Using I-X Process Panels as Intelligent To-Do Lists for Agent Coordination in Personnel Recovery

Gerhard Wickler
AIAI, University of Edinburgh
g.wickler@ed.ac.uk

Stephen Potter
AIAI, University of Edinburgh
s.potter@ed.ac.uk

Austin Tate
AIAI, University of Edinburgh
a.tate@ed.ac.uk

ABSTRACT

The aim of this paper is to describe the I-X system with its principal user interface, the I-X Process Panel, its underlying ontology, <I-N-C-A>, and how this panel can be used as an intelligent to-do list that assists emergency responders in applying pre-defined standard operating procedures in different types of emergencies. In particular, multiple instances of I-X Process Panels can be used as a distributed system to coordinate the efforts of independent emergency responders as well as responders within the same organization. Furthermore, it can be used as an agent wrapper for other software systems, such as web services, to integrate these into the emergency response team as virtual members. The heart of the I-X system is an automated planner that can be used to synthesize courses of action or explore alternative options manually.

In the Co-OPR project that is currently underway the I-X framework has been used to develop a prototypical application to support training exercises for personnel recovery. This paper will describe some of the initial findings that are the result of an experiment conducted to evaluate the suitability and extent to which personnel recovery trainees and trainers can be supported by I-X in so-called “Command Post Exercises”. The result shows that an I-X application can be usefully used in such a scenario eliminating some of the basic problems that often occur.

Keywords

HTN planning; intelligent systems; agent capabilities; domain modeling; agent coordination; emergency response.

INTRODUCTION

There are a number of tools available that help people organize their work. One of these is provided with virtually every organizer, be it electronic or paper-based: the “to-do” list. This is because people are not good at remembering long lists of potentially unrelated tasks. Writing these tasks down and ticking them off when they have been done is a simple means of ensuring that everything that needs to be done does get done, or at least, that a quick overview of unaccomplished tasks is available. In responding to an emergency this is vital, and the larger the emergency is, the more tasks need to be managed.

I-X is a framework that can be used to create an application in which multiple agents adopt a task-centric view of a situation, and which supports the necessary coordination of their activities to respond to that situation. The I-X Process Panel provides the functionality of a to-do list and thus, it is a useful tool when it comes to organizing the response to an emergency. The idea of using a to-do list as a basis for a distributed task manager is not new (Kreifelts, Hinrichs and Woetzel, 1993). However, I-X goes well beyond this metaphor and provides a number of useful extensions that facilitate the finding and adaptation of a complete and efficient course of action.

The remainder of this paper is organized as follows: Firstly, we will describe the ontology underlying the whole system and approach. This is necessary for understanding the philosophy behind I-X Process Panels, the user interface that provides the intelligent to-do list. Next, we will describe how the intelligence in the to-do list part is achieved using a library of standard operating procedures, an approach based on HTN (Hierarchical Task Network) planning (Sacerdoti, 1975; Tate 1977). The HTN planning system built into I-X is seamlessly integrated into the system. I-X is not meant to only support single agents in responding to an emergency, but it also provides mechanisms for connecting a number of I-X Process Panels and supporting a coordinated multi-agent response. The key here is a simple agent capability model that automatically matches tasks to known capabilities for dealing with these tasks.

When the I-X framework is instantiated with a domain-specific model, we refer to it as an I-X application. Such an application has been developed during the Co-OPR project for the task of personnel recovery training. A brief description of this application and the set-up for an experiment aimed at evaluating the potential of I-X will be described next. Finally, the results of this experiment will be discussed and some preliminary conclusions drawn.

USING I-X PROCESS PANELS

I-X Process Panels constitute the primary user interface to an I-X application. A panel more or less directly reflects the ontology underlying the whole I-X system, the <I-N-C-A> ontology (Tate, 2003), which is a generic description of a synthesis task, dividing it into four major components: *Issues*, *Nodes*, *Constraints*, and *Annotations*. When used to describe proceses, nodes are the activities that need to be performed in a course of action, thus functioning as the items in an intelligent to-do list. The other elements contain issues as questions remaining for a given course of action, information about the constraints involved and the current state of the world, and notes such as reports or the rationale behind items in the plan.

The <I-N-C-A> Ontology

In <I-N-C-A>, both processes and process products are abstractly considered to be made up of a set of *Issues* which are associated with the processes or process products to represent potential requirements, questions raised as a result of analysis or critiquing, etc. They also contain *Nodes* (activities in a process, or parts of a physical product) which may have parts called sub-nodes making up a hierarchical description of the process or product. The nodes are related by a set of detailed *Constraints* of various kinds. Finally there can be *Annotations* related to the processes or products, which provide rationale, information and other useful descriptions.

<I-N-C-A> models are intended to support a number of different uses:

- for automatic and mixed-initiative generation and manipulation of plans and other synthesized artifacts and to act as an ontology to underpin such use;
- as a common basis for human and system communication about plans and other synthesized artifacts;
- as a target for principled and reliable acquisition of knowledge about synthesized artifacts such as plans, process models and process product information;
- to support formal reasoning about plans and other synthesized artifacts.

These cover both formal and practical requirements and encompass the requirements for use by both human and computer-based planning and design systems.

Issues

The issues in the representation may give the outstanding questions to be handled and can represent decisions yet to be taken on objectives to be satisfied, ways in which to satisfy them, questions raised as a result of analysis, etc. Initially, an <I-N-C-A> artifact may just be described by a set of issues to be addressed (stating the requirements or objectives). The issues can be thought of as implying potential further nodes or constraints that may have to be added into the specification of the artifact in future in order to address the outstanding issues.

In work on I-X until recently, the issues had a task or activity orientation to them, being mostly concerned with actionable items referring to the process underway – i.e., actions in the process space. This has caused confusion with uses of I-X for planning tasks, where activities also appear as nodes. This is now not felt to be appropriate, and as an experiment we are adopting the gIBIS orientation of expressing these issues as questions to be considered (Selvin, 1999; Conklin, 2003). This is advocated by the *Questions–Options–Criteria* approach (MacLean, Young, Bellotti and Moran, 1991) – itself used for rationale capture for plans and plan schema libraries in earlier work (Polyak and Tate, 1998) and similar to the conceptual mapping approaches used in Compendium (Selvin et. al, 2001).

Nodes

The nodes in the representation describe components that are to be included in the design. Nodes can themselves be artifacts that can have their own structure with sub-nodes and other <I-N-C-A> described refinements associated with them. The node constraints (which are of the form “include node”) in the <I-N-C-A> model set the space within which an artifact may be further constrained. The “I” (issues) and “C” (constraints) restrict the artifacts within that space which are of interest.

Constraints

The constraints restrict the relationships between the nodes to describe only those artifacts within the design space that meet the objectives. The constraints may be split into “critical constraints” and “auxiliary constraints” depending on whether some constraint managers (solvers) can return them as “maybe” answers to indicate that the constraint being added to the model is okay so long as other critical constraints are imposed by other constraint managers. The maybe answer is expressed as a disjunction of conjunctions (using an and/or tree) of such critical or shared constraints. More details on the “yes/no/maybe” constraint management approach used in I-X and the earlier O-Plan systems are available in Tate (1995).

