AFRL-IF-RS-TR-1999-60
Final Technical Report
April 1999

MULTI-PERSPECTIVE PLANNING - Using Domain Constraints to Support the Coordinated Development of Plans

The University of Edinburgh

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. C690

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

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

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

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

AFRL-IF-RS-TR-1999-60 has been reviewed and is approved for publication.

APPROVED: Northrup Fowler

NORTHRUP FOWLER III
Project Engineer

FOR THE DIRECTOR: Eugene C. Blackburn

EUGENE C. BLACKBURN, Chief
Information Technology Division
Information Directorate

MULTI-PERSPECTIVE PLANNING -
Using Domain Constraints to Support the Coordinated Development of Plans

Austin Tate
Jeff Dalton
John Levine

Contractor: University of Edinburgh
Contract Number: F30602-95-C-0022
Effective Date of Contract: 15 June 1995
Contract Expiration Date: 30 September 1998
Short Title of Work: Multi-Perspective Planning

Period of Work Covered: Jun 95 - Sep 98

Principal Investigator: Austin Tate
Phone: UK (+44) 31 650 2732
AFRL Project Engineer: Northrup Fowler III
Phone: (315) 330-3011

This research was supported by the Defense Advanced Research Projects Agency of the Department of Defense and was monitored by Northrup Fowler III, AFRL/IFT, 525 Brooks Road, Rome, NY 13441-4505.

REPORT DOCUMENTATION PAGE			*Form Approved* *OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	April 1999	Final Jul 95 - Sep 98

4. TITLE AND SUBTITLE
MULTI-PERSPECTIVE PLANNING - Using Domain Constraints to Support the Coordinated Development of Plans

5. FUNDING NUMBERS
C - F30602-95-1-0022
PE - 61101E and 62702F
PR - C690
TA - 00
WU - 79

6. AUTHOR(S)

Austin Tate, Jeff Dalton and John Levine

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

University of Edinburgh
Artificial Intelligence Applications Institute
80 South Bridge
Edinburgh EH1 1HN, United Kingdom

8. PERFORMING ORGANIZATION REPORT NUMBER
AIAI-TR-231

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

Defense Advanced Research Projects Agency
3701 North Fairfax Drive
Arlington VA 22203-1714

Air Force Research Laboratory/IFT
525 Brooks Road
Rome NY 13441-4505

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-IF-RS-TR-1999-60

11. SUPPLEMENTARY NOTES

Air Force Research Laboratory Project Engineer: Northrup Fowler III/IFT/(315) 330-3011

12a. DISTRIBUTION AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The project investigated multi-agent mixed initiative interaction between a "task assignment" or "command" agent and a planning agent. Each agent maintains an agenda of outstanding tasks they are engaged in and uses a common representation of tasks, plans, processes and activities based on the notion that these are all "constraints on behavior". Interaction between the agents uses an exchange of explicit task and option management information and explicit agent authorization.

The project has provided a Web-based demonstration of a Course of Action (COA) comparison matrix being used as an interface to an O-Plan plan server to explore multiple qualitatively different plan options. The scenario used for this demonstration is concerned with crisis operations on the island of Pacifica. The interface allows two users acting in designated user roles to maintain their own views on the COA comparison matrix. The two users work together to explore and evaluate several different plan options based on different command-level requirements and different assumptions about the environmental conditions. This work is part of a larger effort to build a comprehensive mixed initiative command, planning and execution support system incorporating human users in designated user roles.

14. SUBJECT TERMS

Planning, Constraint Management, Artificial Intelligence

15. NUMBER OF PAGES
238

16. PRICE CODE

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

Acknowledgements

This research was jointly sponsored by the Defense Advanced Research Projects Agency of the Department of Defense and by the US Air Force Research Laboratory at Rome, and was monitored by Northrup Fowler III, RL (C3C), 525 Brooks Rd., Griffiss AFB, NY 13441-4505, USA.

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Advanced Research Projects Agency or the US Government. The United States Government is authorised to reproduce and distribute reprints of this paper and each of the appendices for government purposes notwithstanding any copyright notation hereon.

The O-Plan project began in 1983. Since that time the following people have participated: Stuart Aitken, Howard Beck, Colin Bell, Ken Currie, Jeff Dalton, Roberto Desimone, Brian Drabble, Mark Drummond, Anja Haman, Peter Jarvis, Ken Johnson, John Kingston, Richard Kirby, John Levine, Steve Polyak, Glen Reece, Arthur Seaton, Judith Secker, Austin Tate, Richard Tobin and Gerhard Wickler.

Prior to 1984, work on Interplan (1972-4) and Nonlin (1975-6) was funded by the UK Science and Engineering Research Council.

From 1984 to 1988, the O-Plan project was funded by the Science and Engineering Research Council on grant numbers GR/C/59178 and GR/D/58987 (UK Alvey Programme project number IKBS/151). The work was also supported by a fellowship from SD-Scicon for Austin Tate from 1984 to 1985.

From 1989 to 1992, the UK Science and Engineering Research Council (grant number GR/F36545 - UK Information Engineering Directorate project number IED 4/1/1320) funded a collaborative project with ICL, Imperial College and other partners in which the O-Plan architecture was used to guide the design and development of a planner with a flexible temporal logic representation of the plan state.

From 1989 to 1992, the O-Plan project was supported by the US Air Force Rome Laboratory through the Air Force Office of Scientific Research (AFOSR) and their European Office of Aerospace Research and Development by contract number F49620-89-C-0081 (EOARD/88-0044) monitored by Northrup Fowler III at the US Air Force Rome Laboratory.

From 1989 to 1993, research on scheduling applications of the O-Plan architecture was funded by Hitachi Europe Ltd. A number of other research and development contracts placed with AIAI have led to research progress on O-Plan.

From 1992 to 1998, the O-Plan project was supported by the Defense Advanced Research Projects Agency (DARPA) and the US Air Force Research Laboratory (AFRL - formerly Rome Laboratory) Knowledge Based Planning and Scheduling Initiative (known as ARPI). The contracts were managed through the US Air Force Research Laboratory at Rome, the Air Force Office of Scientific Research (AFOSR) and the European Office of Aerospace Research and Development by contract numbers F49620-92-C-0042 (EOARD/92-0001) and F-30602-95-1-0022 monitored by Northrup Fowler III at AFRL.

For the period 1994-1999, support to students was provided by Augmentation Awards for Science and Engineering Research Training (AASERT) provided by the Air Force Office of Scientific Research (AFOSR) under grant numbers F49620-93-1-0436 (for Glen Reece) and F49620-96-1-0348 (for Steve Polyak) monitored by Abe Waksman at AFOSR. A studentship was also included in grant number F-30602-95-1-0022 to support Gerhard Wickler.

Additional resources for the O-Plan projects have been provided by the Artificial Intelligence Applications Institute at the University of Edinburgh.

Abbreviations

The following abbreviations are used within the report. This section serves as a reminder of their meaning wherever the context is not clear.

ACP^3 Air Campaign Planning Process Panel – a Java-based Open Planning Process Panel (O-P^3) for ARPI TIE 97-1

ARPI DARPA/Air Force Research Laboratory (Rome) Planning Initiative – earlier called the ARPA/Rome Laboratory Planning Initiative – the Knowledge-based Planning and Scheduling Initiative research and development programme under which the research reported on was funded.

CGI Common Gateway Interface – a mechanism by which a URL is handled by running a program on the server machine.

CLOS Common Lisp Object System – an extension to Common Lisp allowing object-oriented programming.

CM Constraint Manager – a module for handling constraints of a particular type.

COA Course of Action – military terminology for a particular plan option for some given task and assuming certain constraints.

CPE Common Process Editor – a Java-based tool designed to support process management and process translation.

DARPA Defense Advanced Research Projects Agency – earlier called ARPA, the Advanced Research Projects Agency.

EE Elements of Evaluation – criteria used to evaluate a COA.

GOST Goal Structure Table – used to hold conditions associated with a plan and their method of satisfaction.

GPDT Go Places and Do Things – a crisis operations planning domain on the fictional island of Pacifica.

