AFRL-IF-RS-TR-2003-292
Final Technical Report
December 2003

I-WFM FOR REAL-TIME C4I PROCESS MANAGEMENT

Intelligent Systems Technology, Inc.

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. AO F079

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

STINFO FINAL REPORT

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2003-292 has been reviewed and is approved for publication

APPROVED: /s/
RICHARD METZGER
Project Engineer

FOR THE DIRECTOR: /s/
JAMES W. CUSACK, Chief
Information Systems Division
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

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

1. AGENCY USE ONLY (Leave blank)	**2. REPORT DATE** Dec 03	**3. REPORT TYPE AND DATES COVERED** Final Jun 97 – Jun 98

4. TITLE AND SUBTITLE

I-WFM FOR REAL-TIME C4I PROCESS MANAGEMENT

5. FUNDING NUMBERS

C - F30602-97-C-0293
PE - 62301E
PR - TCOM
TA - T2
WU - 02

6. AUTHOR(S)

Azad Madni, Weiwen Lin, and Carla Madni

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Intelligent Systems Technology, Inc.
2800 28th Street, Suite 306
Santa Monica, CA 90405

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

DARPA
3701 North Fairfax Drive
Arlington, VA 22203-1714

AFRL/IFTB
525 Brooks Rd
Rome, NY 13441

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

AFRL-IF-RS-TR-2003-292

11. SUPPLEMENTARY NOTES

AFRL Project Engineer: Richard C. Metzger, IFTB, 315-330-7652, metzgerr@rl.af.mil

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

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 Words)*

Ongoing advances in globally netted C4ISR systems have ushered in a new era in which information technologies (IT) can be exploited to create unprecedented capabilities that will allow dispersed military forces to operate synchronously with great agility and superior situation awareness. The realization of such capabilities will inevitably be accompanied with changes in command processes, organizational structures and doctrine. In particular, IT holds great promise in the rapid design, re-engineering, automation aiding, and management of command processes. The focus of this three-year R&D effort is to develop, demonstrate, and transition components of Intelligent Process Management (IPM) technology in support of C4ISR. The R&D performed in Phase I of this three-year effort is the subject of this report. The overall goal of this three-year effort was to develop, demonstrate, and transition intelligent process management technology. The objectives of Phase I of this effort were to: a) develop a conceptual framework for creating the major building blocks of intelligent process management; b) create a command process (re)design and analysis capability; c) model representative command processes for a target application; d) create a command process monitoring and visualization capability; and e) demonstrate the evolving IPM capabilities at DARPA program workshops and Principal Investigator meetings.

14. SUBJECT TERMS
Process Design, Process Analysis, Intelligent Process Management, Adaptive Workflow, Collaboration Support, Intel Agents, Java, COBRA

15. NUMBER OF PAGES
61

16. PRICE CODE

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

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

ACKNOWLEDGEMENT

We are grateful to Mr. B. Naiem and Mr. V. Carter of ISTI for their constructive inputs to this document and to Mr. Jeffrey Koehler of Northrop Grumman Corporation who provided the subject matter expertise for creating application scenario vignettes.

EXECUTIVE SUMMARY

Overview

Ongoing advances in globally netted C^4ISR systems have ushered in a new era in which information technology (IT) can be exploited to create unprecedented capabilities that will allow distributed military forces to operate synchronously with great agility and superior situation awareness. Inevitably, the realization of such capabilities will be accompanied with changes in command processes, organizational structures, and doctrine. In particular, IT holds great promise in the rapid design, re-engineering, automation, aiding, and management of command processes. The focus of this three-year R&D effort is to develop, demonstrate, and transition components of Intelligent Process Management (IPM) technology in support of C^4ISR. The R&D performed in the first year of this three-year effort is the subject of this report.

Phase I Objectives

The overall goal of this three-year effort is to create and transition intelligent process management technology. The objectives of the first year were to: a) develop a conceptual framework comprising the major building blocks for intelligent process management; b) create a command process (re)design and analysis prototype; c) model representative command processes for a target application; d) create a command process monitoring and visualization capability; and e) demonstrate the evolving IPM capabilities at DARPA Information Systems Office (ISO) workshops and Principal Investigator meetings.

Phase I Accomplishments

In Phase I, we accomplished the following:

- Created a framework for investigating interleaved planning and execution.
- Created a core IPM ontology that integrates key concepts from enterprise process modeling and enterprise process management.
- Conducted a review of commercial-off-the-shelf workflow products to assess their suitability for serving as a backbone for IPM.
- Created and demonstrated an IPM prototype with the following capabilities:
    - Command process modeling and analysis.
    - Command process import from SMEs at remote locations.
    - Process asset library consisting of component command processes that are persistently stored and that can be accessed online, tailored, and reused.
    - Command process monitoring with facilities for "roll up" of status from activity level to process level, and "drill down" of process status to lower levels to identify problematic activities.
- Created an event taxonomy related to the execution of command processes with a view to defining the various types of adaptations required within an IPM.

Technical Issues

The technical issues we addressed during the first year included:

- Creating a conceptual framework for interleaved planning and execution with a view to introducing component technologies (e.g., collaboration support, automated planners, adaptive workflow, decision support, agents and sentinels) within the framework.
- Defining intelligent process management for C^4ISR.
- Identification of C^4ISR process representation requirements.
- Assessing the viability of employing COTS workflow products as a backbone for IPM.
- Acquiring process knowledge electronically from remote Subject Matter Experts (SMEs).
- Creation of an event taxonomy to develop event classification and handling (i.e., adaptation) methods.
- Devising representative mini-scenarios to illustrate various types of adaptations.

Approach

Our overall approach was based on a concurrent strategy for: a) eliciting or modeling representative command processes with a view to identifying the required concepts/semantics for IPM; b) developing IPM requirements for C^4ISR; c) reviewing COTS workflow products from the viewpoint of their suitability for IPM; and d) identifying insertion opportunities for the evolving IPM prototype. These activities were accompanied by rapid prototyping activities in which the IPM prototype consisting of command process design and analysis capabilities were constructed on top of **`ProcessEdge`**™, the company's forthcoming commercial product.

Contributions

The scientific and technical contributions of the first year are:

a) An *innovative IPM system concept* for C^4ISR that combines process design, collaboration support, adaptive workflow, and decision support capabilities for C^4ISR process management. This capability is key to handling automated, interactive, and collaborative processes during replanning and execution.

b) A *core ontology* for IPM that integrates core concepts from enterprise process modeling and enterprise process management. This ontology provides the underpinnings of integrated, interleaved planning and execution.

c) An *event taxonomy* that is key to process dynamism and process adaptation in IPM.

d) A "100%-Java"-based *command process re-engineering capability*. This capability is key to platform-independence and low total cost-of-ownership.

e) A *prototype process library* for persistently storing, retrieving, and customizing component command processes. This capability is key to process reuse and substitution of partial workflows in response to certain types of events.

f) An *email-based process import* facility. Process definitions are encoded in Excel spreadsheets by remote users and submitted through electronic mail. This capability is key to “in-location” knowledge acquisition from remotely located SMEs.

Payoffs

The evolving IPM tool suite provides the foundation technology for command process re-engineering, a key requirement on ISO several programs such as ALP, JFACC, AIM, and a fundamental requirement of the AITS JPO’s ACOA program. The tool suite also satisfies the requirements for command process re-engineering in support of Integrated Battle Force Management (IBFM), an ISO program in its formative stages. The “in-location” knowledge engineering facility of the IPM prototype will save travel costs and circumvent scheduling problems between the knowledge engineer and the SMEs. The capabilities and benefits of the IPM prototype are shown in the Table E-1 below.

Table E-1. Capabilities and Benefits of the IPM Prototype

- Acquire process models (knowledge) in electronic form from geographically distributed subject matter experts
  - *Benefit*: Eliminates rekeying of process models, saves travel costs
- Verify completeness and consistency of the models and correct deficiencies
  - *Benefit*: Detect and correct errors at “build” time rather than “run” time
- Analyze processes in terms of cycle time, resource requirements, and costs
  - *Benefit*: Pre-analyzed processes support comparative analysis when adopting a process
- Persistently store analyzed processes in process library
  - *Benefit*: Support “plug-and-play” of component processes
- Select, tailor, and reuse processes from process library
  - *Benefit*: Dramatic reduction in process design cycle time and cost
- Monitor and visualize processes at multiple levels of abstraction
  - *Benefit*: Rapidly verify progress of executing process by “rolling up” status of individual analysis

    Rapidly locate bottlenecks/impediments by “drilling down” status of high level process if it is “blocked”

Target Insertion Opportunities

Target programs that offer the most immediate insertion opportunities are ACOA and JFACC. ACOA is a particularly appealing target for IPM technology insertion due to the fact that the ACOA is target for insertion in GCCS.

1. INTRODUCTION

1.1 Project Overview

The goal of this incrementally funded three-year R&D project is to develop, demonstrate, and transition an Intelligent Process Management methodology and software prototype for: a) command process design and re-engineering; and b) command process execution monitoring, control, aiding, and adaptation in dynamic C^4ISR environments. The specific objectives of the first year were to:

- Prototype process design, "quick change" and analysis capabilities.
- Create the IPM system concept.
- Demonstrate the evolving IPM prototype at DARPA Information Systems Office (ISO) workshops and Principal Investigator Meetings.

1.2 Phase I Accomplishments

The major accomplishment of this year was the creation and demonstration of an IPM prototype with capabilities for: a) process modeling and analysis; b) process import from Subject Matter Experts (SMEs) at remote locations; c) process visualization from multiple complementary perspectives and at multiple levels of abstraction; d) process monitoring with facilities for "roll up" of status from activity level to process level and "drill down" of process status to lower levels to identify problematic activities; and e) persistent storage of component processes within a process asset library that supports online access, tailoring, and reuse. In addition, we identified transition programs (e.g., ACOA, JFACC) and transition sites (i.e., USCINCPAC, Hawaii; Air and Space C2 Agency, Hampton, Virginia). At the conclusion of the first year we were redirected again by DARPA and AFRL to focus our effort for transition to the ACOA project.

1.3 Report Roadmap

This report is divided into five sections. Section 2 presents the C^4ISR Process Management challenge. Section 3 presents the Process Management problem, key requirements, and the need for Intelligent Process Management. Section 4 presents the IPM prototype in terms of the overall system concept, underlying ontology, development approach, capabilities, and features. Section 5 concludes with a discussion of future direction and insertion strategy.

2. PROCESS MANAGEMENT AND C^{4}ISR

2.1 The Changing Warfighting Environment

Advances in globally netted C^{4}ISR systems offer the unprecedented opportunity to create an IT-enabled integrated warfighting capability that will allow dispersed military forces to operate synchronously with superior agility and situation awareness than ever before possible. Along with this opportunity comes the challenge of managing coordinated command processes in dynamic C^{4}ISR environments. An equally important aspect of the problem is the re-engineering of command processes to exploit IT in ways that make the processes more agile and responsive in support of integrated planning and execution.

2.2 Coordinated (Re)Planning and Execution within Dynamic C^{4}ISR Environments

C^{4}ISR processes are characterized by interleaved (re)planning and execution in dynamic environments. Gaining visibility into and maintaining control of such processes is beyond the current scope and capabilities of traditional workflow technologies. C^{4}ISR processes, by their very nature, are *distributed** in time and space, and therefore require *coordinated* planning and execution. Figure 1 presents a conceptual view of the interleaved nature of planning and execution processes. In particular, it shows that automated (re)planning, mixed-initiative replanning, and collaborative replanning are all part of the C^{4}ISR planning and execution processes.

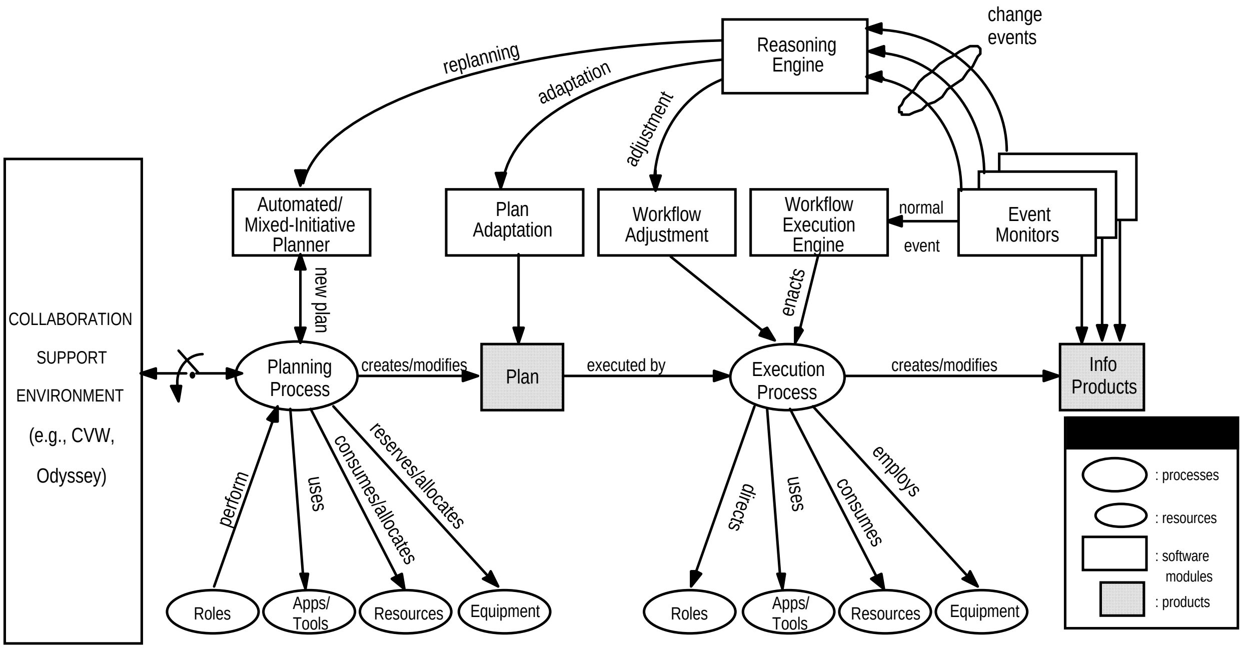


Figure 1. Coordinated Planning and Execution within C^{4}ISR Processes

**Distributed* implies that the participants in the planning effort, their processes, planning resources, and work products are all geographically dispersed.

As shown in this figure, the planning process: a) consists of a sequence of executable activities; b) is performed by roles; c) uses tools; d) consumes/allocates resources; e) reserves/allocates equipment; and f) creates/modifies one or more plans. Thus, a plan is the product of the planning process. The planning process is a combination of automated, mixed-interactive, and collaborative processes. The plan is also an executable process. The plan execution process: a) consists of a sequence of partially ordered activities; b) directs role(s); c) uses tools; d) consumes resources during execution; e) employs equipment during execution; while f) creating/modifying information products. During plan execution, the various resources and information products are monitored by event monitors (to detect normal completion events) and different types of sentinels (to detect deviations from plan or assumptions, and to generate appropriate alerts). The changes detected are evaluated by a reasoning engine that determines the appropriate response: (a) automated workflow adjustment; (b) automated or interactive plan adaptation; (c) automated, mixed-initiative or collaborative replanning.

Working from left to right in Figure 1, the planning process is supported through user interactions and dialogues supported by a customer-specified, open collaboration support environment (e.g., Mitre's CVW, SPAWAR's Odyssey) and automated/mixed-initiative planners that assure proper progress of the planning process. The product of the planning process is the initial plan. The plan is executed by a workflow execution engine. The execution results in the creation/modification of information products and the update of various C^{4}ISR "folders." The execution is monitored by event monitors/sentinels (e.g., temporal sentinels). Event monitors confirm completion events and detect expected events. Sentinels, a special class of agents, detect plan deviations or unusual events and trends, and then trigger a reasoning engine to determine the appropriate response, i.e., automatic adjustment, interactive plan repair or collaborative replanning. It is important to recognize that the need for seamless transition back and forth from automated workflow execution to spontaneous computer-supported human collaboration is beyond the capabilities of existing workflow systems and automated planners. This recognition, in part, provided the impetus for this R&D project. However, introducing process management without first re-engineering command processes to explicit information technologies can result in automation of suboptimal processes.

2.3 The Command Process Re-engineering Imperative

Command and control functions are performed through an arrangement of personnel, equipment, communications, facilities, and procedures employed by a commander in planning, directing, coordinating, and controlling forces and operations in the accomplishment of the mission. *Command Process Re-engineering* [1] is concerned with identifying specific functions and relationships that can be modified or created to:

- Produce greater responsiveness in planning and execution.
- Rapidly generate *executable* courses of action.
- Generate military campaign plans which lead to pre-defined end states.
- Support coordinated force employment in the execution of military campaigns.
- Minimize the potential for mutual interference and fratricide in friendly forces.
- Provide planning and execution "products" on demand.

