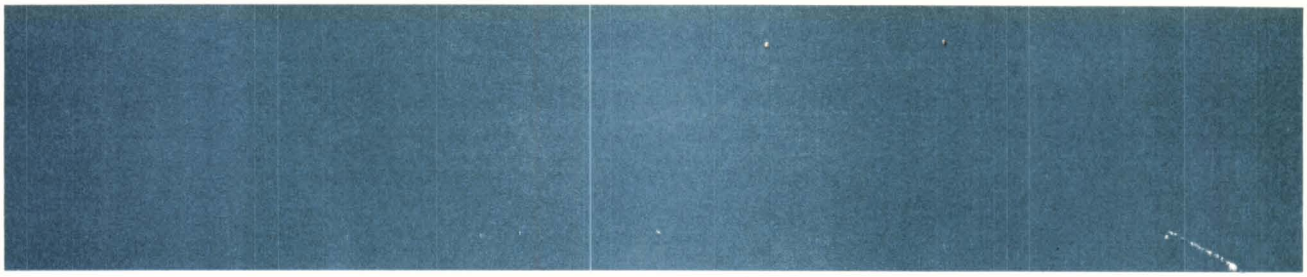
1-1-1995

Integrated Eagle/BDS-D Research Resolution Of Behavior Representation Methods Of CCTT SAF And ModSAF: Ecp 7 Work Plan

Robert W. Franceschini

Michael D. Myjak

INSTITUTE FOR SIMULATION AND TRAINING



Contract Number N61339-92-K-0002
October 2, 1995

ECP 7 Work Plan

Integrated Eagle/BDS-D

Research Resolution of Behavior Representation Methods of CCTT SAF and ModSAF

Institute for Simulation and Training
3280 Progress Drive
Orlando FL 32826

University of Central Florida
Division of Sponsored Research

Integrated Eagle/BDS-D
Research Resolution of Behavior Representation Methods of CCTT SAF and ModSAF

ECP 7 Work Plan

October 2, 1995

Prepared by
Robert W. Franceschini and Michael D. Myjak

Reviewed by
Clark R. Karr

Institute for Simulation and Training - Computer Generated Forces

Contract N61339-92-K-0002 - CDRL A00A

Institute for Simulation and Training - 3280 Progress Drive - Orlando, Florida 32826-0544

University of Central Florida - Division of Sponsored Research and Training

Introduction

The Integrated Eagle/BDS-D project is attempting to integrate constructive, aggregate unit wargame models typically used for analysis, such as Eagle, with virtual, individual vehicle level simulations used for training, such as SIMNET/BDS-D. Within the project, that integration is achieved by making it possible for Eagle units of battalion size or smaller to be automatically disaggregated into their component vehicles in the virtual battlefield. Once disaggregated, the vehicles move and fight in the virtual simulation under the control of a Computer Generated Forces (CGF) system. When their engagement is concluded, the vehicles are removed from the virtual simulation and the Eagle units are reaggregated.

The Integrated Eagle/BDS-D project is a cooperative effort involving the TRADOC Analysis Center (TRAC) and the Institute for Simulation and Training (IST). IST's participation is sponsored by STRICOM under contract N61339-92-K-0002. IST is involved in Computer Generated Forces research relevant to the linkage of constructive and virtual simulations.

This document is IST's Work Plan for ECP 7 to the Integrated Eagle/BDS-D project. ECP 7 has two goals:

1. Research the behavior specification mechanisms used in the Close-Combat Tactical Trainer's Semi-Automated Forces (CCTT SAF) and Modular Semi-Automated Forces (ModSAF) and to investigate approaches to integrate the mechanisms to create a single behavioral specification mechanism. For example, the CCTT SAF includes behaviors determined as a result of an extensive knowledge engineering process. As a research tool, experimental behavioral representation techniques have been developed in ModSAF. This research project will examine these and other aspects of the behavioral representation of these systems.
2. Develop the class structure and instance objects for the Eagle Frame Model. This work involves the development of the CLOS Methods to replace KEE Primitives used by the Eagle runtime infrastructure. Building on previous work at IST, this task requires changes to Eagle code which will touch every file that is a part of the Eagle system, some of them requiring major changes, others only minor changes.