HTML Hypertext Markup Language – a markup language used to define documents on the World Wide Web.

HTTP Hypertext Transfer Protocol – the protocol by which Web pages are sent from the server machine to the client machine.

IFD Integrated Feasibility Demonstrator – used to demonstrate ARPI technologies on military relevant problems.

<I-N-OVA> Issues, Nodes, Orderings, Variables, Auxiliary Constraints Model – used to represent constraints on activity or plans.

KS Knowledge Source – a computational capability in O-Plan.

MTC Modal Truth Criterion – another name adopted by other researchers for a process similar to Question Answering (QA).

NEO Non-combatant Evacuation Operations – military operations to evacuate civilians from a danger zone.

O-P^3 Open Planning Process Panels – panels giving multiple users visualisation of and control over the planning process.

O-Plan The Open Planning Architecture.

PERT Project Evaluation and Review Technique – a way of visualising a plan as a network of tasks.

PMO Plan Modification Operator – a term used to describe the abstract operation of O-Plan in which partially-specified plans are modified by "Operators" during the search for a solution to a given task. PMOs correspond to Knowledge Sources in O-Plan.

PSV Plan State Variable – an object in a plan which is not fully defined.

PRECiS Planning, Reactive Execution and Constraint Satisfaction domain – an experimental application domain to allow demonstration and evaluation of systems for planning, scheduling, constraint satisfaction and reactive plan execution. This domain involves NEOs from the fictional island of Pacifica.

QA Question Answering – the O-Plan support routine which finds the ways in which a plan condition can be satisfied.

TA Task Assigner – a human agent who is concerned with developing briefings on alternative course of action.

TIE Technology Integration Experiment – an experiment to join together two or more technologies from the ARPI to evaluate some given objective.

TF Task Formalism – the domain description language for the O-Plan planner.

TGM TOME/GOST Manager – the Constraint Manager in O-Plan which looks after effects and conditions.

TOME Table Of Multiple Effects – used to hold effects associated with a plan.

TPN Time Point Network – used to hold time points associated with a plan and constraints between these time points.

TPNM Time Point Network Manager – the Constraint Manager in O-Plan which builds and looks after the TPN.

URL Universal Resource Locator – an address giving the location of a document on the World Wide Web.

Contents

Appendices: Attached Papers

List of Figures

1 Executive Summary

An important planning capability is the generation and refinement of alternative Courses of Action (COAs) to respond to a developing crisis requiring military intervention. This research addressed two key areas of importance to the military planning community which also pose significant challenges for the AI planning community:

1. generation of multiple qualitatively different courses of action dependent upon alternative assumptions concerning the emerging crisis;
2. support for mixed initiative plan development, manipulation and use dependent upon different assumptions concerning the level of response to be made and the levels of assets to be assigned.

These generic tasks are vital to support initial generation of COAs and their subsequent refinement, analysis, comparison, selection and use in planning for crisis response in situations such as Non-combatant Evacuation Operations (NEOs) and Air Campaigns – two of the target domains for the DARPA/Air Force Research Laboratory Planning Initiative (ARPI).

The work addressed the development of plans from a number of different *perspectives.* It furthered a number of developments in task specification, knowledge rich plan representation, plan constraint manipulation and explicit workflow management to coordinate user and system input to the planning process. The 4 main technical themes of the project are summarised in figure 1. A combination of these techniques has be demonstrated in realistic scenarios related to NEOs and Air Campaign Planning using domain materials within the ARPI suited to demonstration and evaluation.

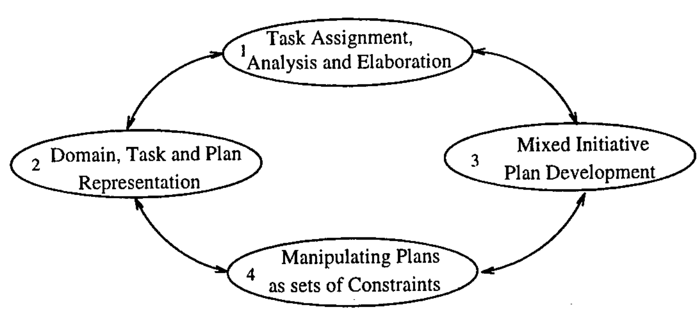


Figure 1: Principal Themes of the Research

The O-Plan Architecture and related technology formed the basis for the work. O-Plan can make use of domain constraint knowledge to direct its search for plans. O-Plan technology has now reached "critical mass" where experience gained with realistic applications points the way towards an approach which can address real problems of concern to the US military.

2 Introduction

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, evaluated and refined, 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 O-Plan Web demonstration illustrates this view. It shows two human agents acting in designated user roles working together with O-Plan to solve a hard planning problem. Decisions have to be made at all levels: the Task Assigner needs to decide what an appropriate response to a crisis is, the Planner User needs to find a plan which addresses the Task Assigner's requirements as closely as possible and O-Plan needs to choose actions and assign resources which satisfy the Task Assigner's COA requirements and which are in keeping with the Planner User's additional constraints and advice.

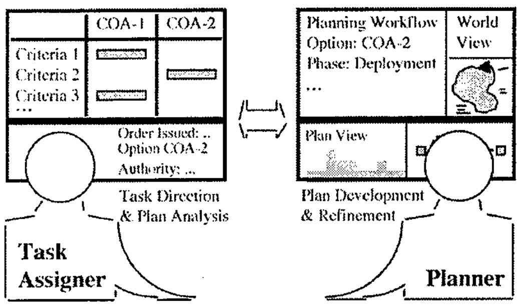
This report describes the work carried out under the O-Plan project. We have constructed a Web-based demonstration of two human agents and one software planning agent working together to populate and explore different options within a Course of Action matrix. The two human agents act in given user roles as Task Assigner (i.e. commander) and Planner User (i.e. planning staff member). Each user is provided with an interface which supports the workflow of the planning process and which facilitates the comparison and visualisation of multiple courses of action according to multiple elements of evaluation. For the demonstration scenario, we are using a general-purpose logistics and crisis operations domain which is an extension of our earlier logistics-related domains based on the fictional island of Pacifica [16, 29].

2.1 Original Aims

Under the University of Edinburgh O-Plan Project [4, 28] which is part of the DARPA/Air Force Research Laboratory (Rome) Planning Initiative – ARPI [11, 21] we are exploring mixed initiative planning methods and their application to realistic problems in logistics, air campaign planning and crisis action response [29] [**see Appendix A**]. In preparatory work, O-Plan has been demonstrated operating in a range of mixed initiative modes on a Non-Combatant Evacuation Operation (NEO) problem [7, 19]. A number of "user roles" were identified to help clarify some of the types of interaction involved and to assist in the provision of suitable support to the various roles [19].

The research and development in this project addressed the development and refinement of a number of alternative Courses of Action (COAs). The COAs are based on different assumptions concerning a developing crisis, the assets to be committed to the mission and the level of response which is being considered. The overall O-Plan research is concerned with three levels: (a) task assignment and mission description; (b) planning; and (c) plan execution support. The current project concentrated on the first two levels and the communication between them. The approach assumed high levels of user interaction within the COA development process, but

CONCEPT



NEW IDEAS

- **Using domain constraints to support the coordinated development of plans.**
- **Communication between users acting as Task Assigner and Planner.**
- **Intelligent workflow model of planning based on "issue" handling (agenda/to-do list).**
- **Simplified planner interfaces to allow "plug and plan" component integration.**
- **Uniform manipulation of plans as a set of constraints (<I-N-OVA> model).**

IMPACT

- **Generation of multiple qualitatively distinct alternative COAs dependent upon alternative assumptions concerning the emerging crisis.**
- **Support for mixed-initiative incremental plan development, manipulation and use.**
- **Promotion of intelligent process management and workflow concepts.**
- **Integration framework for large-scale modular planning systems.**
- **Contribution to shared plan representations.**

MILESTONES

Q4/Y1
- **MIP Demonstration in Pacifica NEO**
- **Initial Evaluation Matrix**
- **Demonstration scenario development**

Q4/Y2
- **Workflow Planning Aid Experiments**
- **TIE with Rochester on Tasking**
- **Release of O-Plan Version 3.1**
- **Evaluation Experiments Interim Report**

Q4/Y3
- **Mixed Initiative Planning Demo.**
- **TIE with USC/ISI on plan evaluation**
- **Release of O-Plan Version 3.2**
- **Final Evaluation Report**

Figure 2: Original Aims of the O-Plan Project - "Quad Chart"

allowed for system initiative where appropriate.

