Task Achieving Agents on the World Wide Web

Austin Tate, Jeff Dalton, John Levine and Alex Nixon

Artificial Intelligence Applications Institute

Division of Informatics

The University of Edinburgh

80 South Bridge, Edinburgh EH1 1HN, UK

{a.tate, j.dalton, j.levine}@ed.ac.uk

http://www.aiai.ed.ac.uk/project/ix/

An important class of problems is related to performing activities, and the planning of future activity, The "doing of things" is at the heart of human endeavour. The WWW has primarily concentrated to date on information storage and retrieval, and the data models and standards mostly relate to such things. More emphasis should now be placed on modelling activity and the collaboration between human and system agents that can be conducted through the WWW.

The planning and process modelling communities have started to develop shared models and ontologies to represent activities, tasks, agent capabilities, constraints, etc. These might form the generic core of a shared ontology to support the movement of information about activities over the WWW.

The paper describes some work on producing collaborative, multi-agent systems with a mix of human and software agents engaging in planning and plan execution support over the WWW. The work includes O-Plan, Process Panels and I-X. The underlying process ontology <I-N-OVA> constraint model of activity and the more general <I-N-CA> constraint model of synthesised artefacts are described. These could provide a robust conceptual model to underlie future web standards for describing task achieving agents on the web and their behaviours.

Introduction and Motivation

The World Wide Web currently acts as a vast electronic library, serving information and providing search facilities for accessing that information. However, given that the Web actually consists of a vast network of task achieving agents (humans and computers), this view of the Web as a static pool of information is only using a small fraction of its real capabilities.

The idea of the Web being a place where you can ask agents to *do* things and to *plan* activities seems intuitively attractive. However, the data models and standards developed to date for the Web mostly relate to information retrieval, rather than activity and the planning of future activity. In order to make the Web a place for "doing things" as well as "finding things", we need shared models and ontologies to represent the entities involved in planning and doing: activities, tasks, plans, agent capabilities, and so on.

The AI planning and the process modelling communities have recently started to develop standards in these areas, for the purpose of working on common models and sharing information about activities and processes (Tate, 1998). These common models and ontologies might form the generic core of a shared ontology to support the movement of information and data relating to activities over the World Wide Web.

This paper is in two parts. In the first part, we describe work towards the creation of a common ontology and representation for plans, processes and other information related to activity. We briefly describe the work going on in two areas: military planning and standards for representing activities and processes.

Our own systems are based on an underlying activity ontology called <I-N-OVA>; this is described, together with the more general <I-N-CA> constraint-based model for representing synthesised artefacts. In both of these models, the I stands for *issues*, which allows us to represent synthesised artefacts which are not yet complete or which have some outstanding issues to address.

The list of outstanding issues is crucial in the communication of partial results between agents, which is clearly needed in multi-agent systems which work together to synthesise solutions.

In the second part, we describe our work on producing collaborative multi-agent systems consisting of human and computer agents engaging in planning and plan execution support over the World Wide Web. These applications are based on a generic interface for web-based task achieving agents called Open Planning Process Panels or O-P^3 (Levine, Tate and Dalton, 2000). These panels are described briefly to introduce the work that follows. Three web-based applications are then described: the O-Plan Web demonstration, the Air Campaign Planning Process Panel (ACP^3) and a version of O-Plan that can run over the Web using a WAP-enabled mobile telephone. These applications are indicative of the kind of systems which we see being deployed in the near future, where the Web site acts as an interface to one or more intelligent agents and the common representation of activity-related information is crucial.

PART 1: Standards for Representing Activities

In the first part of this paper, we describe work towards the creation of a common ontology and representation for plans, processes and other information related to activity.

There are two major stands of work here. In military planning work, there has already been much work in developing shared models for planning and representing plans, such as the KRSL plan representation language, the Core Plan Representation (CPR) and the Shared Planning and Activity Representation (SPAR).

At the same time, work in the standards community has attempted to standardize the terminology for talking about activities and processes: examples include the Process Interchange Format (PIF), NIST Process Specification Language (PSL), and work by the Workflow Management Coalition (WfMC).

Tate (1998) gives an overview and history of all these efforts and shows their relationship to the Shared Planning and Activity Representation (SPAR) developed under the DARPA and USAF Research Laboratory planning initiative (ARPI). Full references are provided in that paper and in its on-line copy [1].

Our own systems are based on the <I-N-OVA> [2] activity ontology; this relates well to the other ontologies of activity described above, such as SPAR, and can be considered as an abstract model which can underlie these. The <I-N-OVA> model is described in the following sections, together with the more general <I-N-CA> model for representing synthesised artefacts.

<I-N-OVA> and <I-N-CA>

This section presents an approach to representing and manipulating plans and other synthesised artefacts in the form of a set of constraints. The <I-N-OVA> (*Issues – Nodes – Orderings/Variables/Auxiliary*) constraints model is used to characterise plans and processes. The more general <I-N-CA> (*Issues – Nodes – Critical/Auxiliary*) constraints model can be used for wider applications in design, configuration and other tasks which can be characterised as the synthesis and maintenance of an artefact or product.

