

## **Activity Representation and Management for Crisis Action Planning**

**James L. Jacobs, Michael C. P. Dorneich, and Patricia M. Jones**  
**Department of Mechanical & Industrial Engineering**  
**University of Illinois at Urbana-Champaign**

### **ABSTRACT**

This work presents the development of a multi-user virtual environment in support of crisis management planning activities. Coupling on-going work in activity representation with analysis of the crisis action planning domain, an extensible domain ontology was developed. Studying the domain, several challenges became apparent: (1) the need to support extensive dynamic and distributed collaboration, (2) the need for a flexible, open architecture, (3) the need for views of the information tailored to the activities of the planning team, and (4) the opportunity to leverage past and current related efforts in the domain.

This project developed a Java-based collaboration architecture around a multi-user domain (MUD) to provide presence and access to collaborative services. The collaborative infrastructure layer provides persistence, user authentication, and access control. Built upon this substrate are collaborative services such as a whiteboard mechanism, chat functions, and the domain-specific Mission Analysis Support Tool (MAST). The target use of MAST is to represent and track workflow in the crisis action planning done by the Operations Planning Team of the U.S. Pacific Command.

The design approach was to base MAST on a reusable class library which implements a rich ontology structure and represents key elements of the domain. Our ontology draws on the SPOT project (Jacobs, et. al. 1997) and the Shared Planning and Activity Representation (SPAR) project. The resulting ontology, implemented in Java, is compliant with the SPAR Reference Object Model Specification. A strength of this work in the extensibility, flexibility and openness both of the representation and the system architecture.

### **1. INTRODUCTION**

Cognitive systems engineering studies human problem solving with cognitive tools (Woods and Roth, 1988). In this paper, we focus on modeling collaborative planning among practitioners and in supporting more effective collaborative planning with advanced information technologies. The domain of practice is military crisis action planning, particularly for humanitarian assistance operations.

The rest of this paper is organized as follows. Section 2 provides more detail on crisis action planning, and Section 3 briefly describes the use case developed and used for our proof-of-concept technology demonstration. Sections 4 through 6 describe this technology demonstration: Section 4 focuses on the object representations, Section 5 on the collaborative infrastructure ("MUDSPOT"), and Section 6 on the crisis action planning tool MAST (Mission Analysis and Support Tool).

### **2. CRISIS ACTION PLANNING**

Crisis action planning refers to a large distributed planning process to cope with crises. Our particular context is humanitarian assistance/disaster relief (HA/DR) operations coordinated by the Operations Planning Team in the Pacific Command. From the time a crisis is detected, a response team is assembled, and a complex planning process is begun to derive a carefully coordinated set of plans and execution orders. The crisis action planning process consists of six stages: (1) Situation development, (2) Crisis assessment, (3) Course of action generation, (4) Course of action selection, (5) Execution planning, and (6) Execution (Jones, 1996).

The coordination of the various elements of this process is facilitated by an individual labeled the OPT Coordinator. This individual is responsible for overseeing the planning process within the OPT. He tracks the progress of each element of the evolving planning document set, the writing assignments, the flows of information into and out of the OPT, and the due dates for the various documents that comprise the emerging plan.

### **3. DOMAIN ANALYSIS AND USE CASE FOR DESIGN**

Some aspects of this domain closely resemble project management activities, while others have a strong workflow management quality. The scenario that was the result of our analysis and the guiding use-case for design was the following:

(1) Operations Planning Team logs into the system. Team consists of minimally the usual Joint staff positions (see [http://www.dtic.mil/jcs/text/joint\\_staff\\_org.html](http://www.dtic.mil/jcs/text/joint_staff_org.html)) J1 (administrative), J2 (intelligence), J3 (operations), J4 (logistics), J5 (strategic plans and policy), J6 (command, control and communications) and the OPT Coordinator. Additional members may be present.

(2) OPT Coordinator creates a new instance of a "Workflow Tracking Object" (WTO) (e.g., for a Warning Order or Operations Order).

[This WTO would have at minimum name and components in which each component has a name, position-assigned-to do this component, status flag, due date (DDD/HH/MM), and pointer to the finished product (e.g., name and path of document on hard drive or Web).]

(3) Team develops plan and OPT Coordinator takes initial notes on plan.

(4) Assignments are made and due dates set by OPT Coordinator.

(5) Team members work in parallel.

(6) OPT Coordinator can check status of WTO to see what's been done and what is still needed to be done.

(7) When team members finish their parts, the WTO gets updated with a new status and pointer.

(8) If product is nearly due or late, the assigned-to person gets reminded.

In conceiving of crisis action planning as a form of project management, we drew on our own previous work on the SPOT project (Jacobs et al., 1997) as well as emerging representations of activity and planning objects. The evolution of our object design is recounted in Section 4.