The choices of which constraints are considered critical and which are considered as auxiliary are decisions for an application of I-X and specific decisions on how to split the management of constraints within such an application. It is not pre-determined for all applications. A temporal activity-based planner would normally have object/variable constraints (equality and inequality of objects) and some temporal constraints (maybe just the simple before {time-point-1, time-point-2} constraint) as the critical constraints. But, for example in a 3D design or a configuration application, object/variable and some other critical constraints (possibly spatial constraints) might be chosen. It depends on the nature of what is communicated between constraint managers in the application of the I-X architecture.

Annotations

The annotations add additional, often human-centric information or design and decision rationale to the description of the artifact. They are normally expressed as “*keyword* = *value*” annotations. This can be of assistance in making use of products such as designs or plans created using this approach by helping guide the choice of alternatives should changes be required.

I-X Process Panels as Intelligent To-Do Lists

The user interface to the I-X system, the I-X Process Panel, shows four main parts that reflect the four components of the <I-N-C-A> ontology just described. They are labeled “Issues”, “Activities”, “State”, and “Annotations”, as shown in figure 1.

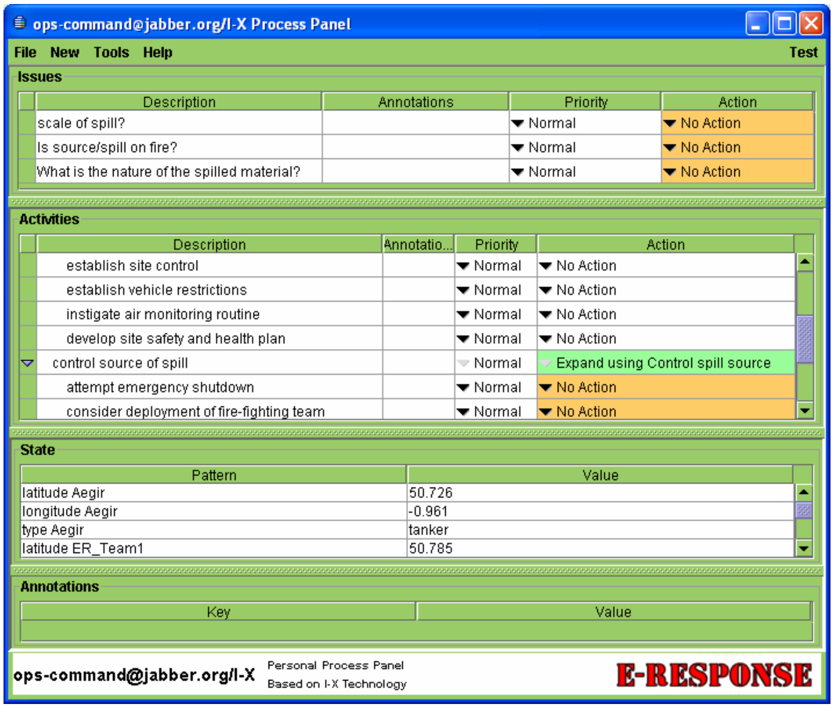

Figure 1: An I-X Process Panel, shown here addressing a simulated oil spill incident.

In the case of the artifact to be synthesized being a course of action, the nodes that will eventually make up the artifact are activities, and these play the central role in the view of an I-X panel as an intelligent to-do list. Users can add an informal or formal description of a task to be accomplished to the activities section of the panel where it will appear as the description of that activity. Each activity consists of four parts listed in the four columns of the activities part of the panel:

- Description: This can be an informal description of a task such as "do this" or it can be a more formal pattern consisting of an activity name (verb) followed by a list of parameters such as
  ```
  (deploy ?team-type)
  ```
  where the words preceded by a question mark are variables that need to be bound before the task can be dealt with.
- Annotation: This can be used to add arbitrary pieces of information to a specific activity.
- Priority: This defines the priority of the activity. Possible values are Highest, High, Normal, Low, or Lowest.
- Action: This field contains a menu that gives the various options that are available to deal with the activity and is the focus of intelligent task synthesis in I-X Process Panels.

It is the last field that allows the user to mark the task as "Done", which corresponds to ticking off an item in a to-do list. Other options that are always available are "No action", the default value until the task has been dealt with, or "N/A" if the activity does not make sense and is "not applicable" in the current context.

The entries in the action menu related to an activity are determined by the activity handlers. These are modules that can be plugged into the I-X system and define ways in which activities can be dealt with. If an activity handler matches an activity it can add one or more entries to the according action menu. The most commonly used activity handler in the context of HTN planning adds "Expand" items to this menu, and this is the point where the to-do list becomes intelligent.

Instead of just being able to tick off an activity, users can use the knowledge in a library of standard operating procedures to break an activity down into sub-activities that, when all performed, accomplish the higher-level task. Of course, sub-activities can themselves be broken down further until a level of primitive actions is reached, at which point the library of procedures no longer contains any refinements that mach the activities. This mechanism supports the user in two ways:

- The library of standard operating procedures may contain a number of different refinements that all match the present activity. All of the applicable procedures are added to the action menu by the activity handler, thus giving the user a comprehensive and quick overview of all the known standard procedures available to deal with this task.
- When a refinement for an activity is chosen, the I-X Process Panel shows all the sub-activities as new items in the to-do list. This ensures that users do not forget to include sub-activities, a common problem especially for infrequently applied procedures.

Both of these problems become only more severe when the user is under time pressure and lives depend on the decisions taken.

Note that the intelligence of the to-do list comes in through the underlying HTN planner that finds applicable refinements in the library and, on demand, can complete a plan to perform a given task automatically, propagating all constraints as it does so. Equally important, however, is the knowledge contained in the library of standard operating procedures. From the perspective of the user this means that I-X can actively suggest ways of performing an activity on the to-do list or I-X can allow the user to explore the set of options currently available.

Other Features

As activities are the nodes that make up a course of action, it is only natural that the activity part of the I-X Process Panel forms the centre of attention for our view of I-X as an intelligent to-do list. In fact, we have implemented a cut-down interface called Post-IX which shows only this part of the panel (and so provides a minimal or 'entry level' interface to the system). We shall now briefly describe the other parts of a panel and how they are used.

World state constraints are used to describe the current state of the world. Essentially, these are a state-variable representation of the form "pattern = value" allowing the user to describe arbitrary features of the world state. They are displayed in the I-X Process Panel in the constraints section. However, it is not expected that users will find this list of facts about the world style representation very useful. Thus, I-X allows for the registration of world state viewers that can be plugged into the system. For example, BBN Openmap (Openmap, 2005) has been used in a number of applications to provide a 2D world map with various features. 3-D virtual reality viewers have also been explored. Most importantly, such world state viewers can be automatically synchronized with the world state constraints such that icons in the map always represent current positions of the entities they represent. Constraints are propagated and evaluated by constraint managers that are plugged into the I-X system.

Issues can be seen as a meta to-do list: instead of listing items that need to be done to deal with an emergency in the real world, they list the questions or outstanding items that need to be dealt with to make the current course of action complete and consistent. Often, these will be flaws in the current plan, but they can also be opportunities that present themselves, or simply facts that need to be verified to ensure a plan is viable. Issues can be either formal, in which case registered issue handlers can be used to deal with them just like activity handlers deal with activities, or they can be informal.

Annotations are used for descriptive elements, such as comments about the course of action as a whole, and are stored as "*keyword* = *value*" patterns.