In this document, the plan for each goal is presented separately for clarity. The plan for Goal 1 was prepared by Robert Franceschini; the plan for Goal 2 was prepared by Michael Myjak.

Goal 1: Research Resolution of Behavior Representation Methods of CCTT SAF and ModSAF

To accomplish this goal, IST will research CCTT's CIS's, their implementation in the CCTT SAF, and ModSAF's behavioral representation techniques. For this work, ModSAF version 1.5.1 will be used until version 2.0 becomes available, and then version 2.0 will be used for the remainder of the project.

Task List

For each task/subtask, the following information is given:
Task Description
Start Date
Duration
Subtasks (if any)
Dependencies

Task 1: Identify CIS's to examine.

Task Description
Identify four to six Combat Instruction Sets to focus on during this research. These CIS's will be selected from the

set of CIS's that have been implemented in the CCTT SAF. These CIS's will be selected so that approximately half of them have some corresponding ModSAF representation, and the remainder will have no corresponding ModSAF representation.

Start Date

10/2/95

Duration

1 week

Dependencies

Predecessors

None

Successors

Task 2: Understand CIS's
Task 3: Study CCTT implementation of CIS's
Task 4: Study ModSAF's behavior control mechanisms

Task 2: Understand CIS's.

Task Description

Study the CIS's selected in Task 1. The goal of this study is to understand the military doctrine behind each task, so that implementations of the CIS's can be analyzed.

Start Date

10/9/95

Duration

12 weeks

Subtasks

2.1 Examine CATTask Database for selected CIS's.
2.2 Gather SME input on CIS's.

Dependencies

Predecessors

Task 1: Identify CIS's to examine.

Successors

Task 5: Construct detailed design for incorporation of CIS's into ModSAF.

2.1 Examine CATTask Database for selected CIS's.

Task Description

The CATTask Database contains information about each of the CIS's implemented in the CCTT program. IST will examine the information in the CATTask Database about each of the CIS's selected in Task 1.

Start Date

10/9/95

Duration

12 weeks

Dependencies

None

2.2 Gather Subject Matter Expert (SME) input on CIS's.

Task Description

Gather appropriate Subject Matter Expert information about the CIS's selected in Task 1, as appropriate. SME's include Dan Mullally, USMC Maj. (Ret.) of IST and SME's identified by STRICOM and TRAC.

Start Date

10/9/95

Duration

12 weeks

Dependencies

None

Task 3: Study CCTT implementation of CIS's.

Task Description

Examine the process developed in the CCTT program for developing and implementing CIS's. The goal will be to understand the behavioral representation techniques used in the CCTT SAF, including the relationship between behavior generation and the terrain representation, and the behavior generation infrastructure.

Start Date

10/9/95

Duration

10 weeks

Subtasks

3.1 Read available papers on CIS development process.
3.2 Examine process of implementing CIS's in CCTT SAF.
3.3 Examine source code of selected CIS's from CCTT SAF.
3.4 Discussions with CCTT SAF development team (focused on questions, lead by IST).
3.5 Prepare deliverable: Presentation.

Dependencies

Predecessors

Task 1: Identify CIS's to examine.

Successors

Task 5: Construct detailed design for incorporation of CIS's into ModSAF.

3.1 Read documentation on CIS development process.

Task Description

Identify and read documentation on the CIS development process. Sources of papers include STRICOM (for technical report deliverables from the CCTT program) and published technical papers (e.g. Computer Generated Forces and Behavioral Representation Conference Proceedings).

Start Date

10/9/95

Duration

10 weeks

Dependencies

Predecessors

None

Successors

3.5 Prepare deliverable: Presentation.

3.2 Examine process of implementing CIS's in CCTT SAF.

Task Description

Study the process of developing CIS's in the CCTT SAF. The CCTT program has developed a formal process for describing military tactics and implementing them in a Computer Generated Forces system. An understanding of this process is essential for understanding the behavior representation techniques of the CCTT SAF.

Start Date

10/9/95

Duration

10 weeks

Dependencies

Predecessors

None

Successors

3.5 Prepare deliverable: Presentation.

3.3 Identify bindings between CCTT SAF architecture and behavioral representation techniques

Task Description

Examine the source code and documentation of the CIS's selected in Task 1. The goals will be to understand the implementation of CIS's in the CCTT SAF and to identify any bindings between the behavioral representation techniques and the CCTT SAF architecture. These bindings must be resolved in the translation process.