There are several reasons why current U.S. Force structures cannot be maintained within forecasted budget levels. First, a shift to smaller, agile, and more capable forces is required to prevent the erosion of U.S. capabilities. This shift, in turn, requires re-engineering of command processes. Second, the full impact of globally netted C^4ISR cannot be felt, until IT has been fully exploited to "parallelize" processes where possible, eliminate extraneous iterations, eliminate non-value-added steps, and automate processes when feasible and advantageous. Third, the early retirement of several forward thinkers continues to erode our process knowledge capital. The bottom line of command process re-engineering is increased capability at less cost. However, to be successful, Command Process Re-engineering requires: a) organizational and doctrinal changes; b) the exploitation of distributed, ubiquitous information technology; and c) a "system of systems" mindset.

It is important to realize that the transformation or re-engineering of organizational processes can be undertaken from a reactive or proactive stance. Examples of reactive circumstances are: threat of imminent bankruptcy or cash flow problems; a recent debacle; or the appearance of a formidable competitor. Examples of proactive transformation or re-engineering are: a strong, visionary, long-term leadership committed to change; the exploitation of an existing slack that permits experimentation without threatening existing organizations; organizational will and motivation resulting from a forward looking management team with a belief in a "learning organization" culture. *In either case, command process re-engineering must precede process automation for process management to be truly effective.*

2.4 Command Process/Workflow Management

Command process/workflow management is the "run-time" component that exploits the improvements achieved during command process re-engineering (the "build time" component). Process/workflow management also contributes to process re-engineering in that data collection from realworld execution can be used to refine, modify or enhance the process models.

The following "mini-scenarios" illustrate the need for process/workflow management in C^4ISR. Each "mini-scenario" illuminates a unique problem that can be tackled by workflow management.

1. *Problem: Tracking Distributed Resources. Commander gives direction to staff and subordinates but then has no way of determining where the different staff, operations and logistics personnel are in the process.*

   Solution: Workflow management can provide realtime tracking of the various roles, individuals, and material/equipment resources, and an appropriately constructed display can provide commanders with the required visualization of the assets/human resources being monitored/tracked. Since human resources perform their duties at various levels in the command hierarchy, the workflow manager should be capable of representing hierarchical workflow while its tracking function should be capable of tracking hierarchical workflow.

2. *Problem Personnel Substitution. For various reasons (e.g., reassignment of certain personnel, change of plans, casualties) certain personnel engaged in executing a plan need to be substituted with others. The new personnel have no way of "getting situated" quickly. Specifically, they need to come up to speed on the context, and the partial/total information products created and left behind by their predecessors.*

   Solution: The workflow manager's visualization interface provides color-coded views of tasks/activities that have been completed, are ongoing, and are pending from both the precedence and decomposition perspectives. In addition, the information management component of the workflow manager manages both content and context and makes these available to the new personnel in terms of appropriate "folders."

3. *Problem Mixed-initiative Execution Tracking. Certain tasks/activities require mixed-initiative plan execution. The workflow manager needs to track the mixed-initiative plan execution process.*

   Solution: The workflow manager "tracks" the handoffs back and forth between the human and automated elements of the system as well as the data exchanged in each handoff, and the "folders" updated with each handoff.

4. *Visibility and Control of Dynamic Workflow Problem: The occurrence of an event at an expected time or unexpected place can disrupt plan execution.*

   Solution: The workflow manager should possess adaptive decision logic to dynamically decide how to respond to a change event during execution. Specifically, the workflow capability should be capable of deciding whether workflow adjustment can be done automatically in response to the change event or whether human intervention is required. If human intervention is called for, then the system should be able to determine whether workflow adaptation can be done by a decisionmaker interacting with the system or whether collaboration among multiple stakeholders is required to respond adequately.

2.5 COTS Workflow Products are Inadequate for C^{4}ISR Process Management

The C^{4}ISR process management challenge is to monitor, control, and adapt processes in the face of continual, relentless change in dynamic environments (Figure 1). In general, dealing with change requires a combination of automated, interactive and collaborative methods. For the latter two, some degree of decision and execution support may also be necessary. In addition, given the heterogeneous platforms in the military, platform-independent approaches are clearly desirable, and, in fact, necessary. These requirements are beyond the capabilities of commercial workflow products [2], [3].

The requirements for process management cut across multiple technology areas and multiple disciplines. Some of the most salient requirements are:

- Wide-area (preferably Web-based) management of multi-component commands and multiple echelons.
- Management of mixed initiative planning and execution processes.
- “On-the-fly” adaptation to changes in the external environment and internal state variables.
- Support for automated, mixed-initiative, and collaborative decisionmaking processes.
- Hierarchical cross-functional workflow modeling and management.
- Interoperability between heterogeneous workflow engines.

Upon examining these requirements, it becomes apparent that they are largely beyond the capabilities of process/workflow products on the market today (see Appendix A for a detailed review of workflow products). The following paragraphs elaborate on this point.

The first requirement is for a wide-area (preferably *Web-based*) *workflow* capability. This can be achieved by managing workflow over the Internet/WWW to increase our geographic reach, and to assure support for inter-organizational workflows, assuming there are multiple compatible workflow servers at each participating site. *Web-based workflow* employs the Internet/WWW as its information infrastructure so that distributed workflow execution and information routing occur over the Web. *Web-enabled workflow*, on the other hand, is what is available in COTS products from vendors such as Action Technologies and InConcert Inc. In Web-enabled workflow systems, existing workflow products are extended to enable: (a) the activation and browsing of Web-based contents or sites; or (b) input or output of workflow model specifications using Web browsers and forms. Thus, in these systems the Web is used as another online medium for conveying or displaying workflow-related information.

The second requirement is for *management of mixed-initiative* planning and execution. To satisfy this requirement we need a mixed-initiative execution engine capable of supporting automated as well as interactive decision-making modes. This capability is not available in any existing commercial-off-the-shelf product or research prototype.

The third requirement is for *intelligent "on-the-fly" adaptation*. This means that: (1) changes can be introduced in executing instances without having to go into the "build" or compile mode; (2) response to changes is more sophisticated than known exception handling; and (3) there is embedded intelligence in the system to determine the scope of the change and then implement the response to the change (i.e., execution parameter adjustment, partial workflow substitution, or collaborative replanning/response generation). Existing workflow products offer only the first type of adaptation. For example, InConcert offers limited adaptation in the form of: dynamic assignment of tasks to users and user pools outside the purview of the process; attachment of data and documents not previously defined in the process; specifying routing sequences; free routing; suspending a process, resuming a suspended process, or completing a process by skipping steps; and modifying the attributes of a process. (See Appendix A).

The fourth requirement is for *wide-area (preferably Web-based) management* of multiple component commands and multiple echelons. This is a two-fold requirement. The first requires management of workflow over a wide-area network such as the Internet/WWW using middleware based on published standards (e.g., CORBA or email). COTS products on the market today provide web-enabled management for only one level (i.e., echelon) through HTTP servers. The second requires *hierarchical, cross-functional workflow* modeling and management. This capability is required to span multiple echelons and multiple component commands. COTS products and research prototypes do not offer this capability.

The fifth requirement is *interoperability between heterogeneous workflow engines*. This capability is necessary when different component commands employ different workflow systems that need to interoperate in a joint mission. This situation could also exist within a particular component command. This problem, in part, is being attacked by the Process Interchange Format (PIF) [4],NISTS' Process Specification Language [5] for process information interchange, and by KQML [6] for communicating between heterogeneous automated planners. ISTI's **`ProcessScript`** [7] is also suitable for this purpose.

Table 1 summarizes the limitations of COTS Workflow Systems.

Table 1. Limitations of COTS Workflow Systems

- Do not capture human decision heuristics.
- Do not promote understanding and communication among users.
- Do not utilize a comprehensive ontology for meaningful analysis.
- Do not manage assumptions, information, decisions.
- Do not provide decision support.
- Do not do self-appraisal of execution performance to trigger appropriate tradeoffs (e.g., timeliness versus completeness or completeness versus cost) that could lead to dynamic replanning.
- Do not offer adaptive control strategies that exploit conventional workflow where possible and employ more sophisticated control to respond to unexpected events.
- Do not provide intelligent summarization or filtering of information.

3. INTELLIGENT PROCESS MANAGEMENT

3.1 The C^4ISR Process Management Problem

The C^4ISR environment of the modern era is characterized by time-compressed decisionmaking and replanning, and distributed interleaved planning and execution that span multiple levels (i.e., echelons) and multiple component commands. The main challenge in this environment is to rapidly and decisively respond to changes to assure timely execution with predictable results [2]. In such environments, there are three types of decision-making modes that have to be supported: automated decision-making; interactive decision-making; and collaborative decision-making [3]. *Automated decision-making* comes into play in routine situations or when responding to changes that can be handled through adjusting plan/workflow parameters during execution (e.g., divert an aircraft to a different landing site) or substituting a partial workflow "on-the-fly" in response to an anticipated change (e.g., tanker abort when aircraft has taken off and is en route). *Interactive decision-making* comes into play when the change is either unanticipated or is such that it requires human confirmation/intervention prior to execution (e.g., mission modification following tanker abort). *Collaborative decision-making* comes into play when responding to unanticipated events (e.g., enemy launching SCUD missiles with chemical/biological warheads) that involve multiple stakeholders (e.g., component commands in a JFACC scenario).

This section presents the issues and questions that need to be answered to ensure the successful planning and execution of military missions within the next generation C^4ISR environment, and describes how intelligent process management (IPM) can assist in answering these questions.

3.2 Recurring Questions

Some of the recurring questions that need answers in the planning and execution of C^4ISR processes within military missions are presented in Table 2.

Table 2. Recurring Problems and Questions

- How do we readily adapt planning and execution to dynamically changing circumstances?
- How do we determine and observe the status of what we have done?
- How do we accelerate effort on a pending activity?
- How do we determine why certain process steps are required?
- How do we determine what needs to be done before we are finished?
- How do we determine what can and cannot be done?
- How do we rapidly ascertain and satisfy prerequisites for upcoming tasks before doing them?
- How do we maintain awareness of where we are to ensure that we can respond to rapidly changing circumstances?
- How do we improve (i.e., make more efficient and effective) the way human planners work together?

Intelligent Process Management (IPM) can play a major role in answering these questions. Specifically, IPM is key to gaining visibility into and maintaining control of C^4ISR processes (Table 3).

Table 3. Why Explicit Process Management is Important

Facilitation
- Improve communication through explicit process representation
- Synthesize, re-engineer, or "quick-change" process to fit circumstances (i.e., tailor process to a new need)
- Maintain observability and visibility of status
- Expedite an activity by changing priorities (e.g., change execution order on path to X)

Explanation
- Why certain steps are necessary
- What you are waiting for
- "Why can't I do X now"

Support
- Selection, definition, and analysis prior to execution
- Status reporting, analysis, and "on-the-fly" adaptation during execution
- Improve coordination and facilitate cooperation between different team members (human, systems)
- Interactive and collaborative decisionmaking

Specifically, IPM supports warfighters by: (a) providing visibility into and tracking the status of troops, military assets and other resources (i.e., facilitation); (b) explaining to the user why certain steps are necessary or why certain activities cannot be performed at a particular point in time (i.e., explanation); (c) detecting changes in the environment and adapting responses and control strategy to fit the circumstances (i.e., execution support); and (d) assisting a single decisionmaker or a group of collaborators during decisionmaking associated with replanning activities (i.e., decision support).

4. THE IPM PROTOTYPE

4.1 IPM Overview

IPM for C^4ISR is the intelligent automated coordination, control, and communication of work, performed by human and/or software agents, through the execution of software distributed across a network of computers. The order of execution is dynamically determined by the state of the shared dynamic information structure that records the execution of the process and the prevailing "business (i.e., C^4ISR) rules." The purpose of IPM is to enable commanders to gain visibility into and maintain control of change-driven C^4ISR processes through a combination of automated/interactive execution, and decision support. Table 4 presents the highlights of intelligent process management along with examples from a JFACC scenario..

Table 4. Intelligent Process Management

- Enable commanders to gain visibility into and maintain control of planning and execution in the face of anticipated and unanticipated changes
- Perform automated reasoning/decision-making to determine the required type of adaptation:
  - Execution adjustment (e.g., divert aircraft to a different air base)
  - Partial workflow substitution (e.g., tanker abort at aircraft takeoff)
  - Suspension of automated execution and handoff with context "folders" to collaborative replanning (e.g., downed crewman)
- Recommend the response to change and either: (a) implement the response automatically, or (b) support human-in-the-loop response generation and execution

4.2 IPM Requirements

The major IPM requirements can be summarized as: an information-centric, scaleable, reconfigurable architecture; stable, predictable operation; and evolveable design and implementation. Table 5 presents a comprehensive set of IPM requirements.

Table 5. IPM Requirements

- Based on an "information-centric" architecture for distributed collaborative workgroups.
- Exhibits stability (i.e., predictable operation) during system execution.
- Operates within a distributed, extensible, tailorable architecture.
- Enables information sharing between human and software agents to assure that critical functions (i.e., high-value, high-priority activities) are carried out expeditiously.
- Enables command process re-engineering and operational flow optimization.
- Scales across multiple commands, multiple echelons, and all types of warfare, operations, and projects.
- Enables "on-the-fly" changes in executing processes, execution priority, and controller operating mode in response to changes.
- Enables evolution in architecture with lessons learned.

4.3 High-Level IPM Concept

As shown in Figure 2 IPM consists of two major components: (a) Process Design and Analysis (PDA) Tool Suite; and (b) Intelligent Control and Decision Support (ICDS) The first component is key to command process design and re-engineering. The second component is key to wide-area process monitoring, adaptive workflow execution, collaborative replanning, and decision support. The first component (shown by the darker shading in Figure 2) was the principal focus of the first year implementation.

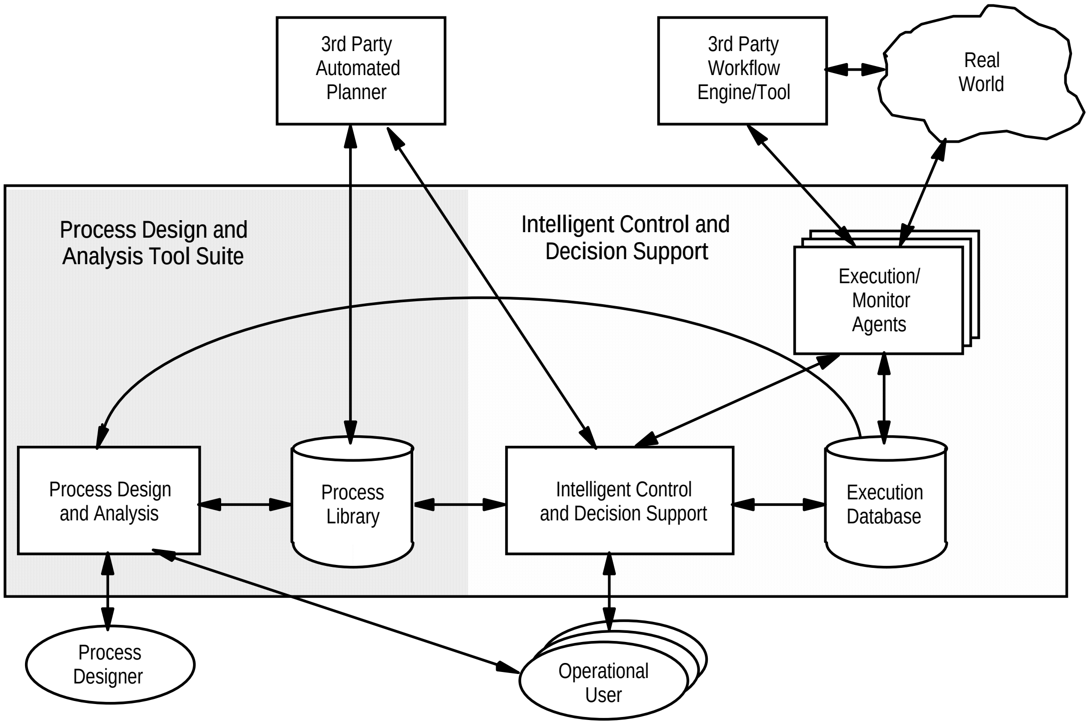

3rd Party Automated Planner
3rd Party Workflow Engine/Tool
Real World
Process Design and Analysis Tool Suite
Intelligent Control and Decision Support
Execution/ Monitor Agents
Process Design and Analysis
Process Library
Intelligent Control and Decision Support
Execution Database
Process Designer
Operational User

Figure 2. IPM Consists of Two Main Interoperable Components

4.4 Process Re-engineering using IPM's Process Design and Analysis Tool Suite

The purpose of command process re-engineering [1] is to:

- Shorten planning and execution cycle time.
- Rapidly generate executable courses of action (COAs).
- Generate military campaign plans that lead to predictable end states.
- Support coordinated force employment in the prosecution of military campaigns.
- Minimize the potential for mutual interference and fratricide in friendly forces.
- Provide planning and execution "products" on demand.

Figure 3 shows the IPM process design and analysis toolsuite. This component provides a full range of process re-engineering capabilities including: in-location knowledge engineering, process definition, verification, visualization, analysis, composition, and reporting. Details of these capabilities along with representative screen shots are presented in Appendix B.