The original project proposal (see summary "Quad Chart" in figure 2) presented a simplified view of the interaction between two people who have different roles in the planning process (see figure 3). These are:

Task Assigner Role: a person supporting the Commander-in-Chief (CINC) who is concerned with developing briefings on alternative COAs and ensuring that recommendations on appropriate responses to an emerging crisis are available for selection and subsequent execution.

Planner Role: a person developing and refining one or more COAs to a given level of detail within a brief specified by the Task Assigner.

Each person is supported notionally by a separate computer system and interaction between the two is in terms of "planning task workflow".

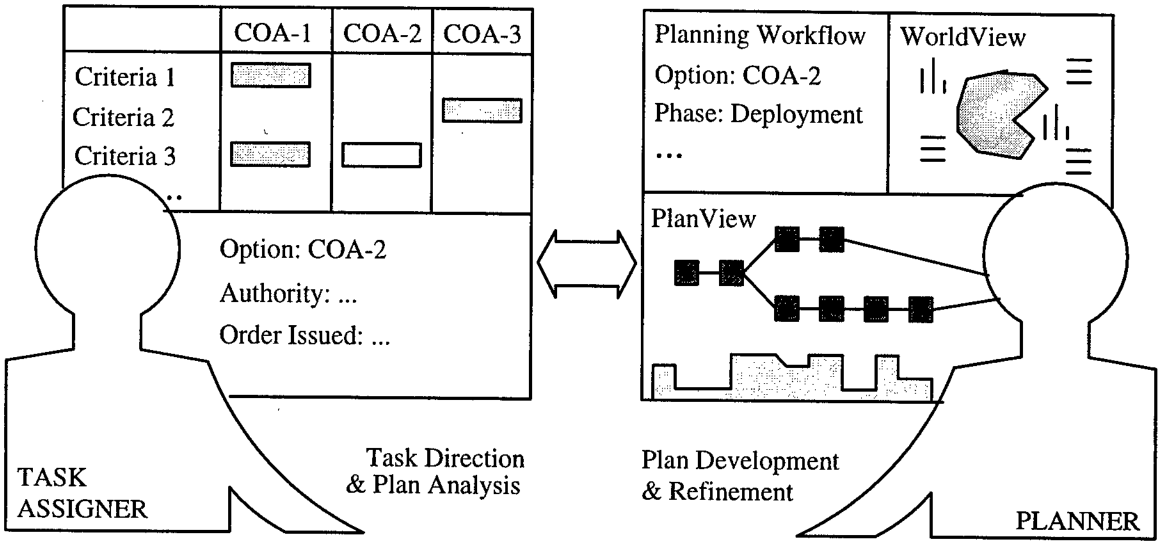


Figure 3: Communication between the Task Assigner and the Planner User

2.2 Mixed Initiative COA Development

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 in given roles to populate a Course of Action (COA) / Elements of Evaluation comparison matrix. The columns of this matrix are alternative options being explored as a potential solution to a (possibly underspecified) problem and the rows are evaluations of the solution being considered. The idea is that the requirements, assumptions and constraints are all refined concurrently using the elements of evaluation (EEs).

We are exploring the links between key user roles in the planning process and automated planning support aids [24, 25] [see **Appendix G**]. The research is exploring a planning workflow control model using:

- a shared model of mixed initiative planning as "mutually constraining the space of behaviour";
- the <I-N-OVA> constraint model of activity as the basis for plan communication;
- explicit management between agents of the tasks and options being considered;
- agent agendas and agenda issue handlers;
- explicit provision of authority for an agent to perform its functions.

Agents maintain their own perspective of their part in the task to hand, while cooperating with other agents who may perform parts of the task.

We envisage two human agents, called the Task Assigner and the Planner, working together to explore possible solutions to a problem and making use of automated planning aids to do

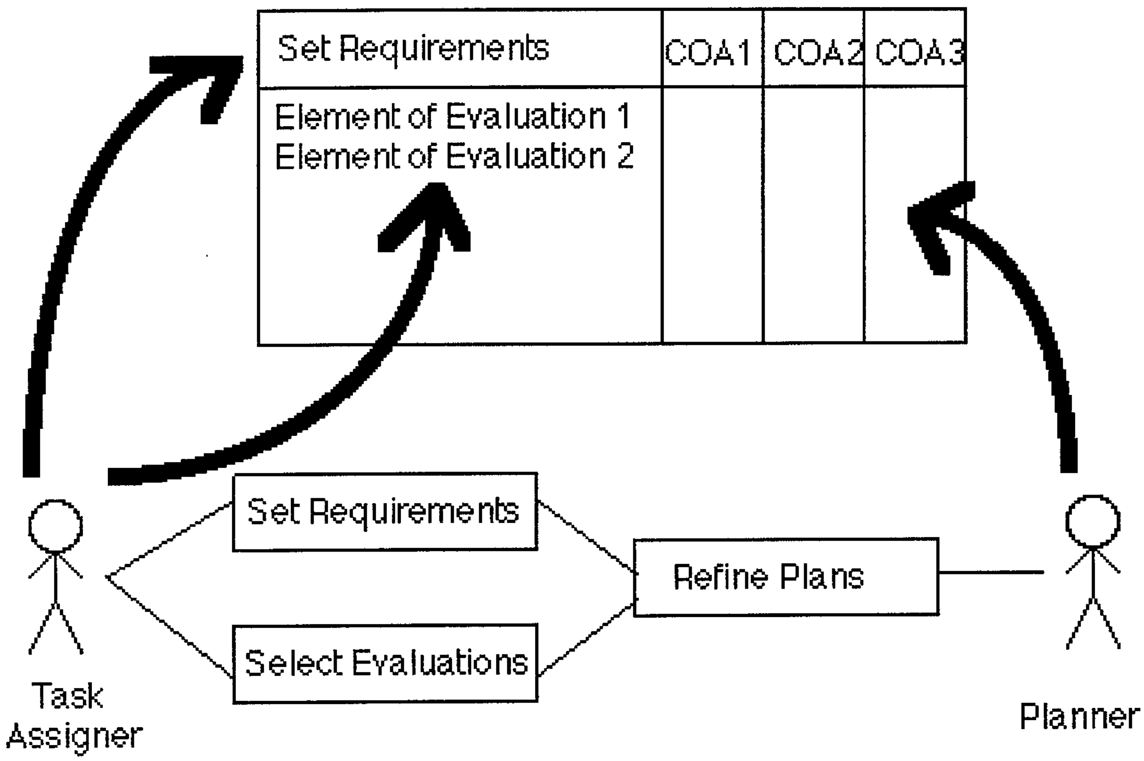


Figure 4: Roles of the Task Assigner and the Planner User

this. Figure 4 shows how the two human agents cooperate to populate the COA comparison matrix. The Task Assigner sets the requirements for a particular Course of Action (i.e. what top level tasks must be performed) and selects appropriate evaluation criteria (elements of evaluation) for the resulting plans. The Planner agent acts to refine the resulting plans by adding further constraints and splitting plans to explore two or more possible options for the same COA requirements.

2.3 Demonstration and Evaluation Scenarios

The basis for our understanding of the domain requirements has come from work on the ARPI IFD-2[1] Tunisian scenario, the IFD-3 Sri Lanka NEO scenario and dramatisations, the Joint Staff Officer's Guide from the Armed Forces Staff College (the "Purple Book", [2]), and our work in developing the Pacifica NEO evaluation domain along with Mitre and ISX Corporation [16, 29]. Discussions were also held with US Transportation Command (USTRANSCOM) and Military Traffic Management Command (MTMC) to understand the availability and format of plan process knowledge for such domains.