Start Date

10/9/95

Duration

10 weeks

Dependencies

Predecessors

None

Successors

3.5 Prepare deliverable: Presentation.

3.4 Discussions with CCTT SAF development team (focused on questions, lead by IST).

Task Description

Focused discussions about specific questions of process, content, or implementation of the CIS's in the CCTT SAF with the SAF development team. Such discussions will occur during and after IST's in-depth study of the available CCTT SAF documentation and source code.

Start Date

10/9/95

Duration

10 weeks

Dependencies

Predecessors

None

Successors

3.5 Prepare deliverable: Presentation.

3.5 Prepare deliverable: Presentation.

Task Description

Prepare a brief presentation summarizing the results of the research in 3.1 through 3.4.

Start Date

12/18/95

Duration

3 weeks

Dependencies

Predecessors

3.1 Read available papers on CIS development process.
3.2 Examine process of implementing CIS's in CCTT SAF.
3.3 Examine source code of selected CIS's from CCTT SAF.
3.4 Discussions with CCTT SAF development team (focused on questions, lead by IST).

Successors

None

Task 4: Study ModSAF's behavior control mechanisms.

Task Description

Examine the representation of behaviors in ModSAF. The goal will be to understand the behavioral

representation techniques used in ModSAF.

Start Date

10/9/95

Duration

10 weeks

Subtasks

4.1 Read available papers on Augmented Asynchronous Finite State Machines used in ModSAF.

4.2 Identify and read available papers on behavior implementations in ModSAF.

4.3 Examine source code of selected behaviors in ModSAF.

4.4 Prepare deliverable: Presentation.

Dependencies

Predecessors

Task 1: Identify CIS's to examine.

Successors

Task 5: Construct detailed design for incorporation of CIS's into ModSAF.

4.1 Read documentation on Augmented Asynchronous Finite State Machines used in ModSAF.

Task Description

Identify and read documentation on Augmented Asynchronous Finite State Machines (AAFSMs) in ModSAF. AAFSMs are the behavioral specification mechanisms used in ModSAF. Sources of documentation include ModSAF internal documentation and published technical papers (e.g. Computer Generated Forces and Behavioral Representation Conference Proceedings).

Start Date

10/9/95

Duration

10 weeks

Dependencies

Predecessors

None

Successors

4.4 Prepare deliverable: Presentation.

4.2 Identify and read documentation on behavior implementations in ModSAF.

Task Description

Identify and read documentation on implementations of behaviors in ModSAF. The goal is to understand how the AAFSM technique is applied in practice to specific behaviors. Possible sources of papers include ModSAF internal documentation and published technical papers (e.g. Computer Generated Forces and Behavioral Representation Conference Proceedings).

Start Date

10/9/95

Duration

10 weeks

Dependencies

Predecessors

None

Successors

4.4 Prepare deliverable: Presentation.

4.3 Examine source code of selected behaviors in ModSAF.

Task Description

Examine the ModSAF source code and documentation for the CIS's selected in Task 1 that have equivalent

ModSAF representations.

Start Date

10/9/95

Duration

10 weeks

Dependencies

Predecessors

None

Successors

4.4 Prepare deliverable: Presentation.

4.4 Prepare deliverable: Presentation.

Task Description

Prepare a brief presentation summarizing the research from 4.1 through 4.3.

Start Date

12/18/95

Duration

3 weeks

Dependencies

Predecessors

4.1 Read available papers on Augmented Asynchronous Finite State Machines.

4.2 Identify and read available papers on behavior implementations in ModSAF.

4.3 Examine source code of selected behaviors in ModSAF.

Successors

None

Task 5: Construct design for incorporation of CIS's into ModSAF.

Task Description

Based on the research in Tasks 2, 3, and 4, design an approach for implementing the CIS's selected in Task 1 in ModSAF. The implementation approach may consist of a mixture of automated tools, manned translation, and recoding of algorithms. The goal is an approach that can be applied to other CIS's in addition to those selected in Task 1; the selected CIS's will serve as an illustration and check of the design.

