



THE UNIVERSITY *of* EDINBURGH

Edinburgh Research Explorer

Using the <I-N-C-A> Constraint Model as a Shared Representation of Intentions for Emergency Response

Citation for published version:

Wickler, G, Tate, A & Potter, S 2006, Using the Constraint Model as a Shared Representation of Intentions for Emergency Response. in 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).

Link:

[Link to publication record in Edinburgh Research Explorer](#)

Document Version:

Peer reviewed version

Published In:

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)

General rights

Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.



Using the <I-N-C-A> Constraint Model as a Shared Representation of Intentions for Emergency Response

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

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

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

ABSTRACT

The aim of this paper is to describe the I-X system with its underlying representation: <I-N-C-A>. The latter can be seen as a description of an agent's intentions, which can be shared and communicated amongst multiple I-X agents to coordinate activities in an emergency response scenario. In general, an <I-N-C-A> object describes the product of a synthesis task. In the multi-agent context it can be used to describe the intentions of an agent, although it also includes elements of beliefs about the world and goals to be achieved, thus showing a close relationship with the BDI agent model which we will explore in this paper. From a user's perspective, I-X Process Panels 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 the 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. At the heart of I-X is a Hierarchical Task Network (HTN) planner that can be used to synthesize courses of action automatically or explore alternative options manually.

Categories and Subject Descriptors

I.2.4 [Artificial Intelligence]: Knowledge Representation Formalisms and Methods – Representation languages;
I.2.8 [Artificial Intelligence]: Problem Solving, Control Methods, and Search – Plan execution, formation, and generation;
I.2.11 [Artificial Intelligence]: Distributed Artificial Intelligence – Multiagent systems.

General Terms

Human Factors, Standardization, Languages, Theory.

Keywords

HTN planning, agent capabilities and coordination, agent modelling.

1 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 very 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.

The I-X system 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 [9]. 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 model underlying the whole system and approach: <I-N-C-A>. 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 Hierarchical Task Network (HTN) planning [14,20]. 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. Finally, we will discuss <I-N-C-A> as a generic artifact model for a synthesis task and show how its components relate the BDI model in the context of planning agents.

2 USING I-X PROCESS PANELS

I-X Process Panels constitute the user interface to the I-X system. They more or less directly reflect the ontology underlying the whole I-X system, the <I-N-C-A> ontology [23], which is a generic description of a synthesis task, dividing it into four major components: Issues, Nodes, Constraints, and Annotations. Of these, nodes are the activities that need to be performed in a course of action, thus functioning as the 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.

2.1 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.

2.1.1 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 [15,3]. This is advocated by the Questions – Options – Criteria approach [10] – itself used for rationale capture for plans and plan schema libraries in earlier work [12] and similar to the mapping approaches used in Compendium [16].

2.1.2 Nodes

The nodes in the specifications 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.

2.1.3 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 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 [21].

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-point1, 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.

2.1.4 Annotations

The annotations add additional human-centric information or design and decision rationale to the description of the artifact. 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.

2.2 I-X Process Panels: 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.

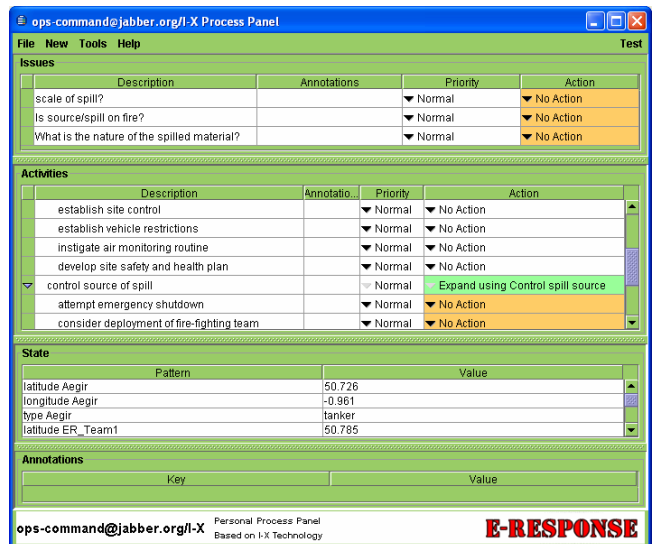


Figure 1. An I-X Process Panel, shown here addressing a simulated oil spill incident.