[1]IFD is an Integrated Feasibility Demonstration.

2.4 Shared Plan Representation

A shared representation for tasks and plans is vital to the success of integrated planning, scheduling and resource management tools. It can enable shared plan use across components of the ARPI and influence future planning aids for the military and other industry sectors. The research has played a large part in progressing the development of a shared planning ontology and plan representations suited to dual use across many types of organisations involved in planning.

2.5 Structure of the Report

Section 3 describes the fundamental science behind O-Plan: the Open Planning Architecture, task and option management, representing plans as a set of constraints on behaviour, the notion of authority to plan, and mixed-initiative planning between agents by mutually constraining the space of future behaviour.

Section 4 describes the O-Plan technology that was used to implement the O-Plan two-user Web demonstration. The first part of this section introduces Open Planning Process Panels (O-P^3) and the Web-based COA evaluation matrices used for the O-Plan demonstration. The section then goes on to report on some of the O-Plan technology elements which were used to implement the Web demonstration.

Section 5 gives the details of the demonstration scenario and the general-purpose crisis operations domain used to demonstrate this work. This section includes a detailed scenario storyboard showing how the two human users work together using O-Plan technology to construct alternative courses of action to respond to a developing crisis situation.

Section 6 evaluates this work. Six different evaluation criteria have been used: meeting our stated vision from the project proposal, an evaluation matrix of domain features and planning technology elements, a consideration of good and bad domains for O-Plan, a critical evaluation of the two-user Web demonstration, a set of scaling experiments carried out on the Task Formalism file for the crisis operations domain, and an assessment of the impact of O-P^3 technology.

Section 7 summarises what has been achieved and assesses the potential impact of this work.

3 O-Plan – the Open Planning Architecture

This section describes the O-Plan architecture and the structure of individual O-Plan agents. Further information on the O-Plan architecture and its use in various applications is given in [3] [**see Appendix B**].

3.1 Generic Systems Integration Architecture

The O-Plan agent architecture to be described in the next section is a specific variant of a generalised systems integration architecture shown in figure 5. This general structure has been adopted on a number of AIAI projects [12]. The architecture is an example of a *Model/Viewer/Controller* arrangement.

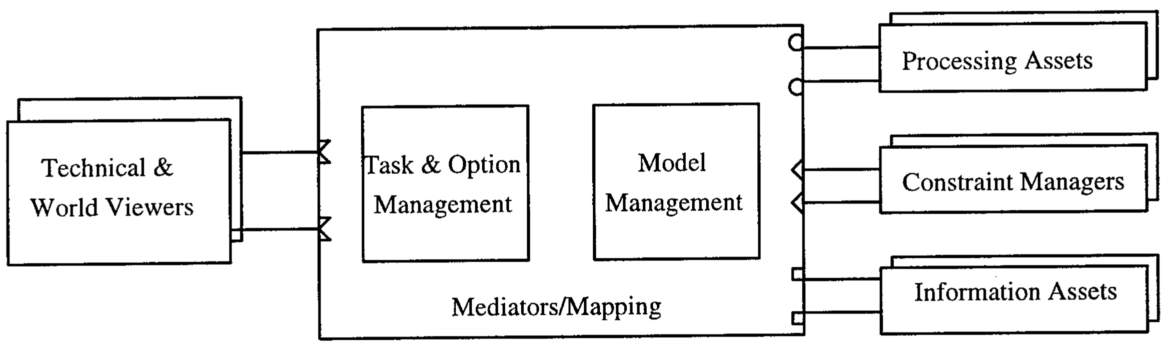


Figure 5: Generic Systems Integration Architecture

The various components "plug" into "sockets" within the architectural framework. The sockets are specialised to ease the integration of particular types of component. The components are as follows:

Viewers: user interface, visualisation and presentation viewers for the model. sometimes differentiated into *technical* model views (e.g. charts, structure diagrams) and *world* model views (e.g. simulations, animations).

Task and Option Management: the capability to support user tasks via appropriate use of the processing and information assets and to assist the user in managing options being used within the model. This is sometimes referred to as the *Controller*.

Model Management: coordination of the capabilities/assets to represent, store, retrieve, merge, translate, compare, correct, analyse, synthesise and modify models.

Mediators: Intermediaries or converters between the features of the model and the interfaces of active components of the architecture. (such as viewers, processing assets, constraint managers and information assets).

Processing Assets: functional components (model analysis, synthesis or modification).

Constraint Managers: components which assist in the maintenance of the consistency of the model.

Information Assets: information storage and retrieval components.

3.2 The O-Plan Architecture

The components of a single O-Plan agent are shown in figure 6.

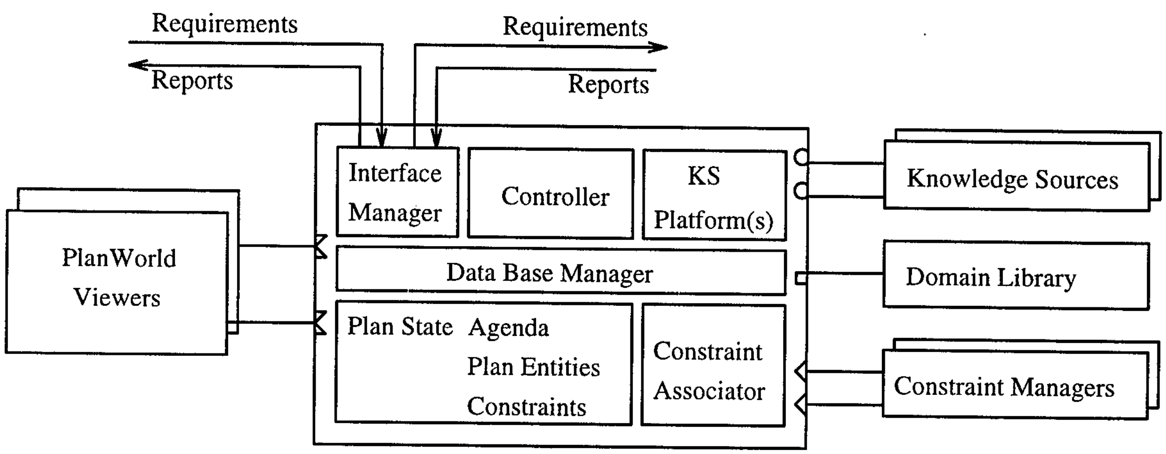


Figure 6: O-Plan Agent Architecture

3.3 Task and Option Management

Task and option management facilities are provided by the *Controller* in O-Plan. The O-Plan Controller takes its tasks from an agenda which indicates the outstanding processing required and handles these with its *Knowledge Sources.*

O-Plan has explicit facilities for managing a number of different options which it is considering. O-Plan has an agent level agenda, and agendas which relate to each option it is considering (in fact, as we shall see later, these are part of the plan representation for these options – the *issues* or I part of <I-N-OVA>). Many of these options are internal to the planning agent, and are generated during the search for a solution. Others are important for the interaction between the planner and a user acting as a task assigner.

3.4 Abstract Model of Planning Workflow – Plan Modification Operators

A general approach to designing AI-based planning and scheduling systems based on partial plan or partial schedule representations is to have an architecture in which a plan or schedule is critiqued to produce a list of issues or agenda entries which is then used to drive a workflow-style processing cycle of choosing a "plan modification operator" (PMO) to handle one or more

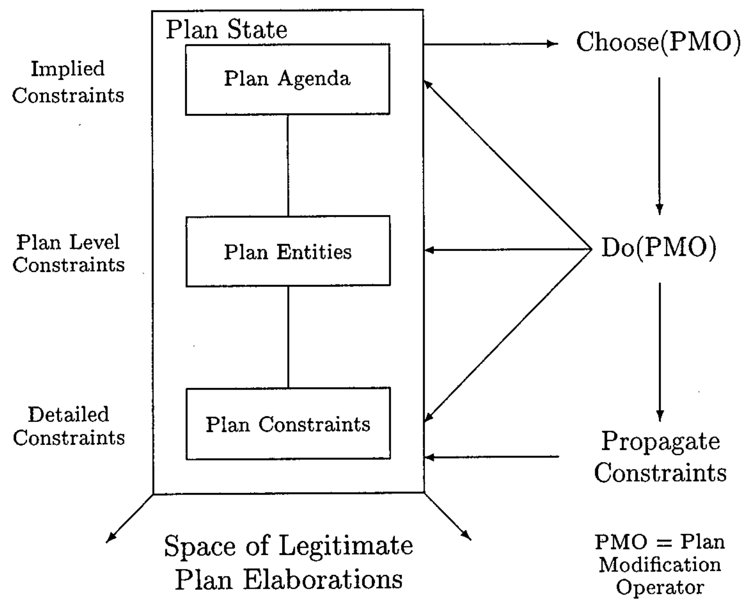


Figure 7: Planning Workflow – Using Handlers for Agenda Issues

agenda issues and then executing the PMO to modify the plan state. Figure 7 shows this graphically.