We also had to cope with issues in the design of collaborative infrastructure. We needed to develop underlying computational mechanisms for persistent collaboration of shared data both in synchronous and asynchronous modes of operation. We based our design on Java MUD (Farley, 1998; also see Jones, 1997) and implemented the MUDSPOT architecture, which is described in Section 5.

Our fundamental hypothesis was that the representation of core domain concepts would provide a powerful organizing element around which to provide tools for planning and coordination. By providing a means to view and manipulate domain models of workflow and the resulting work products (the plan documents themselves), the envisioned tool would assist the OPT Coordinator to manage the workflow associated with the planning process and track the production of the resulting plan documents. This tool is MAST, the Mission Analysis and Support Tool, and is described in Section 6.

### **4. SPOT and SPAR**

In the beginning, there was SPOT, the set of Java objects from our previous work with NCSA on lightweight Java-based project management tools (Jacobs et al., 1997; also see <http://tortie.me.uiuc.edu/spot/>). Of course, there are many research projects on planning and activity management and enterprise coordination (see Jones and Jasek, 1997). We used this crisis action planning project as a means to explore other work and extend and enrich our original SPOT representations.

We iterated several times on object design. Our first redesign relied on the University of Edinburgh's Enterprise Ontology (Uschold et al, 1998; also see <http://www.aiai.ed.ac.uk/~enterprise/enterprise/ontology.html>).

The Enterprise Ontology was developed from studies of workflow in business applications in conjunction with the European Workflow Management Consortium. This work comprises a set of terms that are defined with reference to each other, with strictly defined semantics that are intended to be suitable for representation and reasoning. We restructured the SPOT object library and melded selected representations from the Enterprise Ontology to create a new version of the Spot library (later designated Spot-II A).  
SPAR: Spot-II B

Soon afterwards, we identified the Shared Plan and Activity Representation (SPAR) which is of considerably larger scope (Tate, 1998; also see <http://www.aiai.ed.ac.uk/~arpi/spar/>). SPAR is a product of a very large set of efforts on unifying plan representations across all major threads of planning research. Its focus on providing essential commonality within a modular framework seemed ideal to this project, as well as its heritage of including planning researchers who were concerned with the domain of crisis planning. Though still in very early stages, a draft specification for a "Reference Object Model" had just become available. We used this draft to implement a Java version of the SPAR ROM and extended it to include SPOT objects. This new modular representation became known within the project as the "Spot-II B design". Additional models specific to the needs of the Mission Analysis Support Tool (MAST) were then added on top of these library packages in a separate "mast.lang" package within the evolving object library.

## 5. MUDSPOT

The intention behind building rich object models of activity was that tools based on such models would be available in a collaborative virtual environment such as a MOO or MUD (Jones, 1996, 1997). As a prototype of such an environment, we built MUDSPOT, a Java-based collaborative virtual environment based on

JavaMUD (Flanagan, 1997) and on Java distributed computing concepts such as Java Remote Method Invocation (RMI) (Farley, 1998).

In particular, we expanded Flanagan's JavaMUD base to create full-fledged objects and dramatically extended its core definitions to provide hooks for multiple renderings of objects. MUDSPOT object classes now include people, places and things with fully vectorized coordinate systems for size, position, velocity, and acceleration in addition to mass. We used Java's reflection mechanism to enable dynamic expansion of mediator-types, room-types, and object-types. The resulting system is a virtual environment that provides explicit support for task collaboration through task-tailored 'workrooms' that mediate interaction among collaborators.

MUDSPOT's collaborative infrastructure is organized around the notion of mediators that build upon updating methods (sending and receiving). The generic `MudSpotMediator` class can be specialized for a variety of purposes; for the MAST tool described in the next section, the `MastMediator` class adds clock management and synchronization capabilities.

The libraries that make up the MUDSPOT architecture are:

**spot.spar:** our Java implementation of the SPAR Model Specification

**spot.lang:** organizational modeling concepts

**spot.grammar:** syntactical definitions for SPAR-based objects

**spot.util:** utility objects needed by other SPOT objects

**spot.gui:** general-purpose graphical user interface objects

Application software is build "on top of" the SPOT libraries, as shown in Figure 1.