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

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.

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 match 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.

2.3 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 only shows 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 state 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 [11] has been used in a number of applications to provide a 2D world map with various features. Most importantly, it 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 arbitrary comments about the course of action as a whole, stored as “keyword = value” patterns.

3 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.

3.1 Activity Refinements in HTN Planning

What are known as standard operating procedures to domain experts are called methods in HTN planning [5]. 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
- 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.

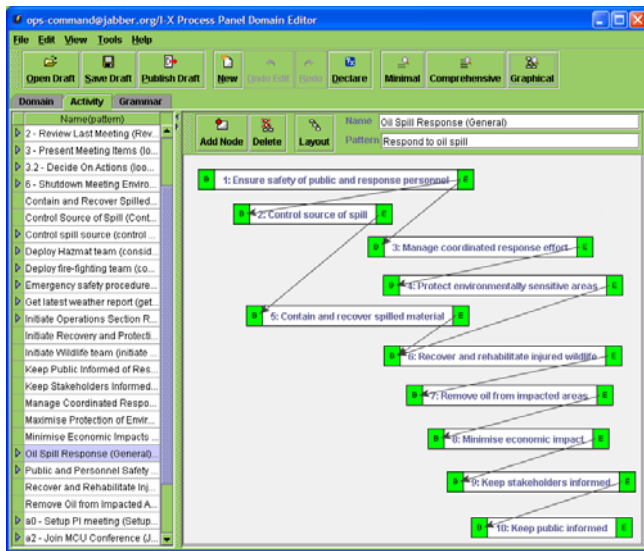


Figure 2. The I-X Domain Editor, here shown modelling an oil spill response standard operating procedure.

3.2 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.

4 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.

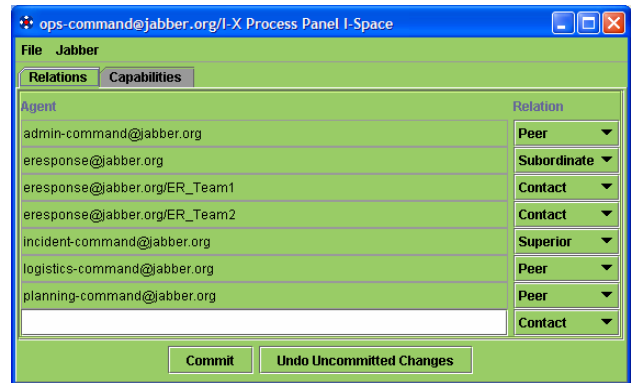


Figure 3. The I-Space Tool. The agents’ relations to each other governs the nature of interactions between them.

4.1 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.

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 [7] is available for using a chat-based 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.

4.2 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 [24].

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 criteria. 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.

4.3 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.

4.4 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 losing the advantages of informal communications.

5 I-X/<I-N-C-A> AND THE BDI MODEL

The idea behind <I-N-C-A> is that it can be used as a generic representation for any synthesized artifact. The nodes are the components that make up the artifact and the constraints restrict the ways in which the components may be synthesized for the design to be successful, i.e. they give relations between the components of the artifact as well as objects in the environment. The issues are the questions that need to be answered before the design is complete and the annotations hold background information of any kind. In the context of planning nodes are actions that need to be synthesized, constraints restrict the way actions can be related to each other, e.g. using the before relation to define a partial order, or what needs to be true in the environment for a plan to be applicable, issues are the items that still need to be worked on before the plan achieves its objective, and annotations hold background information about the plan such as rationale or assumptions. Thus, the task of planning can be described as synthesizing an <I-N-C-A> object, namely a plan which is just an instance of a synthesized artifact. In classical AI planning, a plan is considered to be a solution for a given planning problem if it achieves a goal, i.e. if the performance of the actions in the plan makes the goal condition come true.