STANDARD OPERATING PROCEDURES

As outlined above, standard operating procedures describe the knowledge underlying the intelligent to-do list. The formalism is based on refinements used in HTN planning and will be explained next. However, users are not expected to learn this formalism, but they can use a domain editor and its graphical user interface to define the library of procedures.

Activity Refinements in HTN Planning

What are known as standard operating procedures to domain experts are called methods in HTN planning (Ghallab, Nau and Traverso, 2004). Methods formally describe how a task can be broken down into sub-tasks. The definition of a method consists of four main parts:

- Task pattern: an expression describing the task that can be accomplished with this method;
- Name: the name of this method (there may be several for the same task);

- Constraints: a set of constraints (e.g. on the world state) that must hold for this method to be applicable; and
- Sub-task network: a description of the sub-tasks into which this method refines the given task.

The task pattern of a method is used for matching methods to items in the activity list. If the task pattern matches the activity the method will appear in the action menu of the activity in the panel as a possible expansion. This is also where the name of the method will be used: the menu displays an entry “Expand using <name>” where name is the name of the method. In this way, the user can easily distinguish the different options available. The constraints are used to decide whether the method is applicable in the current world state. If they are satisfied, the method can be selected in the action menu, otherwise the unsatisfied constraints can be seen as issues, namely sub-goals that need to be achieved in some way. Finally, the network contains the list of sub-tasks that will be added as activities to the panel when the method is selected. The ordering constraints between sub-tasks are used to show in the interface those sub-tasks that are ready for tackling at any given time.

The I-X Domain Editor

Figure 2 shows an example of the I-X Domain Editor for defining standard operating procedures. The panel on the left lists all the currently defined procedures by name, and the task pattern they match. One, called “`Oil Spill Response (General)`”, is shown being edited. There are a number of views available to edit a refinement. The one shown is the graphical view which shows all the direct sub-tasks with their begin and end time points. Arrows between these activities indicate temporal ordering constraints, for example, the activity “`Control source of spill`” cannot be started before “`Ensure safety of public and response personnel`” has been completed. However, the activities “`Control source of spill`” and “`Manage coordinated response effort`” can then be performed in parallel. Other views show the conditions and effects that can be defined for refinements.

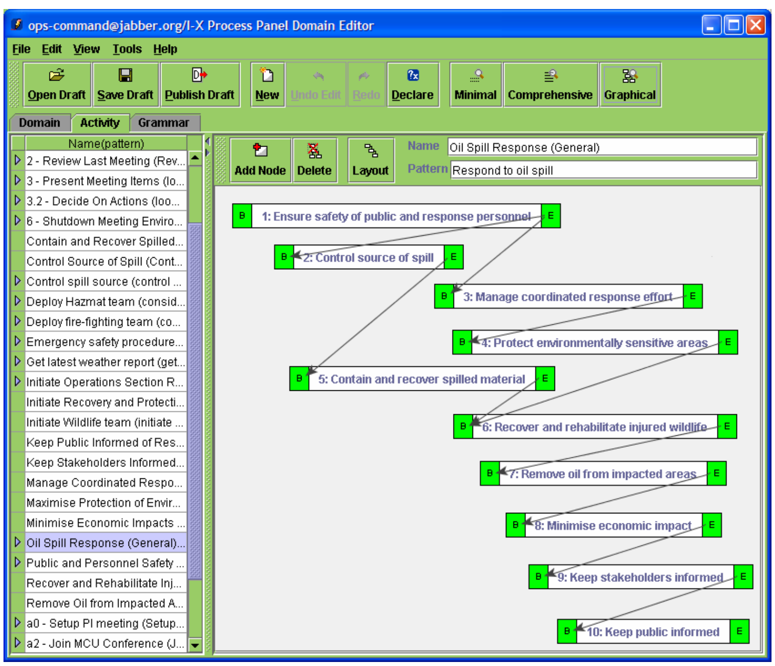

Figure 2: The I-X Domain Editor, here shown modelling an oil spill response standard operating procedure.

AGENT COORDINATION WITH MULTIPLE PANELS

So far we have described I-X as a tool for assisting a single person in organizing and executing the response to an emergency. However, I-X is also a tool that supports the coordination of the response of multiple agents. I-Space is a tool in which users can register the capabilities of other agents. These capabilities can then be used from an I-X panel through inter-panel communication. Augmented instant messaging can be used to directly communicate with other responders via their panels.

I-Space

Every I-X panel can be connected to a number of other I-X agents. Each I-X agent represents an agent that can potentially contribute to the course of action taken to respond in an emergency. The I-Space holds the model of the other agents and can be managed with a simple tool as shown in figure 3.

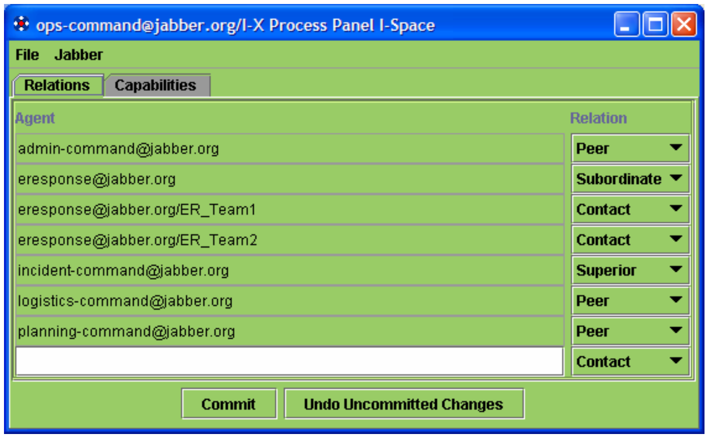

Figure 3: The I-Space Tool. The agents' *relations* to each other governs the nature of interactions between them.

Associated with each agent are one or more communication strategies which define how messages can be sent to this agent. By default, a built-in communication strategy simply sends XML-formatted messages to a given IP-address and socket. Alternatively, a Jabber-strategy (Jabber, 2006) is available for using a instant messaging mechanism for communication. New communication strategies can be added to communicate with agents implemented using different frameworks.

Usually users will not be concerned with the question of how communication takes place as long as the system can find a way, but more with the relationships between the different agents in the I-Space. Within an organization a hierarchical structure is common, so collaborating agents are usually either superiors or subordinates. They can also be modelled as peers, which is also how agents from other organizations can be described. If the agent to be integrated into the virtual organization is a software agent it is described as a (web) service. Finally, a generic relation "contact" is available, but it does not specify what exactly the relationship to this agent is.

Agent Capabilities

At present there is only a relatively simple capability model implemented in I-X. The idea behind this model is that activities are described by verbs in natural language and thus, a task name can be used as a capability description. Parameter values are currently not used to evaluate a capability. Each agent is associated with a number of capabilities that can be called upon.

In the future it will be possible to use a much more sophisticated model. The problem with more complex representations is often that matching capabilities to tasks can be computationally expensive, and when the number of known capabilities becomes large, this can be a problem, which is why the current model is so simple. On the other hand, capabilities can often only be distinguished by a detailed description. One approach to this trade-off is to provide a representation that is flexible, allowing for a more powerful representation where required, but retaining efficiency if the capability description is simple (Wickler, 1999).

Conceptually, the description of a capability is similar to that of an action, which is not surprising as a capability is simply an action that can be performed by some agent. A capability description essentially consists of six components:

- Name: The name of a capability corresponds to the verb that expresses a human-understandable description of the capability.