Start Date

10/9/95

Duration

10 weeks

Subtasks

5.1 Produce plan based on CCTT CIS program.

5.2 Produce rough schedule for implementing selected CIS's in ModSAF.

5.3 Prepare deliverables: Presentation and Report.

Dependencies

Predecessors

Task 2: Understand CIS's.

Task 3: Study CCTT implementation of CIS's.

Task 4: Study ModSAF's behavior control mechanisms.

Successors

None

5.1 Produce plan based on CCTT CIS program.

Task Description

Develop a plan that extends the CCTT SAF development process to allow conversion of SAF CISs to ModSAF

behaviors. This plan may involve taking intermediate CIS representations as the starting point for conversion rather than the final SAF CIS implementation. Automated conversion tools will be identified or specified.

Start Date

1/8/96

Duration

8 weeks

Dependencies

Predecessors

None

Successors

5.3 Prepare deliverables: Presentation and Report.

5.2 Produce rough schedule for implementing selected CIS's in ModSAF.

Task Description

Estimate the effort required to convert each Task 1 CIS to ModSAF. Produce a schedule from these estimates and estimates of project management and supervision.

Start Date

1/8/96

Duration

8 weeks

Dependencies

Predecessors

None

Successors

5.3 Prepare deliverables: Presentation and Report.

5.3 Prepare deliverables: Presentation and Report.

Task Description

Present the results of Task 5. Prepare a Project Report describing the methods and results of this research.

Start Date

3/4/96

Duration

4 weeks

Dependencies

Predecessors

5.1 Produce plan based on CCTT CIS program.

5.2 Produce rough schedule for implementing selected CIS's in ModSAF.

Successors

None

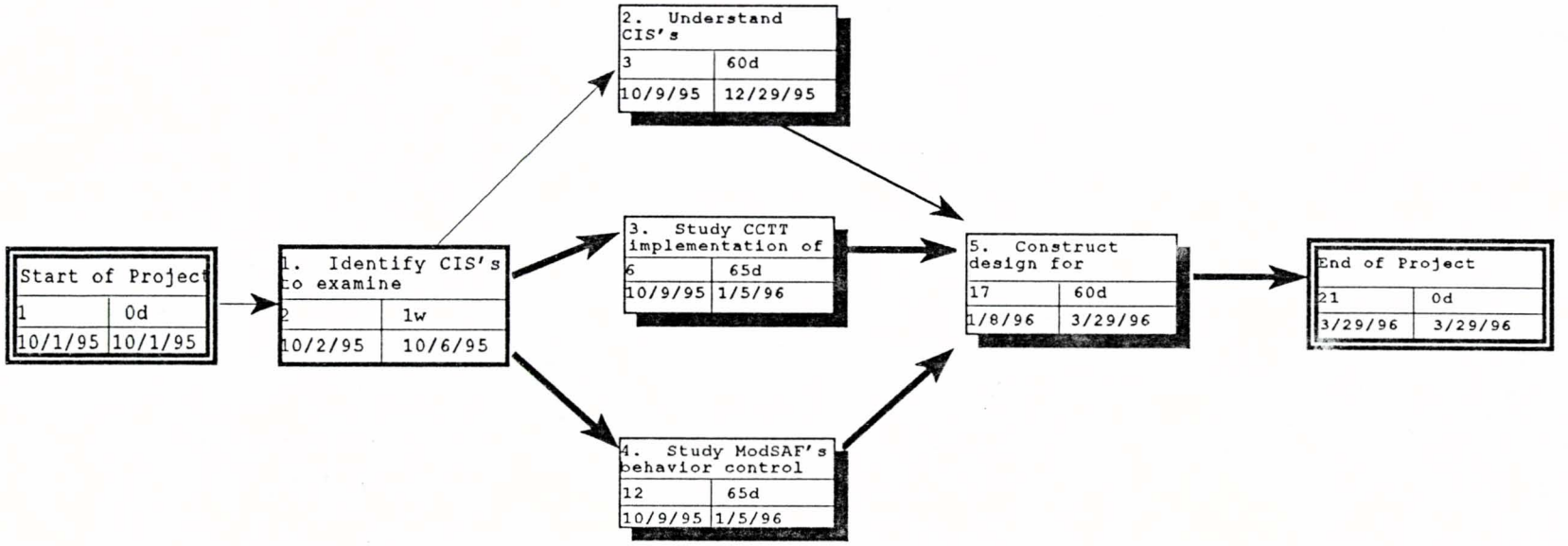
Deliverables

Presentation combining 3.5, 4.4, and 5.3 results.