Two of the properties that are often associated with intelligent agents, amongst others, are that they are situated and that they should exhibit a goal-directed behaviour [13,6]. By "situatedness"

we mean that an agent exists in and acts upon some environment. The agent may be able to sense the environment and therefore hold some *beliefs* about the state of its environment. A goal is a condition that an agent *desires* to hold in its world, and if it is not believed to be true already, the agent may be able to act towards achieving. The (goal-directed) behavior of an agent is made up of the actions it performs and their performance is not just by accident but because it *intends* to do these actions. Beliefs, desires and intentions are the three cognitive primitives that form the basis for the BDI model of agency [19].

At present, the BDI model is probably the most widely used formal model for describing agents. <I-N-C-A> is the model underlying the I-Plan planner in I-X that is based on decades of planning research. Despite the difference in origin, the two models are closely related and we shall now explore this relation in more detail, by comparing a BDI agent with an I-X agent.

We model an I-X agent by its current (possibly partial) plan (an <I-N-C-A> object) and its world state constraints (as described on the I-X panel). We can relate this to the beliefs, desires and intentions of a BDI agent as described below. The task-oriented nature of I-X means that intentions naturally become most prominent, and it is with these that we begin.

5.1 Intentions

Essentially, I-X agents are focused on intentions. In BDI intentions can be considered to be relationships between an agent and a (again, possibly partial) plan; in the I-X 'world' a plan is the principal <I-N-C-A> object. Specifically, the nodes in an <I-N-C-A> plan are the intended actions; the activity constraints in <I-N-C-A> arrange these actions into a plan; the world state constraints in <I-N-C-A> correspond to that subset of the BDI beliefs that must be held if the plan is to be applicable. <I-N-C-A> issues are related to desires as described below.

5.2 Beliefs

Beliefs are relationships between agents and statements about the world. An I-X agent maintains only specific beliefs, namely: 'facts' about the world that are believed to be true, modeled as constraints in the panel; capability descriptions of other agents in the world; and beliefs about how activities affect the state of the world. Note that the task-centric view of I-X agents means that the knowledge of other agents cannot be easily represented.

5.3 Desires

Desires are not explicitly represented in <I-N-C-A>, but we can say there is a function that can map a given set of BDI desires and an intended partial plan to a set of unresolved or outstanding issues. This means that, in a given context, we can take a BDI description and map it to an <I-N-C-A> object. Correspondingly, given a set of issues and a partial plan, we can derive a super-set of the agent's desires. Initially, when there are no activities then the set of issues correspond to the desires, and eventually, when the plan is complete (and hence, will fulfill the agent's desires), the set of issues will be empty. At any intermediate point, the set of issues will correspond to those desires that the current partial plan will not, as yet, fulfill. Annotations can be used to capture the relationship between satisfied desires and the elements of the plan that satisfy them.

5.4 Summary

This shows that the I-X model of agency and the BDI model are quite similar in many respects. The main difference is rooted in the task-centric view taken by the I-X agent. The <I-N-C-A> model is more specific when it comes to representing plans and activities, but focuses on activity-related beliefs. While this is not a restriction imposed by the <I-N-C-A> model, it is so in the I-X architecture with its specific syntax for representing world state constraints. This is of course necessary to build practical planners for efficient problem solving in real world applications.

6 APPLICATIONS

I-X has been applied to a number of application scenarios in the area of emergency response. In this section we survey some of the current applications.

6.1 Co-OPR

Personnel recovery teams operate under intense pressure, and must take 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 [2,22]. Co-OPR integrates the Compendium sensemaking-support tool for real-time information and argument mapping, using the I-X 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, with Compendium being very flexible with few constraints on terminology and content, to the knowledge-based approach of I-X, relying on rich domain models and formal conceptual models (ontologies). In a personnel recovery experimental simulation of a UN peacekeeping operation, with roles played by military planning staff, the Co-OPR tools were judged by external evaluators to have been very effective.

6.2 I-Rescue

Siebra and Tate [18] have used I-X to support coordination of rescue agents within the RoboCup Rescue simulation [8]. Strategic, Tactical and Operational levels of decision-making were modelled. Their work shows the integration of an activity-oriented planner with agent collaboration using the <I-N-C-A> framework, enabling the easy development of activity handlers that are customized according to the tasks of each decision-making level.