- Inputs: These are the objects that are given as parameters to the capability. This may be information needed to perform the capability, such as the location of a person to be recovered, objects to be manipulated by the capability, such as paper to be used in a printing process, or resources needed to perform the capability.
- Outputs: These are objects created by the capability. Again, this can be information such as references to hospitals that may have been sought, or they can be new objects if the capability manufactures these.
- Input constraints: These are effectively preconditions, consisting of world state constraints that must be true in the state of the world just before the capability can be applied. Usually, they will consist of required relations between the inputs.
- Output constraints: These are similar to effects, consisting of world state constraints that are guaranteed to be satisfied immediately after the capability has been applied. Usually, they will consist of provided relations between the outputs.
- I-O constraints: These cross constraints link up the inputs with the outputs. For example, a prioritization capability might order a given list of options according to some set of criterions. A cross constraint, referring to both the situation before and after the capability has been applied is necessary to say that the given list of options and the prioritized list contain the same elements.

This capability model can be used to describe the abilities of real-world agents that ultimately must be deployed to do things, or for software agents that provide information that can be used to guide the activity in the physical world.

Handling Activities through Task Distribution

From a user's perspective, task distribution is integrated into the user interface through the "action" menu in the activities part of the panel as just another option available to deal with an activity. The agent relationship is used to determine in which way the activity can be passed to another agent, for example, if the other agent is a subordinate the activity can simply be delegated to the agent.

The capability model is used to filter the options that are listed in the action menu. Currently there is the option of specifying no capabilities for an agent in which case the agent will always be listed. If there is a list of capabilities associated with an agent than these options will only be listed if there is an exact match of the verb capability.

Structured Instant Messaging

Another tool that is widely used for the coordination of efforts in response to an emergency is instant messaging. Like a to-do list, it is very simple and intuitive, but it lacks the formal structure that is needed when the scale of the event that needs to be addressed increases. As for the to-do list, I-X builds on the concept of instant messaging, extending it with the <I-N-C-A> ontology, but also retaining the possibility of simple and informal messages. Thus, users can use structured messaging when this is appropriate, or continue to use unstructured messaging when this is felt to be more useful.

The structured version can be activated by selecting a message type: issue, activity, constraint or annotation, rather than a simple chat message. An <I-N-C-A> object with the content of the message will then be created and sent to the receiving I-X agent. Since all messages between agents are <I-N-C-A> objects, the receiving agent will treat the instant messenger generated message just like any other message from an I-X panel, e.g. the message generated when a task is delegated to a subordinate agent. In this way, structured instant messaging can be seamlessly integrated into the I-X framework without loosing the advantages of informal communications.

APPLICATIONS

I-X has been applied to a number of application scenarios in the area of emergency response. Two projects are currently under way in this context: Co-OPR and FireGrid.

Personnel recovery teams must operate under intense pressure, taking into account not only hard logistics, but "messy" factors such as the social or political implications of a decision. The Collaborative Operations for Personnel Recovery (Co-OPR) project has developed decision-support for sensemaking in such scenarios, seeking to exploit the complementary strengths of human and machine reasoning (Buckingham Shum, Selvin, Sierhuis, Conklin, Haley and Nuseibeh, 2006; Tate, Dalton, and Stader, 2002). Co-OPR integrates the Compendium sensemaking-support tool for real time information and argument mapping, with the I-X artificial intelligence planning and execution framework to support group activity and collaboration. Both share a common model for dealing with issues, the refinement of options for the activities to be performed, handling constraints and recording other information. The tools span the spectrum from being very flexible with few constraints on terminology and content, to knowledge-based relying on rich domain models and formal conceptual models (ontologies). In a personnel recovery experimental simulation of an UN peacekeeping operation, with roles played by military planning staff, the Co-OPR tools were judged by external evaluators to have been very effective.

An example project which needs, and is helping to develop, an integrated and inter-disciplinary approach for emergency response, is FireGrid (Berry et. al, 2005; FireGrid, 2005). This is a UK project to address emergency response in the built environment, where sensor grids in large-scale buildings are linked to super-real time grid-based simulations, and used to assist fire responders to work with the building's internal response systems and occupants to form a "team" to deal with the emergency.

FireGrid will integrate several technologies, extending them where necessary:

- High Performance Computing involving fire models and structural models
- Wireless sensors in extreme conditions with adaptive routing algorithms, including input validation and filtering
- Grid computing including sensor-guided computations, mining of data streams for key events and reactive priority-based scheduling
- Command and Control using knowledge-based planning techniques with user guidance

THE CO-OPR APPLICATION AND EXPERIMENT C

Personnel Recovery (PR) is the sum of military, diplomatic and civil efforts to effect the recovery and reintegration of isolated personnel. During any military operation Joint Force Commanders and Staff are responsible for and must be prepared to accomplish the PR tasks throughout a specified operational area or else determine and accept the risk of not doing so (Joint Publication 3-50, 2005). In order to be prepared, the USJFCOM/JPRA Personnel Recovery Education and Training Center (PRETC) in Fredericksburg, VA, trains US military personnel in the execution of PR tasks. This training consists of classroom sessions in which the necessary knowledge is taught, and it consists of Command Post Exercises (CPX) in which the students have to perform PR tasks in a simulated fictitious military operation.

The aim of "Experiment C" was to emulate one half-day round of a CPX usually held at the PRETC. Experiments A and B were held during an initial phase of the Co-OPR project. Such exercises were observed by the project team and researchers in October 2005, and materials were provided to enable research and experimentation. The experimentation was designed to demonstrate and stress the I-X technology components in response to various individual events in sample missions and events provided by JPRA/PRETC. Following a number of progressively more realistic trials held in AIAI's experimental Emergency Response Coordination Center (e-RCC) during April and May 2006, Co-OPR Experiment C was held on June 1st 2006 following trials of the experimental setup and Internet collaboration software on 30th May 2006.

Command Post Exercises

Command Post Exercises (CPX) are performed at the Personnel Recovery and Training Center (PRETC) as part of the Personnel Recovery course. The course consists of classroom teaching sessions and the CPX in which students are divided into groups, each group playing the role of a rescue center that has to respond to some incidents that are emulated by the trainers.

The context for the incidents and rescue missions that need to be launched is a generic military operation which is set in an area corresponding to the generic map shown in figure 4. In the figure, County-1 represents the country that is being assisted and that is in conflict with its immediate neighbours. A shared coastline makes the involvement of the Navy possible. Country-1 also has rural as well as urban areas that make for an interesting variety of potential incidents. Finally, a neutral country provides some overseas base that may play a role.