Motivation

As shown in Figure 1, the <I-N-OVA> and <I-N-CA> constraint models are intended to support a number of different uses:

- for automatic manipulation of plans and other synthesised artefacts and to act as an ontology to underpin such use;
- as a common basis for human communication about plans and other synthesised artefacts;
- as a target for principled and reliable acquisition of plans, models and product information;
- to support formal reasoning about plans and other synthesised artefacts.

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

The <I-N-OVA> model is a means to represent plans and activity as a set of constraints. By having a clear description of the different components within a plan, the model allows for plans to be manipulated and used separately from the environments in which they are generated. The underlying thesis is that activity can be represented by a set of constraints on the behaviours possible in the domain being modelled and that activity communication can take place through the interchange of such constraint information.

<I-N-OVA>, when first designed (Tate, 1996b), was intended to act as a bridge to improve dialogue between a number of communities working on formal planning theories, practical planning systems and systems engineering process management methodologies. It was intended to support new work then emerging on automatic manipulation of plans, human communication about plans, principled and reliable acquisition of plan information, mixed-initiative planning and formal reasoning about plans. It has since been used as the basis for a number of research efforts, practical applications and emerging international standards for plan and process representations. For some of the history and relationships between earlier work in AI on plan representations, work from the process and design communities and the standards bodies, and the part that <I-N-OVA> played in this, see Tate (1998).

Representing Plans in <I-N-OVA>

A plan is represented as a set of constraints which together limit the behaviour that is desired when the plan is executed. The set of constraints are of three principal types with a number of sub-types reflecting practical experience in a number of planning systems.

INSERT FIGURE 2 NEAR HERE

The node constraints (these are often of the form "include activity") in the <I-N-OVA> model set the space within which a plan may be further constrained. The I (issues) and OVA constraints restrict the plans within that space which are valid. Ordering (temporal) and variable constraints are distinguished from all other auxiliary constraints since these act as *cross-constraints* [3], usually being involved in describing the others – such as in a resource constraint which will often refer to plan objects/variables and to time points or ranges.

In Tate (1996b), the <I-N-OVA> model is used to characterise the plan representation used within O-Plan (Currie and Tate, 1991; Tate, Drabble and Dalton, 1994) and is related to the plan refinement planning method used in O-Plan.

We have generalised the <I-N-OVA> approach to design and configuration tasks with I, N, CA components - where C represents the "critical constraints" in any particular domain - much as certain O and V constraints do in a planning domain. We believe the approach is valid in design and synthesis tasks more generally - we consider planning to be a limited type of design activity. <I-N-CA> is used as an underlying ontology for the I-X project [4].

Rationale for the Categories of Constraints within <I-N-OVA>

Planning is the taking of planning decisions (I) which select the activities to perform (N) which creates, modifies or uses the plan objects or products (V) at the correct time (O) within the authority, resources and other constraints specified (A). The Issues (I) constraints are the items on which selection of Plan Modification Operators is made in agenda based planners.

Others have recognised the special nature of the inclusion of activities into a plan compared to all the other constraints that may be described. Khambhampati and Srivastava (1996) differentiate Plan Modification operators into "progressive refinements" which can introduce new actions into the plan, and "non-progressive refinements" which just partitions the search space with existing sets of actions in the plan. They call the former genuine planning refinement operators, and think of the latter as providing the scheduling component.

If we consider the process of planning as a large constraint satisfaction task, we may try to model this as a Constraint Satisfaction Problem (CSP) represented by a set of variables to which we have to give a consistent assignment of values. In this case we can note that the addition of new nodes ("include activity" constraints in <I-N-OVA>) is the only constraint which can add variables dynamically to the CSP. The Issue (I) constraints may be separated into two kinds: those which may (directly or indirectly) add nodes to the plan and those which cannot. The I constraints which can lead to the inclusion of new nodes are of a different nature in the planning process to those which cannot.

Ordering (temporal) and variable constraints are distinguished from all other auxiliary constraints since these act as cross-constraints, usually being involved in describing the others – such as in a resource constraint which will often refer to plan objects/variables and to time points or intervals.

INSERT FIGURE 3 NEAR HERE

Sorted First Order Logic Base, and XML

<I-N-OVA> and <I-N-CA> are meant as conceptual models which can underlie any of a range of languages which can describe activities, plans, processes and other synthesised artefacts. For example, O-Plan is based on <I-N-OVA>, but utilises the Task Formalism domain description language which has a simple keyword introduced syntax.

It is anticipated that any <I-N-OVA> or the more general <I-N-CA> model in whatever language or format it is expressed can be reduced to a conjunctive set of statements in first order logic with strong requirements on the type of the terms involved in each statement – i.e. a sorted first order logic. See Polyak and Tate (1998) for further details, and for a use described in a planning domain modelling support system.