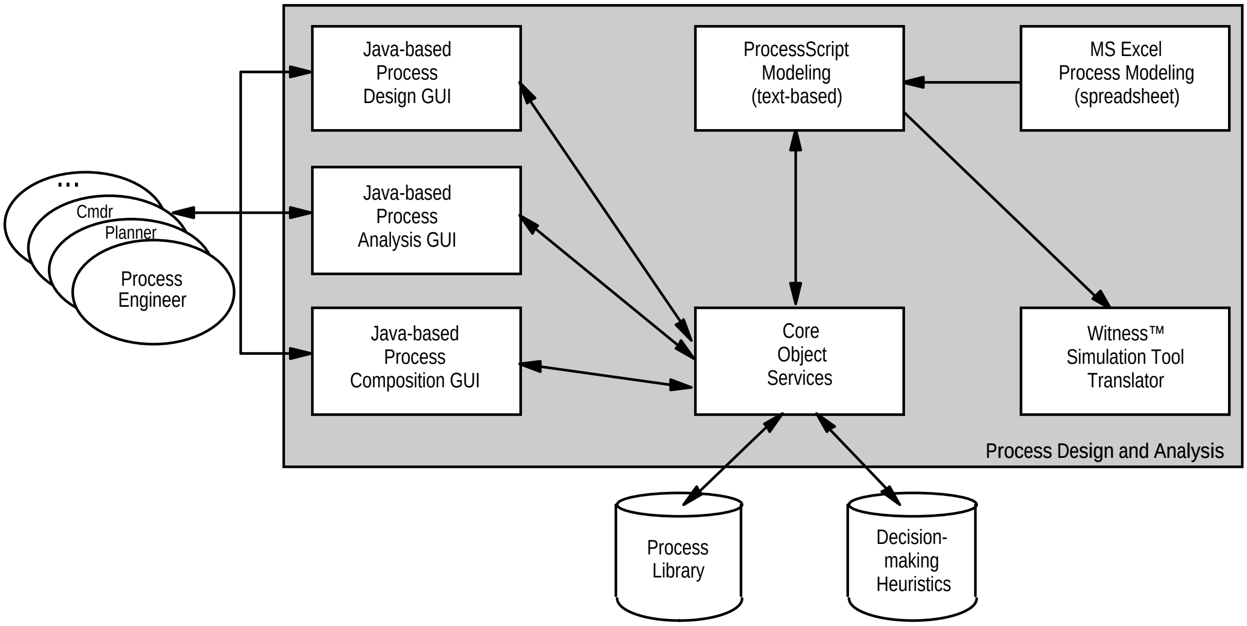

...
Cmdr
Planner
Process Engineer
Java-based Process Design GUI
Java-based Process Analysis GUI
Java-based Process Composition GUI
ProcessScript Modeling (text-based)
MS Excel Process Modeling (spreadsheet)
Core Object Services
Witness™ Simulation Tool Translator
Process Design and Analysis
Process Library
Decision-making Heuristics

Figure 3. IPM's Process Design and Analysis Component

4.5 C^4ISR Process Management using IPM's Intelligent Control and Decision Support System

IPM's Intelligent Control and Decision Support (ICDS) System is the "run-time" component of IPM. The ICDS system concept is shown in Figure 4. The ICDS is the component that is responsible for dynamic adaptation in response to change events. It performs automatic workflow adjustments, assists the user in interactively substituting a component process for an invalidated portion in an executing workflow, and determines when to handoff to collaborative replanning. In collaborative replanning, ICDS provides execution support.

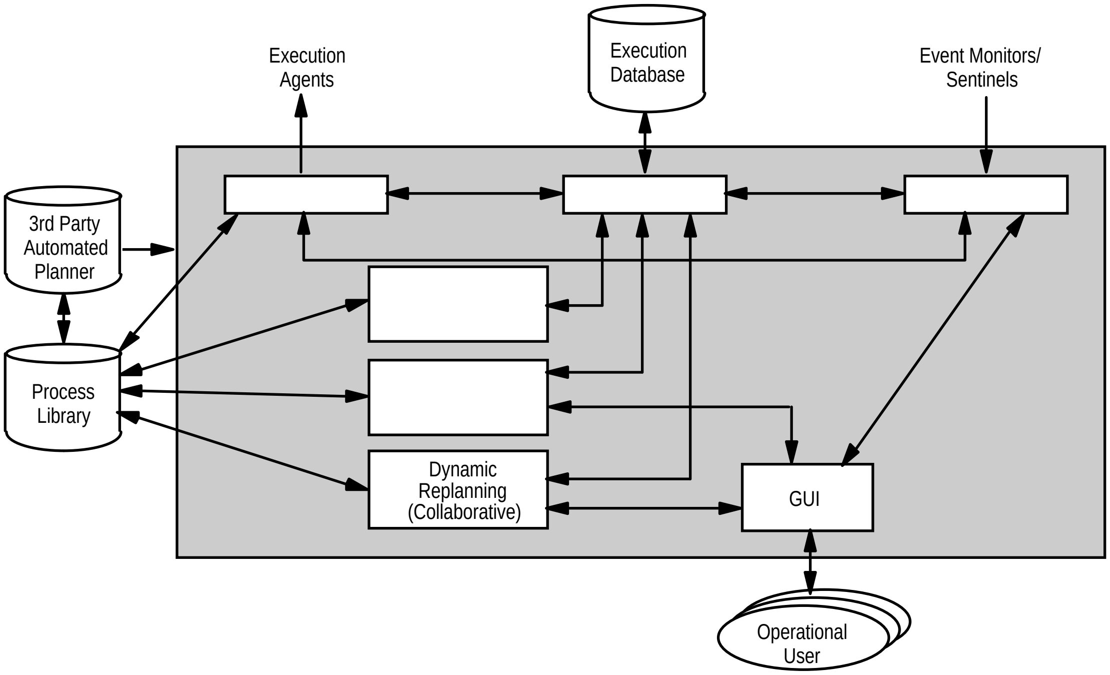

Execution Agents
Execution Database
Event Monitors/ Sentinels
3rd Party Automated Planner
Process Library
Dynamic Replanning (Collaborative)
GUI
Operational User

Figure 4. Intelligent Controller and Decision Support

4.6 IPM Execution and Replanning Concept

Figure 5 presents the IPM execution and replanning concept. IPM starts operation once it receives an initial plan for execution. When the IPM starts executing the plan, its execution results in updates to the world model. During execution, deviations from the plan (i.e., expectations) can occur. Such deviations (e.g., a variable exceeds a threshold, a resource becomes unavailable, an activity is taking longer than expected), when they occur, are detected by event monitors/sentinels [3]. This information is used by the intelligent controller to decide the appropriate type of response (i.e., automated workflow adjustment, interactive plan adaptation, or collaborative replanning).

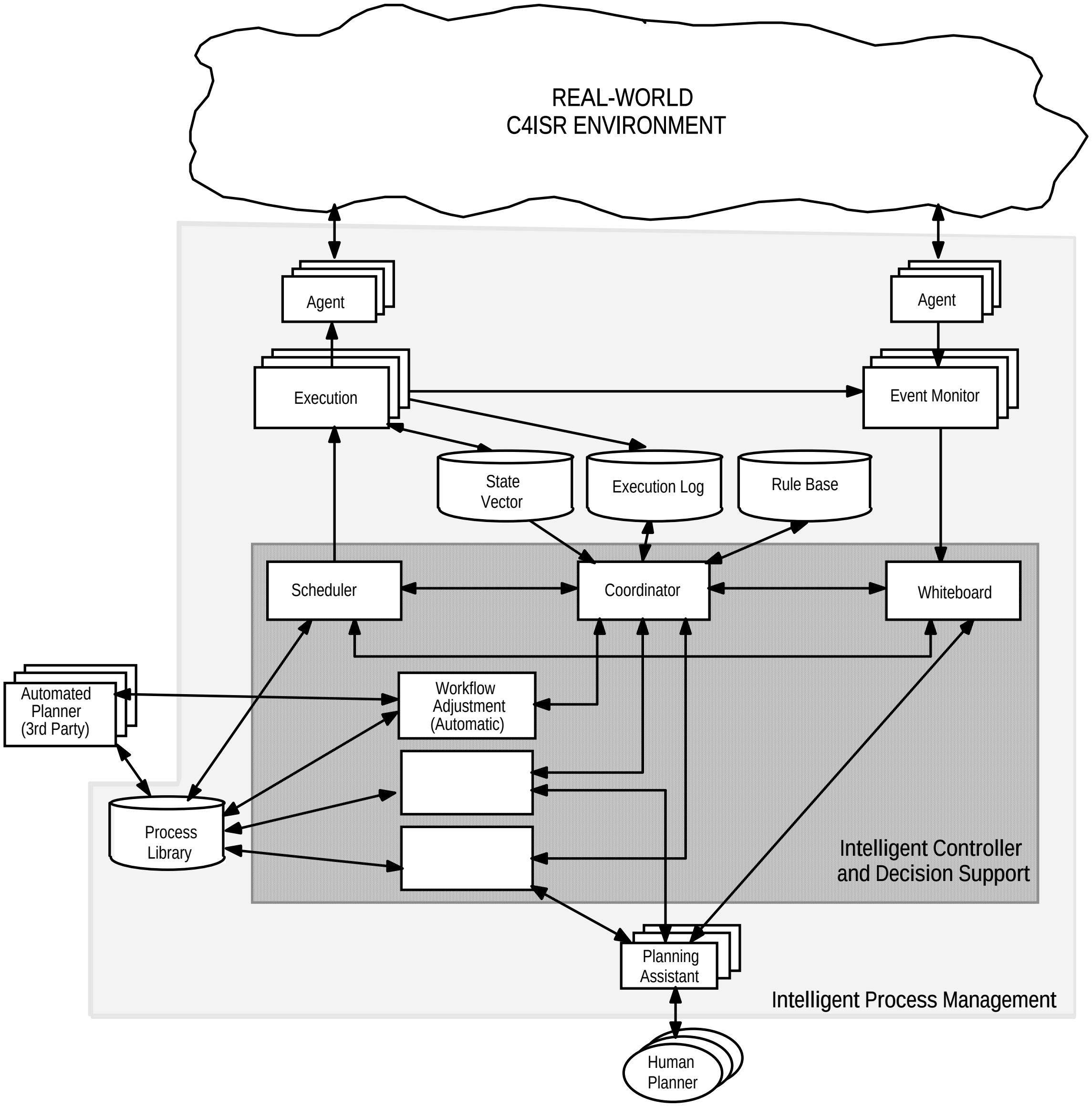


Figure 5. IPM Execution and Replanning Concept

As shown in Figure 5 the execution and replanning (i.e., “run-time”) component of IPM employs three types of agents: a) execution agents; b) event monitor agents; and c) human planners. The execution agent is an adapter to any software tool that might be invoked by the workflow execution engine. The event monitor agent is an adapter to one or more external data sources (e.g., JOPES, Gsorts, JTAV) or environmental sensors. The planning assistant agent helps human planners in planning, execution monitoring, and replanning. The automated planner is any third-party planning system that can automatically generate plans (e.g., SRI’S MPA). The process library is an online repository of component processes, process instances (i.e., plans) and partial workflows. The whiteboard enables information sharing between human and software agents.

The scheduler in the intelligent controller is responsible for scheduling a plan. The coordinator in the intelligent controller is responsible for orchestrating all components in the intelligent controller. It continually monitors events posted on the “whiteboard”, and uses state vector data, execution log status, and the IPM Rule Base to make planning-related decisions. It relies on third-party automated planners to generate a new plan and instructs the scheduler to modify the schedule when necessary. Details of the terminology used in Figure 5 are presented in Appendix C.

The coordinator decides the correct response to a change event. The response could be one of four classes. Type 0 response implies maintaining the status quo. Type 1 response implies automatic adjustment of plan/workflow parameters in the executing workflow. Type 2 response implies interactive selection and substitution of a component procsss/partial workflow during execution. Type 2 response is aided by an online library of reusable, customizable, and instantiable component processes/partial workflows. Type 3 response implies handoff to collaborative replanning with “context folders.” Collaborative replanning is supported by a collaboration support tool such as Mitre’s Collaborative Virtual Workspace (CVW) [8] or SPAWAR’s Odyssey [9]. Since workflow execution is a combination of automated steps and human interventions, the execution engine is described as “mixed-initiative.” Also, as execution proceeds, the new state information is fed back to the planner (outside the purview of IPM) for planning beyond the original plan. Plan refinements from the planner are input to the execution engine as and when such inputs become available.

4.7 Highlights of the Execution Replanning Component

Most commercial workflow products are based on executing a sequence of operations in accordance with a predefined reference process model or script [10]. [11]. However, while such approaches might be applicable to specific regimes in the C^{4}ISR process, they don’t support unexpected event handling or collaborative replanning activities. Therefore, our approach is to design an event-driven architecture that can intelligently handle all kinds of events, expected or unexpected, prescribed or postscribed.

The IPM execution component will be implemented within a flexible architecture in which the default operational mode will be that of traditional workflow control. It is only when unexpected events occur that a more sophisticated control strategy will be invoked. In such cases, either automatic workflow adjustment, interactive component plan substitution, or collaborative replanning will be performed.

The Coordinator within the Intelligent Controller decides the correct response to a change event based on event type, current workflow status and context, user-supplied rules, and system-generated rules from execution history. For prescribed events, the system should be able to perform user-defined event handling actions or workflow adjustment without human intervention. For those events that the system cannot handle by itself, it will interact with the human planners to make plan modification decisions. In doing so, it will assist the human users with visualization of the status and context information, and provide the user with: (a) the execution history of previous similar situations; (b) the reusable component process asset library; (c) transparent access to 3rd party automated planner(s), and complex process simulation and analysis tools. The implementation of this flexible architecture will be accomplished using information technologies including agents, CORBA and other prevailing standards, "publish-subscribe" and other proven design patterns.

4.8 Event Model and Event Handling for Intelligent Process Management

A key feature of the Intelligent Process Management (IPM) is its ability to handle events dynamically and intelligently [3]. There is no single event handling method today that can handle the spectrum of events adequately. This section presents work to date on the conceptual design for the event model in IPM. In Table 6, we present a taxonomy of events that encompasses the various classes of events that can occur during integrated planning and execution. This event taxonomy will be used to create the different handling schemes/approaches for each event type.

Table 6. IPM's Event Taxonomy

- Internal event (an event that is triggered by other events in the system)
    - Process event
        - Process status change event
            - Process ready event
            - Process start execution event
            - Process finish execution event
            - Process suspended event
            - Process terminated event
        - Process property (or attribute) change event
    - Resource event
        - Resource availability change event
            - New available resource event
            - Resource no longer available event
        - Resource property (or attribute) change event
    - Goal event
        - Goal accomplished event
        - Goal cancelled event
    - Timing event (an event triggered by the system clock)
- External event (an event that is triggered by user or environment)
    - Expected event (an event that is defined by the user with prescribed handling function)
    - Unexpected event
        - Known event (an event similar to other events in the execution history)
        - Unknown event

The following paragraphs present a definition of the events, event response modes, event handling mechanisms, and a brief overview of future work (based on follow-on funding).

4.8.1 Event Definition

In IPM, certain events are defined by the system with built-in event handling mechanisms. In addition, events may be defined by the users through the following mechanism:

- A primitive event can be defined as (a op b) where
  - a is an attribute of an object (process or resources object) or a variable defined by the user,
  - op is any comparison operator (i.e., = =, <>, >, >=, <, <=),
  - b is an attribute of an object, a variable defined by the user, or a constant set by the user.
- A composite event may be defined by combining primitive events or other composite events using logical operators (i.e. AND and OR).

For any event defined by a user, the user can define the event handling function for it. For any event in the system, a user can subscribe to it and the system will deliver a message to the user when the event happens. The user can then make the event handling decision at run time.

4.8.2 Event Response Mode

The IPM execution engine will be able to respond to events using one of the following four modes:

- *Predefined execution propagation*. In this mode, the event handling function is built into the system or prescribed by the users.
- *Automatic adjustment*. In this mode, the system performs automatic workflow adjustment without human intervention. The system may automatically invoke an automatic planner or scheduler for this purpose.
- *Interactive substitution of component plan/workflow*. In this mode, the system interacts with a user to choose and customize a component plan/workflow from the process library (or one supplied by an automatic planner).
- *Collaborative dynamic replanning*. In this mode, the system interacts with multiple human planners to develop a new plan collaboratively. The new plan may be built from scratch or using component processes available from the process library or from automatic planners.

Depending on the event type and the context of the event, the system will choose the appropriate response mode for the event. In some cases, the system may fork multiple threads with each thread in a different mode. For example, when a goal is accomplished by a process, the system may terminate other processes that were invoked to achieve the same goal. In the mean time, the system may interact with a human planner to perform replanning.

4.8.3 Event Handling Mechanism

This section discusses the event handling mechanism required to respond to each event type.

Process ready event. This happens when all the preconditions of a process are satisfied. The system will:

- inform the assigned actor of this process;
- collect related information about the process in a folder for the actor;
- update the actor's work list.

Process start execution event. This happens when the actor starts executing a process. The system will:

- inform the registered listeners;
- update the resource availability list (i.e. remove the resource from the available pool);
- update the precondition status for the successor processes (i.e. check if the successor processes are enabled by this event).