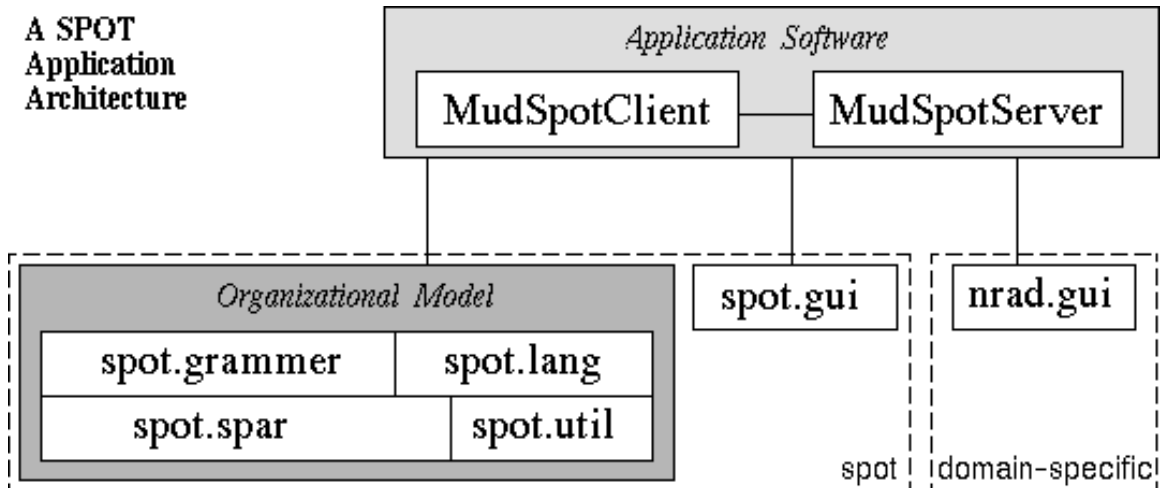


Figure 1. SPOT Application Server

Notice that the `spot.gui` library does not depend on the other libraries; these generic GUI objects provide views into SPOT library objects. An application built upon the SPOT libraries will build up GUI objects that are specific to the domain, needs, and perspectives of the application area. The MAST tool, described next, uses the `nrad.gui` objects shown in Figure 1.

## 6. MAST

MAST, the Mission Analysis Support Tool, is the tool we developed for the OPT Coordinator. We worked closely with the OPT Coordinator to develop the use-case scenario and engage in iterative participatory design of MAST capabilities and user interface. Interestingly, he was aware of SPAR before we were, and had already contributed comments to the SPAR working group. Thus, he was familiar enough

with object representation issues to give us very clear guidance on the needs of the underlying representations, the character of the graphical user interface, and also agreed with our focus on the flexibly ability to view and edit data structures.

The OPT Coordinator's vision of a graphical user interface was a "data cube" of several interrelated views of the planning process. This vision contributed heavily to the MAST design, and based on it we added Workflow and Task classes to `spot.lang` and created visual representations of planning levels and documents. The OPT Coordinator made it clear that a visual representation of the Plan Document space, (the levels, the document categories, and the individual document sections) was the most critical cognitive element, and to him, the most important visual element in the interface.