This approach is taken in O-Plan. The approach fits well with the concept of treating plans as a set of constraints which can be refined as planning progresses. Some such systems can act in a non-monotonic fashion by relaxing constraints in certain ways. Having the implied constraints or "agenda" as a formal part of the plan provides an ability to separate the plan that is being generated or manipulated from the planning system itself.

3.5 Representing Plans as a Set of Constraints on Behaviour

The <I-N-OVA> (*Issues – Nodes – Orderings / Variables / Auxiliary*) Model is a means to represent and manipulate plans 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.

Work on the O-Plan Project has led to an ontology [22] [**see Appendix D**] for activities, processes and plans which has been used as input to a range of efforts intended to standardise the terminology and concepts used for planning and activity management. This has included input to the Workflow Management Coalition's Process Description Language, DARPA's work on the Process Interchange Format, The National Institute of Standards and Technology's Process Specification Language, The Object Model Working Group (OMWG) Core Plan Representation, the AITS WarPlan Effort, and, most recently, DARPA's efforts towards a Shared Planning and Activity Representation (SPAR) [27] [**see Appendix F**].

Using the ontological basis explained in [22], the <I-N-OVA> model [23] **[see Appendix E]** has been developed to characterise the plan representation used within O-Plan The <I-N-OVA> work is related to emerging formal analyses of plans and planning. This synergy of practical and formal approaches can stretch the formal methods to cover realistic plan representations as needed for real problem solving, and can improve the analyses that are possible for production planning systems.

<I-N-OVA> is 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 is intended to support new work on automatic manipulation of plans, human communication about plans, principled and reliable acquisition of plan information, and formal reasoning about plans. 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.

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 8: <I-N-OVA> Constraint Model of Activity

The node constraints (these are often of the form "include activity") in the <I-N-OVA> model create the space within which a plan may be further constrained. The I (issues) and OVA constraints restrict the plans within that space to those which are valid. Ordering (temporal) and variable constraints are distinguished from all other auxiliary constraints since these act as *critical constraints* or *cross-constraints*[2] 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.

The <I-N-OVA> constraint model of activity allows planning process state as well as the current state of the plan generated to be communicated between agents involved in the planning process. This is done via the Issues part of <I-N-OVA> – which can be used to amend the task and option specific agenda which a planning agent is using for its problem solving.

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

3.6 Authority to Plan

As described in [18], it is intended that O-Plan support authority management in a comprehensive and principled way. *Changes* of authority are possible via Task Assignment agent communication to the Planner agent. This may be in the context of a current plan option and task provided previously or it is possible to give defaults which apply to all future processing by the planner agent. The authorities may use domain related names that are meaningful to the user and may refer to the options, sub-options, phases and levels of tasks and plans known to O-Plan.

Ways to authorise agents to take initiative in the problem solving process are being explored. This can be done by communicating the types of agenda entry or issue which the planning agent may handle and giving limitations on which types of constraint may be manipulated and the extent to which they may be manipulated while problem solving.

3.7 Mutually Constraining Plans for Mixed Initiative Planning and Control

Our approach to Mixed Initiative Planning in O-Plan assists in the coordination of planning with user interaction by employing a shared model of the plan as a set of constraints at various levels that can be jointly and explicitly discussed between and manipulated by any user or system component in a cooperative fashion.

The model of Mixed Initiative Planning that can be supported by the approach is *the mutual constraining of behaviour* by refining a set of alternative partial plans. Users and systems can work in harmony though employing a common view of their roles as being to constrain the space of admitted behaviour. Further detail is given in [19].

Workflow ordering and priorities can be applied to impose specific styles of authority to plan within the system. One extreme of user driven plan expansion followed by system "filling-in" of details, or the opposite extreme of fully automatic system driven planning (with perhaps occasional appeals to an user to take predefined decisions) are possible. In contrast with this, our goal is to establish a mixed initiative form of interaction in which users and system components proceed by mutually constraining the plan using their own areas of strength.

Coordination of problem solving must take place between users and the automated components of a planning system. In joint research with the University of Rochester (whose work is described in [1]) we have explored ways in which the O-Plan controller can be given specific limitations on what plan modifications it can perform, and the specific plan options or sub-options it is working on can be coordinated with those being explored by a user supported by a suitable interface.

4 O-Plan Technology

This section describes our current implementation of the abstract O-Plan architecture and introduces Open Planning Process Panels (O-P^3). These panels are based on explicit models of the planning process and used to coordinate the development and evaluation of multiple courses of action. We describe the generic ideas behind O-P^3 technology, a general methodology for building O-P^3 interfaces and the current O-Plan Web demonstration, based around COA evaluation matrices tailored for each user role. O-P^3 technology has also been used to build the Air Campaign Planning Process Panel (ACP^3) for ARPI TIE[3] 97-1 [31] **[see Appendix K]**.

We begin with a description of generic O-P^3 ideas, followed by the specific O-Plan implementation of these concepts. We then describe some of the detailed O-Plan technology elements which have been used to implement the O-Plan two-user Web demonstration.

4.1 Generic O-P^3 Technology

The generic O-P^3 is based on an explicit model of the planning process, which would be encoded using an activity modelling language such as IDEF3 [15]. 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 nodes 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;

[3] TIE is a Technology Integration Experiment.

- 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 IDEF3.
- 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 a O-P^3 interface which is tailored to the overall nature of their task.

Generic O-P^3 design rules are used to inform the construction of the O-P^3 interfaces (see example in figure 9):

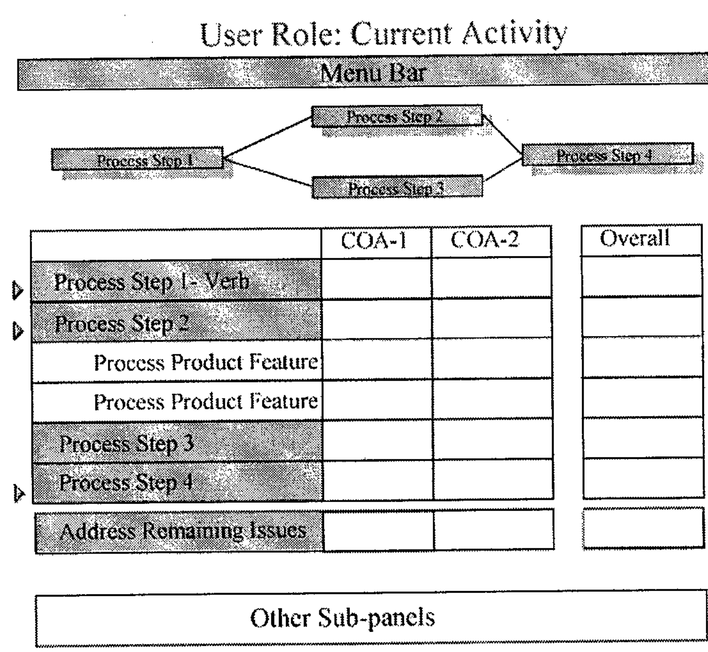


Figure 9: The Generic O-P^3 Interface

- Each user role in the planning process is provided with a panel which is tailored to activities and needs of that role.

