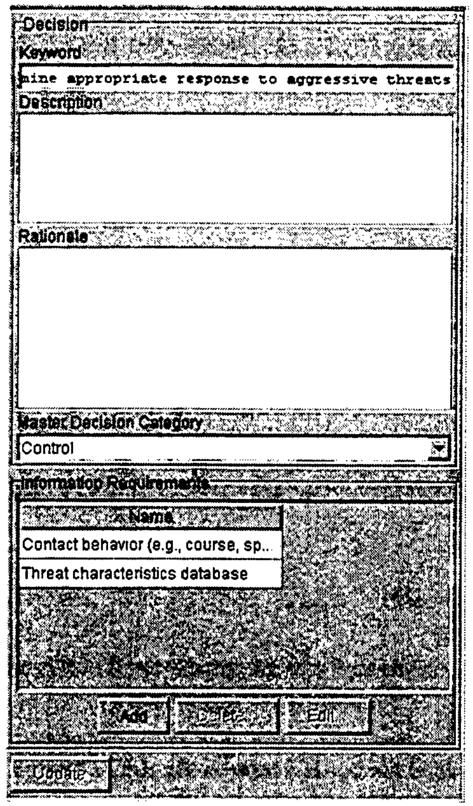
  The Decision section is brought to the top of the display, and shows two additional fields, Description and Rationale, and two additional sections, Master Decision Category and Information Requirements. (See the illustration on the following page.)

- Click in the text field provided and type a **Description**.

- Click in the text field provided and type a **Rationale**.

  If you do not wish to enter additional detail:

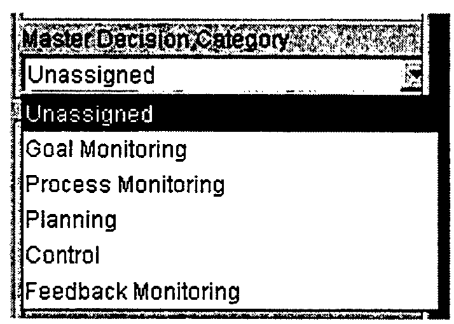
  - Click on the **Update** button at the bottom of the Design Thread window.



Decision Section of Design Thread Window

Entering a Master Decision

When entering decision details, you may subscribe to the section for entering the decision selected into the master decision category.



Master Decision Category Pull-down List

To do so:

- Click in the text field provided, or on the pull down arrow at the end of the **Master Decision Category** field.

  A pull-down list of categories appears.

- Click on the name of the **category** in the list.

  That name appears highlighted in the Master Decision Category text field.

If you do not wish to enter additional detail:

- Click on the **Update** button at the bottom of the Design Thread window.

Entering Information Requirements

Information requirements may be entered for a selected decision.

To enter information requirements:

- Click on the **Add** button in the Information Requirements section of the Design Thread window.

  The text field area below the field Name appears white.

- Click in the text field provided and type the **Information Requirement**.

- Press the **Enter** key on the keyboard.

  Once you have added an information requirement, you may enter additional detail, including data elements.

  If you do not wish to enter additional detail:

  - Click on the **Update** button at the bottom of the Design Thread window.

Entering Information Requirement Detail

Once you have entered an information requirement, you may select it and enter additional details.

To enter information requirement details:

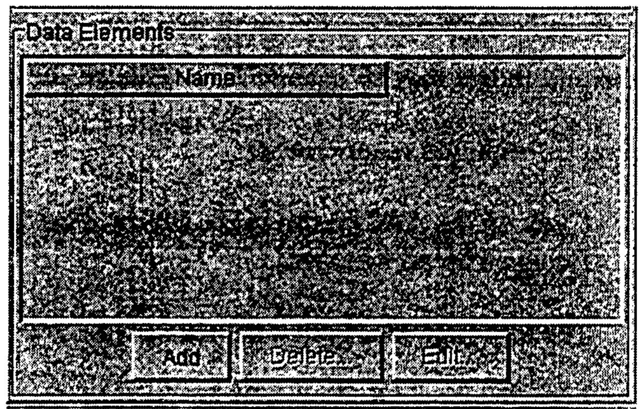
- Click on the name of the **Information Requirement** you wish to edit.

  If more than one information requirement is in the list and you do not select one, details are provided for the last item in the list. The pull down arrow at the end of the field provides you with the full list.

  The selected item in the list appears highlighted in the Information Requirement text field.

- Click on the **Edit** button under the Information Requirement section.

  The Information Requirements section is brought to the top of the display, and shows two additional fields, Description and Rationale, and one additional section, Data Elements.



Data Elements Section of Design Thread Window

- Click in the text field provided and type a **Description**.

- Click in the text field provided and type a **Rationale**.

  If you do not wish to enter additional detail:

  - Click on the **Update** button at the bottom of the Design Thread window.

Entering Data Elements

Data elements may be entered for information requirements.

To enter Data Elements:

- Click on the **Add** button in the Data Elements section of the Design Thread window.

  The text field area below the field Name appears white.

- Click in the text field provided and type the **Data Element**.

- Press the **Enter** key on the keyboard.

  Once you have added a data element, you may enter details for that data element.

  If you do not wish to enter additional detail:

  - Click on the **Update** button at the bottom of the Design Thread window.

Entering Data Element Details

Once you have entered a data element, you may select it and enter additional details.

To enter data element details:

- Click on the name of the **Data Element** you wish to edit.

  If more than one data element is in the list and you do not select one, details are provided for the last item in the list. The pull down arrow at the end of the field provides you with the full list.

  That selected item in the list appears highlighted in the Data Element text field.

- Click on the **Edit** button under the Data Element section.

  The Data Element section is brought to the top of the display, and shows four additional fields, Description, Rationale, Type and Constraints.

- Click in the text field provided and type a **Description**.
- Click in the text field provided and type a **Rationale**.
- Click in the text field provided and enter the **Type** of data element.
- Click in the text field provided and type **Constraints** for the data element.
- Click on the **Update** button at the bottom of the Design Thread window.

Entering Processes

Use the scroll bar to the right of the Design Thread window to view the process entry field.

To enter a process:

- Click on the **Add** button in the Processes section of the Design Thread window.

  The area below the field Name appears white.

- Click in the text field provided and type the **Process**.

- Press the **Enter** key on the keyboard.

  Once you have created a process description, you may enter process detail and create a process diagram.

  If you do not wish to create a process diagram:

  - Click on the **Update** button at the bottom of the Design Thread window.

Entering Process Detail

To enter process detail:

- Click on the **Edit** button under the Process section.

  The Graphical Form section is brought to the top of the display, and shows three additional fields, Description, Rationale and Process Diagram.

- Click in the text field provided and type a **Description**.

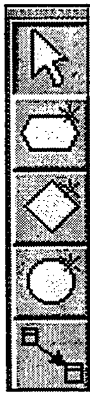
- Click in the text field provided and type a **Rationale**.

  If you do not wish to create a process diagram:

  - Click on the **Update** button at the bottom of the Design Thread window.

Creating a Process Diagrams

Once you have created a process description, you may create a process diagram. The Process Diagram section of the Design Thread window contains five tools that allow you to create the properties of the process. See the table below.



Selection Tool	Click on the Selection tool to highlight process nodes in the Process Diagram section.
New Source Tool	Click on the New Source tool to enter a process source.
New Transport Tool	Click on the New Transport tool to enter a process transport.
New Target Tool	Click on the New Target tool to enter a process target.
New Relationship Tool	Click on the New Relationship tool to link one process to another in order to show support.

Process Diagram Toolbar Definition Table

Entering a Process Source

To enter a process source:

- Click on the **New Source** tool.
- Click in the **Process Diagram** display.

  The process source node is displayed. You may now change the name and enter detail for the process source.

  If you do not wish to enter additional detail:

  - Click on the **Update** button at the bottom of the Design Thread window.

Entering Process Source Detail

To enter process source detail:

- Double-click on the **process source node**.

  The Source section is brought to the top of the display, and shows two additional fields, Description and Rationale.
- Click in the text field provided and type a **Description**.

- Click in the text field provided and type a **Rationale**.

  You may wish to create additional sources, targets, transports or relationships. (See the previous and following sections.)

  When you are finished with the process diagram:

  - Click on the **Update** button at the bottom of the Design Thread window.

  To return to the top of the Process section:

  - Double-click on a **goal node** in the Design Thread list.

Entering a Process Transport

To enter a process transport:

- Click on the **New Transport** tool.
- Click in the **Process Diagram** display.

  The process transport node is displayed. You may now change the name and enter detail for the process transport.

  If you do not wish to enter additional detail:

  - Click on the **Update** button at the bottom of the Design Thread window.

Entering Process Transport Detail

To enter process transport detail:

- Double-click on the **process transport node**.

  The Transport section is brought to the top of the display, and shows two additional fields, Description and Rationale.

- Click in the text field provided and type a **Description**.