Process finish execution event. This happens when the actor finishes executing a process. The system will:

- inform the registered listeners;
- update the resource availability list (i.e. release the resources used or created by this process);
- update the precondition status for the successor processes (i.e. check if the successor processes are enabled by this event);
- trigger automatic plan adjustment if the difference between the actual finish time and the schedule finish time exceeds the threshold (note: this may be handled by an intelligent sentinel agent).

Process suspended event. This happens when an authorized user decides to temporarily suspend a process. The system will:

- inform the actors of this process;
- inform the registered listeners;
- update the resource availability list (i.e. release the resources);
- propagate the suspend command to the sub-processes;
- review overall process status and advise user on plan adjustment or re-planning.

Process terminated event. This happens when an authorized user decides to terminate a process or triggered by another internal event (e.g. the goal of this process has been accomplished by another process). The system will:

- inform the actors of this process;
- inform the registered listeners;
- update the resource availability list (i.e. release the resources);
- propagate the terminate command to the sub-processes;

- if the event is triggered by a user, review overall process status and advise user on plan adjustment or re-planning;
- if the event is triggered by another internal event, perform user-defined handling function if it exists;
  - otherwise, review overall process status and either:
    - perform automatic plan adjustment;
    - inform a designated user to start an interactive plan substitution/modification session; or
    - start a collaborative dynamic re-planning session.

Process property (or attribute) change event. This happens when a property (attribute) of a process is modified by a user or as a result of the execution of another process (e.g. scheduling process). The system will:

- inform the registered listeners,
- perform user-defined handling function if it exists;
- otherwise, review overall process status and may decide to perform automatic plan adjustment.

New available resource event. This happens when a new resource is created or released by a process. The system will:

- inform the designated process if this resource is allocated (reserved) to that process;
- otherwise, inform all the other "waiting" processes (i.e. the processes with all the "predecessor" conditions satisfied and waiting for resources).

Resource no longer available event. This happens when an available resource is "reserved" by a process. The system will:

- check if there is a contention for this resource;
- if yes, the system will either:
  - perform automatic plan adjustment; or
  - interact with a designated user to perform plan modification.

Resource property (or attribute) change event. This happens when a property (attribute) of a resource is modified by a user or a process. The system will:

- inform registered listeners;
- check if the change will affect the "usability" of the resource (e.g., a new skill learned by a person will enable the person to qualify for additional processes; a role revocation will make a person be ineligible to perform certain processes);
- inform designated resource manager (a software or a human user) about the changes in usability.

Goal accomplished event. This happens when a goal is accomplished by a process. The system will:

- inform registered listeners;

- if there are “alternative” processes that were invoked to achieve the same goal,
  - terminate those processes;
  - may perform automatic plan adjustment or start interactive re-planning session.

Goal cancelled event. This happens when a goal is cancelled by a user. The system will:

- terminate those processes that were invoked to achieve this goal;
- propagate the cancel command to the sub-goals;
- review overall process status and advise user on plan adjustment or re-planning.

Timing event. This is triggered by the system clock. The system will:

- trigger the designated sentinel agents to:
  - review overall process status;
  - start, suspend, or terminate designated processes;
  - alert designated users;
  - perform automatic plan adjustment;
  - start an interactive plan modification session;
  - start a collaborative dynamic re-planning session.

  (Note: The sentinel agent could decide to perform one or more of the actions listed.)

Expected external event. This event is defined by a user with prescribed handling function. The event is triggered by a user (e.g. an intelligence officer) or an external system (e.g. a sensor) through an event notification API. The system will:

- inform registered listeners;
- execute the prescribed handling function.

Unexpected known external event. This is a new event that is not defined in the system. But the system has determined that there are similar events in the execution history through pattern matching. The event is triggered by a user (e.g. an intelligence officer) or an external system (e.g. a sensor) through an event notification API. The system will:

- perform detail analysis about this event and the previous history and decide to:
  - invoke the handling function for previous event; or
  - inform a designated user to start an interactive planning session.

Unknown external event. This is a brand new event that is unknown to the system (i.e. the event is not defined in the system and no similar events are in the execution history). The event is triggered by a user (e.g. an intelligence officer) or an external system (e.g. a sensor) through an event notification API. The system will:

- notify a group of designated users to start a collaborative dynamic re-planning session.

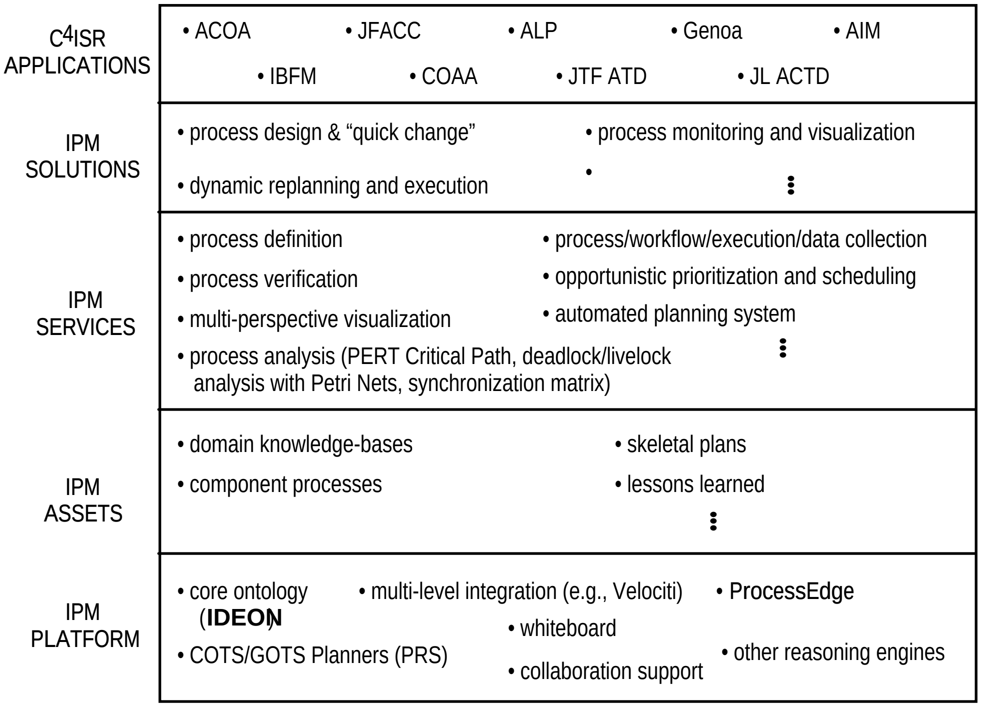
4.8.4 Future Work

To finish the complete design of the event handling mechanism, we plan to continue with the detailed design of the following based on availability of follow-on funds:

- pattern matching of events,
- sentinels for situation monitoring,
- decision rationale for choosing the appropriate event response mode.

4.9 Disciplined Development Approach

We will follow a disciplined, “building block” approach to create the IPM software in a staged series of “capability-adds.” Figure 6 shows the IPM conceptual layers. An IPM solution is created by developing/adapting the components from the bottom three layers to satisfy the solution requirements of the top layer.



C4ISR APPLICATIONS	• ACOA • JFACC • ALP • Genoa • AIM • IBFM • COAA • JTF ATD • JL ACTD
IPM SOLUTIONS	• process design & “quick change” • process monitoring and visualization • dynamic replanning and execution • ⋮
IPM SERVICES	• process definition • process verification • multi-perspective visualization • process analysis (PERT Critical Path, deadlock/livelock analysis with Petri Nets, synchronization matrix) • process/workflow/execution/data collection • opportunistic prioritization and scheduling • automated planning system ⋮
IPM ASSETS	• domain knowledge-bases • component processes • skeletal plans • lessons learned ⋮
IPM PLATFORM	• core ontology (IDEON) • COTS/GOTS Planners (PRS) • multi-level integration (e.g., Velociti) • whiteboard • collaboration support • ProcessEdge • other reasoning engines

Figure 6. Component-based Structured Development

4.10 Sample Scenarios

In the following paragraphs, sample scenario vignettes are presented from the JFACC domain to illustrate several key aspects of IPM. The mini-scenarios are geared to illustrating: (a) continuous re-entrant planning loop; (b) change of information; (c) switch to conventional workflow; (d) change of goal; and (e) change of plan. These mini-scenarios follow.

<u>Mini-scenario #1: Continuous re-entrant planning loop</u>

- Scenario from Desert Storm.
- Operation: SCUD Hunt
- Goal: To find and destroy the Iraqi SCUD missile launchers.
- This is an unprecedented situation. Therefore, there is no existing plan that can be used.
- The plan is developed collaboratively by automated planners and human decision makers with the aid of process components (partial process sequences).
- The plan is scheduled into op orders.

- The op orders are executed and results are monitored.
- The execution results are analyzed.
- The analysis results are used in re-planning.
- New op orders are scheduled as a result of the new plan.
- Iterations of Plan, Op orders, and execution can continue until the goal is achieved.

Mini-scenario #2: Change of information

- Goal: To provide enough fuel (additional 15000# of fuel is needed) for a flight of four F-16s returning to the base from an INTERDICTION mission.
- Plan: Dispatch a tanker.
- Op order: Tanker A will take off at 10:00 with 24000# of fuel.
- Event: The on-board air crew unit notices some problem and decides to abort the takeoff at 09:55.
- The Abort Report is sent to the Tanker Rep.
- The Tanker Rep finds another tanker that is available.
- The existing Op order is replaced with a new one: Tanker B is scheduled to take off at 10:30 with 18000# of fuel.

Mini-scenario #3: Switch to conventional workflow

- Goal: To provide adequate fuel for a flight of four F-16s returning to the base from a routine training mission.
- This is typically a routine operation.
- The Tanker Rep creates a plan using: (1) the process component from the library, (2) the amount of fuel needed for this mission.
- The op order is created and executed.

Mini-scenario #4: Change of goal

- Goal: To destroy enemy's military facilities.
- Intelligence has identified 10 weapon storage facilities; these targets are the sub-goals.
- Plan: Send two B-2 bombers to destroy the 10 targets.
- Event: The intelligence found a new chemical weapon facility.
- This new facility is established as a new target with top priority.
- Try to add a new plan to destroy this new target and find out that there are no "additional" unplanned resources available. Therefore, a replan is needed. (i.e. the original plan needs to be modified.)
- New goal: This new chemical weapon facility plus 8 of the original 10 targets.
- New plan: Send two B-2 bombers to destroy the new set of 9 targets.

Mini-scenario #5: Change of plan

- Scenario from Desert Storm.
- Goal: To destroy the bunkers built by Iraqi soldiers in Kuwait.
- Plan: Divide the area into boxes. For each box, send one airplane. The airplane will survey the assigned box, and destroy the bunkers found. Shifts of airplanes will be sent to destroy all the bunkers.

- Assessment after a few attack “iterations”: The bunkers found by different airplanes assigned to same box are almost identical. Therefore, the "undiscovered" bunkers remain undiscovered.
- New plan: The "original" airplanes will not be responsible for surveying the area and finding the target. Instead, another group of airplanes (killer scouts) are sent for this purpose and they will instruct those shifts of attack airplanes to designated targets.
- This new plan is created while the original plan is still in execution.
- This new plan will negate the old plan and op orders.

4.11 Architectural Strategy

The IPM architecture is based on a layered structure (Figure 7) consisting of: the core IPM ontology; application-specific extensions; and implementation-specific mappings [2], [3].

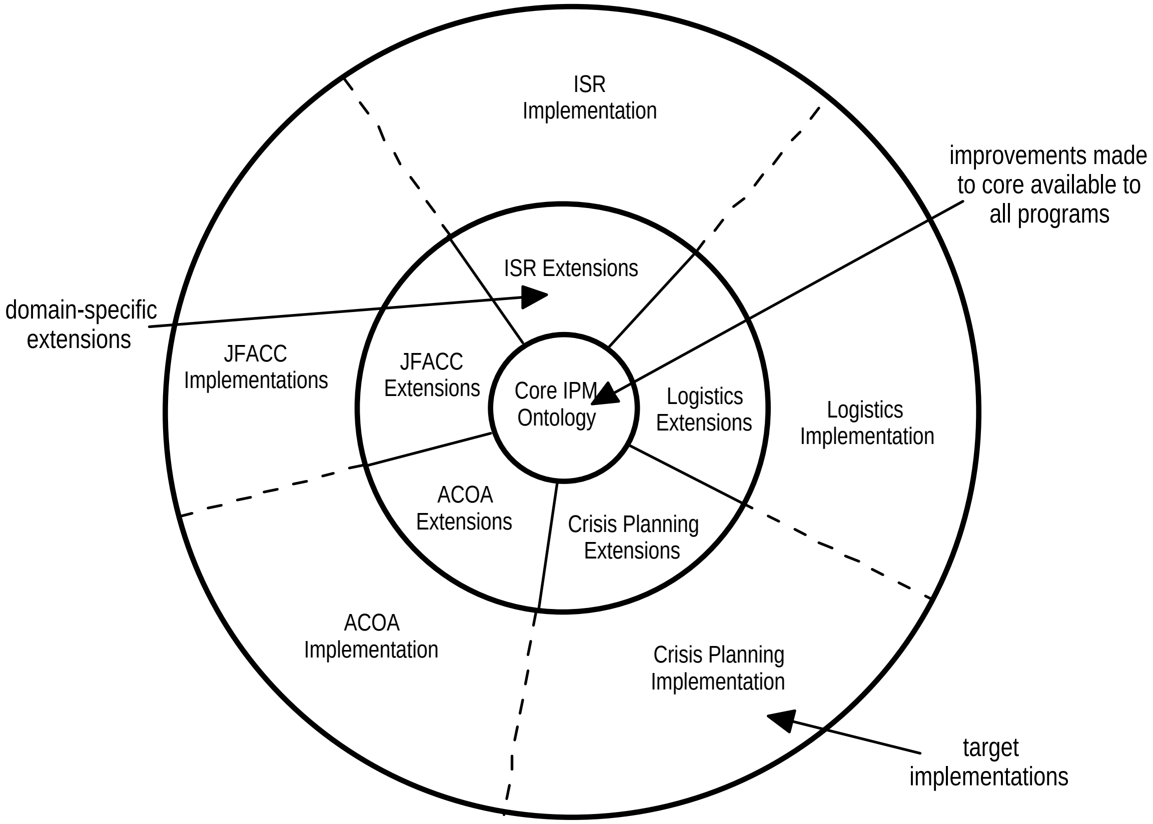
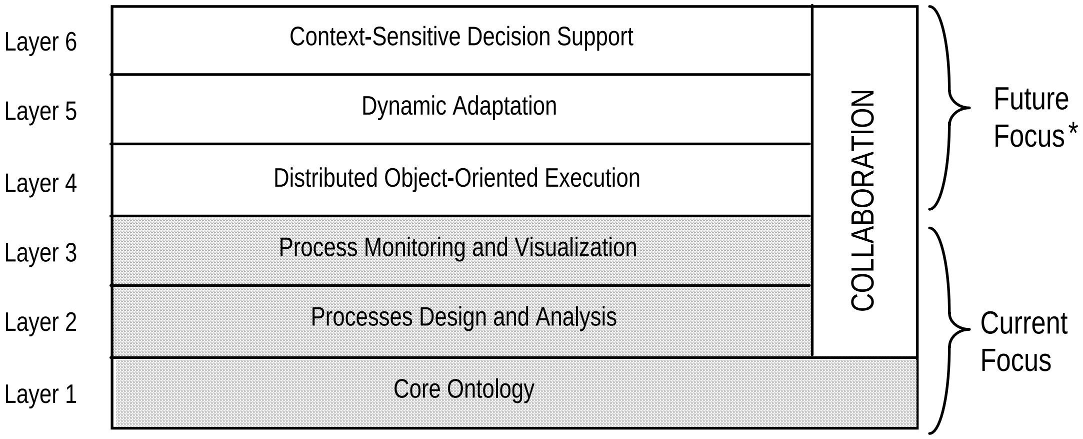


Figure 7. IPM Architectural Strategy

As shown in Figure 7, the core IPM ontology consist of key concepts and relationships that are central to IPM and, therefore, common to all applications. The domain-specific extensions to the ontology are in the layer adjacent to and outside the core ontology. The outermost layer is the implementation layer that maps the core ontology and domain-specific extensions to a target implementation environment (e.g., CORBA IDLs, DCOM, KIF).

4.12 Implementation Strategy

IPM, as conceptualized in this report, combines several technologies to create an innovative solution to the C^4ISR process management problem. Figure 8 presents a six layer "building block" approach for creating a "full blown" IPM.


Layer 6
Context-Sensitive Decision Support
Layer 5
Dynamic Adaptation
Layer 4
Distributed Object-Oriented Execution
Layer 3
Process Monitoring and Visualization
Layer 2
Processes Design and Analysis
Layer 1
Core Ontology
COLLABORATION
Future Focus*
Current Focus

**based on follow-on funding*

Source: Madni A.M. (1998): **ProcessEdge**™ *Implementation Strategy (Copyright ©, 1998)*

Figure 8. Building Block Approach

The first layer is the *core ontology* for Intelligent Process Management. It is the foundation for all subsequent development. The core ontology for IPM is **IDEON**™ [12], ISTI's Distributed Enterprise Ontology. **IDEON** is specifically designed to support both enterprise modeling and process/ workflow management. This ontology can be adapted to the C^4ISR domain, and then specialized for various applications such as ACOA, ALP, Genoa, and JFACC.