- Each user role is assigned a colour to distinguish between the roles. This is used, for example, as a background colour for the header of the panel. Since a given user may act in more than one distinct user role, this acts as a useful visual cue as to which user role is being enacted at any one time.

- The generic O-P^3 panel consists of three parts: a graph sub-panel (PERT chart), a matrix sub-panel (COA comparison matrix) and other sub-panels (e.g. information on assumed environmental conditions). The graph sub-panel and the other sub-panels are optional items (depending on how useful they are for a given application).

- The graph sub-panel contains a partially-ordered graph showing the activity model of the planning planning process. Since the activity model may be large and may apply for each COA being developed, it may not be possible to show the whole network, so some sort of navigation based on decompositions and switching between COAs may be needed.

- The actions shown in the graph sub-panel are annotated with colours to show their current status in the *state model* (see above). The colours used are adapted from other ARPI plan visualisation work [17].

- The matrix sub-panel is a table which contains two types of rows and and two types of columns. The rows are process steps (verb phrases) and COA descriptors (noun phrases). The process steps labels are coloured with the user role background colour and the COA descriptors are white. The columns are the individual COAs being developed (labelled COA-N) and a column reflecting the overall workflow (labelled "Overall").

- The process steps in the matrix sub-panel are an appropriately flattened form of the activity model of the planning process. The status of the actions can be shown using the same colours as are used in the graph sub-panel. The currently active workflow fringe (i.e. what can be done next) is shown using active hyperlinks – clicking on a hyperlink initiates the action.

- The rows are arranged in three parts, running from top to bottom. The first section is concerned with process steps prior to plan synthesis, such as setting the COA requirements. The middle section consists of the COA descriptors and is filled out when a COA has been synthesised. The final section consists of process steps which come after plan synthesis, such as addressing any outstanding issues and viewing the resulting COA in various ways.

- The COA descriptors relate to the COA products produced by the steps of the planning process, such as the minimum duration of the plan and the effectiveness. These can be provided by separate plan evaluators, simulators, etc. The COA descriptors can be selected by the users to show only the critical elements of evaluation. Colours are used to show whether the result is acceptable and raises no issues (green), is possibly acceptable but has some issues to note (orange) or is not acceptable unless the user is prepared to relax the initial requirements or make other necessary changes (red).

- The other sub-panels can contain other useful information such as tables showing the COA objectives and assumed environmental conditions for each COA.

An example of the panel design showing all these features can be seen in figure 10.

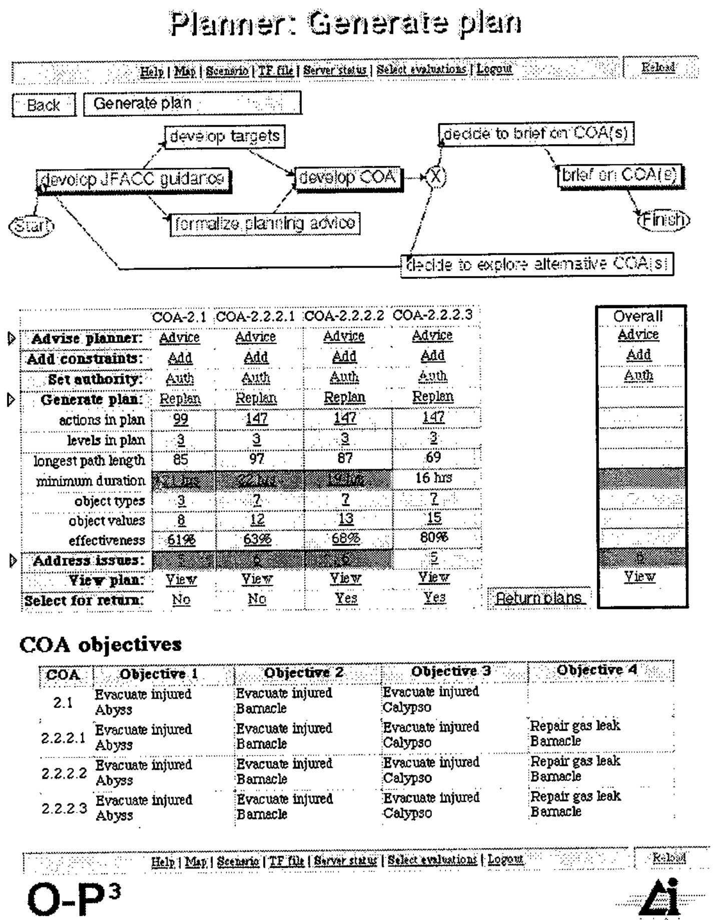


Figure 10: An Example O-P^3 Interface

4.2 The O-Plan COA Matrix

The O-Plan project is concerned with providing support for mixed-initiative planning. The project demonstration 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.

The O-P^3 agent interfaces allow the human agents to play their part in the overall planning

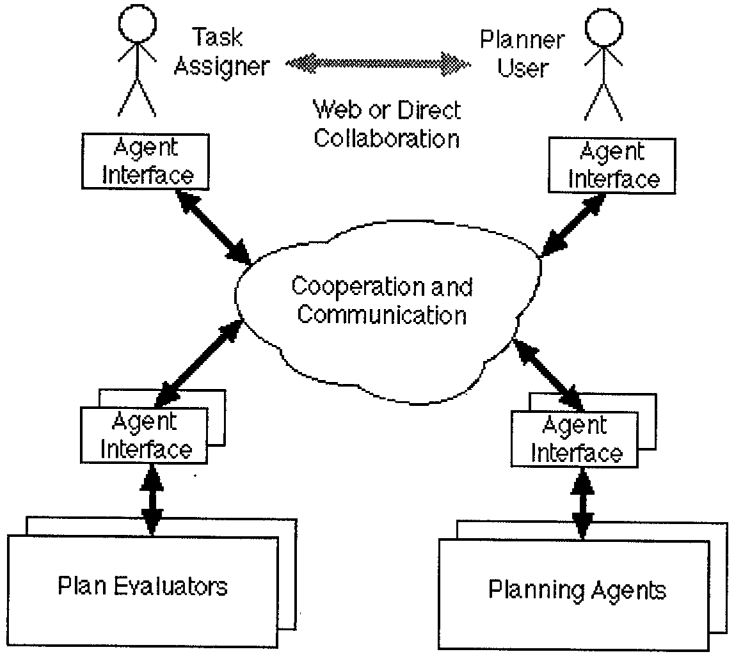


Figure 11: Using O-P^3 Interfaces

process, alongside the system agents, which will be AI planners, schedulers, plan evaluators and so on. This is illustrated in figure 11.

The overall planning task is 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 TA 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 TA. The two users can work in parallel, as will be demonstrated in the example scenario in Section 5 of this report.

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 on the fictional island of Pacifica [16]. This domain, together with the O-Plan Task Formalism (TF) file which implements it is described in Section 5 of this report.

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, Macintosh) 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 TA 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).