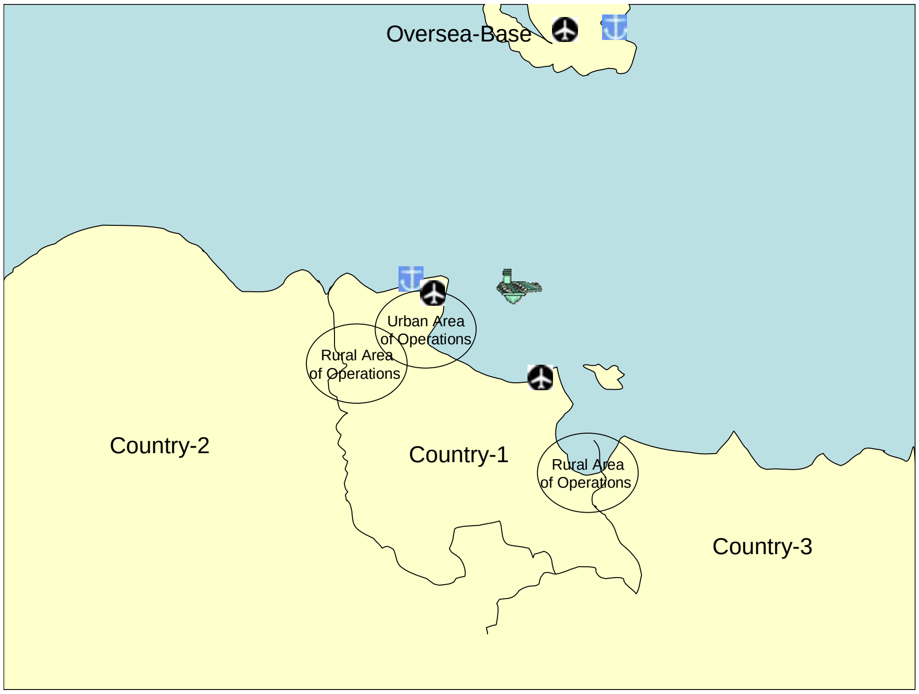

Figure 4: Generic Scenario Map

The students are divided into four groups and placed in different rooms where they act out the activities performed by the different Rescue Coordination Centers (RCCs). In the CPX the Joint Personnel Recovery Center (JPRC) is co-located with the Air Force RCC. All other agents are role-played by the trainers at the PRETC. An overview of the organizational relationships between the different agents is given in figure 5.

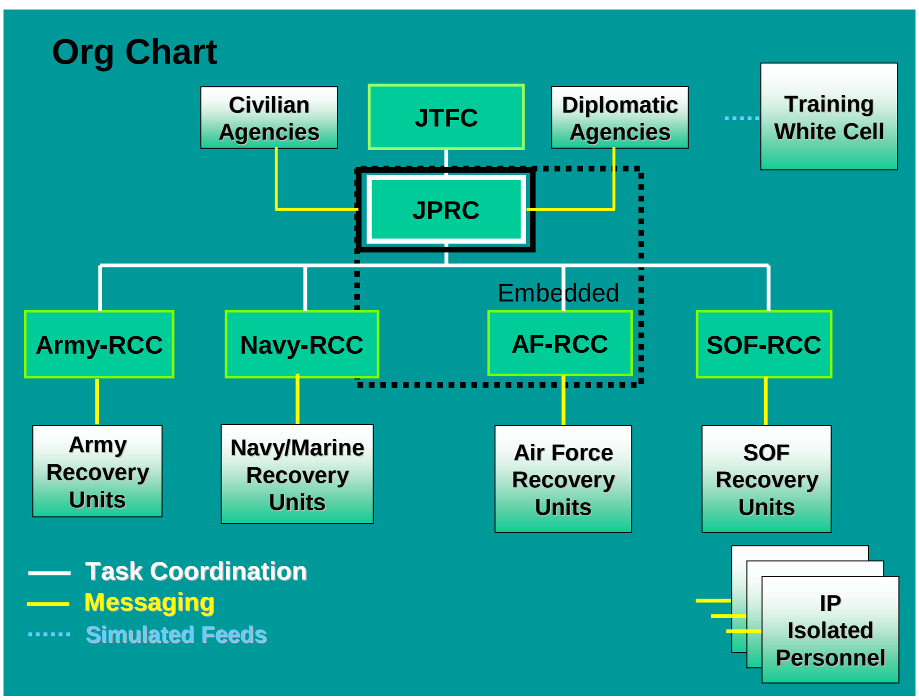

Figure 5: Organization of Agents in the Scenario

The Co-OPR Application

The first step in developing an I-X application consists of deciding which agents to support. For the Co-OPR application it was clear that the most important agent is the JPRC which coordinates the efforts of the different RCCs. Two roles in the JPRC of particular importance are that of the director, who has to manage the centre and make sure everything that needs to be done gets done, and the controller who manages the recovery assets and has to come up with plans for individual recovery missions. Two I-X Process Panels were used to support these two roles. Only the second of these, the one for the controller, had the I-X option management facility enabled (not described in this paper) which can be used to explore possible courses of action and compare different recovery plans (see figure 8). Other RCCs were supported by a single panel only.

Another agent that plays an important role in the training scenario is the “white cell” that drives the scenarios and simulates the events that lead to the incidents the JPRC has to deal with. An I-X Process Panel was used to support this role by allowing for an additional communication channel with the other agents supported by panels. Finally, some other agents that play only minor roles in the different scenarios were included, e.g. the Joint Task Force Commander (JTFC) that has to give authorization for certain missions. Thus, the organization of all the agents in the application is as shown in figure 5.

To implement the task support it was necessary to model a set of standard operating procedures that could be used as refinements in the I-X Process Panel as described above. The refinements used were derived from two sources. Firstly, the U.S. manual for Personnel Recovery (Joint Publication 3-50, 2005) was used as a base for knowledge engineering. Secondly, the checklists used by the PRETC during a CPX were imported into I-X using an experimental import facility. However, the resulting model still required some knowledge engineering, in this case using the I-X Domain Editor.

The application so far can be considered as a simple customization of I-X for the task at hand. However, during the real CPX a number of other tools were used to support the JPRC and other RCCs. It was felt that these were needed for the I-X application too, and corresponding extensions to I-X were implemented.

Whiteboards: The JPRC and RCCs make heavy use of wall mounted whiteboards, maps, overlays on maps, and “pin-board” material such as codes, phone lists, etc. We have implemented whiteboard and map orientated “viewers” that can all simultaneously share the same state in a single panel for display and sharing. We are now exploring ways in which the state underlying specific views can easily be shared with other users and I-X panels, and ways in which variances between the incoming and current believed state on any panel can be highlighted, such that the changes can initiate issues, activities, constraints or notes that need to be incorporated into the local plan.

White-Cell Support: We have created a white cell support panel to assist the trainers in a CPX. This will allow:

- Driving a simulation of the world in which the training takes place, including starting and stopping moving assets such as fuel tankers, trucks, planes and ships.
- Assisting in logging, noting training issues for report back, etc.

Experiment C

Experiment C concentrates on a number of personnel recovery incidents that arise during a military operation which nominally takes place on some given dates in June/July 2000. The experiment covers setting up a JPRC which is co-located with an Air Force RCC and checks with associated RCCs for the Navy, Army and Special Operation Forces (SOF) that they are ready for operations, prior to declaring to the JTFC that the JPRC is active. Incidents of various kinds are dealt with, and a final operation is to prepare a shift change briefing. The aim of the experiment was to allow for an evaluation of the I-X technology as a support tool for both trainers and trainees. At this stage the evaluation was performed with an observer from USJFCOM/J9. It is planned that an evaluation with real users can take place later in the project.

Some screen shots (figures 7-9) illustrating the progress during the experiment from the point of view of the JPRC can be found in the appendix. This double-screen setup was projected in the room such that all members of the JPRC could see the shared information displays, e.g. the electronic whiteboards. Internet application sharing technology was used to let observers remotely view the operations.