The second layer is the *process design and analysis* layer. It is the layer that is central to command process re-engineering. It consists of: a) in-location knowledge engineering; b) process definition; c) verification; d) visualization with "drill down" and "roll up;" and e) process analysis in terms of critical path, activity-based cost analysis, and process mismatch analysis. This layer provides the foundation for significantly greater types of analysis given the expressive power of **IDEON** (e.g., simulation, resource analysis, training analysis). Wide-area collaborative process design requires a transaction model that varies in sophistication depending on the degree of collaboration (i.e., multiple viewing only, multiple view-single edit, multiple view-multiple edit of different portions, multiple view-multiple edit of the same process).

The third layer is the *process monitoring and visualization* layer. This layer allows users to monitor the progress of the overall process in terms of status at the process level, which is derived by "rolling up" the status from the bottom-most activity level. The activity level is the level that interfaces with the automated tools. It is also the level where manual activities are performed by human users.

The fourth level is *the distributed federated execution management* layer. This layer is the one that enables "global" distributed process execution and management across multiple echelons and geographic locations. This layer allows each echelon to control their "local" processes while participating in the "global" processes. The key enabling technology for this layer is a reliable, secure, flexible, real time "messaging" system.

The fifth layer is the *dynamic adaptation* layer. This layer provides the "dynamism" in IPM. It includes: a) automatic workflow adjustment; b) interactive tailoring and substitution of a component process/workflow during execution; and c) collaborative replanning in the face of an unprecedented contingency event or when multiple stakeholders are involved in response to the contingency. Third party tools such as SPAWAR's Odyssey or Mitre's CVW are candidates for the coordination support environment although commercial products featuring the required capabilities are beginning to appear on the market. From a technical perspective, this type of process adaptation is event-driven, non-static (i.e., unprescribed), automatic or semi-automatic (i.e., with human user in-the-loop), and context-sensitive. The dynamic adaptation is partially driven by users at both "design" time (with user-defined control rules) and run time (in collaborative dynamic replanning). The dynamic adaptation is also partially driven by the intelligent controller in the system, using agent, sentinel, pattern matching, and other AI technologies.

The sixth layer provides *decision support* during interactive workflow adaptation and during collaborative planning. In interactive workflow adaptation, it assists the user in selecting suitable components, tailoring them, and inserting them in an ongoing process to replace invalidated portions of the workflow. In collaborative replanning, the decision support capability guides users in application/tool selection, directs the users to appropriate "folders" and work products needed to perform their activities, records the results of the collaborative session, and prompts the user with pending activities that need collaborative resolution.

It is important to note that the collaboration component cuts across layers 2 to 6. This is because collaboration capabilities come into play at each of these levels.

5. CONCLUDING REMARKS

5.1 Current Uses of IPM Prototype

The current IPM prototype can be used on several ongoing DARPA ISO programs (e.g., JFACC, ALP, Genoa, AIM) as well as on DARPA-DISA JPO's ACOA Program. The current capabilities and their benefits are presented in Table 7.

Table 7. Current Uses of the IPM Prototype

- Acquire process models (knowledge) in electronic form from geographically distributed subject matter experts
  - – *Benefit*: Eliminates rekeying of process models, saves travel costs
- Verify completeness and consistency of the models and correct deficiencies
  - – *Benefit*: Detect and correct errors at "build" time rather than "run" time
- Analyze processes in terms of cycle time, resource requirements, and costs
  - – *Benefit*: Pre-analyzed processes support comparative analysis when adopting a process
- Persistently store analyzed processes in process library
  - – *Benefit*: Support "plug-and-play" of component processes
- Select, tailor, and reuse processes from process library
  - – *Benefit*: Dramatic reduction in process design cycle time and cost
- Monitor and visualize processes at multiple levels of abstraction
  - – *Benefit*: Rapidly verify progress of executing process by "rolling up" status of individual analysis

    Rapidly locate bottlenecks/impediments by "drilling down" status of high level process of it is "blocked"

5.2 Future Directions

At the time of preparation of this report, the ACOA program was recommended by our DARPA sponsor as the target insertion environment for IPM. To this end, we expect to transition command process design and analysis, and command process monitoring and visualization capabilities to ACOA. We expect to integrate with selected tool(s) in the ACOA toolsuite in the process. Based on the availability of follow-on incremental funding, we plan on continuing with IPM development to cover layers 4-6 in Figure 7.

5.3 Other Insertion Opportunities

Several other opportunities for IPM insertion exist with DARPA ISO and DARPA's Joint Logistics Technology Office (JLTO). ISO programs that could benefit from IPM include: JFACC, AIM, and Genoa. JLTO programs that stand to benefit from IPM include ALP and Joint Logistics ACTD.

REFERENCES

1 Integrated Battleforce Management, Command Process Re-engineering and Military Utility Panel, May 8-9, 1998.

2 Madni, A.M. "Intelligent Process Management: The Key to Replanning and Execution Cycle Time Reduction in Dynamic C4I Environments," ISTI White Paper, ISTI-WP-12/97, Santa Monica, California, Dec, 1997.

3 Madni, A.M. "Intelligent Process Management: The Vision and Conceptual Framework," ISTI White Paper, ISTI-WP-2/98, Contract #F30602-97-C-0293, February 5, 1998.

4 Lee, J., Gruninger, Jin, M., Malone, T., Tate, A., Yost, G., and other members of the PIF Working Group, "The PIF Process Interchange Format and Framework." May 24, 1996, URL: http://ccs.mit.edu/CCSWP194/CCSWP194.html#8

5 Schlenoff, C., Knutilla, A., Ray, S., Unified Process Specification Language: Requirements for Modeling Process, NISTIR 5910, National Institute of Standards and Technology, Gaithersburg, MD (1996).

6 Fritzson, R., Finin T., McKay D., and McEntire R. "KQML - A Language and Protocol for Knowledge and Information Exchange," 13th International Distributed Artificial Intelligence Workshop, Seattle WA., July 28-30, 1994.

7 ProcessScript™ User Guide, Intelligent Systems Technology, Inc., March 31, 1998.

8 Kordash, J.L., Deus, L.M., and Rhode, E. Collaborative Virtual Workspace (CVW), 1998 URL: http://www.mitre.org/resources/centers/advanced_info/g04e/cvw.html.

9 Odyssey/facilitate.com (http//www.facilitate.com.

10 Madni, A.M. and Madni, C.C. "An Adaptive Wide-area Design Process Manager for Collaborative Multichip Module Design," Proceedings of the 1997 IEEE Multi-Chip Module Conference, Santa Cruz, CA, February 1997.

11 Madni, C.C. and Madni, A.M. "Web-enabled Collaborative Design Process Management: Application to Multichip Module Design" (invited paper) Proceedings of the 1998 IEEE International Conference on Systems, Man, and Cybernetics, San Diego, California, October 11-14, 1998.

12 Madni, A.M., Madni, C.C., and Lin, W. ISTI Distributed Enterprise Ontology (**IDEON**™): An Overview, ISTI White Paper, ISTI-WP-5/97-1, Intelligent Systems Technology, Inc., Santa Monica, California, 1997-1998.

13. Madni, A.M., Lin, W., and Madni, C.C. "**IDEON**™**/IPPD**: An Ontology for Systems Engineering Process Design Management," (invited paper) Proceedings of the 1998 IEEE International Conference on Systems, Man, and Cybernetics, San Diego, California, October 11-14, 1998.

APPENDIX A
COTS Workflow Products Survey

Following is a survey of four COTS WORKFLOW products.

1. ActionWorks Metro 4.0 by Action Technologies, Inc.
   http://www.actiontech.com

   Entities

   - Process definition:
     - Process Name
     - What actions need to happen
     - Who is involved (person or role)
       - customer
       - performer
     - How long it takes (day/hour/minute)
     - What it costs
     - Conditions of satisfaction (string)
   - Each step is divided into four distinct phrases:
     - Preparation
     - Negotiation
     - Performance
     - Acceptance
   - Rendezvous, splitter, and conditional flow objects
   - Person
   - Role (a job category)

   Claimed Functionality

   - Distributed dynamic work activities across multiple servers
   - Support LDAP (Lightweight Directory Access Protocol)
   - Provide one-click navigation between work-related email messages and rich, web-based forms.
   - Built-in Business Process Integrity features that perform simultaneous updates to multiple servers, promulgate changes across all affected work, and rollback transactions where necessary.
   - Users receive all work in their Web-based WorkBox or directly into their e-mail inbox, via Metro WorkLinks.
   - ActionWorks Process Builder: Metro applications are developed in a graphical, object-oriented environment processes are modeled using drag-and-drop icons, templates wizards

- By identifying a customer for each task in a process, the Analyst positions Customer Satisfaction at the core of every step.
- Consistency check:
  - improper links
  - unassigned customer or performer roles
  - missing conditions of satisfaction
  - cycle time problems (project cycle time in primary loop is not equal to the sum of cycle times in the secondary loop)
  - cycle value problems
- Other problems checked
  - incomplete processes
- Business rule definition
  - MS VB-compatible scripting language
  - Expression editor for building formula
  - Support data attributes of must fill, editable, read-only and hidden

Implementation

- Web-based
- NT 4.0 server with MS IIS 3.0 or Netscape Enterprise Server 3.0, MS SQL Server 6.5, Microsoft Exchange or Netscape Messaging Server
- Development Interfaces:
  - HTML, DHTML, XML, Java, JavaScript, VB Script
  - Active X, COM
  - Precision Internet Forms

Interoperability

- Unknown

Remarks:

A simple process model confined by their 4-step (preparation, negotiation, performance, acceptance) loops model which emphasize on the "customer" satisfaction.

2. Forte Application Environment (including Conductor) by Forte
http://www.Forte.com

Claimed Functionality

- Location transparent
  - The run-time system provides the location for any object through a naming service
- Integrated transaction
  - Any Forte component can be defined as transactional by checking a box on the object's property sheet.

  - Within an application, any number of these transactional components can be included in a business transaction.
- Multi-user support
  - The Forte partitioning system can automatically generate the necessary code for multi-threading and concurrency control when the application id deployed.
- Fault tolerance and load balancing
  - Can load balance and/or failover across heterogeneous machines (e.g. failing over from a UNIX to an NT server.)
- Provides integrated support for various Web browsers
- Portable database support
  - Vendor-neutral ANSI SQL
- Portable OS and networking support
  - Includes portability abstractions for operating systems and networks
- Integrated application management: software distribution, management, configuration, etc.
- Development environment
  - Development repository supports tram development with check-in/check-out, versioning, and branching.
  - The repository also stores an environment definition which includes information on each machine.
  - 4GL-based development
    - 4GL interpreter (Rapid Application Development support)
    - 4GL to C++ code generation
    - Multitasking 4GL debugger
  - Generate application from RDBMS data schema or an imported object model from object oriented design tools (Select, Rational, Cayenne, Platinum/Paradigm Plus)
    - The generated distributed application can be further enhanced with custom business logic.

Implementation

- The Forte run-time system consists of an integrated infrastructure of application services. Forte developers construct an application as a collection of components that are subsequently deployed in a distributed environment supported by these services.
- Forte use the publish-and-subscribe event model to enable applications to respond to external stimuli such as real-time alerts of low inventory or the arrival of a high priority call.
  - Events are specified on Forte objects, which in turn keep track of all other objects that express an interest in that event.

- Forte includes a router that can send remote method invocations to more than one server. This is a key enabling technology that underlies Forte's load balancing for scalability and failover for highly available applications.
- C programming interface to Forte run-time system
  - Will support Java in the next major release.
- Each Forte application partition contains a collection of system management and monitoring agents.
  - Management agents enable the system administrators to do such things as start and stop remote servers, give permissions to new users, and reconfigure the application.
  - Monitoring agents instrument more than 200 system variables – including the message traffic counts, queue depths, and message packet size – for each distributed components.
  - The agent infrastructure also has a programmable interface to monitor any other component, such as the status of each modem in a modem bank.
- Windows workshop for defining windows, panels, and canvassed using a standard drag-and-drop interface for manipulating screen widgets.
  - At partitioning time, Forte uses this information to render the screen in the native look and feel of the target GUI environment, using the native GUI tool kit or HTML.
- HTTP state information is enforced using cookies, URL encoding, and hidden fields.
- Partitioning engine
  - The partitioning plan specifies how to break the application into modules and where to place each module in the distributed environment.
  - Mapping environment-neutral application code into the environment-specific technology infrastructure of operating systems, networks, GUIs and RDBMSs.
  - Moving each module to its targeted platform and performing all the necessary bindings to enable the application to run.
  - If the developer modify (with a drag-and-drop visual tool) the partitioning scheme, Forte will recompile and re-distribute the application components.

Claimed Interoperability

- WebEnterprise supports the publication of Forte application services as CORBA, ActiveX/SCOM, IIOP or JavaBeans components
- Supports integration with Tuxedo, CICS, and Encina.
- Communicate with other non-Forte applications using IIOP, DCOM, CORBA, DCE, and sockets
- Support third-party HTML editors, HTML extensions, and third-party Java clients.
- Can publish collected data to HP/Overview and IBM/Tivoli system management systems.

Forte Conductor

Entities

- process
  - process attributes to tell the client how to retrieve data from database
- activity (smallest unit of work performed within a process)
  - priority
  - resource pool of people (roles) or complex rule
  - attached data/documents
- user
  - can be assigned multiple roles
  - associated with one manager
- process routing rules
  - triggers
    - specify what other activities must have completed, or what data relationships must exist, before an activity is started
  - timers
    - deadline timers
      - e.g. start a new process at the end of the month
    - elapsed timers
      - e.g. raise priority if a customer call has not been worked on for more than two weeks

Claimed Functionality

- Graphical process designer.
- Distributed workflow engines.
- Flexible reliable work queuing
  - Work offer model
    - User selects a task from a list of work.
  - Heads down shared queue
    - User is assigned a task to work next
  - Automatic
    - Performed by a machine
- Information routing.
- Work tracking and analysis
- Capable of automating business processes that span all existing systems.
- Flexible work assignment rule
  - Support traditional role-based authentication model.

        - Customizable user-profile allows for the objects that control the assignments to be modified by the developer. (The assignment rule validation logic can be overridden or augmented with custom validation code.).
            - e.g. this transaction must be approved by the bank officer at this location who has sufficient sign-off authority and who is not on vocation

Implementation

- Distributed workflow engines.
    - Every engine has a backup engine that will automatically take over activity coordination if the primary engine fails.
    - Centralized console to monitor and control all process engines.
    - Each workflow engine can also have a local console.
- Use Forte's publish and subscribe event model for work list notification via asynchronous events. (Reduce performance problem created by client polling.)
- All active flow states are cached in memory. (Reduce database intensive I/O.)
- API supports for access to the process engine
    - Forte API for access from Forte applications
    - ActiveX API
    - C++ API for access from any C++ program
    - Web API for access from standard HTML forms via HTTP
    - Java API for access from standard HTML forms via HTTP
    - CORBA API for access from any application via CORBA
- The user profile can call existing security systems or directory services, allowing the security system to handle all the validation logic for Conductor.
- Records all process instance information in a RDBMS.

Claimed Interoperability

- Supports all major RDBMSs (Oracle, SQL Server, Sybase, Informix, etc.)
- Support ODBC.

Remarks:

Forte has a solid product the offers good performance and reliability.
Forte Conductor supports a simple process model.

3. FlowMark by IBM
http://www.software.ibm.com/ad/flowmark/

Entities

- People
  - Include Role, Organization, Authorization (the function and process categories the person is authorized to use), Substitute (substitute person when this person is absent)
- Role
- Organization
- Level
- A process diagram is a directed graph – that is, a representation of the logic of your process composed of nodes (activities, blocks, subprocesses, and source and sink data containers) and directional connectors (control connectors and data connectors).
- Activity
  - Include input data, output data, and support tools (additional programs that end users can start from their work list window to help complete an activity
- Process category

Claimed Functionality

- Process definition
- Simulation
- Execution
  - User logon to FlowMark runtime by specifying user ID, password, database name, and server name
  - Provides the responsible person with the relevant activity and the necessary data at the right time in the process
  - The FlowMark Work Lists folder contains your work lists for the databases that you are currently logged on to.
  - You can also create additional work lists according to the process categories.

Implementation

- Part of IBM MQSeries product family
- FlowMark components
  - Graphical process definition; staff, program and data registration, verification and animation (simulation).
  - Process execution and navigation, process monitor, work list management.
  - Audit trail, import/export.
  - API, integration building blocks, program invocation.
- TCP/IP

Claimed Interoperability

- Unknown

4. InConcert by InConcert, Inc.
http://www.inconcertsw.com

Entities

- A predefined process definition describes the basic steps that occur in a particular type of process, the order of steps and who performs each one.
- A task may consist of a hierarchy of subtasks with dependencies on the tasks of others.
- A task has attributes which are user-defined or built-in name-value pairs, such as priority, start and finish date, due date, costs, etc.
- A task may be in one of several states, including waiting, ready, bypassed, activated/acquired, in-process, or done.
- All users in the same pool are eligible to play the same role.