<I-N-OVA> and <I-N-CA> constraint sets lend themselves very well to being used in eXtendible Markup Language (XML) representations of synthesised artefacts, especially when these are still in the process of being designed or synthesised. The processes that are used to do this synthesis and the collaborations and capabilities involved can also be described in <I-N-OVA> and/or <I-N-CA>.

PART 2: Web-Based Applications

In the second part, we describe our work on producing collaborative multi-agent systems consisting of human and computer agents engaging in planning and plan execution support over the World Wide Web. These applications are based on a generic interface for web-based task achieving agents called Open Planning Process Panels or O-P^3 (Levine, Tate and Dalton, 2000). These panels are described briefly to introduce the work that follows. Three web-based applications are then described: the O-Plan Web demonstration, the Air Campaign Planning Process Panel (ACP^3), and a version of O-Plan that can run over the Web using a WAP-enabled mobile telephone. These applications are indicative of the kind of systems which we see being deployed in the near future, where the Web site acts as an interface to one or more intelligent agents and the common representation of activity-related information is crucial.

Open Planning Process Panels

Real world planning is a complicated business. Courses of action to meet a given situation are constructed collaboratively between teams of people using many different pieces of software. The people in the teams will have different roles, and the software will be used for different purposes, such as planning, scheduling, plan evaluation, and simulation. Alternative plans will be developed, compared and evaluated, and more than one may be chosen for briefing. In general, planning is an example of a multi-user, multi-agent collaboration in which different options for the synthesis of a solution to given requirements will be explored.

The process of planning is itself the execution of a plan, with agents acting in parallel, sharing resources, communicating results and so on. This planning process can be made explicit and used as a central device for workflow coordination and visualisation.

We have used this idea to create Open Planning Process Panels (O-P^3). These panels are used to coordinate the workflow between multiple agents and visualise the development and evaluation of multiple courses of action (COAs). The generic notion of O-P^3 has been used to implement an O-Plan two-user mixed-initiative planning Web demonstration and an Air Campaign Planning Process Panel (ACP^3). In the former, O-P^3 technology is used to enable the development and evaluation of multiple COAs by a commander, a planning staff member and the O-Plan automated planning agent. In the latter, O-P^3 is used to build a visualisation panel for a complex multi-agent planning and evaluation demonstration which uses 11 different software components and involves several users.

O-P^3 technology could have an impact on several important research areas:

- Automated planning: O-P^3 shows how automated planning aids such as AI planners can be used within the context of a wider workflow involving other system agents and human users.
- Computer-supported cooperative work (CSCW): O-P^3 uses explicit models of the collaborative planning workflow to coordinate the overall effort of constructing and evaluating different courses of action. This is generalisable to other team-based synthesis tasks using activity models of the task in question (e.g. design or configuration).

- Multi-agent mixed-initiative planning: O-P^3 facilitates the sharing of the actions in the planning process between different human and system agents and allows for agents to take the initiative within the roles that they play and the authority that they have (Tate, 1993).
- Workflow support: O-P^3 provides support for the workflow of human and system agents working together to create courses of action. The workflow and the developing artefact (i.e. the course of action) can be visualised and guided using O-P^3 technology.

The kind of planning system that we envisage O-P^3 being used for is one in which the planning is performed by a team of people and a collection of computer-based planning agents, who act together to solve a hard, real world planning problem. Both the human and the software agents will act in given roles and will be constrained by what they are authorised to do, but they will also have the ability to work under their own initiative and volunteer results when this is appropriate. When the planning process is underway, the agents will typically be working in parallel on distinct parts of the plan synthesis. The agents will also be working in parallel to explore different possible courses of action; for example, while one COA is being evaluated, another two may be in the process of being synthesised.

This section introduces O-P^3 technology. It begins with a description of the generic O-P^3 ideas, based on the central notion of an explicit shared model of the activities involved in creating a plan – the planning process.

Generic O-P^3 Technology

The generic O-P^3 is based on an explicit model of the planning process, which is encoded using an activity modelling language, such as <I-N-OVA>. This represents the planning process as a partially-ordered network of actions, with some actions having expansions down to a finer level of detail (i.e. to another partially-ordered network).

The purpose of O-P^3 is to display the status of the steps in the planning process to the users, to allow the users to compare the products of the planning process (i.e. the courses of action) and to allow the users to control the next steps on the "workflow fringe" (i.e. what actions are possible next given the current status of the planning process). In the context of creating plans, O-P^3 is designed to allow the development of multiple courses of action and the evaluation of those courses of action using various plan evaluations.

A generic O-P^3 panel would have any of a number of "sub-panels", which can be tailored to support specific users or user roles. These include:

- A course of action comparison matrix showing:
  - COAs vs elements of evaluation, with the plan evaluations being provided by plug-in plan evaluators or plan evaluation agents;
  - the steps in the planning process (from the explicit process model), the current status of those steps (the *state model*), and control for the human agent of what action to execute next;
  - the *issues* outstanding for a COA that is being synthesised and which must be addressed before the COA is ready to execute;
- a graphical display showing the status of the planning process as a PERT chart, which is a useful alternative view of the planning process to that given by the tabular matrix display;
- other visualisations, such as bar charts, intermediate process product descriptions, and textual description of plans.