6.3 FireGrid

FireGrid [1,4] is a multi-disciplinary UK project to address emergency response in the built environment, where sensor grids in large buildings are linked to faster-than-real-time grid-based simulations of a developing fire, and used to assist human responders to work with the building's internal response systems and occupants to form a team to deal successfully with the emergency.

The goal of FireGrid is to integrate several technologies, extending them where necessary:

- High Performance Computing applied to the simulation of fire spread and structural integrity.

- 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 I-X technology is to be applied at this level.

This command and control element essentially provides an integrating ‘knowledge layer’ to the system. By using <I-N-C-A> to formalize the interactions between the various participating agents (which, as can be seen from the above description, are drawn from quite different fields and cultures) we hope to harness their various capabilities to provide a seamlessly integrated, response-focused system from the perspective of the human controller.

6.4 AKT e-Response

The Advanced Knowledge Technologies (AKT – see www.actors.org) project is an inter-disciplinary applied research project involving a consortium of five UK universities, concentrating on ‘next generation’ knowledge management tools and techniques, particularly in the context of the semantic web. Emergency response has been chosen as an appropriate task to act as a focus for an integrated demonstrator of a number of AKT technologies.

To this end, we are currently developing a scenario that builds upon the RoboCup-Rescue project “Kobe earthquake” simulator [8]. This project was begun in the wake of the devastating 1995 earthquake to promote applied research to address the inadequacies of the then available IT systems to cope with the demands of the situation. The Kobe simulator was developed to provide a focus to this effort; it models the immediate aftermath of the earthquake, with fires spreading across a district of the city, injured and trapped civilians, and blocked roads hindering response units. Researchers from various fields are invited to participate in the project as they see fit; for instance, the ideas of multi-agent systems researchers can be applied to the coordination of the available (firefighter, police, ambulance) rescue units to attempt to produce an effective response to the disaster. Indeed, this task has become something of a test-piece for researchers interested in agent coordination, with regular competitions to evaluate the relative success (in terms of minimizing overall human and material cost) of different strategies.

However, since the AKT project is focused less on multi-agent systems than on more ‘semantic’ open systems centred on and around humans, for the purposes of the integrated demonstrator we are addressing the task of supporting the high-level strategic response to the emergency. In particular, we aim to provide an ‘intelligence unit’ for the strategy-makers that maintains an overview of the current state of the emergency and the response to it; allows them to access relevant ‘real’ information about the affected locations; lets them explore available options and revise the strategy; and provides a means by which to enact this strategy by relaying orders, reports and other information up and down the chain of command. Since we are looking beyond the simulated world and aim to exploit existing resources and information to guide the response, we have taken the pragmatic decision to

relocate the emergency to London, and in particular the central City of London region, because a number of the AKT technologies are geared towards mining English-language WWW resources for information. (Furthermore, the earthquake has now become a civilian aircraft crash affecting the area, earthquakes of destructive magnitude being rare in the UK.)

The demonstrator is to be underpinned by semantic web technologies. The intelligence unit is supported by a ‘triple-store’ database of RDF ‘facts’ described against OWL ontologies describing types of buildings, medical resources, agents, events, phenomena, and so on. This database is to be populated in part by mining WWW pages. A semantic web service-based architecture [17] will be used to provide a flexible and open framework by which, for example resource management, expertise location, situation visualization and matchmaking services can be invoked. Compendium will again be used as the principal interface to the system, providing an ‘information space’ in which the state of the response is described as it evolves, and from which the various services can be invoked. Alongside this, and building on the *I-Rescue* work, I-X will be used to provide a process-oriented view of the response, with calls to libraries of standard operating procedures providing plans for dealing with archetypal tasks, and activities delegated to agents further down the command-chain, down to and including rescue units ‘on the ground’, also modelled as I-X agents. <I-N-C-A> will be used to formalize the information passed between the agents, and allow it to be located appropriately within the information space.