Claimed Functionality

- Support for structured and ad hoc processes which can be modified on the fly when they are still active
- Standard template
- Attributes can trigger process branches, send email, launch an application, or start a new workflow
- Agents can be set to perform tasks automatically
- Manage the documents associated with the tasks

Implementation

- 3-tier architecture
  - client (including web-based)
  - server
  - RDBMS (Oracle, Sybase, Informix, MS SQL Server)
- Communication between client and server: RPC
- Communication between server and DBMS: networked SQL
- Object-oriented API

Claimed Interoperability

- CACI Product Company's SIMPROCESS, an enterprise modeling, analysis and simulation tool
- Microsoft Project for Windows 95
- Documentum's Docbase document management repository

Remark

A simple process model with role-based task assignment.

REFERENCES

ActionWorks Metro 4.0 by Action Technologies, Inc., URL: http://www.actiontech.com

Forte' Application Environment by Forte', URL: http://www.Forte.com

FlowMark by IBM, URL: http://www.software.ibm.com/ad/flowmark

InConcert by InConcert, Inc., http://www.inconcertsw.com

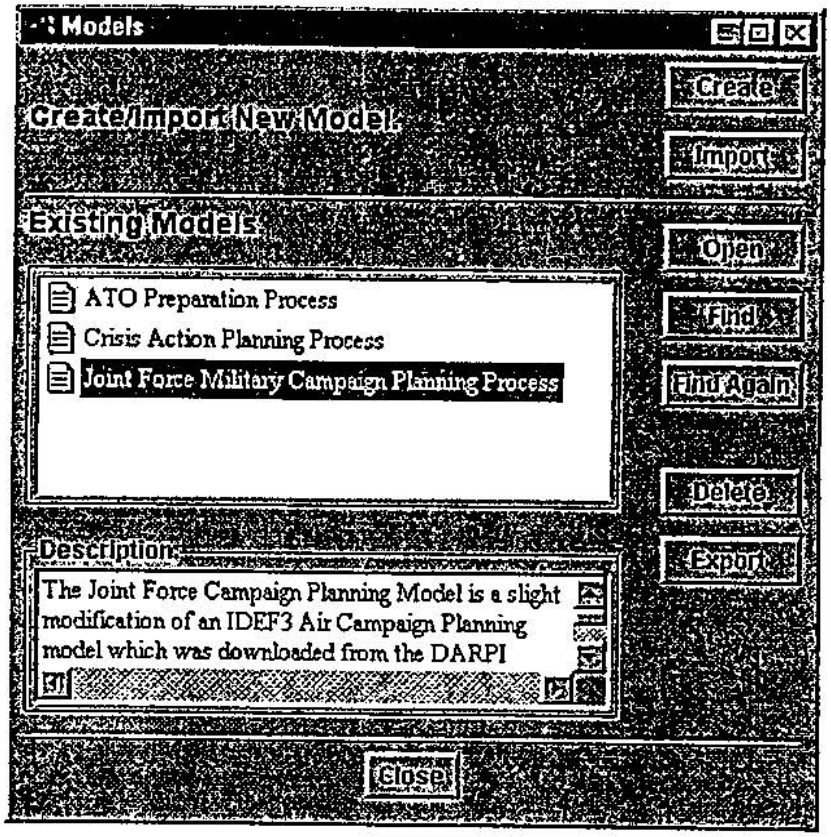
APPENDIX B
Process Design and Analysis Tool Suite (Built on Top of ProcessEdgeTM)

This Appendix presents the functionality of IPM's Process Design and Analysis tool suite which is built on top of ProcessEdge™ **.** The current prototype offers:

- IPM Asset Library
- Process Definition
- Activity Attributes and Data Assignment
- Process Verification
- Process Visualization
- Process Analysis
- Process Composition
- Process Composition
- Process Mismatch Analysis
- Process Reporting

IPM Asset Library

The IPM Asset Library (Figure B1) is used to create, persistently store, and retrieve component process models. The IPM library offers a notepad feature for the user to provide qualitative information about each process model. This GUI is also used for creating new process models, importing process models from and Excel template for ISTI's ProcessScript, and exporting models to ProcessScript which can be used to interface with third party products (e.g., simulation engines such as Lanner Group's Witness)



ATO Preparation Process

Crisis Action Planning Process

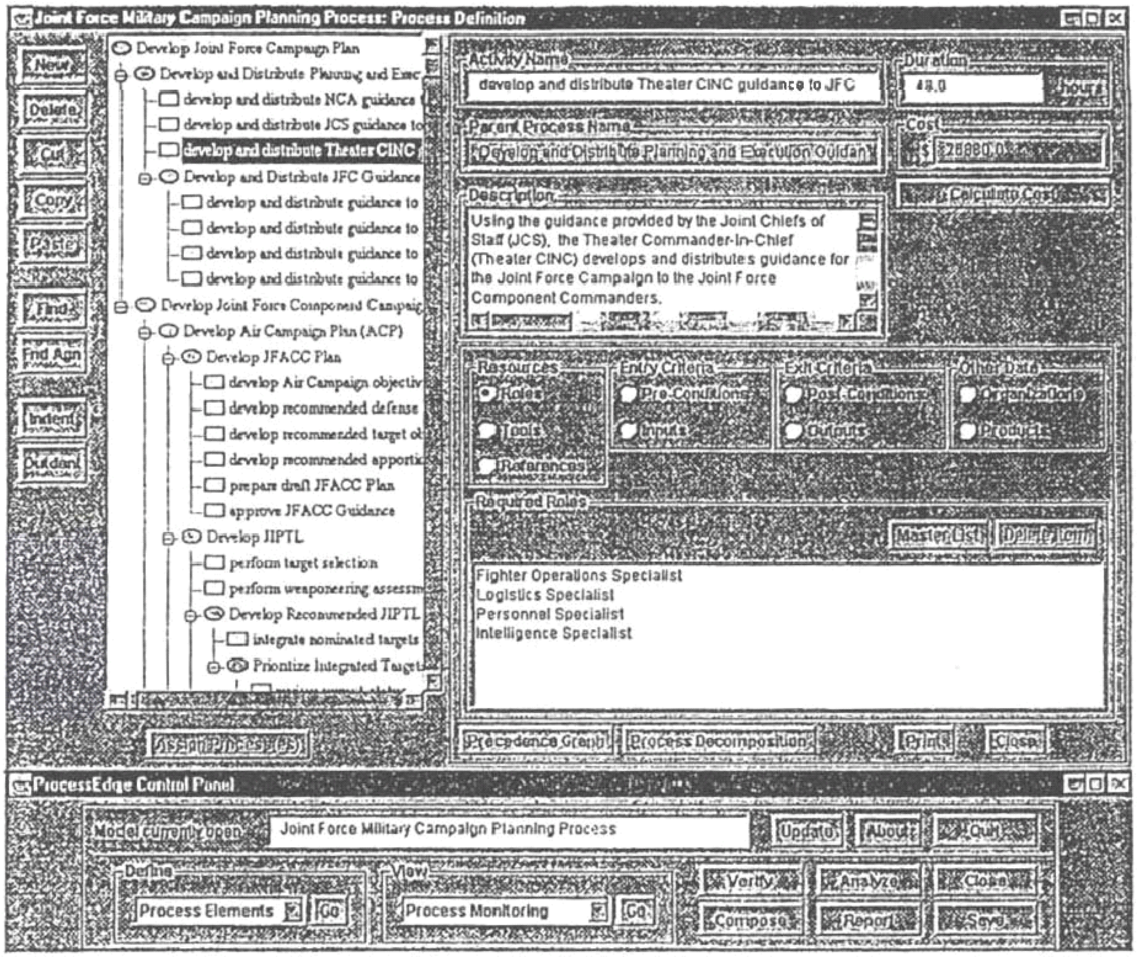
Joint Force Military Campaign Planning Process

The Joint Force Campaign Planning Model is a slight modification of an IDEF3 Air Campaign Planning model which was downloaded from the DARPI

Figure B1: Process Asset Library

Process Definition

This form based interface (Figure B2) is used to define process models. Process models can be hierarchically decomposed into subprocesses, down to activities. For each activity, the user can assign pertinent attributes such as Roles, Tools, Reference Material, Pre-Conditions/Post-Conditions, Inputs/Outputs, Organizations, and Products. In addition, the user can assign Duration and Cost values for each activity for simulation/analysis purposes,



Joint Force Military Campaign Planning Process: Process Definition

- Develop Joint Force Campaign Plan
  - Develop and Distribute Planning and Exec
    - develop and distribute NCA guidance
    - develop and distribute JCS guidance to
    - develop and distribute Theater CINC
  - Develop and Distribute JFC Guidance
    - develop and distribute guidance to
    - develop and distribute guidance to
    - develop and distribute guidance to
    - develop and distribute guidance to
- Develop Joint Force Component Campaig
  - Develop Air Campaign Plan (ACP)
    - Develop JFACC Plan
      - develop Air Campaign objectiv
      - develop recommended defense
      - develop recommended target ob
      - develop recommended apportio
      - prepare draft JFACC Plan
      - approve JFACC Guidance
    - Develop JIPTL
      - perform target selection
      - perform weaponeering assessm
      - Develop Recommended JIPTL
        - integrate nominated targets
        - Prioritize Integrated Target

Activity Name: develop and distribute Theater CINC guidance to JFC

Duration: 48.0 hours

Parent Process Name: Develop and Distribute Planning and Execution Guidan

Cost: $ 28880.0

Description: Using the guidance provided by the Joint Chiefs of Staff (JCS), the Theater Commander-In-Chief (Theater CINC) develops and distributes guidance for the Joint Force Campaign to the Joint Force Component Commanders.

Resources: ● Roles ○ Tools ○ References

Required Roles:

Fighter Operations Specialist
Logistics Specialist
Personnel Specialist
Intelligence Specialist

ProcessEdge Control Panel

Model currently open: Joint Force Military Campaign Planning Process

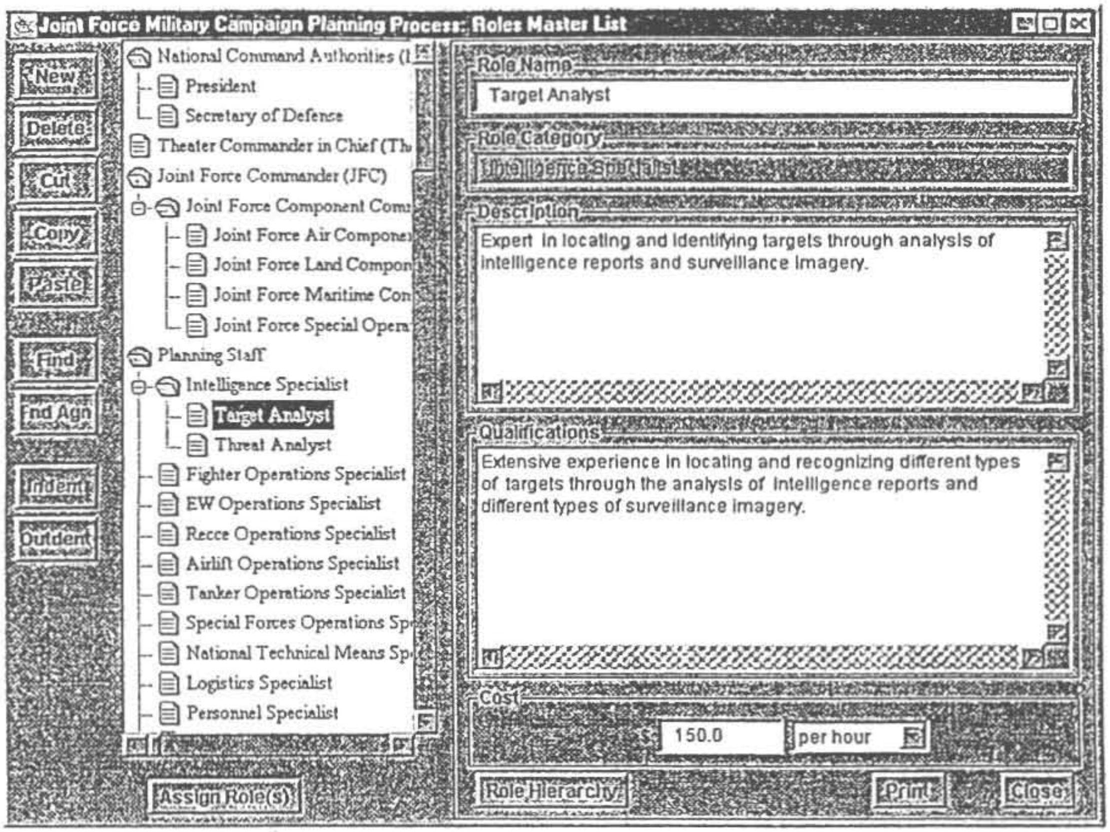
Define: Process Elements

View: Process Monitoring

Figure B2: Process Definition

Activity Attributes and Data Assignment

Master lists are used to enter attribute data for each activity. Figure B3 shows the Master Lists for Roles, Master Lists for other attributes are similar to this GUI in both their presentation and functionality. Attributes are entered and then assigned (i.e., related) to each activity by highlighting the attribute and clicking the Assign button.



Joint Force Military Campaign Planning Process: Roles Master List
New
Delete
Cut
Copy
Paste
Find
Fnd Agn
Indent
Outdent
National Command Authorities (l
President
Secretary of Defense
Theater Commander in Chief (Th
Joint Force Commander (JFC)
Joint Force Component Com
Joint Force Air Compone
Joint Force Land Compon
Joint Force Maritime Con
Joint Force Special Opera
Planning Staff
Intelligence Specialist
Target Analyst
Threat Analyst
Fighter Operations Specialist
EW Operations Specialist
Recce Operations Specialist
Airlift Operations Specialist
Tanker Operations Specialist
Special Forces Operations Sp
National Technical Means Sp
Logistics Specialist
Personnel Specialist
Assign Role(s)
Role Hierarchy
Print
Close

Role Name: Target Analyst

Role Category: Intelligence Specialist

Description: Expert in locating and identifying targets through analysis of intelligence reports and surveillance imagery.

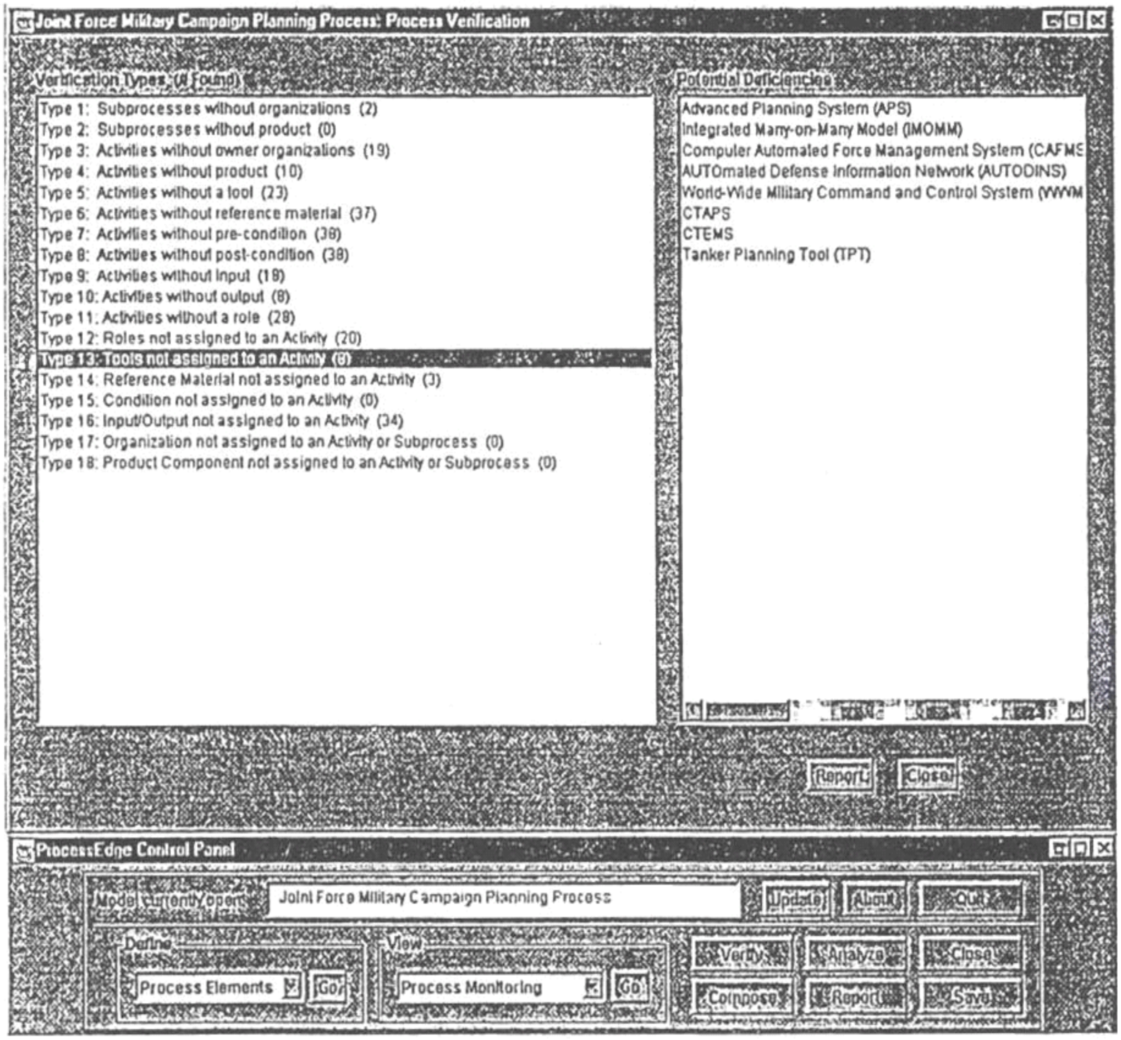
Qualifications: Extensive experience in locating and recognizing different types of targets through the analysis of intelligence reports and different types of surveillance imagery.