- Click in the text field provided and type a **Rationale**.

  You may wish to create additional sources, targets, transports or relationships. (See the previous and following sections.)

  When you are finished with the process diagram:

  - Click on the **Update** button at the bottom of the display.

  To return to the top of the Process section:

  - Double-click on a **goal node** in the Design Thread list.

Entering a Process Target

To enter a process target:

- Click on the **New Target** tool.
- Click in the **Process Diagram** display.

  The process target node is displayed. You may now change the name and enter detail for the process target.

  If you do not wish to enter additional detail:

  - Click on the **Update** button at the bottom of the Design Thread window.

Entering Process Target Detail

To enter process target detail:

- Double-click on the **process target node**.

  The Target section is brought to the top of the display, and shows two additional fields, Description and Rationale.

- Click in the text field provided and type a **Description**.

- Click in the text field provided and type a **Rationale**.

  You may wish to create additional sources, targets, transports or relationships. (See the previous and following sections.)

  When you are finished with the process diagram:

  - Click on the **Update** button at the bottom of the Design Thread window.

  To return to the top of the Process section:

  - Double-click on a **goal node** in the Design Thread list.

Defining Process Relationships

You define relationships to make one process node link to another by connecting them using the New Relationship tool. Click on the New Relationship tool to link one process to another in order to show process flow.

To define relationships between process diagram properties:

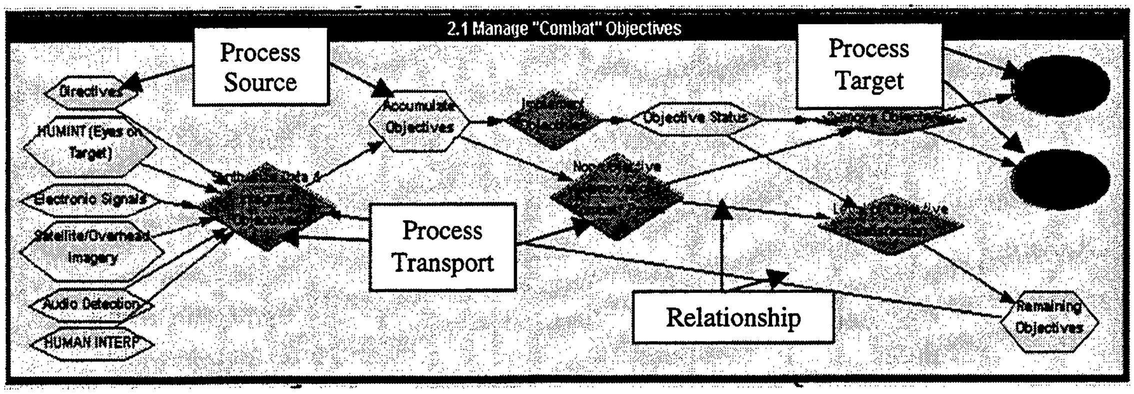
- Click on the **New Relationship** tool in the Process Diagram section.
- Click on the **up-stream process node.**
- Click on the **down-stream process node** to complete the relationship.

  The process diagram nodes are connected. (See the explanation on the following page.)

  You must <u>not</u> move the cursor AT ALL when you click.

  A relationship <u>cannot</u> be drawn to itself.

  You <u>must</u> click on the Selection tool or another tool in the toolbar to discontinue placing relationships, (i.e., turn off the New Relationship tool).



Process Diagram Explanation

Adding Decisions to Processes

By adding a decision to processes at this point in the hierarchy, you may use this a key way to access the different levels of details.

To enter a process decision:

- Click on the **process node** in the left side of the Design Thread window.
- Click on the **Add** button in the Decision section of the Process Properties region of the Design Thread window.

  The text field area below the field Name appears white.

- Click in the text field provided and type the **Decision.**
- Press the **Enter** key on the keyboard.

  Once you have added a decision, you may categorize the decision in the master decision list as well as enter information requirements for the decision.

  If you do not wish to enter additional detail:

  - Click on the **Update** button at the bottom of the Design Thread window.

Otherwise, to enter process decision details:

- Click on the name of the **process decision** to highlight it.

- Click on the **Edit** button under the Decision section of the Design Thread window.

  The Decision section is brought to the top of the display, and shows two additional fields, Master Decision Category and Information Requirements.

Entering the Process Decision as a Master Decision

To enter the process decision into the master decision category:

- Click in the text field provided, or on the pull down arrow at the end of the **Master Decision Category** field.

  A pull-down list of categories appears. (See page 23 of this user's manual.)

- Click on the name of the **category** in the list.

  That name appears highlighted in the Master Decision Category text field.

  If you do not wish to enter additional detail:

  - Click on the **Update** button at the bottom of the Design Thread window.

  To return to the top of the Process section:

  - Double-click on a **goal node** in the Design Thread list.

Entering Information Requirements for a Process Decision

To enter Information Requirements for a process decision:

- Click on the **Add** button in the Information Requirements section.

  The text field area below the field Name appears white.

- Click in the text field provided and type the **Information Requirement**.
- Press the **Enter** key on the keyboard.

Once you have added an information requirement, you may enter a data element for it.

If you do not wish to enter additional detail:

- Click on the **Update** button at the bottom of the Design Thread window.

Entering Data Elements

To enter Data Elements:

- Click on the **Add** button in the Data Elements section.

The text field area below the field Name appears white.

- Click in the text field provided and type the **Data Element**.
- Press the **Enter** key on the keyboard.

Once you have added a data element, you may enter details for that data element, including a description, rationale, type and constraints.

If you do not wish to enter additional detail:

- Click on the **Update** button at the bottom of the Design Thread window.

Entering Data Element Details

To enter data element details:

- Click in the text field provided and type a **Description**.
- Click in the text field provided and type a **Rationale**.
- Click in the text field provided and type a **Type**.

- Click in the text field provided and type **Constraints**.

To save all the detailed information:

- Click on the **Update** button at the bottom of the Design Thread window.

To return to the top of the Process section:

- Double-click on a **goal node** in the Design Thread list.

Updating Information

At many points while you are adding information to an object that you wish to keep for the analysis, you should update the information. You definitely should update it once you entered additional details concerning the goal.

To save Design Thread information:

- Click on the **Update** button at the bottom of the Design Thread window.

  To return to the top of the hierarchy:

  - Click on a **goal node** in the left side of the Design Thread window.

EDITING THE DESIGN THREAD

To edit text entries for Decisions, Processes, Information Requirements and Data Elements:

- Click on the name of the **Goal**, **Decision**, **Process**, **Information Requirement** or **Data Element** (in the left side of the Design Thread window) that contains the object you wish to edit.

  Since the Information Requirements and Data Elements fields are shareable, the pull-down arrow at the end of these fields provides you with the full list.

  That name appears highlighted in the text field.

- Click on the **Edit** button for that section.

  You may now change the information for the object that you are editing.

Editing a Process Diagram

To edit a process diagram for a selected process:

- Click on the (process diagram icon) in the left side of Design Thread window.

  A process is entered at the goal level.

  The Process Details section of Design Thread window appears.

- Click in the **text field** you wish to edit and type.

Or:

- **Edit** the diagram.

  If you delete a relationship in the Process Diagram section of the Design Thread window, then click on the Update button, it will not show the updating of that removal in the FAH window automatically.

  To see the changes via the FAH window:

  - ***Close the FAH window by clicking in the close box***
  - ***Click on the Open Functional Abstraction Hierarchy button in the display toolbar.***

Saving Edit Changes

To save the editing changes:

- Click on the **Update** button at the bottom of the Design Thread window.

Deleting

To delete text entries for Decisions, Processes, Information Requirements and Data Elements:

- Click on the name of the **Decision**, **Process**, **Information Requirement** or **Data Element** you wish to delete.

  Since the Information Requirements and Data Elements fields are shareable, the pull-down arrow at the end of these fields provides you with the full list.

  That name appears highlighted in the text field.

- Click on the **Delete** button for that section.

  The Delete? window appears.

  To delete:

  - Click on the **Yes** button.

    The property is deleted and the Design Thread and FAH windows are updated automatically.

  If you do not wish to delete:

  - Click on the **No** button.

WORKSPACE DESIGN : BUILDING THE USER INTERFACE STRUCTURE

To support multiple levels of visualization design, the CACSE Tool provides you with a graphical tool for defining work space design. This graphical tool has the capability to model parallel (information presented together within a coordinated view) or serial (total replacement of one coordinated view with another) displays of data. It allows you to define graphical forms that comprise coordinated views.