EVALUATION

The initial evaluation focused on the cognitive tasks that the JPRC director and JPRC controller performed when working in tandem to respond to the incidents that came into the JPRC as an emergency response coordination centre. This evaluation was necessarily limited in that, without a corresponding analysis of the performance with and use of the current *in-situ* systems and (manual) processes, a comparative assessment of the influence and worth of the I-X system as a whole is not possible. However, an analysis of the results throws up some interesting insights.

Evaluation Methodology

The evaluation methodology was straightforward. The director and the controller roles were played by two members of the I-X development team. In addition to being familiar with the use of I-X systems and with its deployment for this particular domain, these two have gained a basic competence in the objectives, approaches and working practices of the JPRC through observation and completion of basic training courses. An independent observer, a non-participant in the exercise (and also a member of the I-X team), was to observe their behaviour (aided and augmented by self-reporting by the subjects), determine the nature of the task that was currently being performed and the time at which the task began and ended, plus any additional comments or observations. In addition, the exercise was being video-taped, which would allow a retrospective analysis, perhaps with the assistance of the 'director' and 'controller', of any points during the exercise where the precise nature of the immediate task in hand was not clear. Importantly, the experiment was also observed by a member of the sponsoring organization familiar with personnel recovery and with systems evaluation. This was done remotely using Internet collaboration and desktop sharing tools including video teleconferencing.

Once this was done, in an attempt to generalize the various tasks that had been performed where appropriate each task was classified into one of several course-grained 'cognitive categories', namely:

- *information-gathering*: these tasks involved searching for information that was required before the overall activity of the JPRC could be moved forward. In certain cases, this may involve looking up information in on-line databases, or paper-based manuals, or it may involve, say, (simulated) phone-calls to appropriate colleagues.
- *sense-making*: these tasks involved an analysis and interpretation of information with the aim of understanding the problem, enumerating the different options that were available, listing the pros and cons of possible courses of action, and so on.
- *decision-making*: these tasks involved the subject making a clear choice from among competing possible activities *that would serve to achieve the objectives of the JPRC by effecting activity in other agents* and then enacting this activity. So, for example, deciding to send a rescue helicopter to a particular destination and issuing the appropriate orders would be an example of a decision point, whereas deciding to look at a map would not, since it has no affect on other agents (and, instead, would probably be an instance of information-gathering).
- *housekeeping*: these tasks involved the initial set-up of the JPRC environment, documentation of decisions, logging of calls, etc.

The first three of these categories (the *housekeeping* category being an artifact arising from the need to manage the JPRC and the 'paperwork' it generates) emerge from consideration of several different 'best practice' approaches to command and control and decision-making in general. For instance, Boyd's well-known OODA loop – *Observe*, *Orient*, *Decide*, *Act* – can be seen to correspond with these three tasks: *observe* is essentially synonymous in this context with *information-gathering* and *orient* is synonymous with *sense-making*, and since enacting most of the decisions that are taken by the JPRC staff is done by issuing commands to others (i.e., in I-X terms, sending an activity to another agent) and this is done on the click of a mouse button, for our analysis we do not attempt to differentiate the *decide* and *act* activities, but instead we conflate these two OODA tasks into the single *decision-making* category. Similarly, Wohl's SHORe (*Stimulus*, *Hypothesis*, *Option*, *Response*) framework (Wohl, 1981) can be seen to be analogous to our categories, with *stimulus* (Wohl's shorthand term for the information correlation and fusion phase) corresponding to *information-gathering*, *hypothesis* (Wohl's situation analysis phase) corresponding to *sense-making*, and the *option* and *response* phases being conflated into the single *decision-making* task (and for the same reason outlined above).

The correspondence between these different models is summarized in Table 1. The fundamental concept underlying all of these models is that a methodical approach to each cycle of the command and control 'loop', based on assembling information, interpreting that information, appraising possible courses of action and making and enacting decisions should lead to clear, consistent, and – ultimately – correct behaviour in situations where the pressure is great and time is short. Our empirical hypothesis here is that the use of the I-X system can encourage its users to adopt such a methodical approach to their task.

Phase	OODA	SHORe	"JPRC Experiment C" Analysis
1	*observe*	*stimulus*	*information-gathering*
2	*orient*	*hypothesis*	*sense-making*
3	*decide*	*option*	*decision-making*
4	*act*	*response*	

Table 1. Comparison of different Command-and-Control frameworks as they apply in this context; only part of the *act* (OODA) and *response* (SHORe) activities occurs within the context of the JPRC.

Evaluation Results

A fragment of the task analysis performed on the activities observed during Experiment C can be seen in figure 6.

37	0:51:15		decision-making	request launch authority
38	0:53:00	JD	decision-making	mission establish
39	0:55:00	JD	decision-making	send plan to whitecell
40	0:55:45			[wait for confirmation from whitecell]
41	0:57:02	JD	housekeeping	updates state assuming mission complete
42	1:00:00	GW	housekeeping	rescue figgy, note reporting agency, establ
43	1:01:00	GW	information-gathering	request info - incident location
44	1:02:13	GW	information-gathering	receive location
45	1:02:36	GW	information-gathering	validate incident (WoD)
46	1:03:30	GW	information-gathering	awaiting message
47	1:04:00	GW	housekeeping	broadcast WoD update to all subordinates
48	1:04:46	GW	information-gathering	ISOPREP - VOC
49	1:06:58	GW	sense-making	identify threats - I-X mesgs
50	1:09:47	GW	sense-making	ensure conditions, ensure OSC
51	1:11:10			[interrupt] duress word
52	1:13:00	GW	sense-making	reviews plan (threats) so far

Figure 6: Fragment of Co-OPR task analysis.

Notwithstanding the provisos noted above about the inability at the time of writing to perform a full comparative evaluation, the analysis is encouraging for the use of the I-X in this task. In general, the use of SOPs encouraged a methodical approach to the overall JPRC activity: instances of information-gathering where followed by instances of sense-making which led to decision-making episodes, with no instances of, for instance, a decision-making activity being interrupted or abandoned due to the lack of a crucial piece of information. In addition, at several times during the exercise, important messages arrived which interrupted the current activity and diverted the cognitive attention of the director or controller. Such interruptions can serve to disrupt the flow of the Center, but in the majority of cases, the framework provided by the SOPs allowed a quick resumption of activity once the message had been dealt with. The arrival of new information which effectively means that the conclusions reached earlier need to be re-appraised

In addition, the analysis highlighted some areas where further support might prove helpful. In addition to dealing with interruptions, the arrival of new information that demands that the decisions made earlier in the process need to be re-appraised (and, in one case during the experiment, wholly abandoned, with rescue resources 'recalled') is currently not difficult to handle using within the SOP framework (and would seem to require something akin to 'exception-handling' procedures). Successfully dealing with such situations seems to rely overly heavily on the experience and initiative of the human in question. This would seem to be a general problem with any SOP-based system rather than with I-X *per se*, but technology that can offer more support would obviously be of great benefit.

Consideration of the time devoted during the experiment to each of the task categories is also interesting. While roughly the same amount of time was spent in information-gathering, sense-making and decision-making during the exercise, a surprisingly large amount of time was spent housekeeping – twice as long, in fact, as the time spent for any of the other categories. This is due, in part, to the time required to initialize the JPRC and check that its procedures and communications are in place, and then later to produce a report summarizing the session activities for the next duty officer. Providing automated assistance for these tasks may reduce the workload of the humans involved while also ensuring a more rapid and efficient establishment of the Center and hand-over of duty.