Cost: $ 150.0 per hour

Figure B3: Attribute and Data Assignment

Process Verification

Process Verification (Figure B4) is performed to verify the syntactic correctness (i.e., completeness, consistency, traceability) of a process model. This capability is designed to detect a set of errors that are frequently made by users when constructing a process. Examples of typical errors are activities without a resource (e.g., roles, tool, reference material), and activities without inputs. This capability serves a two-fold purpose. First, when users believe they have completed the process model, it provides them with a list of potential deficiencies. Second when users are in the middle of developing a model, it helps re-establish context whenever they have to stop and resume modeling (e.g., hours or days later). By clicking the Verify button in the Control Panel, information that has already been entered into the model and what still needs to be entered is displayed



Joint Force Military Campaign Planning Process: Process Verification

Verification Types: (# found)

Type 1: Subprocesses without organizations (2)
Type 2: Subprocesses without product (0)
Type 3: Activities without owner organizations (19)
Type 4: Activities without product (10)
Type 5: Activities without a tool (23)
Type 6: Activities without reference material (37)
Type 7: Activities without pre-condition (38)
Type 8: Activities without post-condition (38)
Type 9: Activities without input (18)
Type 10: Activities without output (8)
Type 11: Activities without a role (28)
Type 12: Roles not assigned to an Activity (20)
Type 13: Tools not assigned to an Activity (8)
Type 14: Reference Material not assigned to an Activity (3)
Type 15: Condition not assigned to an Activity (0)
Type 16: Input/Output not assigned to an Activity (34)
Type 17: Organization not assigned to an Activity or Subprocess (0)
Type 18: Product Component not assigned to an Activity or Subprocess (0)

Potential Deficiencies

Advanced Planning System (APS)
Integrated Many-on-Many Model (IMOMM)
Computer Automated Force Management System (CAFMS
AUTOmated Defense Information Network (AUTODINS)
World-Wide Military Command and Control System (WWM
CTAPS
CTEMS
Tanker Planning Tool (TPT)

ProcessEdge Control Panel

Model currently open: Joint Force Military Campaign Planning Process

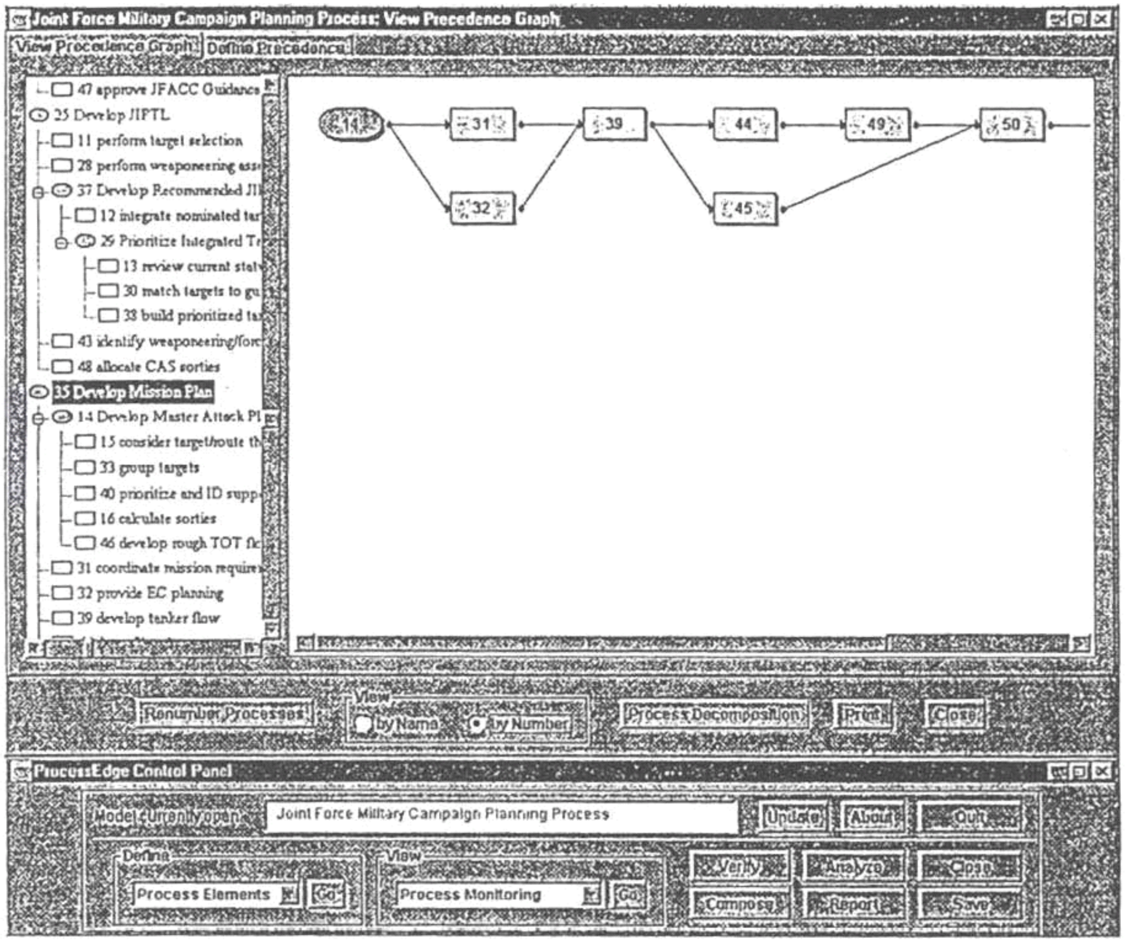
Define: Process Elements

View: Process Monitoring

Figure B4: Process Verification

Process Visualization

This capability allows the user to view the process from different complementary perspectives. The activity dependency graph in Figure B5 shows the causal relationships, at various levels in the hierarchy. The Process Decomposition hierarchy in Figure B5 shows the work break down structure at multiple levels of abstraction. These views help collaborative teams in communicating effectively and resolving differences arising from different "mental models" of the process.



Joint Force Military Campaign Planning Process: View Precedence Graph

- □ 47 approve JFACC Guidance
- ○ 25 Develop JIPTL
  - □ 11 perform target selection
  - □ 28 perform weaponeering ass
  - ○ 37 Develop Recommended JI
    - □ 12 integrate nominated tar
    - ○ 29 Prioritize Integrated Tr
      - □ 13 review current stat
      - □ 30 match targets to gu
      - □ 38 build prioritized ta
  - □ 43 identify weaponeering/forc
  - □ 48 allocate CAS sorties
- ○ **35 Develop Mission Plan**
  - ○ 14 Develop Master Attack Pl
    - □ 15 consider target/route th
    - □ 33 group targets
    - □ 40 prioritize and ID supp
    - □ 16 calculate sorties
    - □ 46 develop rough TOT fl
  - □ 31 coordinate mission require
  - □ 32 provide EC planning
  - □ 39 develop tanker flow

View: ○ by Name ● by Number

ProcessEdge Control Panel

Model Currently open: Joint Force Military Campaign Planning Process

Define: Process Elements

View: Process Monitoring

Figure B5: Activity Dependency Graph

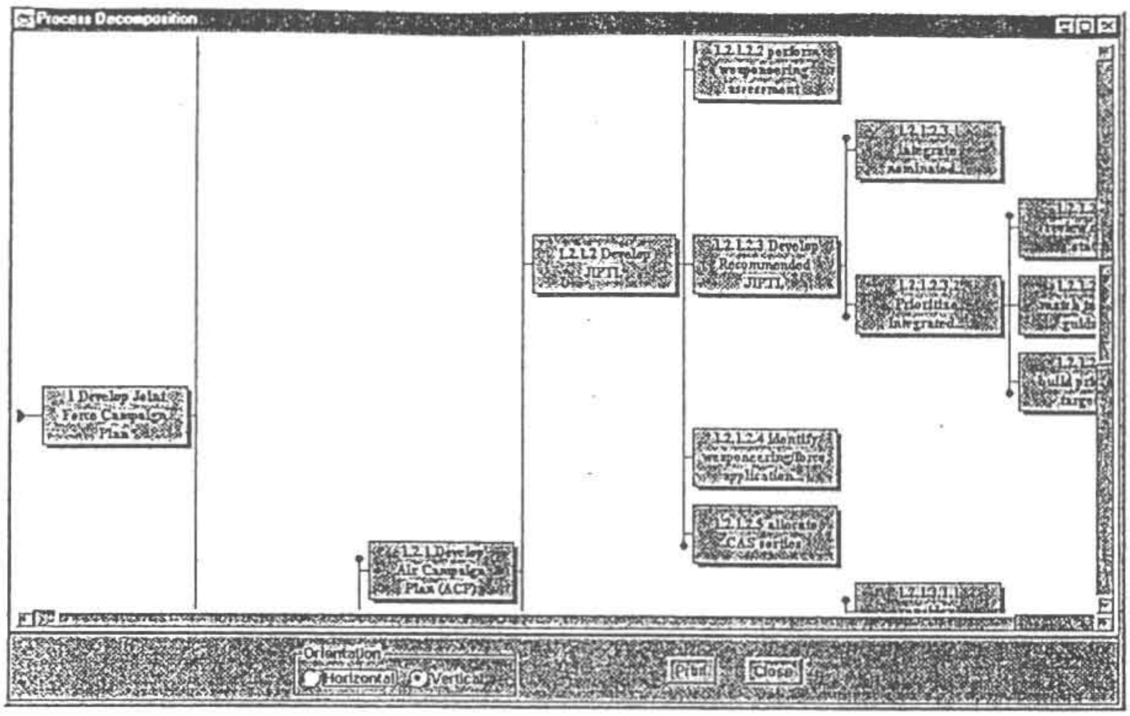
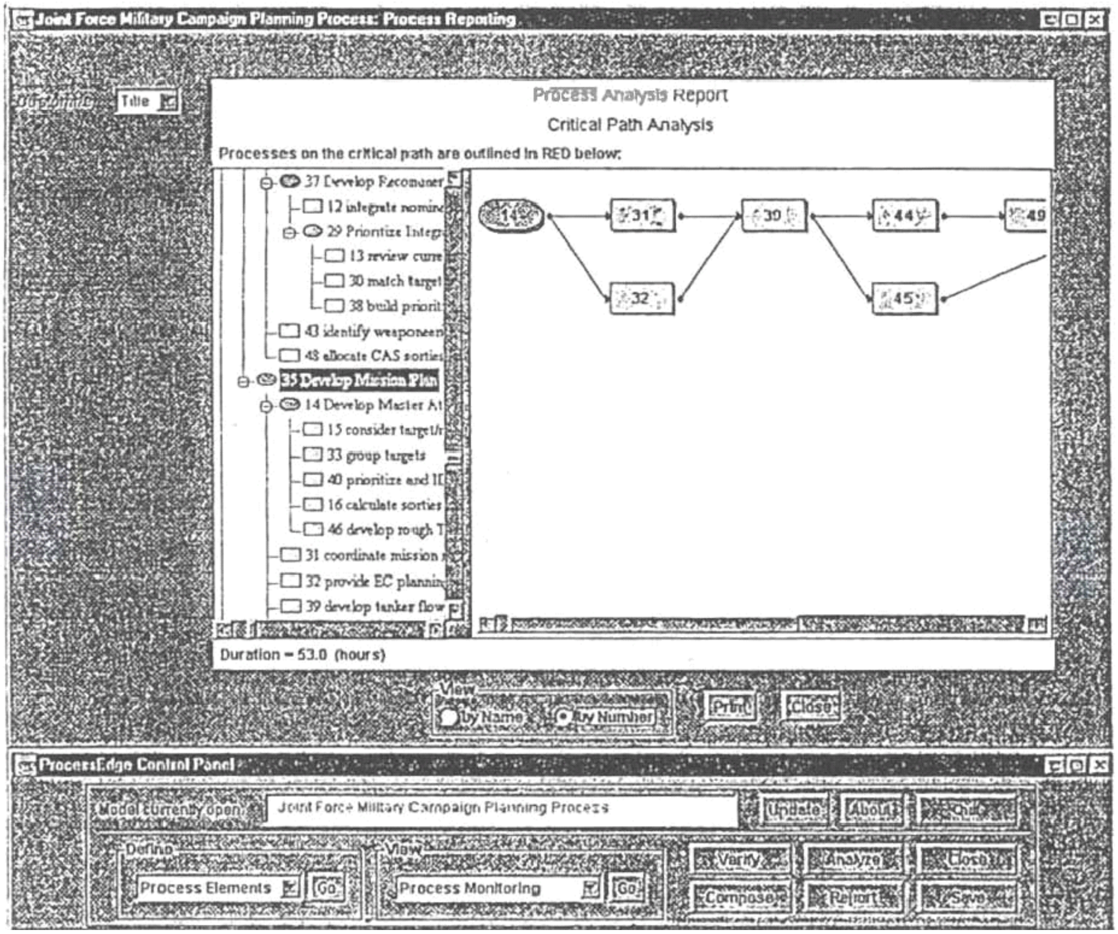

Process Decomposition
1 Develop Joint Force Campaign Plan
1.2.1 Develop Air Campaign Plan (ACP)
1.2.1.2 Develop JIPTL
1.2.1.2.2 perform weaponeering assessment
1.2.1.2.3 Develop Recommended JIPTL
1.2.1.2.4 Identify weaponeering/force application...
1.2.1.2.5 allocate CAS sorties
1.2.1.2.3.1 Integrate nominated...
1.2.1.2.3.2 Prioritize integrated...
Orientation
Horizontal
Vertical
Print
Close

Figure B6: Process Decomposition

Process Analysis

Process Analysis provides the capability to both conduct critical tasks and dependencies between the path analysis and process-based cost analysis. Critical Path Analysis is a common approach for evaluating process/workflow(Figure B7).When there are n tasks and dependencies between the tasks, the question that must be answered is “What set of tasks determine the minimum (i.e., earliest completion time) for the whole process?” This is the critical path.

Process-based cost analysis (Figure B8) displays the cost attributed to each activity in the process model. Cost roll-up is supported from the activity level to the subprocess and process level.