The generic O-P^3 methodology for building Open Planning Process Panels consists of the following steps:

- Consider the agents (human and system) who are involved in the overall process of planning. Assign roles and authorities to these agents.

- Construct an activity model of the planning process, showing the partial ordering and decomposition of the actions and which agents can carry out which actions. This activity model could be represented using an activity modelling language such as <I-N-OVA>.
- Build a model of the current state of the planning process and an activity monitor which will update this state model as actions in the planning process take place.
- Construct appropriate O-P^3 interfaces for each of the human agents in the planning process, taking into account the role which they play in the interaction. This means that each different user role will have an O-P^3 interface which is tailored to the overall nature of their task.

The O-P^3 agent interfaces then allow the human agents to play their part in the overall planning process, alongside the system agents, which will be AI planners, schedulers, plan evaluators and so on. This is illustrated in Figure 4.

Application 1 – O-Plan on the Web

The O-Plan project (Tate, Drabble and Dalton, 1996; Tate, Dalton and Levine, 1998) was concerned with providing support for mixed-initiative planning. The web-based demonstration described here [5] shows interaction between two human agents and one software planning agent (the O-Plan plan server). The overall concept for our demonstrations of O-Plan acting in a mixed-initiative multi-agent environment is to have humans and systems working together to populate the COA matrix component of the O-P^3 interface.

As shown in Figure 5, we envisage two human agents acting in the user roles of Task Assigner and Planner User, working together to explore possible solutions to a problem and making use of automated planning aids to do this. Figure 6 shows how the two human agents work together to populate the matrix. The Task Assigner sets the requirements for a particular course of action (i.e. what top level tasks must be performed), selects appropriate evaluation criteria for the resulting plans and decides which courses of action to prepare for briefing. The Planner User works with O-Plan to explore and refine the different possible course of action for a given set of top level requirements. The two users can work in parallel, as is demonstrated in the example scenario (Levine, Tate and Dalton, 2000).

The overall planning task is thus shared between three agents who act in distinct user and system roles. The Task Assigner (TA) is a commander who is given a crisis to deal with and who needs to explore some options. This person will be given field reports on the developing crisis and environmental conditions. The Planner User is a member of staff whose role is to provide the Task Assigner with plans which meet the specified criteria. In doing this, the Planner User will make use of the O-Plan automated planning agent, whose role is to generate plans for the Planner User to see. The Planner User will typically generate a number of possible course of action using O-Plan and only return the best ones to the Task Assigner.

For our current demonstration, we are using a general purpose logistics and crisis operations domain which is an extension of our earlier Non-Combative Evacuation Operations (NEO) and logistics-related domains (Reece *et al.*, 1993). This domain, together with the O-Plan Task Formalism (TF) implementation, is described in detail by Tate, Dalton and Levine (1998).

The two human users are provided with individual O-P^3 panels which are implemented using a CGI-initiated HTTP server in Common Lisp and which therefore run in any World Wide Web browser – the Common Lisp process returns standard HTML pages. This way of working has many advantages:

- the two users can be using different types of machine (Unix, PC, Mac) and running different types of Web browser (Netscape, Internet Explorer, Hotjava, etc.);
- the only requirement for running O-Plan is a World Wide Web connection and a Web browser (i.e. no additional software installation is needed);

- the two users can be geographically separate – in this case, voice communication via the telephone or teleconferencing is all that is required in addition to the linked O-P^3 interfaces.

The planning process for the Task Assigner and the Planner User is made explicit through the hypertext options displayed in the process parts of the O-P^3 panels. These are either not present (not ready to run yet), active (on the workflow fringe) or inactive (completed). Further parts of the planning process are driven by *issues* which O-Plan or the plan evaluation agents can raise about a plan under construction and which can be handled by either or both of the human agents. Because the planning process is made explicit to the two users through these two mechanisms, other visualisations of the planning process itself are not required. However, the products of the planning process (the courses of action) are complex artefacts for which multiple views are needed. In the current version, the courses of action can be viewed as a PERT network, as a textual narrative, or as a plan level expansion tree (all at various levels of detail).

The user roles are arranged such that the Task Assigner has authority over the Planner User who in turn has authority over O-Plan. This means that the Task Assigner defines the limits of the Planner User's activity (e.g. only plan to level 2) and the Planner User then acts within those bounds to define what O-Plan can do (e.g. only plan to level 2 and allow user choice of schemas). Other aspects of what the two users are authorised to do are made explicit by the facilities included in their respective panels.

The COA Comparison Matrix

The two panels for the Task Assigner and Planner User are shown in Figures 7 and 8. Each user has control over the plan evaluation elements which are shown, to enable the critical elements of evaluation to be chosen. In the example scenario given later, the Task Assigner is only interested in the minimum duration and the effectiveness, so only these are selected. On the other hand, the Planner User wants a variety of data to pick the best COA, so all evaluations are shown.