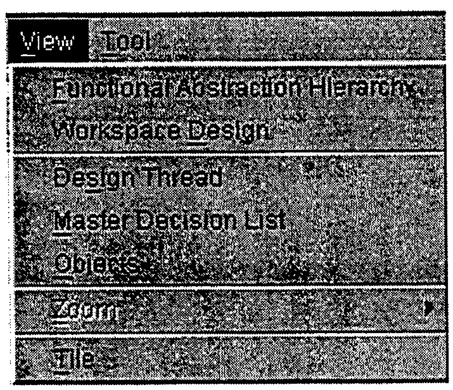
The Workspace Design graphical tool provides you with a way to incorporate your analysis into the display design by using the Workspace Design window to: 1) build and modify process views in the workspace; 2) define the layout of these views; 3) define graphical forms that comprise the coordinated process views; and 4) define the information content of the process views or graphic forms (using the data you entered in the Design Thread). A process view is a single window or display; however, the display has multiple components. The layout level allows you to place where these parts will appear on the window/display. Finally, the graphical form section allows you to identify specific components of the display.

When you are ready to design the workspace:

- Click on the **Workspace Design** button below the menu bar.

Or

- Click on the **View Menu**.



View Menu

- Click on **Workspace Design**.

  The Workspace Design window opens in the left side of the display. The Design Thread window should be concentrated on the folder for Workspace Design. The Workspace Design window depicts its first level, Work Space, and contains six tools: Selection tool, New Process View tool, New Process View Layout tool, New Graphic Form tool, New Inclusive OR tool and a New Relationship tool

Understanding the Workspace Design Levels

The three sections that divide the Workspace Design window horizontally are display levels. There are essentially three types of levels; however, because process view objects may be placed in the Process View Layout level, the Process View and Process View Layout levels may appear more than once. These display levels are explained in the table below.

Work Space	This level exists via programming and cannot be deleted, moved or selected. Only process view objects may be created in this level.
Process View	This level becomes visible when once the Show Level option is selected from the Process View pop-up menu; this menu is accessible once a process view object is created in the Work Space level. Only process view layout objects may be created in this level. It is possible to have multiple Process View Levels. This level may be resized (smaller/larger).
Process View Layout	This level becomes visible when once the Show Level option is selected from the Process View Layout pop-up menu; this menu is accessible once a process view layout object is created in the Process View level. Process view, inclusive OR and graphic form objects may be created in this level. Because a process view object may be placed in this level, it is possible to have multiple Process View Layout levels. This level may be resized (smaller/larger).

Display Levels Explanation Table

See pages 42, 46 and 47 of this user's manual for instructions on using these pop-up menus.

Changing the Size of the Display Level

The Process View and Process View Layout levels of the Workspace Design window may be resized (expanded or collapsed) vertically.

To change the size of a level of the Workspace Design window:

- Click on the **Selection** tool.

- Position the Selection tool (mouse) over the **separator** (line at the top of the level) you wish to expand or collapse until the cursor takes the form of a crosshair.

- **Drag** the separator to the desired location.

  When you resize a level, if the resizing is too small for the object, the level will reset to its previous size. Also, if you size an object vertically larger than the space provided for the level, the level will grow in size to accommodate.

Placing Objects in the Display Level

You may move certain objects into certain display levels. Refer to the Display Levels Explanation Table for restrictions. Levels may appear nested. For example, you may place a process view in the Process View Layout level; therefore, another Process View level would appear below that Process View Layout level.

To place an (allowed) object in a display level:

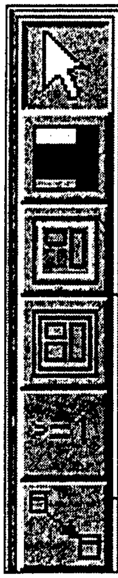
- Click on the object (**Process View**, **Process View Layout**, **Graphic Form**, or **Inclusive OR**) in the Workspace Design window to highlight it.

  Highlighting is indicated by solid red squares surrounding the perimeter of the object.

- **Drag** the object to the desired level.

Using the Workspace Design Tools

The six tools shown along the upper left side of the Workspace Design window are defined in the table below.



Selection Tool	Click on the Selection tool to highlight items in the Workspace Design window to select, move or resize (PVL only) an object.
New Process View	Click on the New Process View tool to create views in the Workspace Design window. A process view may be created only in the Work Space and Process View Layout levels.
New Process View Layout	Click on the New Process View Layout tool to create a layout of the process view. A process view layout may be created only in the Process View Layout level.
New Graphic Form	Click on the New Graphic Form tool to create a graphic form in the layout. A graphic form may be created only in the Process View Layout level.
New Inclusive OR	Click on the New Inclusive OR tool to crate an inclusive OR. An inclusive OR may be created only in the Process View Layout level.
New Relationship	Click on the New Relationship Tool to link either a process view or graphic form to an inclusive OR in order to show support

Workspace Design Window Toolbar Definition Table

Creating a New Process View

To create a new process view:

- Click on the **New Process View** tool.
- Click in the **Work Space** level of the Workspace Design window.

  A rectangle appears containing the name ProcessView#.

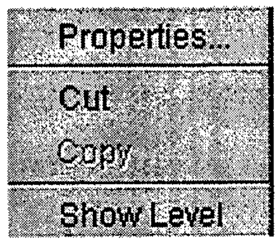
  The number is updated based upon how many process views you have created, i.e., ProcessView1.

  If you at least one process view object already exists in the Work Space level, and one process view layout object already exists in the Process View level, you also may place process view objects in the Process View Layout level.

To define the properties of a process view:

- Using the **Selection** tool, click on a **process view** node using the right mouse button.

  The **Process View** pop-up menu appears.



Process View Pop-up Menu

- Click on **Properties...** .

  The properties of that view appear in the right section of the Design Thread window, with the Process View section appearing at the top of the display. (See the example on the following page.)

- Click in the text field provided and type a **Name** for the view.

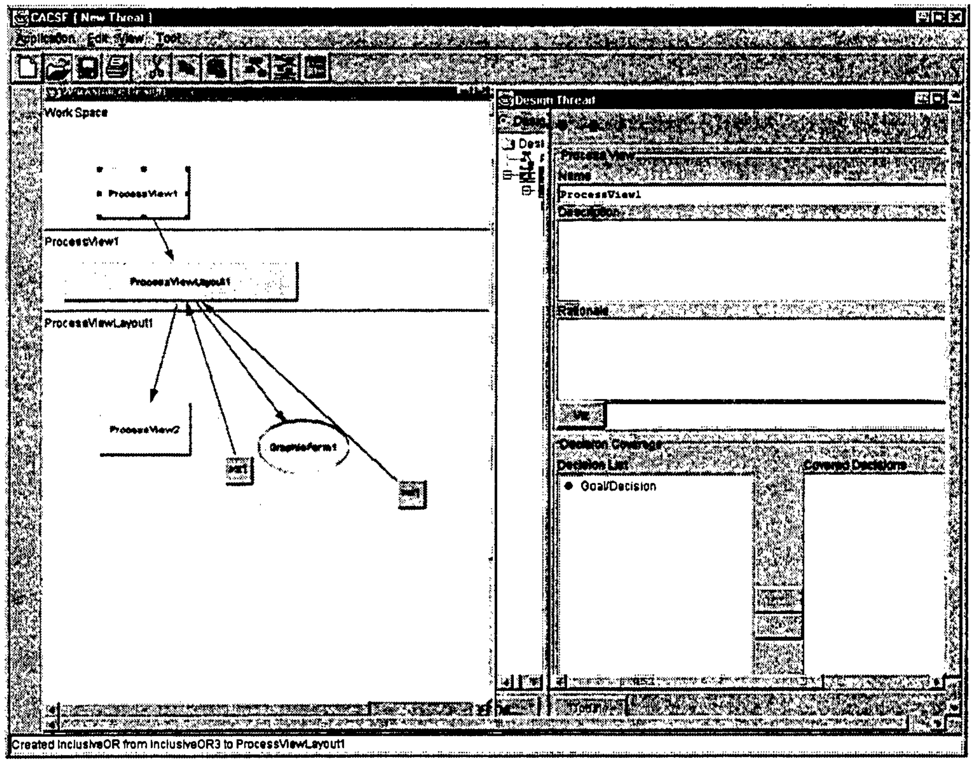
- Click in the text field provided and type a **Description**.

- Click in the text field provided type a **Rationale**.

  You also may enter choose to create a depiction of the view by attaching a Microsoft® PowerPoint file via the VIZ button, as well as define the decision coverage.

  If you do not wish to add additional detail at this time:

- Click on the **Update** button at the bottom of the right side of the Design Thread window.



Workspace Design Window &
Process View Section of Design Thread Window

Creating a Visual Display

To create a visual display for a view:

- In the text field provided next to the Viz button, type a **name** for the display you wish to create, or enter the name of one already created to modify it.

- Click on the **Viz** button.

  If you do not type the name first and click on the Viz button, the "Viz Filename Missing" window appears, prompting you to enter a name; if this happens, simply

  - ***Click on the OK button and enter the name first.***

  If the file is new, the New File window appears prompting you to remember to save the file.