Joint Force Military Campaign Planning Process: Process Reporting

Process Analysis Report

Critical Path Analysis

Processes on the critical path are outlined in RED below:

- 37 Develop Reconnaer
  - 12 integrate nomin
  - 29 Prioritize Integr
    - 13 review curre
    - 30 match target
    - 38 build priorit
  - 43 identify weaponeer
  - 48 allocate CAS sortie
- 35 Develop Mission Plan
  - 14 Develop Master At
    - 15 consider target/r
    - 33 group targets
    - 40 prioritize and II
    - 16 calculate sorties
    - 46 develop rough T
  - 31 coordinate mission
  - 32 provide EC plannin
  - 39 develop tanker flow

Duration = 53.0 (hours)

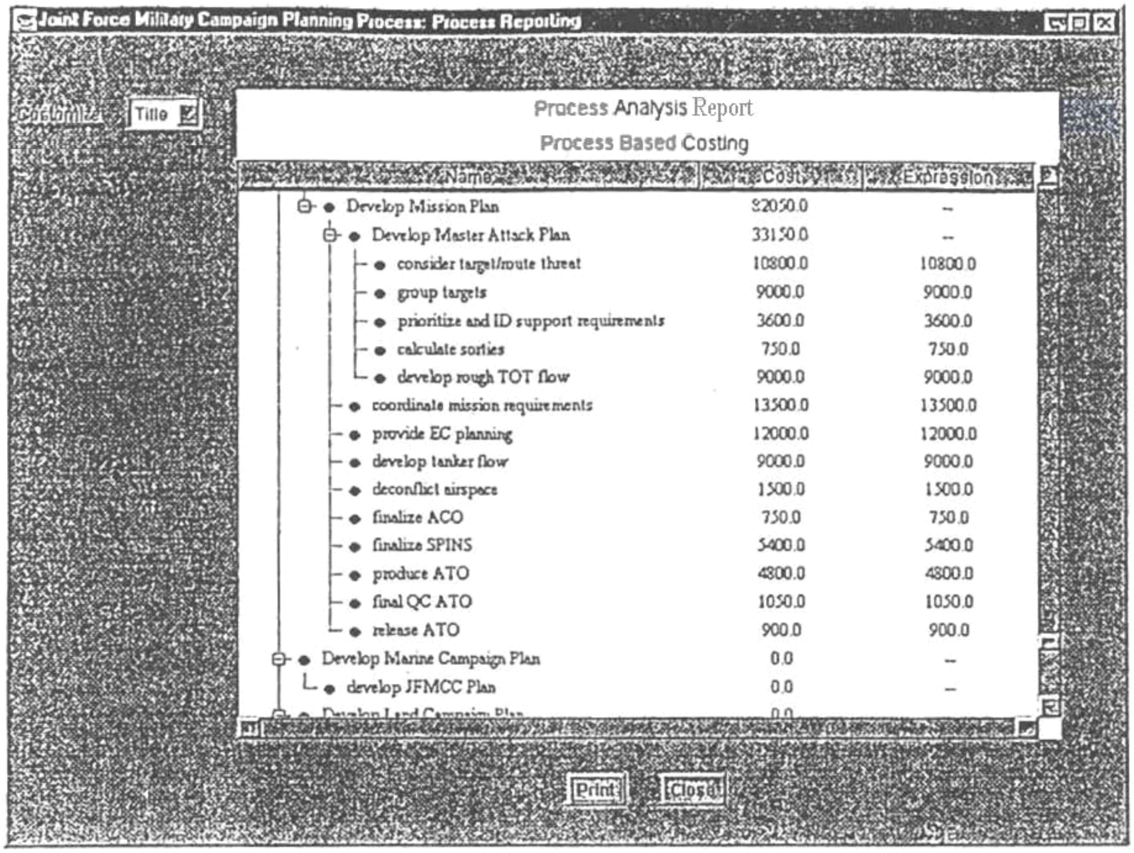
ProcessEdge Control Panel

Model currently open: Joint Force Military Campaign Planning Process

Define: Process Elements

View: Process Monitoring

Figure B7: Critical Path Analysis Report


Joint Force Military Campaign Planning Process: Process Reporting
Customize
Title ☑
Print
Close

Process Analysis Report

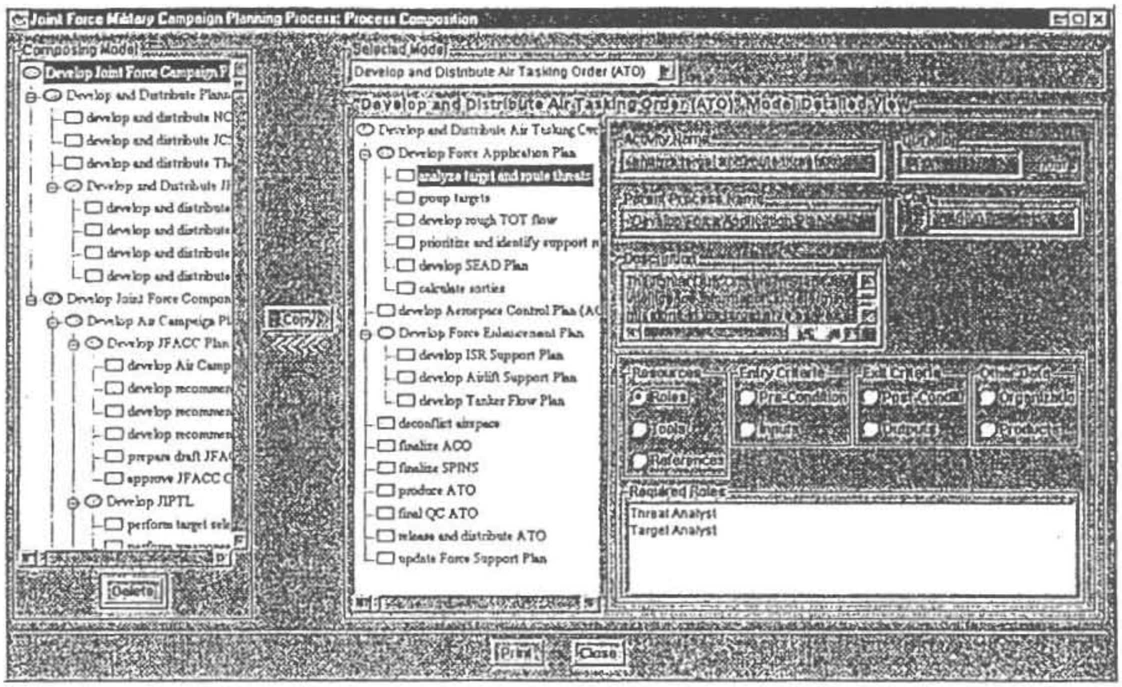
Process Based Costing

Name	Cost	Expression
Develop Mission Plan	82050.0	--
Develop Master Attack Plan	33150.0	--
consider target/route threat	10800.0	10800.0
group targets	9000.0	9000.0
prioritize and ID support requirements	3600.0	3600.0
calculate sorties	750.0	750.0
develop rough TOT flow	9000.0	9000.0
coordinate mission requirements	13500.0	13500.0
provide EC planning	12000.0	12000.0
develop tanker flow	9000.0	9000.0
deconflict airspace	1500.0	1500.0
finalize ACO	750.0	750.0
finalize SPINS	5400.0	5400.0
produce ATO	4800.0	4800.0
final QC ATO	1050.0	1050.0
release ATO	900.0	900.0
Develop Marine Campaign Plan	0.0	--
develop JFMCC Plan	0.0	--
Develop Land Campaign Plan	0.0	

Figure B8: Process-Based Costing Report

Process Composition

Process Composition (Figure B9) is performed by using reusable process component that are stored in the Process Asset Library. This is accomplished by viewing, selecting, and copying desired process components into the composing model. In doing so, all the information attributes of the process components are also transferred to the composing model.



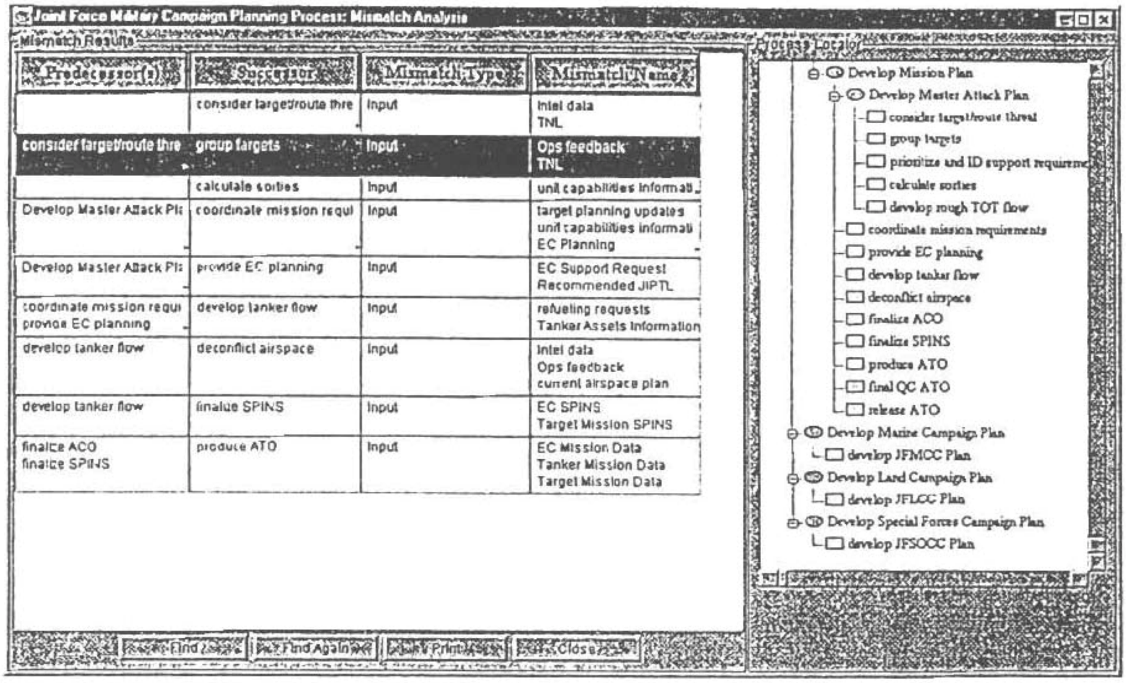
Selected Model: Develop and Distribute Air Tasking Order (ATO)

Required Roles:
Threat Analyst
Target Analyst

Figure B9: Process Composition

Mismatch Analysis

When performing Process Composition, there can be several types of mismatches between the composed processes, e.g., data mismatches between the outputs and inputs and pre-conditions and post-conditions. Figure B10 displays the results of mismatch analysis and identifies source of mismatches.

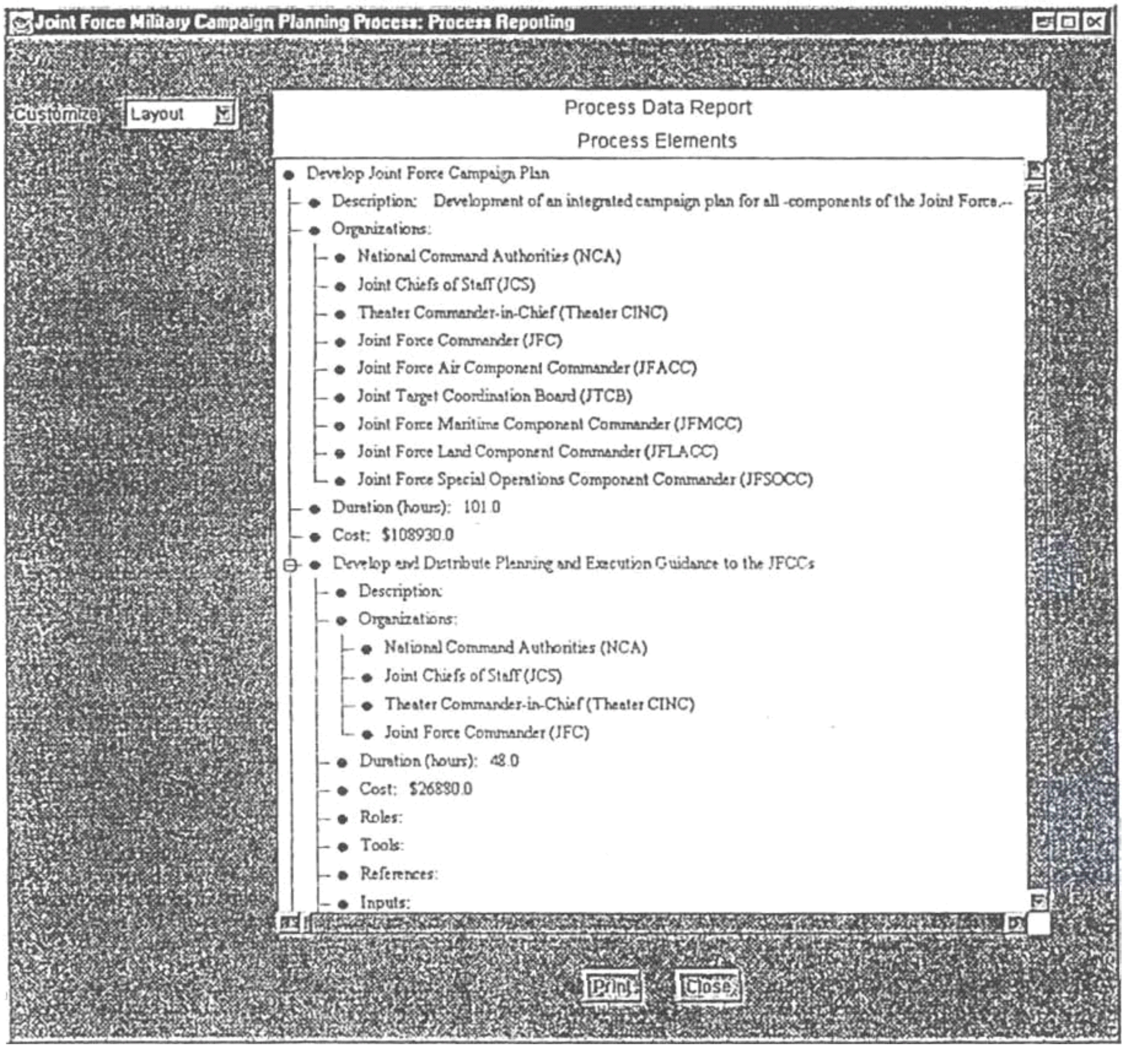


Predecessor(s)	Successor	Mismatch Type	Mismatch Name
	consider target/route thre	Input	Intel data TNL
consider target/route thre	group targets	Input	Ops feedback TNL
	calculate sorties	Input	unit capabilities informati
Develop Master Attack Pl:	coordinate mission requi	Input	target planning updates unit capabilities informati EC Planning
Develop Master Attack Pl:	provide EC planning	Input	EC Support Request Recommended JIPTL
coordinate mission requi provide EC planning	develop tanker flow	Input	refueling requests Tanker Assets Information
develop tanker flow	deconflict airspace	Input	Intel data Ops feedback current airspace plan
develop tanker flow	finalize SPINS	Input	EC SPINS Target Mission SPINS
finalize ACO finalize SPINS	produce ATO	Input	EC Mission Data Tanker Mission Data Target Mission Data

Figure B10: Mismatch Analysis

Process Reporting

Process Reporting allows the user to customize views into the process model to review, examine, verify, understand, and utilize the information contained within the model. The Process Reporting module obtains data from the Data Repository and formats the information far display based on user inputs. The reporting function generates raw data or a structured narrative of data or the following: (a)Model Components(e.g., Activities, Roles, Tools, Inputs/Outputs); (b)Model Verification; (c)Critical Path Analysis; (d) Process-based Cost Analysis. Figure BI 1shows a sample of a report on JFACC's Process Components.



Joint Force Military Campaign Planning Process: Process Reporting
Customize
Layout
Print
Close

Process Data Report

Process Elements

```
● Develop Joint Force Campaign Plan
├─ ● Description:   Development of an integrated campaign plan for all -components of the Joint Force.--
├─ ● Organizations:
│  ├─ ● National Command Authorities (NCA)
│  ├─ ● Joint Chiefs of Staff (JCS)
│  ├─ ● Theater Commander-in-Chief (Theater CINC)
│  ├─ ● Joint Force Commander (JFC)
│  ├─ ● Joint Force Air Component Commander (JFACC)
│  ├─ ● Joint Target Coordination Board (JTCB)
│  ├─ ● Joint Force Maritime Component Commander (JFMCC)
│  ├─ ● Joint Force Land Component Commander (JFLACC)
│  └─ ● Joint Force Special Operations Component Commander (JFSOCC)
├─ ● Duration (hours):  101.0
├─ ● Cost:  $108930.0
⊖─ ● Develop and Distribute Planning and Execution Guidance to the JFCCs
   ├─ ● Description:
   ├─ ● Organizations:
   │  ├─ ● National Command Authorities (NCA)
   │  ├─ ● Joint Chiefs of Staff (JCS)
   │  ├─ ● Theater Commander-in-Chief (Theater CINC)
   │  └─ ● Joint Force Commander (JFC)
   ├─ ● Duration (hours):  48.0
   ├─ ● Cost:  $26880.0
   ├─ ● Roles:
   ├─ ● Tools:
   ├─ ● References:
   ├─ ● Inputs:
```

Figure B11: Process Reporting

APPENDIX C
Glossary of Terms

Agent - Execution
Adapter to any software tool that might be invoked by the workflow execution engine, e.g. adapter for an automatic controller of a mill.

Agent - Event Monitor
Adapter to environmental sensors, e.g. adapter for a temperature sensor of a reactor.

Agent - Human Planner
Planning assistant that helps the human planners in planning, execution monitoring, and replanning.

Automated Planner
Any third party tool that can generate plans automatically.

Automatic Plan Adjustment
A component that is invoked by the coordinator to facilitate automatic plan generation/modification.

Collaborative Dynamic Replanning
A component that is invoked by the coordinator to facilitate collaborative plan generation/modification.

Coordinator
The component that is responsible for orchestrating all components in the Intelligent Controller. It constantly monitors the events posted on the whiteboard, uses the data in the State Vector, Execution Log, and Rule Base to make planning decisions. It relies on third-party components to generate a new plan and uses that information to instruct the Scheduler to modify the schedule.

Event Monitor
A "listener" for internal and external events that is also responsible for classification of events and posting them on the whiteboard.

Execution
Distributed workflow enactment engine.

Execution Log
An audit log that maintains the execution history.

Interactive Component Plan Substitution
A component that is invoked by the coordinator to facilitate interactive plan generation/modification.

Process Library
A repository (database) of component processes/partial plans that can be reused.

Scheduler
A component that is responsible for scheduling the plan.

State Vector
A database that maintains the current status of the executing processes.

Whiteboard
A shared bulletin board that supports multiple agents.

Rule Base
A set of decision rules that support the coordination and replanning function.