The role of the Task Assigner is to set up the top level requirements for a course of action. Once this is done, the COA is passed across to the Planner User, whose matrix is initially blank. The Planner User then explores a range of possible COAs for the specified requirements and returns the best ones to the Task Assigner. When the Planner User returns a COA to the Task Assigner, the column for that COA appears in the Task Assigner's matrix. The Planner User and the Task Assigner can be working in parallel, as demonstrated in the scenario.

Application 2 – ACP^3

One of the integrated demonstrations from the US DARPA ARFL/Rome Planning Initiative (ARPI) (Tate, 1996a) brings together eleven, separately developed, software systems for planning and plan evaluation. When the demonstration is run, these systems work together to create and evaluate multiple courses of action in the domain of Air Campaign Planning. The systems communicate with each other by exchanging KQML messages (Finin, Labrou and Mayfield, 1997). Finding out what is happening at any given time could (in theory) be done by watching these KQML messages, but this was obviously less than ideal as these messages use technological terms which are far removed from the terminology used by the user community.

Our aim was to use O-P^3 technology to build a visualisation component for this demonstration which would allow the target end users to view the current state of the planning process in process terms they are familiar with. This has resulted in ACP^3 – the Air Campaign Planning Process Panel.

Modelling the Planning Process

The software components of the ARPI demonstration can be described as performing activities such as planning, scheduling, simulation and plan evaluation. Going into more detail, we can talk about hierarchical task network planning and Monte Carlo simulation methods. However, end users are more likely to conceive of the processes of Air Campaign Planning in more general, domain-related terms, such as "develop JFACC guidance" and "create support plan". The gaps in terminology and in levels of description can be bridged by building models of the planning process which are rooted in established ACP terminology. We have therefore made use of the previously elicited and verified ACP process models of Drabble, Lydiard and Tate (1997) as our source of terminology and as the basis of our IDEF3 models of the planning process for the ARPI demonstration. The full models used for building ACP^3 are described in Aitken and Tate (1997).

Building ACP^3

The ACP^3 viewer is shown in Figure 9. The purpose of ACP^3 is to track the overall planning process and display this to the viewers of the ARPI demonstration in a meaningful way using appropriate military process terminology. The planning process is shown in two separate sub-panels. The tabular COA comparison matrix shows COAs being developed (columns) against a tree-based view of the planning process. The graph viewer sub-panel shows the planning process as a PERT network. Since the planning process consists of many nodes with expansions, the graph viewer can only display one individual graph from the planning process for one COA. Other graphs may be reached by clicking on nodes with expansions, and the end user can choose which COA to view.

The two views are required because the planning process in the ARPI demonstration is a complex artefact. It is possible to see the whole process for every COA in the COA matrix, but information about the partial ordering of the actions in a graph is lost when the graph is converted to a tree structure. The graph viewer shows the full partial ordering but space considerations mean that only a single graph for a single COA can be shown at one time.

The ACP^3 process monitor works by watching for certain KQML messages which it can relate to the status of certain nodes in the ACP process models. As the demonstration proceeds, the status of actions in the model progress from white (not yet ready to execute), to orange (ready to execute), then to green (executing) and finally blue (complete). The final column in the COA matrix is labelled "overall" and summarises the overall status of the COA creation and evaluation process.

The panel is written entirely in Java to form the basis for future Web-based process editors and activity control panels.

Application 3 – WOPlan

The aim of this application (Nixon, 2000; Nixon, Levine and Tate, 2000) was to create a mobile, limited media interface onto the O-Plan system. What was envisaged was the case of the mobile human agent, equipped with a small, hand-held wireless device, attempting to access a planning server in order to request some kind of course of action dependent on that user's current situation. Available web-based demonstrations of O-Plan (Tate, Dalton and Levine, 1998; Levine, Tate and Dalton, 2000) propose problem domains involving various military disaster relief and evacuation operations, and it was thought that a mobile telephone or *Personal Digital Assistant* (PDA) could be a tool for plan delivery to mobile units in such a situation. Alternatively, a lone mobile user could access O-Plan to retrieve a plan to assist in a situation in which that user had insufficient experience. Someone who had no experience of engineering, for example, could retrieve a checklist to perform in the event of their car breaking down. The utility of such a system would depend not only on the design of the system itself, but also on the identification of a suitable problem domain, in particular a domain in which the users of the planner are likely to be on the move and in need of a course of action to solve an immediate task, and otherwise with no access to more conventional interfaces such as PCs. The name *WOPlan* (for "Wireless O-Plan") was given to the system.