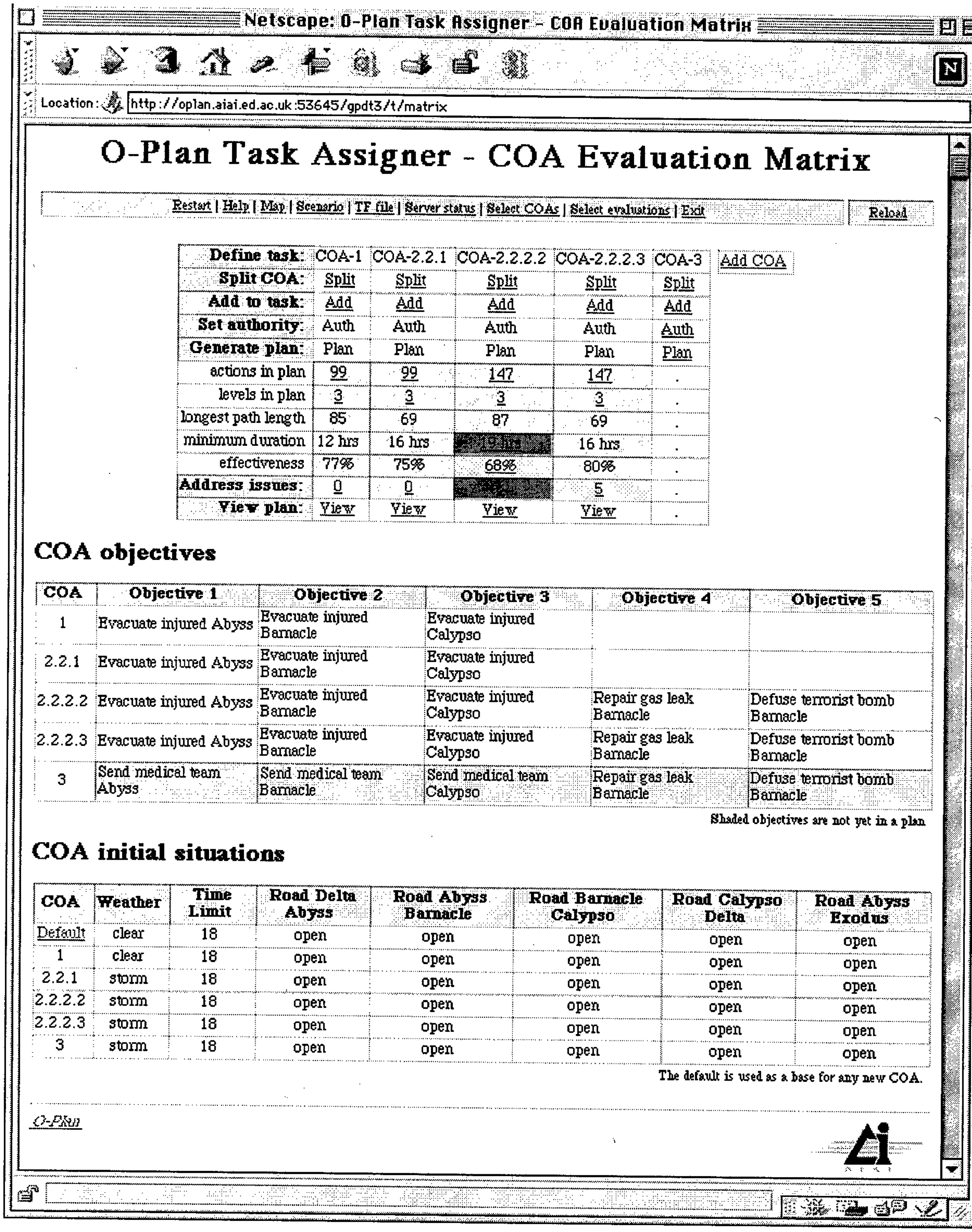
The user roles are arranged such that the TA has authority over the Planner User who in turn has authority over O-Plan. This means that the TA defines the limits of the Planner User's activity (e.g. only plan to level 2) and the Planner User then acts within those bounds to define what O-Plan can do (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 two panels for the Task Assigner and Planner User are shown in figures 12 and 13. 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 TA 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 TA 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 TA. 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.

In summary, the O-Plan Web demonstration illustrates mixed-initiative interaction between two human agents and one system planning agent engaged in the process of developing multiple qualitatively different courses of action. O-P^3 interfaces are provided for the two human users which are tailored to their individual user roles.

The remainder of this section of the report discusses some of the detailed O-Plan technology elements which been used to implement this demonstration, namely the implementation of the plan server, interfacing the plan server to the COA matrix, and handling authority to plan, plan options and plug-in constraint managers.


Netscape: O-Plan Task Assigner - COA Evaluation Matrix

Location: http://oplan.aiai.ed.ac.uk:53645/gpdt3/t/matrix

O-Plan Task Assigner - COA Evaluation Matrix

Restart | Help | Map | Scenario | TF file | Server status | Select COAs | Select evaluations | Exit   Reload

Define task:	COA-1	COA-2.2.1	COA-2.2.2.2	COA-2.2.2.3	COA-3	Add COA
Split COA:	Split	Split	Split	Split	Split	
Add to task:	Add	Add	Add	Add	Add	
Set authority:	Auth	Auth	Auth	Auth	Auth	
Generate plan:	Plan	Plan	Plan	Plan	Plan	
actions in plan	99	99	147	147	.	
levels in plan	3	3	3	3	.	
longest path length	85	69	87	69	.	
minimum duration	12 hrs	16 hrs	19 hrs	16 hrs	.	
effectiveness	77%	75%	68%	80%	.	
Address issues:	0	0	[illegible]	5	.	
View plan:	View	View	View	View	.	

COA objectives

COA	Objective 1	Objective 2	Objective 3	Objective 4	Objective 5
1	Evacuate injured Abyss	Evacuate injured Barnacle	Evacuate injured Calypso		
2.2.1	Evacuate injured Abyss	Evacuate injured Barnacle	Evacuate injured Calypso		
2.2.2.2	Evacuate injured Abyss	Evacuate injured Barnacle	Evacuate injured Calypso	Repair gas leak Barnacle	Defuse terrorist bomb Barnacle
2.2.2.3	Evacuate injured Abyss	Evacuate injured Barnacle	Evacuate injured Calypso	Repair gas leak Barnacle	Defuse terrorist bomb Barnacle
3	Send medical team Abyss	Send medical team Barnacle	Send medical team Calypso	Repair gas leak Barnacle	Defuse terrorist bomb Barnacle