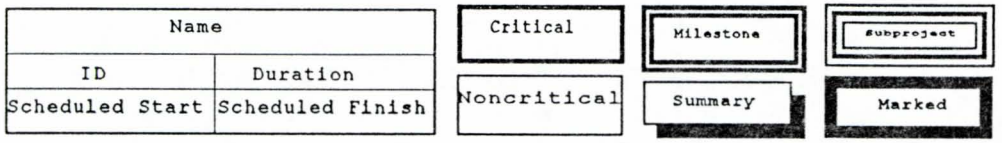
Final report (5.3)

Gantt and PERT Charts

		4th Quarter			1st Quarter			2nd Quarter			3rd Quarter			4th Quarter		
Name	Duration	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Start of Project	0d															
1. Identify CIS's to examine	1w															
2. Understand CIS's	**60d**															
2.1 Examine CATTask Database for selected CIS's	12w															
2.2 Gather SME input on CIS's	12w															
3. Study CCTT implementation of CIS's	**65d**															
3.1 Read documentation on CIS development process	10w															
3.2 Examine process of implementing CIS's in CCTT SAF	10w															
3.3 Identify bindings between CCTT SAF architecture and	10w															
3.4 Discussions with CCTT SAF development team	10w															
3.5 Prepare deliverable: Presentation	3w															
4. Study ModSAF's behavior control mechanisms	**65d**															
4.1 Read documentation on AAFSMs used in ModSAF	10w															
4.2 Read documentation on behavior implementations in ModSAF	10w															
4.3 Examine source code of selected behaviors in ModSAF	10w															
4.4 Prepare deliverable: Presentation	3w															
5. Construct design for incorporation of CIS's into ModSAF	**60d**															
5.1 Produce plan based on CCTT CIS program	8w															
5.2 Produce rough schedule for implementing selected CIS's	8w															
5.3 Prepare deliverables: Presentation and Report	4w															
End of Project	0d															



Project: CIS in ModSAF
Date: 10/2/95

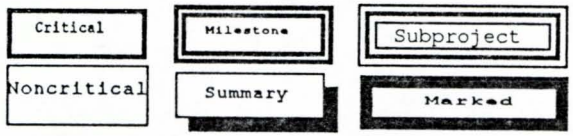


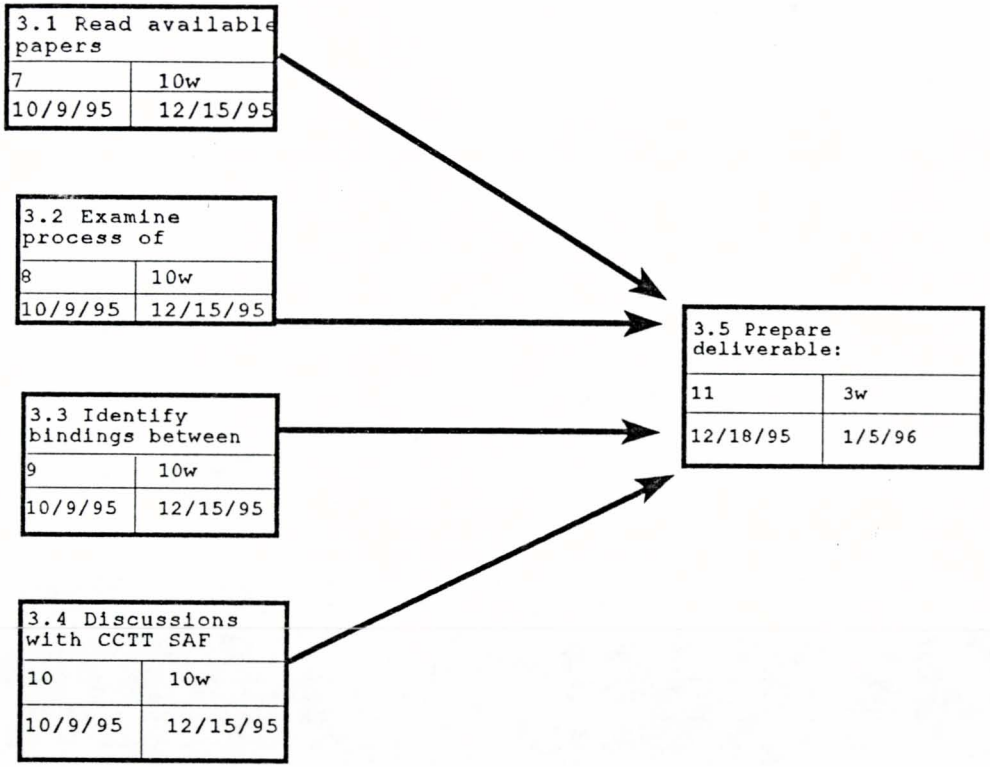
2.1 Examine CATTask Database	
4	12w
10/9/95	12/29/95