Looking beyond AKT, we aim to make the modified simulation and the associated semantic resources available to the wider research community, the intention being to provide a test-bed for (and challenge to) semantic web and knowledge management researchers. By engaging these researchers in this manner, we hope to contribute to the RoboCup-Rescue project and its laudable aim of advancing the state-of-the-art in disaster management and response technologies.

7 CONCLUSIONS

In this paper we have described the I-X system which can be seen as a distributed and intelligent to-do list for agent coordination in emergency response. In this view, the system can be used as an extension of a familiar and proven concept, integrating new technologies in a seamless way. Most importantly, it provides an HTN planner that uses methods (standard operating procedures) to define ways in which tasks can be accomplished, and a capability model that describes other agents in a virtual organization. Together these technologies are used to effectively support emergency responders in organizing a collaborative response quickly and efficiently.

A fundamental conceptualization underlying the I-X architecture is the <I-N-C-A> model of a synthesized artifact. This shows up in the internal representation used by I-Plan, in the structure of messages exchanged between I-X agents, and in the user interface, the I-X Process Panels. <I-N-C-A> was developed in the context of AI planning as plan representation but can be generalized to generic synthesis tasks. Furthermore, we have shown that it is closely related to the BDI model of agency, thus providing further evidence that <I-N-C-A> is indeed a good basis for the I-X agent architecture which combines AI planning technology with agent-based system design into an practical

framework that has been and is being applied to several emergency response domains.

8 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.

9 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] Huhns M., Singh M. (1998) Agents and Multi-Agent Systems: Themes, Approaches, and Challenges. In: Huhns M., Singh M. (eds) *Readings in Agents*, pp. 1-23, Morgan Kaufman.
- [7] Jabber (2006) Jabber: Open Instant Messaging and a Whole Lot More, Powered by XMPP. <http://www.jabber.org/>
- [8] Kitano H., and Tadokoro S. (2001) RoboCup Rescue: A Grand Challenge for Multiagent and Intelligent Systems, *AI Magazine* 22 (1): Spring 2001, 39-52
- [9] 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.
- [10] 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.
- [11] Openmap (2005) Open Systems Mapping Technology. <http://openmap.bbn.com/>
- [12] Polyak S. and Tate A. (1998) Rationale in Planning: Causality, Dependencies and Decisions. *Knowledge Engineering Review*, Vol.13(3), pp 247-262.
- [13] Russell S. and Norvig P. (2003) *Artificial Intelligence—A Modern Approach*, 2nd edition, Prentice Hall.
- [14] Sacerdoti E. (1975) The Nonlinear Nature of Plans. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pp 206-214.
- [15] Selvin A.M. (1999) Supporting Collaborative Analysis and Design with Hypertext Functionality, *Journal of Digital Information*, Volume 1 Issue 4.
- [16] 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.
- [17] Shadbolt N., Lewis P., Dasmahapatra S., Dupplaw D., Hu B. and Lewis H. (2004) MIAKT: Combining Grid and Web Services for Collaborative Medical Decision Making. In *Proceedings of AHM2004 UK eScience All Hands Meeting*, Nottingham, UK.
- [18] Siebra C. and Tate A. (2005) Integrating Collaboration and Activity-Oriented Planning for Coalition Operations Support. In *Proceedings of the 9th International Symposium on RoboCup 2005*, 13-19 July 2005, Osaka, Japan.
- [19] Singh M., Rao A. and Georgeff M. (1999) Formal Methods in DAI: Logic-Based Representation and Reasoning. In: Weiss G. (ed) *Multiagent Systems*, pp. 331-376, MIT Press.
- [20] Tate A. (1977) Generating Project Networks. . In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pp 888-893.
- [21] Tate A. (1995) Integrating Constraint Management into an AI Planner. *Journal of Artificial Intelligence in Engineering*, Vol. 9, No.3, pp 221-228.
- [22] Tate A., Dalton J., and 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.
- [23] 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.
- [24] Wickler G. (1999) *Using Expressive and Flexible Action Representations to Reason about Capabilities for Intelligent Agent Cooperation*. PhD thesis, University of Edinburgh.