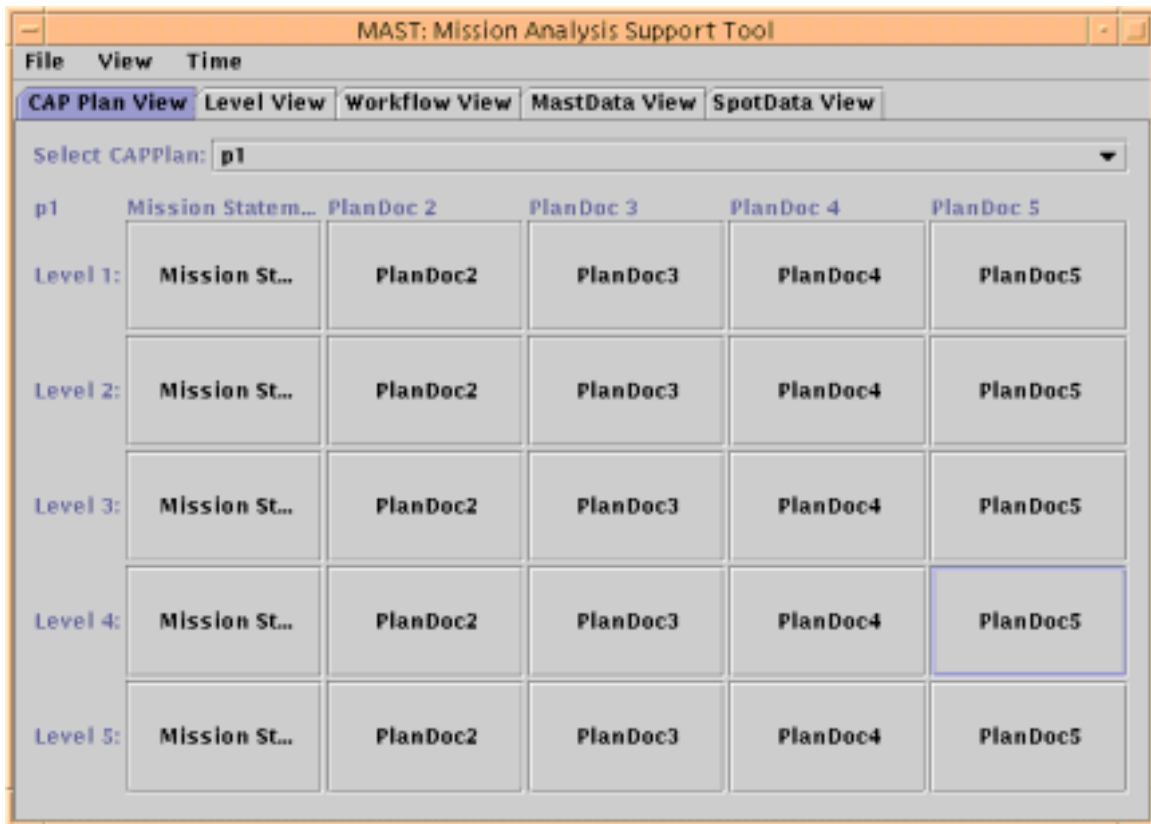


Figure 2. Generic user interface to MAST, showing the Crisis Action Plan view. Each level of planning has a mission statement and documents. Other views provide more detail at planning levels, show workflow relationships, and provide access to data.

Figure 2 shows a screendump of one of the initial MAST screens that represents one face of the “data cube”. The labels in the screendump are generic, but illustrate the layered view of planning at different echelons (e.g., National Command Authority, strategic, tactical) which each have associated mission statements and plan documents. Individual documents can be accessed from this screen, and the tabs at the top of the interface represent other faces of the “data cube” that provide related views of the data.

In fact, in MAST we conceived of two such cubes. Each of the elemental data pieces -- a particular plan document -- has workflow task associated with it. Each of these tasks has a performer, a due-time and a status. The data-cube is mirrored by a parallel workflow cube. Each of the cubes is represented in a tree-view, following the precedence of the cube dimensions from Level to Document to Document-Section. For the workflow cube-tree, each task label

displays its status and due-time along with the current time. Every domain object, either data or workflow, can be more closely examined and edited by clicking the associated label in the higher-level frame. For some domain objects, specific editing views were implemented. However, for more general objects in the mast.lang, spot.lang, and spar.lang packages, we built a few generic editors that rely of Java language’s reflection mechanism to perform introspection on the specific instance to determine how it should be displayed.

## ACKNOWLEDGMENTS

This research was supported by the Navy Space and Warfare Center (SPAWAR), formerly the Naval Command, Control, and Ocean Surveillance Center (NCCOSC), under contract number N66001-97-M-2077.

## REFERENCES

- Farley, J. (1998). *Java distributed computing*. O'Reilly & Associates, Inc.
- Flanagan, D. (1997). *Java examples in a nutshell: A tutorial companion to Java in a Nutshell*. O'Reilly & Associates, Inc.
- Jacobs, J. L., Dorneich, M. C., Jones, P. M., O'Keefe, B. J. and Contractor, N. (1997). SPOT: Using collaborative technologies for developing collaborative technologies. *Proceedings of the 1997 IEEE International Conference on Systems, Man, and Cybernetics*, Volume 1, pp. 690-695. Orlando FL, October 1997.
- Jones, P. M. (1997). Towards a collaborative virtual environment for distributed crisis action planning: Communicative and representational functions of a MOO infrastructure. HCCPS Technical Report 9701, Human-Computer Cooperative Problem Solving Laboratory, Department of Mechanical and Industrial Engineering, University of Illinois at Urbana-Champaign.
- Jones, P. M. (1996). Collaborative virtual environments: Issues and directions for C2MUVE and VCL. Final Report, Navy/ASEE Summer Faculty Fellow Program, NRaD, August 9, 1996.
- Jones, P. M. and Duffy, L. T. (1996). Using a MOO infrastructure for distributed collaborative crisis action planning. Position paper presented at the Workshop on "Design and use of MUDs for serious purposes", ACM International Conference on Computer-Supported Cooperative Work (CSCW'96), Boston.
- Jones, P. M. and Jasek, C. A. (1997). Intelligent support for activity management (ISAM): An architecture to support distributed supervisory control. *IEEE Transactions on Systems, Man, and Cybernetics*, Special issue on Human Interaction in Complex Systems, Vol. 27, No. 3, May 1997, 274-288.
- Tate, A. (1998). Roots of SPAR - Shared Planning and Activity Representation. *The Knowledge Engineering Review*, Vol. 13(1), Special Issue on Putting Ontologies to Use, to appear.
- Uschold, M., King, M., Moralee, S. and Zorgios, Y. (1998) The Enterprise Ontology. *The Knowledge Engineering Review*, Vol. 13, (1), Special Issue on Putting Ontologies to Use, to appear.
- Woods, D. D. and Roth, E. M. (1988). Cognitive engineering: Human problem solving with tools. *Human Factors*, 29, 415-430.