2.2 Gather SME input on CIS's	
5	12w
10/9/95	12/29/95

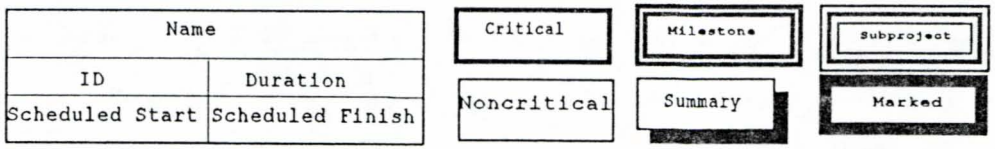
Project: CIS in ModSAF
Date: 10/2/95

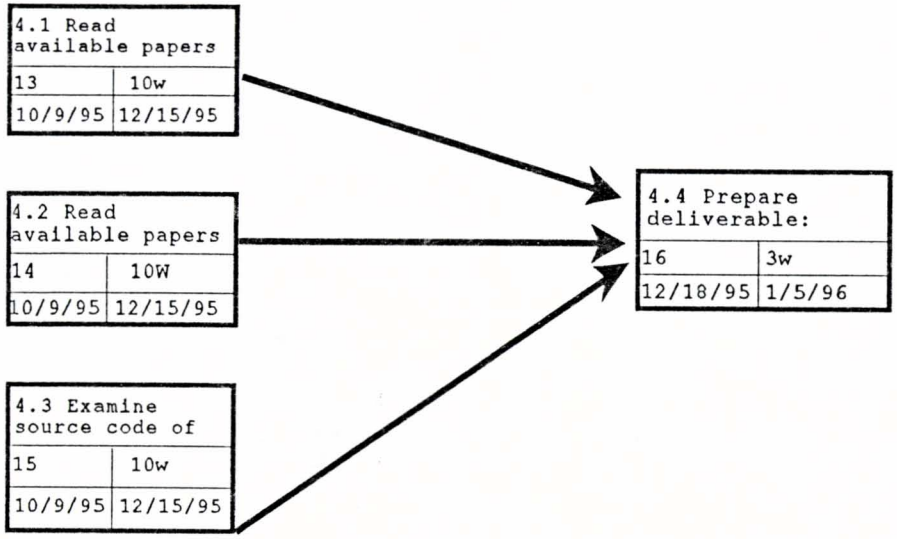
Name	
ID	Duration
Scheduled Start	Scheduled Finish