The design of a mobile interface onto O-Plan is made more difficult by the limited screen sizes of mobile devices, especially in the case of the mobile telephone. Development of interfaces onto O-Plan to date has allowed for the luxury of a full-sized terminal screen (Tate, Dalton and Levine, 1998; Levine, Tate and Dalton, 2000). Some issues of human-computer interaction which may not be critical when using a full-sized colour terminal interface become problematic in the case of the limited media interface. Generally users of mobile devices expect their interaction with the device to be brief, whereas a user sitting down at a workstation is prepared for a more prolonged session. Browsing with a mobile device, especially with a mobile telephone, is (with current devices) slow and cumbersome; data entry is difficult and should be kept to a minimum. A mobile telephone system needs only be slightly poorly designed to be rendered unusable; this is especially true if the system is attempting to serve long lists of data (such as delivering a plan description), as long pages increase download times and make navigation even slower and more frustrating.

WOPlan was developed as a web application which communicates with an instance of the core O-Plan engine and delivers *Wireless Mark-up Language* (WML) to a connected client. The client may be any device with has a browser which conforms to the *Wireless Application Protocol* (WAP), such as a WAP-enabled mobile telephone [6]. The WOPlan web application is a *Java Servlet*. In development and testing the *Nokia WAP Toolkit* WML browser emulator [7] was used in place of a physical WAP device, and the servlet was hosted within the *Tomcat Jakarta* webserver [8]. The client initiates a session by connecting to WOPlan, which connects to the O-Plan server and initialises the service, and provides the user (on the WAP device) with a list of available problem domains. The user is prompted to choose a planning *domain* (defined as a *Task Formalism* file), then choose a *task* within that domain, and then to view, execute or evaluate the resulting *plan*, or to get a different plan that fulfils the same specified task.

The architecture of the WOPlan system is shown in Figure 10. The *WAP Client* is the component with which the WOPlan user interacts. The user activates WOPlan by initiating an internet session on the WAP Client and navigating to the internet address of the second component, the WOPlan servlet. The WAP client could be any device with a WAP-enabled browser and internet connectivity, although WOPlan has only been tested in use with WAP emulator browsers running on a workstation. Features to notice are the very limited screen size (only four lines of text), and the user interface objects. Directly below the screen are two arrow buttons (used for scrolling up and down a WML page), in between which is a single `Select` button (used for selecting whatever item is currently highlighted in the WML page). Below and left of the screen is the `Options` button, which when available and selected should display a context-sensitive list of options. Below and right of the screen is the `Back` button, which when available and selected should navigate the user back to the previous screen.

The *WOPlan Servlet* sits between the WAP Client and the core O-Plan system. It accepts WAP requests from the client (or from multiple clients simultaneously) and communicates with O-Plan, initially connecting to O-Plan as required and sending and receiving messages through the the standard O-Plan Task Assignment interface [9]. The development work for the WOPlan system has focused on the WOPlan servlet; it is in this component that the logic specific to this implementation resides. The servlet dynamically creates WML pages, depending on the responses it is receiving from O-Plan, which are then sent to the WAP Client for browsing. These WML pages may themselves contain logic such as navigational directives or actions to perform after a certain length of time has passed. Although these directives are executed by the WAP Client, their source is the servlet.

The *O-Plan Server* sits in the bottom tier of the architecture, responding to requests from the WOPlan servlet.

In user trials (Nixon, 2000), it was found that WOPlan provides reasonably stable, scalable and usable access to the O-Plan system through a mobile telephone. Although it does not provide all of the functionality which O-Plan, and in particular the standard Task Assignment interface, has to offer, it provides a useful subset of this functionality, and has addressed the core issues of plan review and execution through the narrative and execution facilities (shown in Figures 11 and 12).

The investigation into the possible use of properties specific to mobile devices was a secondary aim of the project. Two such properties which have been discussed are voice technology and mobile positioning technology. No provision for voice or location integration exists in currently available WAP devices, although they will certainly become available in the near future. One possibility for the former would involve the use of *VoiceXML*, an XML variant intended to "make Internet content and information accessible via voice and phone". VoiceXML has the backing of industry giants IBM, AT\&T and Motorola. The Motorola *Mobile ADK* is a development environment for integrating VoiceXML and WML services [10]. The provision of *Location Services* (LCS) as a standard for mobile devices is still currently at the design stage. It is likely that some kind of service based on *Global Positioning System* (GPS) technology will be available to GSM (*Global System for Mobile Communications* – the current European standard) telephones in the near future.

The execution facility provided with this version of WOPlan is little more than a prototype, but it offers interesting possibilities for further development and research. Firstly it could certainly be improved, augmented and made more usable within the context of the WOPlan system. Perhaps more importantly, however, its simple, ordered, one-dimensional format, with action items emphasised according to what may done with those items, could provide a basic template for any system with a mobile limited media interface which is attempting to deliver courses of action (COAs) to human agents on the move.

SUMMARY

In this paper, we have argued that the World Wide Web should be seen as a place for "doing things" as well as "finding things". In order to do this, we need shared models and ontologies to represent plans, processes, activities, tasks and issues. We have described work towards this aim, concentrating on the <I-N-OVA> constraint model of activity and the <I-N-CA> constraint model of synthesised artefacts. These are designed to relate strengths from a number of different communities: the AI planning community with both its theoretical and practical system building interests; the issue-based design community, those interested in formal ontologies for processes and products; the standards community; those concerned with new opportunities in task achieving agents on the world wide web; etc. We have described three web-based applications which use such models and have been implemented to "do things" on the Web: the O-Plan Web demonstration, the Air Campaign Planning Process Panel (ACP^3), and O-Plan use via a WAP phone – WOPlan. In the future, we envisage many more such applications, with the possibility that the individual planning applications can communicate with each other using the <I-N-OVA> issue-based constraint-based models of activity described in this paper.