  - ***Click on the OK button.***

You are prompted to save the file with the ".ppt" extension; then the Microsoft® PowerPoint program is opened allowing you to create and link a graphic file to this view.

You also may search for a ".ppt" already created to link to this view.

Defining Functional Coverage for a Process View

The Decision Coverage section of the View display provides end-to-end support for the design thread. To define functional "coverage" for each display, move decisions (only) from the Decision Lists to the Covered Decisions. This "move" shows which decisions will be covered by the selected view.

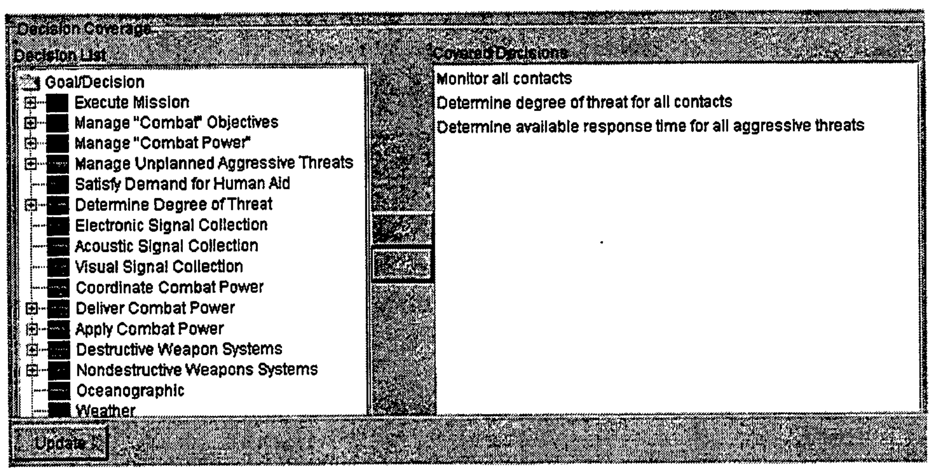
To cover decisions in a view:

- Click on a **Decision** in the Decision List in the Decision Coverage section.

  Only decisions may be moved.

- Click on the >> button to move the decision to the Covered Decisions list.

  The decision now appears in the Covered Decisions list



Decision Coverage Section

To remove a decision from a view:

- Click on a **Decision** in the Covered Decisions list in the Decision Coverage section.

- Click on the << button to move the decision out of the Covered Decisions list.

  The decision is removed from the Covered Decisions list.

- Click on the **Update** button at the bottom of the right side of the Design Thread window.

Creating a Process View Layout

To create a Process View layout of the new view:

- Using the **Selection** tool, click the **right mouse button** on a **process view** node.

  The Process View pop-up menu appears. (See page 42.)

- Click on **Show Level.**

  The ProcessView# level of the display appears below the Work Space level. A teal rectangle appears containing the name ProcessViewLayout#. (See the example on page 43.)

  The number is updated based upon how many process view layouts you have created, i.e., ProcessViewLayout1.

Resizing the Process View Layout Object

When you create a process view layout object, it is placed in the display at a predetermined size by programming. The process view layout object's size may be resized (enlarged or reduced); this is the only object in the workspace design display, which may be altered in his way.

To change the size of a process view layout object:

- Using the **Selection** tool, click on the **process view layout** node that you wish to resize.

  The node becomes highlighted (solid red squares).

- Position the cursor over one of the **red squares** until it takes the shape of a double-headed arrow.

- **Drag** the **red square** until the node is the size you desire.

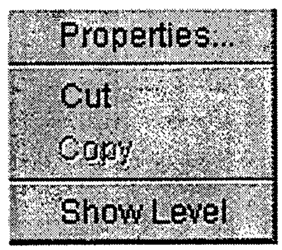
To define the properties of a layout region:

- Using the **Selection** tool, click on a **process view layout** node using the right mouse button.

  The **Process View Layout** pop-up menu appears. (See the following page.)

- Click on **Properties...** .

  The properties of that view appear in the right section of the Design Thread window, with the Process View section appearing at the top of the display.



Process View Layout Pop-up Menu

- Click in the text field provided and type a **Name** for the view.
- Click in the text field provided and type a **Description**.
- Click in the text field provided type a **Rationale**.
- Click on the **Update** button at the bottom of the display.

Creating the Graphic Form

To create a Graphical Form of the new view:

- Using the **Selection** tool, click the **right mouse button** on a **process view layout** node.

  The Process View Layout pop-up menu appears.

- Click on **Show Level.**

  The ProcessViewLayout# level of the display appears below the Work Space level. A green oval appears containing the name GraphicForm#.

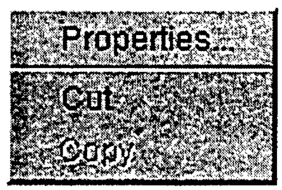
  The number is updated based upon how many graphic forms you have created, i.e., GraphicForm1.

Defining Properties of a Graphic Form

To define the properties of a control:

- Using the **Selection** tool, click on a **graphic form** using the right mouse button.

  The **Graphic Form** pop-up menu appears.



Graphic Form Pop-up Menu

- Click on **Properties...** .

  The properties of that view appear in the right section of the Design Thread window, with the Graphical Form section appearing at the top of the display.

- Click in the text field provided and type a **Name** for the graphical form.

- Click in the text field provided and type a **Description**.

- Click in the text field provided type a **Rationale**.

  You also may enter choose to create a depiction of the graphic form by attaching a Microsoft® PowerPoint file via the VIZ button, as well as define the decision coverage.

If you do not wish to add additional detail at this time:

- Click on the **Update** button at the bottom of the right side of the Design Thread window.

Creating a Graphic Form Visual Display

To create a visual display for a graphic form:

- In the text field provided next to the Viz button, type a **name** for the display you wish to create, or enter the name of one already created to modify it.
- Click on the **Viz** button.

If you do not type the name first and click on the Viz button, the "Viz Filename Missing" window appears, prompting you to enter a name; if this happens, simply

- ***Click on the OK button and enter the name first.***

If the file is new, the New File window appears prompting you to remember to save the file.

- ***Click on the OK button.***

You are prompted to save the file with the ".ppt" extension; then the Microsoft® PowerPoint program is opened allowing you to create and link a graphic file to this view.

You also may search for a ".ppt" already created to link to this view.

Defining Functional Coverage for a Graphic Form

The Decision Coverage section of the Graphical Form display provides end-to-end support for the design thread. To define functional "coverage" for each display, move decisions (only) from the Decision Lists to the Covered Decisions. This "move" shows which decisions will be covered by the selected view.

To cover decisions in a view:

- Click on a **Decision** in the Decision List in the Decision Coverage section.

Only decisions may be moved.

- Click on the >> button to move the decision to the Covered Decisions list.

The decision now appears in the Covered Decisions list. (See the example on page 44.)

To remove a decision from a view:

- Click on a **Decision** in the Covered Decisions list in the Decision Coverage section.

- Click on the << button to move the decision out of the Covered Decisions list.

  The decision is removed from the Covered Decisions list.

- Click on the **Update** button at the bottom of the right side of the Design Thread window.

Understanding Automatic Relationships

Automatic relationships are defined by programming. They are relationships that are "automatically" drawn by the Workspace Design tool once certain objects are place in the Workspace Design window. Automatic relationships exist between the following pairs of objects, as depicted in the illustration on the following page: process view to process view layout; process view layout to graphic form; process view layout to process view; inclusive OR to process view layout.

These relationships cannot be selected, moved or deleted.