Aside from an analysis of the cognitive tasks performed by the system users, the experimentation also highlighted a number of open issues with the current prototype. Firstly, support for the white cell was rather limited at this stage. Only the structured messaging feature was a real advantage provided by I-X. However, the way the scenario was driven was adapted to this way of delegating tasks, which does not correspond well to the way the real CPX works. This in effect removes a large part of the sense-making task from the problem and shifts the focus onto the planning activities, an area in which I-X is strong. Secondly, the two panels used by the director and the controller are equipped with independent <I-N-C-A> models which may lead to inconsistent world state representations within the JPRC. While this did not occur during the experiment, it is a potential problem that was noted. Finally, a few problems with the user interface need to be addressed for future versions, e.g. the lack of a mechanism to draw the user’s attention immediately to new, incoming activities.

CONCLUSIONS

In this paper we have described the I-X framework and I-X Process Panels, which can be seen as providing a distributed and intelligent to-do list for agent coordination in emergency response. The to-do list analogy provides users with a familiar metaphor that should make an I-X application easy to understand. However, I-X extends this concept in two important ways.

- Firstly, items on the to-do list can be expanded using pre-defined standard operating procedures. Such procedures are available in many scenarios but usually only in the form of books or manuals that, even if they are to hand, are often too cumbersome to use in a real emergency. The encoding of such standard operating procedures in I-X is supported by a graphical domain editor. The intention is, of course, that this takes place before an emergency occurs. As a result, this knowledge is at hand and can be used when it is most needed. The HTN planner that is available in I-X uses the library of standard operating procedures to update the Process Panel, showing the user the various ways in which an item on the to-do list can be refined. Thus, the apparent intelligence of the panel is in fact based on the knowledge encoded by a domain expert before an emergency occurs.
- The second extension provided by I-X is the capability model. This allows for a number of panels to be linked to respond in related ways to an emergency. For the user this means that the panel can suggest other agents that may be able to deal with an item if they choose to advertise a matching capability. Furthermore, the panel provides support for the management of such task distribution by sending activities with their parameters and keeping track of reports relating to that activity as they come back.

Both these extensions are integrated into the panel in a seamless way. Together these technologies are used to effectively support emergency responders in organizing a collaborative response quickly and efficiently.

Of the I-X applications currently under development at AIAI, the Co-OPR application was chosen as a test case and an experiment was performed in which the Co-OPR application was used to support the task of personnel recovery training. As a first result, this shows that I-X can indeed be used to build applications that support task-centric activities in the this domain, and that the two features focused on in this paper, namely intelligence through integrated standard operation procedures, and coordination support through linked panels, are useful in supporting the overall activity of a JPRC. More specifically, an analysis of the experiment shows that the hierarchical structure of the tasks in the to-do list helps users to focus their efforts and avoid distractions, and if interrupted, it helps them to quickly continue with important decision making without having to repeat information-gathering or sense-making activities that have already been completed.

The experimentation also served to highlight a number of weaknesses and shortcomings of the current system and its application to problems such as personnel recovery. Work is already underway to address these issues, and an experiment D is scheduled to take place at a US military facility in early October 2006.

ACKNOWLEDGMENTS

The I-X project is sponsored by the Defense Advanced Research Projects Agency (DARPA) under agreement number F30602-03-2-0014. Parts of this work are supported by the Advanced Knowledge Technologies (AKT) Interdisciplinary Research Collaboration (IRC) sponsored by the UK Engineering and Physical Sciences Research Council by grant no. GR/N15764/01. The University of Edinburgh and research sponsors are authorized to reproduce and distribute reprints and on-line copies for their purposes notwithstanding any copyright annotation hereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of other parties.

REFERENCES

1. Berry, D., Usmani, A., Terero, J., Tate, A., McLaughlin, S., Potter, S., Trew, A., Baxter, R., Bull, M. and Atkinson, M. (2005) FireGrid: Integrated Emergency Response and Fire Safety Engineering for the Future Built Environment, UK e-Science Programme All Hands Meeting (AHM-2005), 19-22 September 2005, Nottingham, UK.
2. Buckingham Shum, S., Selvin, A., Sierhuis, M., Conklin, J., Haley, C. and Nuseibeh, B. (2006). Hypermedia Support for Argumentation-Based Rationale: 15 Years on from gIBIS and QOC. In: *Rationale Management in Software Engineering* (Eds.) A.H. Dutoit, R. McCall, I. Mistrik, and B. Paech. Springer-Verlag: Berlin
3. Conklin J. (2003) Dialog Mapping: Reflections on an Industrial Strength Case Study. In: P.A. Kirschner, S.J. Buckingham Shum and C.S. Carr (eds.) *Visualizing Argumentation: Software Tools for Collaborative and Educational Sense-Making*. Springer-Verlag: London, pp. 117-136.
4. FireGrid (2005) FireGrid: The FireGrid Cluster for Next Generation Emergency Response Systems. http://firegrid.org/
5. Ghallab M., Nau D., and Traverso P. (2004) *Automated Planning – Theory and Practice*, chapter 11. Elsevier/Morgan Kaufmann.
6. Jabber (2006) Jabber: Open Instant Messaging and a Whole Lot More, Powered by XMPP. http://www.jabber.org/
7. Joint Publication 3-50 (2005) Joint Doctrine for Personnel Recovery, 2nd Draft, March 2005.
8. Kreifelts Th., Hinrichs E., and Woetzel G. (1993) Sharing To-Do Lists with a Distributed Task Manager. In: de Michelis G. and Simone C. (eds.) *Proceedings of the 3rd European Conference on Computer Supported Cooperative Work*, pp 31-46, Milano, 13-17 September 1993, Kluwer, Dordrecht.
9. MacLean A., Young R., Bellotti V. and Moran T. (1991) Design space analysis: Bridging from theory to practice via design rationale. In *Proceedings of Esprit '91*, Brussels, November 1991, pp 720-730.
10. Openmap (2005) Open Systems Mapping Technology. http://openmap.bbn.com/
11. Polyak S. and Tate A. (1998) Rationale in Planning: Causality, Dependencies and Decisions. *Knowledge Engineering Review*, Vol.13(3), pp 247-262.
12. Sacerdoti E. (1975) The Nonlinear Nature of Plans. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pp 206-214.
13. Selvin A.M. (1999) Supporting Collaborative Analysis and Design with Hypertext Functionality, *Journal of Digital Information*, Volume 1 Issue 4.
14. Selvin A.M., Buckingham Shum S.J., Sierhuis M., Conklin J., Zimmermann B., Palus C., Drath W., Horth D., Domingue J., Motta E. and Li G. (2001) Compendium: Making Meetings into Knowledge Events. *Knowledge Technologies* 2001, Austin TX, USA, March, pp 4-7.
15. Tate A. (1977) Generating Project Networks. . In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pp 888-893.
16. Tate A. (1995) Integrating Constraint Management into an AI Planner. *Journal of Artificial Intelligence in Engineering*, Vol. 9, No.3, pp 221-228.
17. Tate, A., Dalton, J., and J. Stader, J. (2002) I-P2- Intelligent Process Panels to Support Coalition Operations. In *Proceedings of the Second International Conference on Knowledge Systems for Coalition Operations* (KSCO-2002). Toulouse, France, April 2002.
18. Tate A. (2003) <I-N-C-A>: An Ontology for Mixed-initiative Synthesis Tasks. *Proceedings of the Workshop on Mixed-Initiative Intelligent Systems (MIIS)* at the International Joint Conference on Artificial Intelligence (IJCAI-03), Acapulco, Mexico, August 2003, pp 125-130.
19. Wickler G. (1999) *Using Expressive and Flexible Action Representations to Reason about Capabilities for Intelligent Agent Cooperation*. PhD thesis, University of Edinburgh.