Acknowledgements

The O-Plan and I-X projects are sponsored by the Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory Command and Control Directorate under grant number F30602-99-1-0024 and the UK Defence Evaluation Research Agency (DERA). The US Government, DERA and the University of Edinburgh are authorised to reproduce and distribute reprints 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 official policies or endorsements, either express or implied, of DARPA, the Air Force Research Laboratory, the US Government, DERA, or the University of Edinburgh.

References

Aitken, S. and Tate, A. (1997). Process Modelling of the TIE 97-1 Demonstration: Modelling Complex Techniques Using ACP Terminology. ISAT Technical Report ISAT-AIAI/TR/6, Version 1, December 1997.

Currie, K.W. and Tate, A. (1991). O-Plan: the Open Planning Architecture. *Artificial Intelligence* Vol. 51, No. 1, Autumn 1991, North-Holland.

Drabble, B., Lydiard, T. and Tate, A. (1997). Process Steps, Process Product and System Capabilities. ISAT Technical Report ISAT-AIAI/TR/4, Version 2, April 1997.

Finin, T., Labrou, Y. and Mayfield, J. (1997). KQML as an Agent Communication Language. In Bradshaw, J. (ed.) *Software Agents*, 291–316. MIT Press, Cambridge, MA.

Khambhampati, S. and Srivastava, B. (1996). Unifying Classical Planning Approaches. Arizona State University ASU CSE TR 96-006, July 1996.

Levine, J., Tate A. and Dalton, J. (2000). O-P^3: Supporting the Planning Process using Open Planning Process Panels. *IEEE Intelligent Systems* Vol. 15, No. 5, September/October 2000, 56–62.

Nixon, A. (2000). Limited Media Interface for AI Planning System. M.Sc. project dissertation, Division of Informatics, University of Edinburgh, September 2000.

Nixon, A., Levine, J. and Tate, A. (2000). Limited Media Interface for AI Planning System. *Proceedings of the 19th Workshop of the UK Planning and Scheduling Special Interest Group (PLANSIG 2000)*, Open University, Milton Keynes, UK.

Polyak, S. and Tate, A. (1998). A Common Process Ontology for Process-Centred Organisations. *Knowledge Based Systems*, to appear. Earlier version published as University of Edinburgh Department of Artificial Intelligence Research paper 930, 1998.

Reece, G.A., Tate, A., Brown, D. and Hoffman, M. (1993). The PRECiS Environment. Paper presented at the ARPA-RL Planning Initiative Workshop at AAAI-93, Washington D.C., July 1993.

Tate, A. (1993). Authority Management – Coordination between Planning, Scheduling and Control. Workshop on Knowledge-based Production Planning, Scheduling and Control at the International Joint Conference on Artificial Intelligence (IJCAI-93), Chambery, France, 1993.

Tate, A. (1994). Mixed Initiative Planning in O-Plan2. *Proceedings of the ARPA/Rome Laboratory Planning Initiative Workshop*, Tucson, Arizona, USA. Morgan Kaufmann, Palo Alto, CA.

Tate, A., Drabble, B. and Dalton, J. (1994). Reasoning with Constraints within O-Plan2. *Proceedings of the ARPA/Rome Laboratory Planning Initiative Workshop*, Tucson, Arizona, USA. Morgan Kaufmann, Palo Alto, CA.

Tate, A. (ed.) (1996a). *Advanced Planning Technology - Technological Achievements of the ARPA/Rome Laboratory Planning Initiative (ARPI)*. AAAI Press, Menlo Park, CA.

Tate, A. (1996b). Representing Plans as a Set of Constraints – the <I-N-OVA> Model. *Proceedings of the Third International Conference on Artificial Intelligence Planning Systems (AIPS-96)*, 221–228. AAAI Press, Menlo Park, CA.

Tate, A., Drabble, B. and Dalton, J. (1996). O-Plan: a Knowledge-Based Planner and its Application to Logistics. In Tate, A. (1996a), 259–266.

Tate, A. (1998). Roots of SPAR - Shared Planning and Activity Representation. *Knowledge Engineering Review* Vol. 13, No. 1, March 1998. Cambridge University Press, Cambridge, UK. See also http://www.aiai.ed.ac.uk/project/spar/

Tate, A., Dalton, J. and Levine, J. (1998). Generation of Multiple Qualitatively Different Plan Options. *Proceedings of Fourth International Conference on AI Planning Systems (AIPS-98)*, Pittsburgh, USA, 27–34. AAAI Press, Menlo Park, CA.