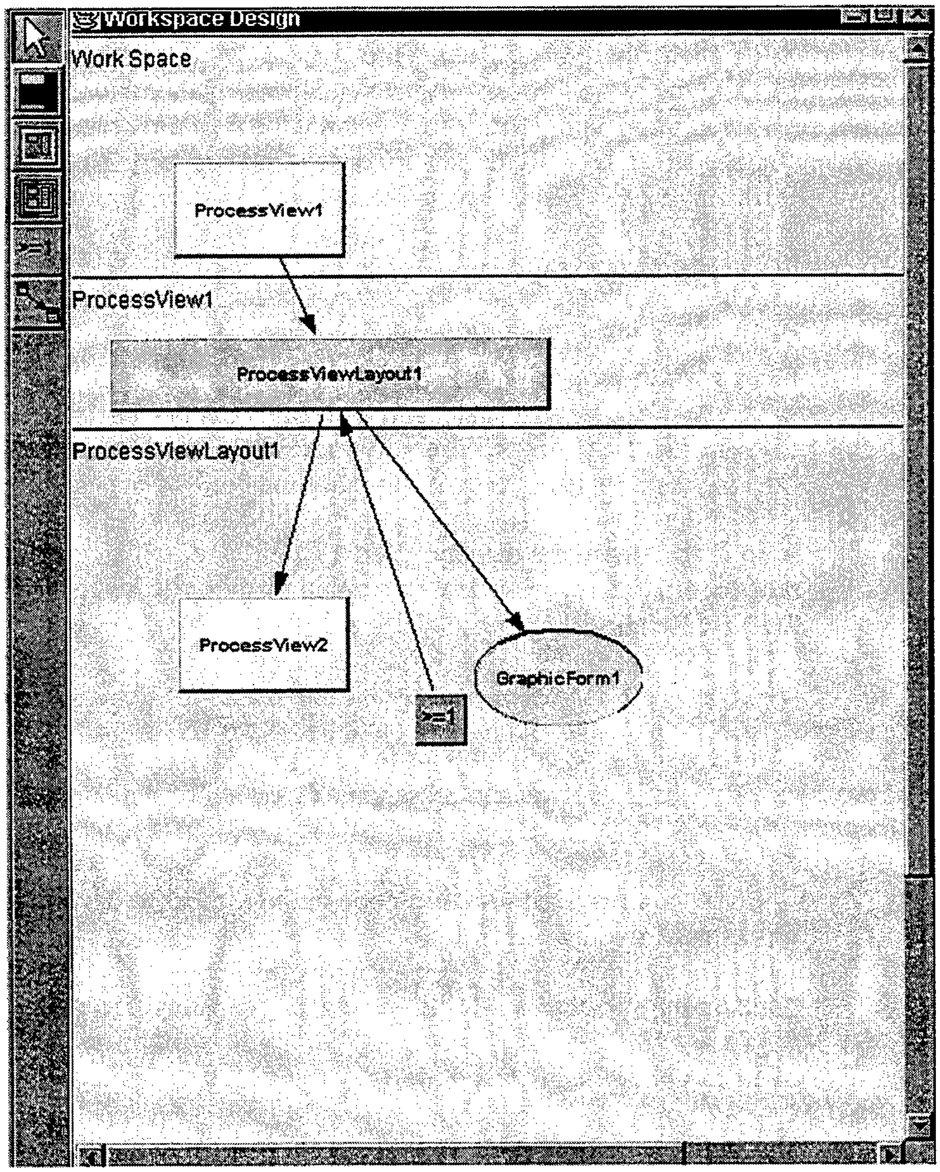
Defining Relationships for the Display

You may define relationships between certain objects by drawing them using the New Relationship tool in the Process View Layout level of the Workspace Design window. Only two object pairs may have supporting relationships, process view to inclusive OR and graphic form to inclusive OR. Once you have defined relationships between objects, the automatic relationships become hidden from view; however, if you delete the relationships you have created, the automatic relationships are revealed again.

To define relationships between views:

- Click on the **New Relationship** tool in the Workspace Design window.

- Click on either a **process view** node or an **graphic form** node.



Workspace Design Window Explaining Automatic Relationships

- Click on an **inclusive OR** node to complete.

The nodes are connected.

You must not move the cursor AT ALL when you click.

A relationship cannot be drawn to itself.

You must click on the Selection tool or another tool in the toolbar to discontinue placing relationships, (i.e., turn off the New Relationship tool).

Defining a New Inclusive OR

You may choose to "or" the relationship using the New Inclusive OR tool with the New Relationship tool. The inclusive OR establishes that at least one of any number of supporting process views or graphic forms may occupy the specified region in the process view layout. Use the New Relationship tool to link supporting views or forms to the inclusive OR; the inclusive OR is automatically linked to the supported view layout.

To create a new inclusive OR:

- Click on the **New Inclusive OR** tool to the left of the Workspace Design window.

- Click inside the **Process View Layout** level of the Workspace Design display where you wish to place the "or".

  A node appears in the display. You may either continue creating "or"s, or connect either a process view or graphic form to the inclusive OR using the New Relationship tool.

You must click on the Selection tool or another tool in the toolbar to discontinue placing "or"s, (i.e., turn off the New Inclusive OR tool).

To define relationships between process view or graphic form nodes and the inclusive OR:

- Click on the **New Relationship** node in the toolbar to the left of the Workspace Design display.

- Click on the **supporting process view** or **graphic form** node.

  Click on the **inclusive OR** node.

  This defines the first of the supporting view/form in the OR relationship.

- **Repeat** this process for additional nodes that are to be included in the OR relationship.

- Click on the **inclusive OR** node.

  The nodes are connected.

  (See notes on the previous page.)

New and inclusive OR relationships that you have defined (i.e., not automatic relationships) may be deleted.

To delete a relationship from the Workspace Design window:

- Using the Selection tool, click on the **relationship arrow** or **inclusive OR** node you wish to delete.

  The arrow/node becomes highlighted (solid red squares).

- Click on the **Edit** menu in the menu bar.

- Click on **Cut**.

  The relationship arrow is removed from the display.

  You also may use the cut option on the pop-up menus.

Moving Workspace Design Objects

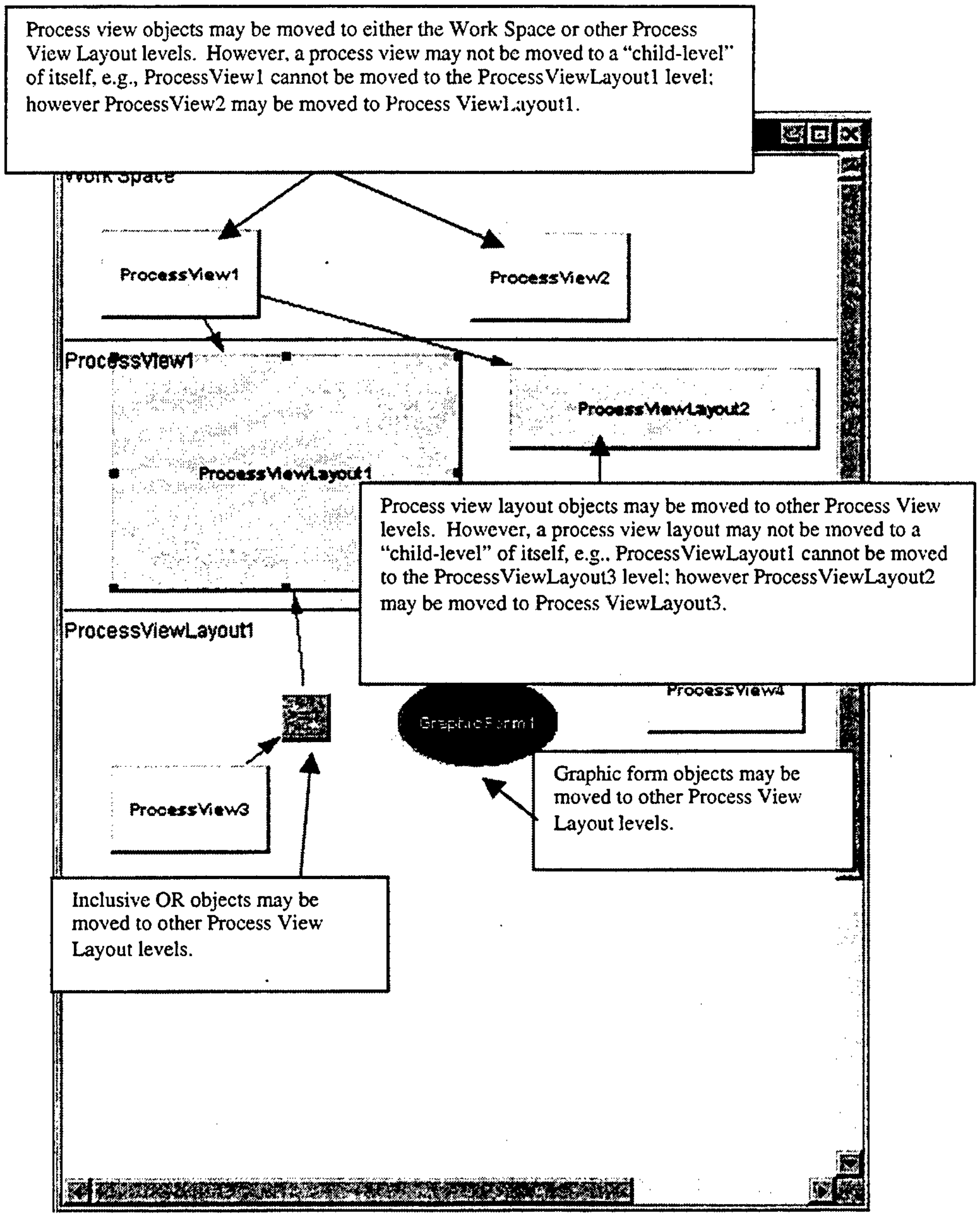
The Workspace Design objects may be moved to different levels of the display, as long as the rules displayed in the illustration on the following page are applied.

To move an object:

- Click on the **object** to highlight it (solid red squares).

- **Drag** the **object** to the position or level you desire.

  The object along with any relationship connected to it is moved.