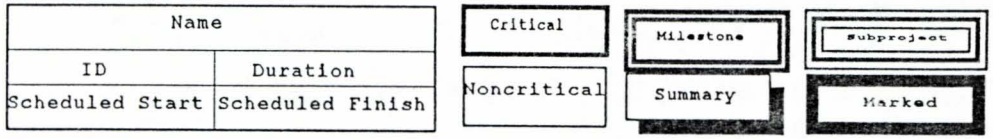


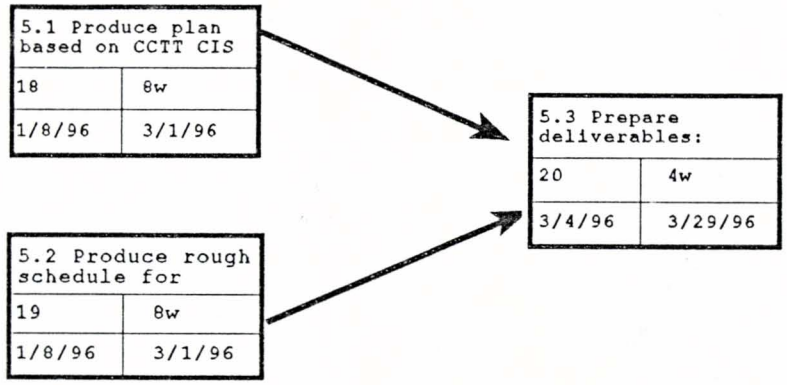
Project: CIS in ModSAF
Date: 10/2/95



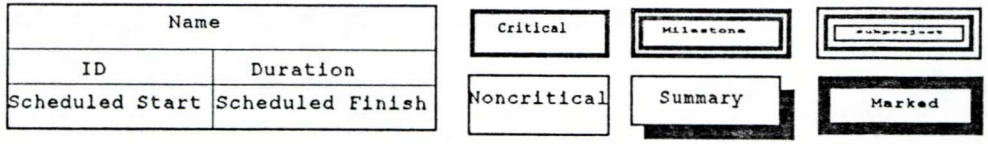


Project: CIS in ModSAF
Date: 10/2/95





Project: CIS in ModSAF
Date: 10/2/95



Goal 2: Develop the Class Structure and Instance Objects for the Eagle Frame Model

Description

This task requires changes to Eagle code which will touch every file that is a part of the Eagle system, some of them requiring major changes, others only minor changes.

The initial loading of the Eagle model is performed by a "build" process, in which the files containing specification data for all of Eagle's structures are loaded. The Eagle "build" process must be studied and documented, and the code which performs this function must be isolated and converted. Similarly, the preprocessor goes through a build process, loading many of the same files as the model and adding files needed to allow the user to generate scenarios.

Existing data must be loaded into the knowledge bases once their structure has been built. This data is found in class data files, scenario input files, and history output files. Each of these types of files has a different format, which must be read and converted to the appropriate Eagle format. User (class) data files are *.data (lowercase) files. Scenario input files and history output files, are *.DATA (uppercase) files. A separate translation utility may be written to convert scenarios now being used by Eagle to a format more suitable for the revised Eagle. This will allow all scenarios now being used to continue being run after completion of this revision.

As the data is read from a data file, the object structure (frame) for that class provides a template for storing the data. Data representing attributes of the object must be collected and stored as slot values by using the CLOS DEFCLASS. Data representing method names for the object must trigger the definition of those methods by means of the CLOS DEFMETHOD.

In an object-oriented environment like CLOS, methods behave like functions. The principal difference between a function and a method is that a method operates only on the object or class of objects for which it is defined. For example, a method FOO for object BAR may or may not be a completely different method from the method FOO for the object BAZ. Determining which method belongs with which object is determined by the definition of a generic function, which acts as a template for defining methods, and a set of methods specialized through their arguments to particular classes of objects.

Methods may be inherited by subclasses and instances, just as attributes are. Some work will have to be done to understand the hierarchy of Eagle objects. Understanding this structure is necessary to properly implement method inheritance.

KEE uses a primitive called UNITMSG to implement message-passing between objects. It provides an identifier for the object (which may be the name of the object or a pointer to the structure representing the object), the name of the method to be called, and the arguments for that method. CLOS allows methods to be called directly, if applicable to the object, instead of having a message-passing scheme. This is used by Eagle to build itself; telling an object where to locate slots containing file names or method names which in turn cause the loading of files and the construction of the knowledge bases.

Significant Sub-tasks

The building of the Eagle model consists of building the structure to store the data and building the operations that will manipulate that structure. The significant sub-tasks are:

Analyze and document the Eagle "build" process.

Convert object attribute data to slot values for use by DEFCLASS.

Define CLOS methods using DEFMETHOD for data representing method names.

Implement the CLOS methods needed to support the build process.

Test the new CLOS method.

Task Schedule

Estimated duration of this task is 100 working days. Two people will work on this task, each working half-time. This task is scheduled to begin on 2 October 1995 and should be completed by 1 March 1995. The start and end dates for this task include 4 days (two for travel and two for technical) for an Eagle SAR interim presentation.