Footnotes

[1] http://www.aiai.ed.ac.uk/project/spar/

[2] <I-N-OVA> is pronounced as in "Innovate".

[3] Temporal (or spatio-temporal) and object constraints are cross-constraints specific to the planning task. The cross-constraints in some other domain may be some other constraint type.

[4] I-X is the successor project to O-Plan – see http://www.aiai.ed.ac.uk/project/ix/

[5] You can try this demonstration out yourself at http://www.aiai.ed.ac.uk/project/oplan/

[6] See http://www.wapforum.org/

[7] See http://www.forum.nokia.com/wapforum/

[8] See http://jakarta.apache.org/tomcat/

[9] This is the standard API provided for external programs to communicate with O-Plan.

[10] See http://www.motorola.com/spin/mix/faqs.html

Figure Captions

Figure 1: Uses of <I-N-OVA> and <I-N-CA>

Figure 2: <I-N-OVA> Constraint Model of Activity

Figure 3: I-X - Two Cycles of Processing - Handle Issues, Respect Constraints

Figure 4: Using O-P^3 Interfaces

Figure 5: Communication between TA and Planner

Figure 6: Roles of the Task Assigner and the Planner

Figure 7: O-Plan Task Assigner's Panel

Figure 8: O-Plan Planner User's Panel

Figure 9: The ACP^3 Viewer

Figure 10: Architecture of *WOPlan*

Figure 11: Example of WOPlan Narrative Display

Figure 12: Example of WOPlan Execution Display

Title:
tate-fig1.dvi
Creator:
dvips(k) 5.82 Copyright 1998 Radical Eye Software
Preview:
This EPS picture was not saved
with a preview included in it.
Comment:
This EPS picture will print to a
PostScript printer, but not to

Figure 1: Uses of <I-N-OVA> and <I-N-CA>

```
Plan Constraints
   I   - Issues (Implied Constraints)
   N   - Node Constraints (on Activities)
   OVA - Detailed Constraints
         O - Ordering Constraints
         V - Variable Constraints
         A - Auxiliary Constraints
             - Authority Constraints
             - Condition Constraints
             - Resource Constraints
             - Spatial Constraints
             - Miscellaneous Constraints
```

Figure 2: <I-N-OVA> Constraint Model of Activity

Title:
/hame/johnl/00-DAGSTUHL/ix-two-cycles.ps
Creator:
XV Version 3.10a Rev: 12/29/94 - by John Bradley
Preview:
This EPS picture was not saved
with a preview included in it.
Comment:
This EPS picture will print to a
PostScript printer, but not to
other types of printers.

Figure 3: I-X - Two Cycles of Processing - Handle Issues, Respect Constraints

Title:
web-interfaces.EPSF
Creator:
GraphicConverter
Preview:
This EPS picture was not saved
with a preview included in it.
Comment:
This EPS picture will print to a
PostScript printer, but not to
other types of printers.

Figure 4: Using O-P^3 Interfaces

Title:
f-2-people.fig
Creator:
fig2dev
Preview:
This EPS picture was not saved
with a preview included in it.
Comment:
This EPS picture will print to a
PostScript printer, but not to
other types of printers.

Figure 5: Communication between TA and Planner

Title:
ta-to-plan.epsf
Creator:
GraphicConverter
Preview:
This EPS picture was not saved
with a preview included in it.
Comment:
This EPS picture will print to a
PostScript printer, but not to
other types of printers.

Figure 6: Roles of the Task Assigner and the Planner

Title:
a4-ta.epsf
Creator:
GraphicConverter
Preview:
This EPS picture was not saved
with a preview included in it.
Comment:
This EPS picture will print to a
PostScript printer, but not to
other types of printers.

Figure 7: O-Plan Task Assigner's Panel

Title:
a4-planner.epsf
Creator:
GraphicConverter
Preview:
This EPS picture was not saved
with a preview included in it.
Comment:
This EPS picture will print to a
PostScript printer, but not to
other types of printers.

Figure 8: O-Plan Planner User's Panel

Title:
acp3-step4-a.epsf
Creator:
GraphicConverter
Preview:
This EPS picture was not saved
with a preview included in it.
Comment:
This EPS picture will print to a
PostScript printer, but not to
other types of printers.

Figure 9: The ACP^3 Viewer

Title:
woplan_architecture.eps
Creator:
fig2dev Version 3.2 Patchlevel 0-beta3
Preview:
This EPS picture was not saved
with a preview included in it.
Comment:
This EPS picture will print to a
PostScript printer, but not to
other types of printers.

Figure 10: Architecture of *WOPlan*

Title:
(narr1.eps)
Creator:

Preview:
This EPS picture was not saved
with a preview included in it.
Comment:
This EPS picture will print to a
PostScript printer, but not to
other types of printers.

Figure 11: Example of WOPlan Narrative Display

Title:
(exec1.eps)
Creator:

Preview:
This EPS picture was not saved
with a preview included in it.
Comment:
This EPS picture will print to a
PostScript printer, but not to
other types of printers.

Figure 12: Example of WOPlan Execution Display