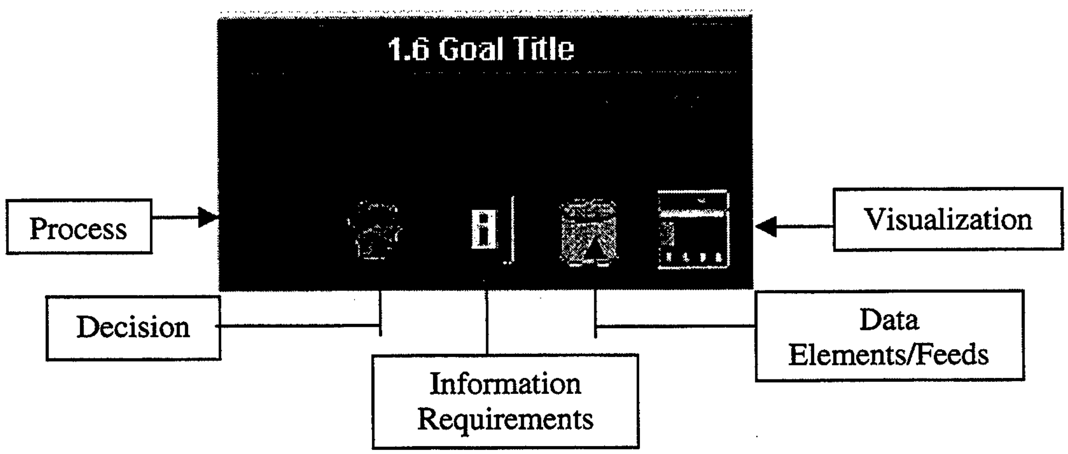
Object Movement Limitations Explanation

EXPLORING VIEWING OPTIONS TO TRACK COMPLETENESS

CACSE supports several ways of displaying the information you have entered into the FAH, the Design Thread and Workspace Design windows. By viewing the information in different ways, you can track the completeness of your analysis.

Viewing Detail in the FAH

The FAH window can show the properties of a goal using three different views: icon view, detail view, and process view. The icon view displays the name of the goal preceded by its hierarchical numbering. The process view expands upon the icon view, to include the process diagram for the goal. Finally, the detail view expands upon the icon view, to include the icons for each of the properties of the goal. (See the explanation below.)



Explanation of Detail View Icons

Once a property has been entered for a goal, the representing icon in the detail view shows an outlined red check mark. This indicates that some information has been filled in.

Once you feel the information is complete:

- Click on the **outlined checkmark**.

  It will become filled in, indicating completeness; this is a user-defined object. Once you have selected to fill in the check mark, a dialog prompts you with "Completed?".

To confirm:

- Click on the **dialog box**.

  The series of five circles shown on the Icon View represent, in a compact form, the same information as in the Detail View. A half-filled circle is the same as an outlined red check mark. A solid circle is the same as a filled-in check mark.

You cannot make changes in the completeness at the Icon View.

You will not be able to select "Process View" if a process diagram has not been defined.

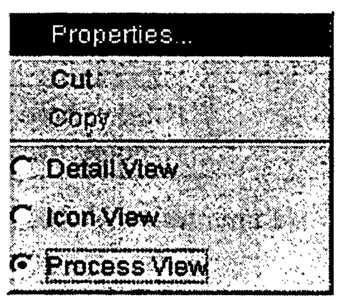
Changing Views of the FAH Window

To change the "view" of the goal node:

- Position the mouse on the **goal node** you wish to view.

- Click on the **right mouse button**.

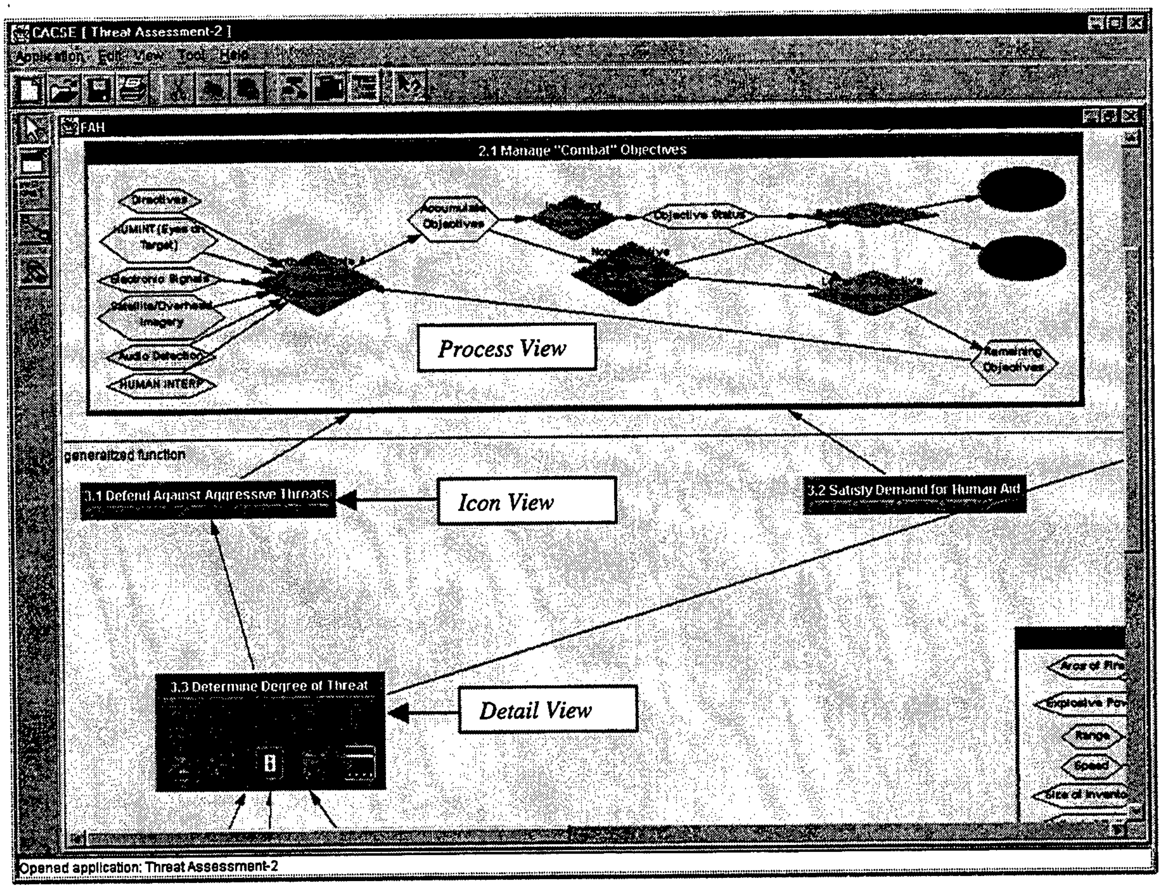
  The Goal pop-up menu with Properties, Editing and Viewing options appears.



Goal Pop-up Menu

- Click in the radio button preceding the **view** you wish to use.

  The selected view appears. (See the following page.)



FAH Window Showing CACSE's Three Views

Using Graphic Overlay to View Visualization in the FAH

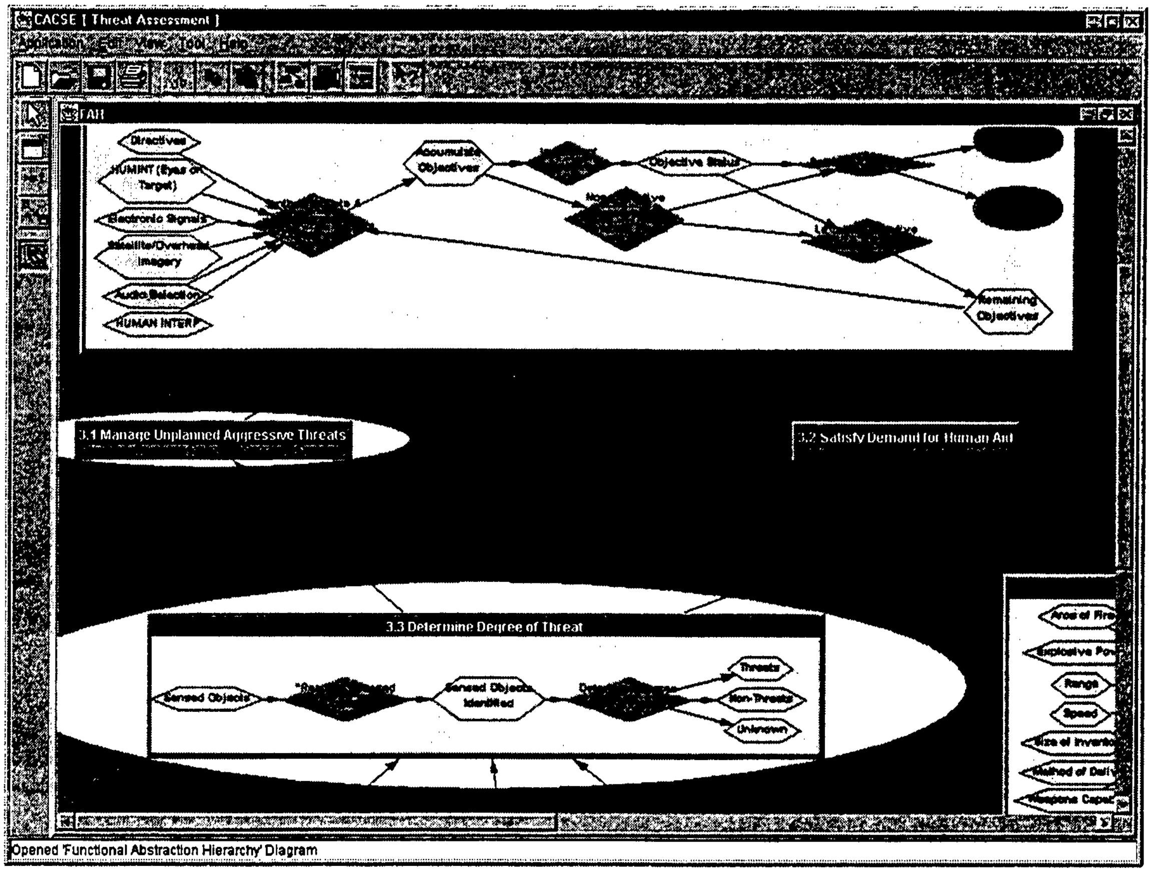
The Graphic overlay tool allows you to highlight goals that contain decisions covered by a specific visualization in the workspace. These decisions were defined into the display using the Decision Coverage section of either a Process View or Graphical Form display.