20. Wickler, G., Tate, A. and Potter, S. (2006) Using the <I-N-C-A> Constraint Model as a Shared Representation of Intentions for Emergency Response, Proceedings of the First International Workshop on Agent Technology for Disaster Management (ATDM), at the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006), Future University, Hakodate, Japan, May 8-12, 2006.

21. Wohl, J.G. (1981) Force Management Decision Requirements for Air Force Tactical Command and Control. *IEEE Transactions on Systems, Man, and Cybernetics*, 1981, 11.

APPENDIX

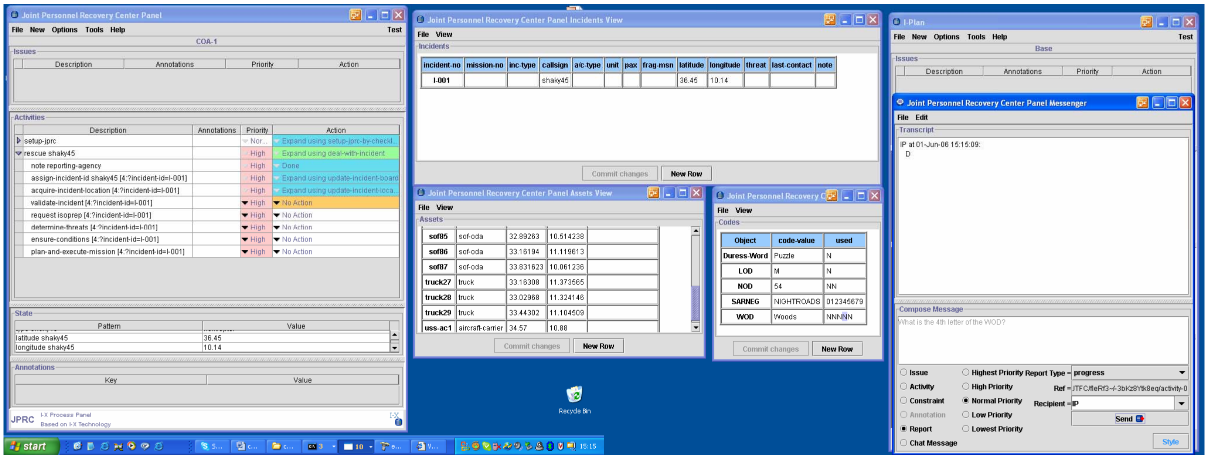

Figure 7: JPRC Panel with rescue in progress (setup of JPRC completed; rescuing shaky45 in progress), Object views ("whiteboards") for incidents, assets and codes; Instant Messenger Window

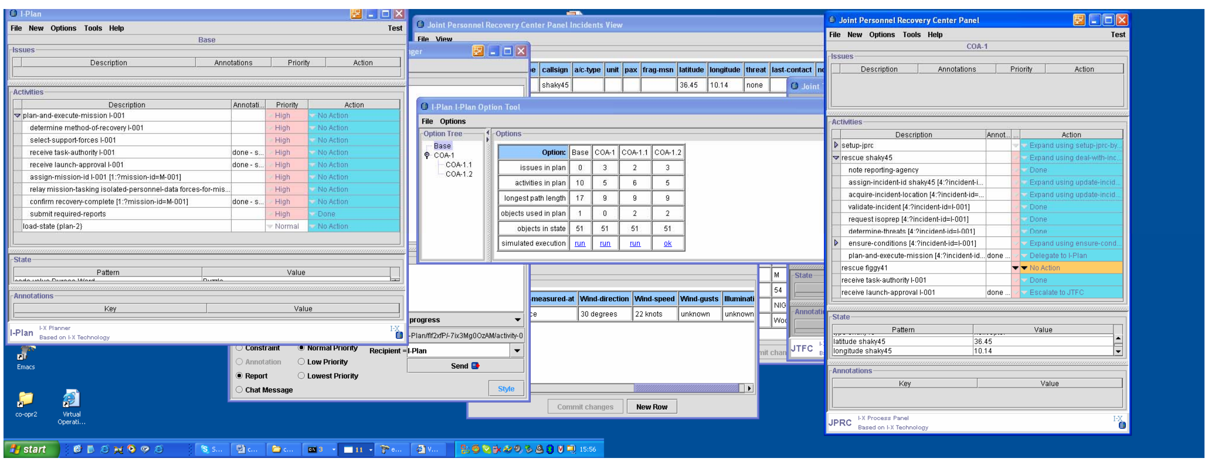

Figure 8: I-Plan Panel with plan completed (all activities' action menus blue); Option Tool with tree structure and comparison matrix; JPRC panel with second incident in progress

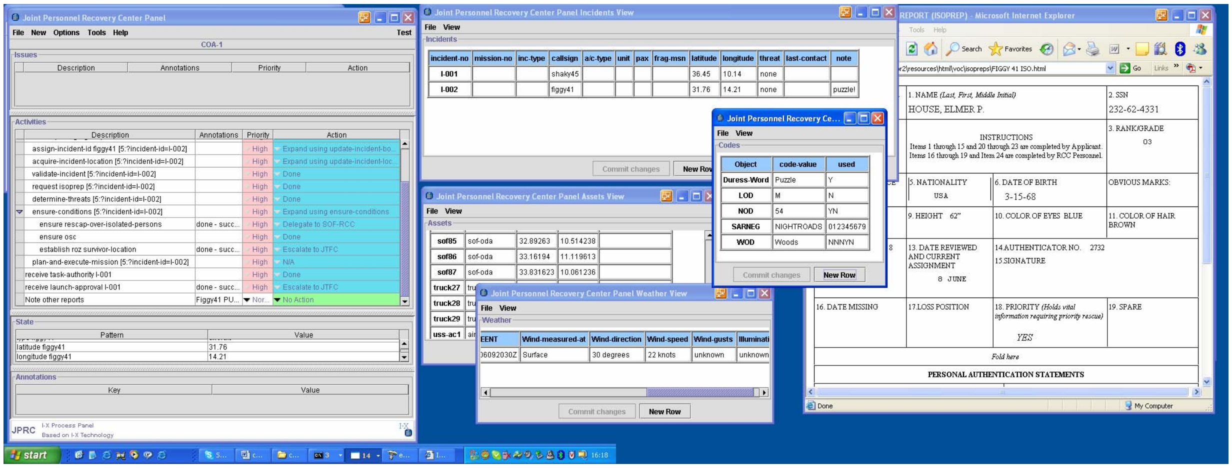

Figure 9: JPRC Panel with second incident completed; Object views (whiteboards) for incidents, assets, weather and codes; Virtual Operation Center showing Isolated Personnel Report (Isoprep) for current incident