For more information, refer to the respective sections Workspace Design: Building the User Interface, beginning on page 38 of this user's manual.

To highlight the graphic overlay:

- Click on the **Graphic Overly** tool in the FAH window.

  The display background color turns gray, with highlighted circles of white surrounding goals containing decisions covered by a display. (See the example on the following page.)



Graphic Overlay Example

To turn off the graphic overlay:

- Click on the **Selection** tool in the FAH window.

Viewing the Design Thread

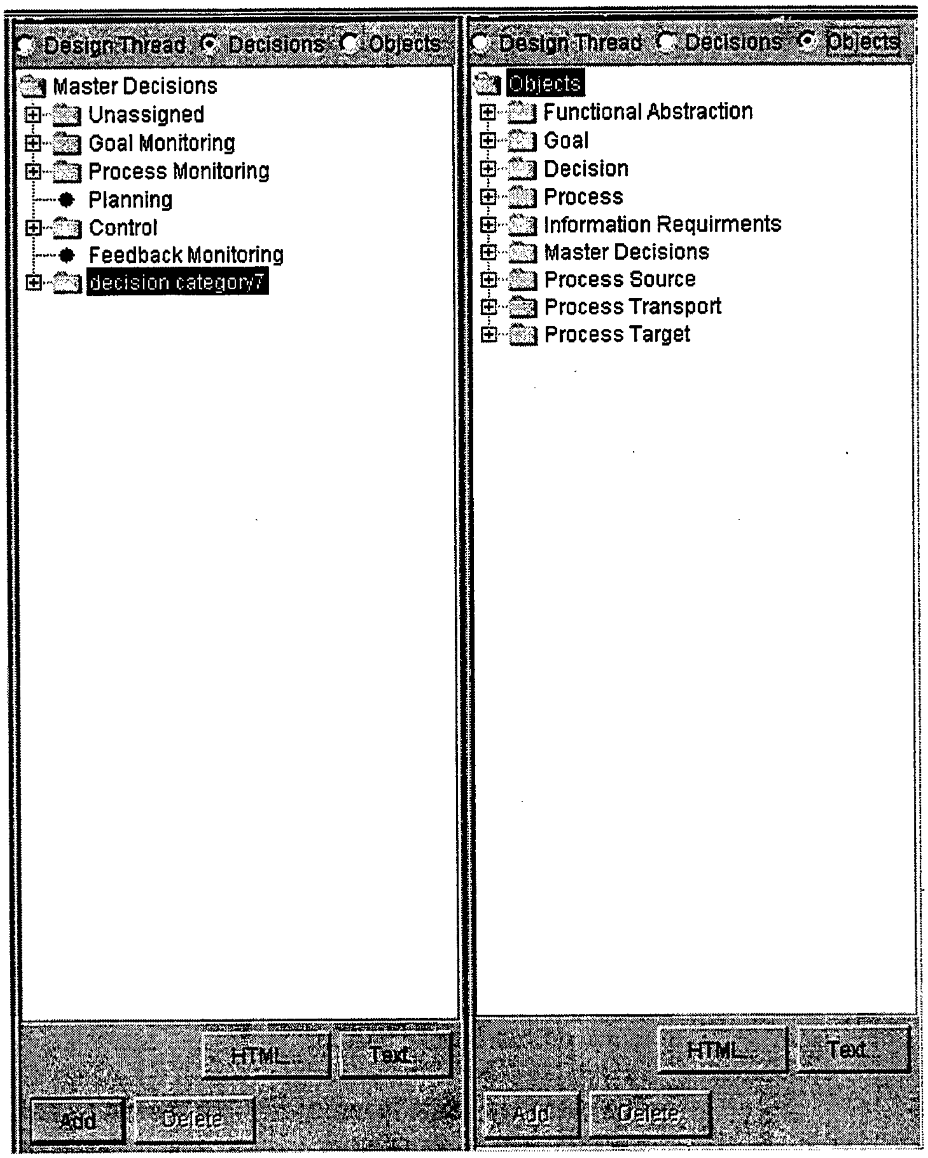
The default view in the Design Thread window for the listing of objects in the application is the Design Thread view; this view shows each object and its properties according to the object hierarchy. You may choose to view the objects according to their category or grouping by selecting the Objects view, or, you may view the decisions according to the master decision category by selecting the Decisions view.

To change the view of the Design Thread window:

- Click on the **radio button** in the top left side of the Design Thread window representing the view you wish to have displayed (either Design Thread [default], Decisions or Objects).

  The selected view is displayed. (See the examples below.)

  You also may select the respective view from the View menu.



Example Decision and Object Lists

You may not add or delete items to the Design Thread or Objects lists. However, the Add and Delete buttons which appear inactive (grayed out) when viewing the Design Thread and Objects displays become active when viewing the Decisions list.

To add a Master Decision Category:

- Click on the **radio button** for Decisions in the top left side of the Design Thread window.

  The Decisions view is displayed.

- Click on the **Master Decisions** folder.

  The Add button at the bottom of the display becomes active.

- Click on the **Add** button.

  A new decision category appears at the bottom of the list.

  To add a decision to that new category:

  - Click on the **Add** button again.

    A decision is added to the category you have created. You may now edit that decision by clicking on it in the list to display its properties. (Refer to the "Entering Decision Details", "Entering a Master Decision" and "Updating Information" sections of this user's manual, on pages, 22, 23 and 35 respectively.)

  To delete a Master Decision Category that you have created:

  - Click on the name of a **Master Decision Category** (folder) that in the Decisions listing in the left side of the Design Thread window.

    The Delete button at the bottom of the display becomes active.

  - Click on the **Delete** button.

    The category name is removed from the list as are any decisions placed under that category.

    The categories that come with CACSE cannot be deleted.

Querying the Database

CACSE's graphical tool provides checks for identifying gaps in the design thread by providing you with a way to identify 1) decisions not associated with a goal, 2) goals and decisions not covered by displays and 3) displays not associated to decisions. You may view this graphically in a "top-down" manner in the FAH window by using the Detail view of the goal, and looking for indicators. Or, you may query the design thread using the "bottom up" view arrow on the right side of the Design Thread window. Queries may be done beginning at the following levels: goals, processes, decisions, information requirements and data elements.

To query the design thread for gaps in the analysis:

- Click on any of the following objects in the left side of the Design Thread window to show its properties: **Goal, Process, Decision, Information Requirement** or **Data Element.**

- Click on the  (up arrow) on the right side of the Design Thread window.

  The arrow's color changes to red, indicating that the "bottom up"/Query view is now being displayed. The (Object) Information section appears at the top of the display. It contains Search Results and Search Criteria sections. (See the example on the following page.)

  To return to the "top down" display (and end querying):

  - Click on the **down arrow.**

    The down arrow will appear red, indicating the top down view is displayed.

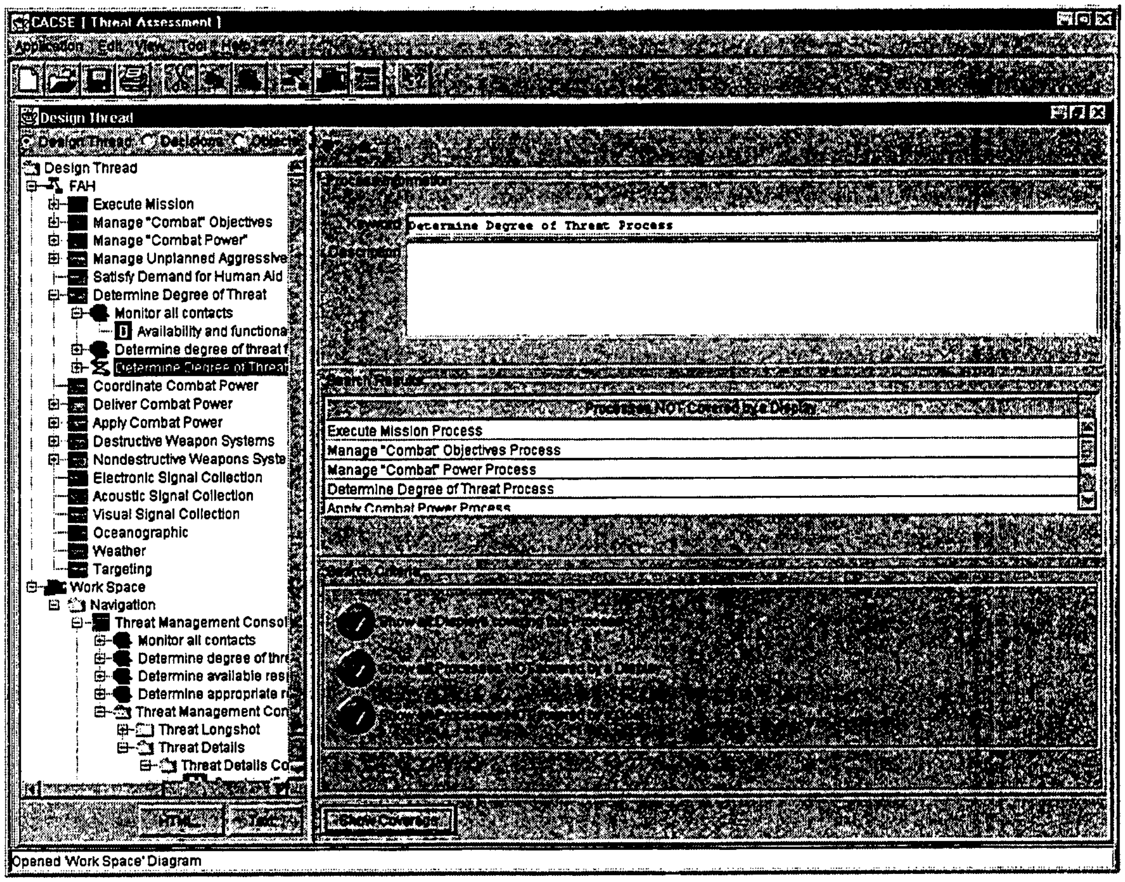
Otherwise, to query the design thread:

- Click on the  (upside down triangle) to the left of the statement representing the query you wish to select.

  The triangle's color changes to green, indicating that the representing statement has been selected.

- Click on the **Show Coverage** button at the bottom of the display.

  The queried results appear in the Search Results section of the display. (See the example on the following page.)



Example Query of Process Displays

Viewing the Workspace

To view the Workspace Design window:

- Click on the (Open Workspace Design button) in the display toolbar across the top of the CACSE displays.

Or:

- Click on the **View menu**. (See page 38.)

- Click on **Workspace Design**.

  The Workspace Design window appears on top of the FAH window. The Design Thread window should be concentrated on the folder for Workspace Design.

OUTPUTTING THE CACSE APPLICATION RESULTS

The CACSE tool supports the generation of the text (ASCII format) and HTML files via the buttons at the bottom of the Design Thread window. It also supports outputting files in systems requirements specification (SRS) format and document view, via the options in the Tool menu.

Creating a Text File

You may create a text file of everything in the object hierarchy, or of a specific (selected) object node and its "children".

To create an ASCII text file of the entire CACSE object hierarchy:

- Click on the **FAH icon** in the left side of the Design Thread window.

Or, to create an ASCII text file of a selected (highlighted) object node in the CACSE object hierarchy:

- Click on the **icon** representing the specific node (and it's "children") in the left side of the Design Thread window.

- Click on the **Text...** button at the bottom of the Design Thread window.

  A window appears prompting you to name the file for placement in the CACSE/bin folder.

- Click in the text field provided and type a **Filename**.

- Click on the **Save** button.

  The file should be saved in the CACSE/bin folder with the name you've given it.

Creating an HTML File

You may create an HTML file of everything in the object hierarchy, or of a specific (selected) object node and its "children".

To create an HTML file of the entire CACSE object hierarchy:

- Click on the **FAH icon** in the left side of the Design Thread display.

  OR

To create an HTML file of a selected (highlighted) object node in the CACSE object hierarchy:

- Click on the **icon** representing the specific node (and it's "children") in the left side of the Design Thread window.
- Click on the **HTML...** button at the bottom of the Design Thread window.

  A window appears prompting you to name the file for placement in the CACSE/bin folder.

- Click in the text field provided and type a **Filename** with a ".HTM" extension.
- Click on the **Save** button.

  The file should be saved in the CACSE/bin folder with the name you've given it.

GENERATING AN SRS

The Tool menu contains an option for outputting the CACSE results called Generate System Requirements. If you generate systems requirements, they will be output in SRS format.

To output your CACSE application analysis in SRS format:

- Click on the **Tool** menu.
- Click on **Generate System Requirements.**

  A window appears prompting you to name the file for placement in the CACSE/bin folder.

- Click in the text field provided and type a **Filename** with a ".HTM" extension..
- Click on the **Save** button.

  The file should be saved in the CACSE/bin folder with the name you've given it.

Printing CACSE FAH

You may print the FAH to either a regular printer or plotter.

To print the FAH:

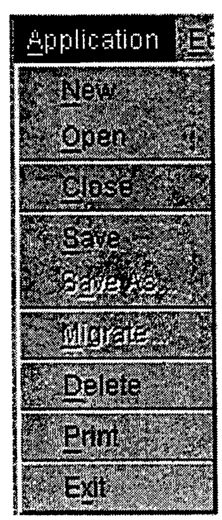
- Click on the **Application** menu.
- Click on **Print.**

  The Page Setup window appears.

You may wish to check page specifications and printer selection.

- Click on the **OK** button.

  The Print window appears.



Application Menu

- Click on the **OK** button.

APPENDIX: MIGRATING THE DATABASE

As new versions of CACSE are released, there will become a need to update any application files created in a previous version of CACSE to the most recent. This is accomplished by using the "Migrate Database" feature.

To migrate database files

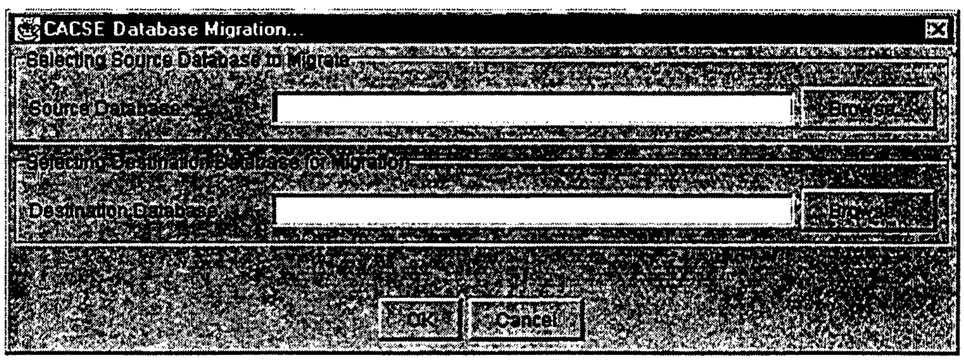
- **Close** any open Applications.

  You <u>cannot</u> migrate the database when an application is opened.

- Click on the **Application** menu.

- Click on **Migrate...** .

  The CACSE Database Migration... window appears. You may "browse" the CACSE/bin folder to locate the name of the database you wish to either use as your source or destination.



CACSE Database Migration Window

- Click in the text field provided and type the name of a **Source Database/**

Or:

- Click on the **Browse...** button to locate one.

  For all applications created in CACSE, there are three files created. Each has the same filename with different extensions: ".ODB", ".ODF", and ".ODT".

For migrating a database:

- Click on the filename with the "**.ODB**" extension.
- Click in the text field provided and type the name of a **Destination Database.**

Or:

- Click on the **Browse...** button to locate one.

  For all applications created in CACSE, there are three files created. Each has the same filename with different extensions: ".ODB", ".ODF", and ".ODT".

- Click on the filename with the "**.ODB**" extension.

Once you have entered the database names:

- Click on the **OK** button in the CACSE Database Migration... window.

  Upon successful database migration, a confirmation dialog appears. You now are able to open the database with the current version of CACSE.