Shaded objectives are not yet in a plan

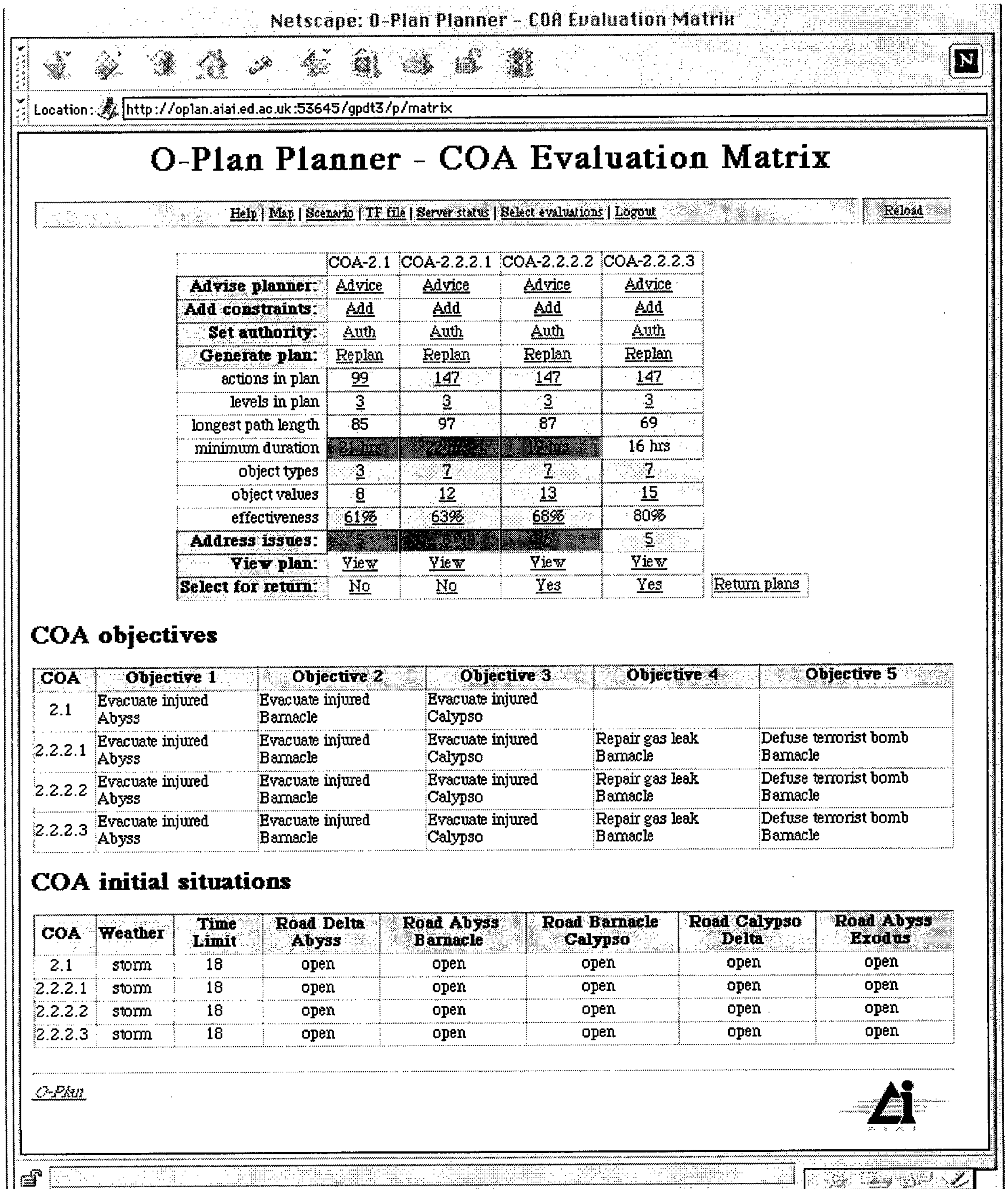
COA initial situations

COA	Weather	Time Limit	Road Delta Abyss	Road Abyss Barnacle	Road Barnacle Calypso	Road Calypso Delta	Road Abyss Exodus
Default	clear	18	open	open	open	open	open
1	clear	18	open	open	open	open	open
2.2.1	storm	18	open	open	open	open	open
2.2.2.2	storm	18	open	open	open	open	open
2.2.2.3	storm	18	open	open	open	open	open
3	storm	18	open	open	open	open	open

The default is used as a base for any new COA.

O-Plan

Figure 12: The Task Assigner's Panel



Netscape: O-Plan Planner - COA Evaluation Matrix

Location: http://oplan.aiai.ed.ac.uk:53645/gpdt3/p/matrix

O-Plan Planner - COA Evaluation Matrix

Help | Map | Scenario | TF file | Server status | Select evaluations | Logout   Reload

	COA-2.1	COA-2.2.2.1	COA-2.2.2.2	COA-2.2.2.3
Advise planner:	Advice	Advice	Advice	Advice
Add constraints:	Add	Add	Add	Add
Set authority:	Auth	Auth	Auth	Auth
Generate plan:	Replan	Replan	Replan	Replan
actions in plan	99	147	147	147
levels in plan	3	3	3	3
longest path length	85	97	87	69
minimum duration	[redacted]	[redacted]	[redacted]	16 hrs
object types	3	7	7	7
object values	8	12	13	15
effectiveness	61%	63%	68%	80%
Address issues:	[redacted]	[redacted]	[redacted]	5
View plan:	View	View	View	View
Select for return:	No	No	Yes	Yes

Return plans

COA objectives

COA	Objective 1	Objective 2	Objective 3	Objective 4	Objective 5
2.1	Evacuate injured Abyss	Evacuate injured Barnacle	Evacuate injured Calypso		
2.2.2.1	Evacuate injured Abyss	Evacuate injured Barnacle	Evacuate injured Calypso	Repair gas leak Barnacle	Defuse terrorist bomb Barnacle
2.2.2.2	Evacuate injured Abyss	Evacuate injured Barnacle	Evacuate injured Calypso	Repair gas leak Barnacle	Defuse terrorist bomb Barnacle
2.2.2.3	Evacuate injured Abyss	Evacuate injured Barnacle	Evacuate injured Calypso	Repair gas leak Barnacle	Defuse terrorist bomb Barnacle

COA initial situations

COA	Weather	Time Limit	Road Delta Abyss	Road Abyss Barnacle	Road Barnacle Calypso	Road Calypso Delta	Road Abyss Exodus
2.1	storm	18	open	open	open	open	open
2.2.2.1	storm	18	open	open	open	open	open
2.2.2.2	storm	18	open	open	open	open	open
2.2.2.3	storm	18	open	open	open	open	open

O-Plan

Figure 13: The Planner User's Panel

4.3 Design and Implementation of the Plan Server

O-Plan version 3.1 (delivered in January 1997) embodied a number of significant changes from earlier versions of O-Plan and reflected a change in emphasis from plan execution and repair [6] [see **Appendix H**][4] to task-assignment and to the use of O-Plan to provide a planning service to other agents. (The agent controlling O-Plan as a planner is called the "task-assignment agent". It might be a person or a program.)

The interfaces that allows other systems to act as task-assignment agents, and hence to interact with O-Plan and to instruct O-Plan in planning tasks, were extended and fully documented. There are defined interfaces at two levels. The *task-assignment interface* specifies the messages that can be exchanged with an O-Plan planning agent. It is used internally by the simple task-assigner that's packaged with O-Plan, and it can be used by external agents that wish to use O-Plan as a planner. The *program interface* allows messages to be sent and received by Common Lisp code that is running together with O-Plan and, in effect, using O-Plan as a subroutine. It defines low-level procedures for sending and receiving messages, as well as higher-level procedures that perform a series of message exchanges. The program interface can be used to build new interfaces. For instance, it is used our Web-based demonstrations to process information from HTML forms and to allow O-Plan to act as an HTTP server.

The principal extensions concern *authority*, so that task-assignment can control what the planner is allowed to do, and *options*. Plan options can be used to create variations on a task, to remember plans while finding other plans, to add constraints to plans, and to ask "what if" questions.

An important internal change was a new, object-oriented way of supporting "plug-in" constraint managers, reflecting our emphasis on uniform protocols for software components and on general, constraint-based, models of plan representation, in particular the <I-N-OVA> model [23] [see **Appendix E**].

Another significant internal change was extensive rewriting of the TF compiler's analysis phase to prepare for later enhancement of its analysis capabilities. The compiler now constructs a number of tables in a uniform fashion, using standard algorithms such as transitive closure rather than the more *ad hoc* methods sometimes used in the past. A side-effect was the removal of a long-standing bug.

Finally, O-Plan 3.1 extended O-Plan's coverage of the Task Formalism language to include "computed" introduction of actions (e.g. to add an evacuate action for each city in a list determined during planning) and multiple-answer compute conditions (so that a Lisp procedure or an external system can return several alternative values for use in a plan).

O-Plan version 3.2 (delivered in October 1998) extended 3.1 in several ways. Internal changes included the ability to work with a greater range of Common Lisp implementations, such as Allegro Common Lisp, the elimination of some long-standing bugs in the handling of variables on the right hand (value) side of world-state conditions, more sophisticated matching of variables that did not yet have values, support for variables in time-window constraints (so that, for instance, the duration of an action can be computed during planning), and earlier evaluation of

[4] This is a recent paper reporting work carried out on a previous contract.

certain world-state conditions (to better handle cases involving variables that can have many possible values).

Other changes enhanced O-Plan's ability to work with other agents via the task-assignment and program interfaces. In particular, O-Plan was given the ability to ask questions of an interacting agent in cases where that agent, rather than O-Plan, had the authority to make certain decisions, and it became possible to merge new task specifications into existing plans and to add new actions at any temporal point within a plan.

The third major area of work, which ran in parallel with the work on O-Plan versions 3.1 and 3.2, was the development of a Web-based, COA-matrix interface as part of the larger project of providing more effective ways of using planners, while also exploring, both conceptually and in practice, the issues and user roles that arise in task-assignment and in mixed-initiative planning.

4.4 Interfacing O-Plan Services to the COA Evaluation Matrix

The matrix interface makes O-Plan available as a planning agent that can be used on the World-Wide Web via standard Web browsers [26] **[see Appendix I]**.

This system uses the program interface (described above) to interact with the O-Plan planner but also provides a substantial intermediate layer oriented towards COA-based planning and supporting two user roles: Task-Assigner and Technical Planning Expert. This layer makes extensive use of O-Plan's authority and option features and provides an additional authority capability of its own: O-Plan can be given the authority to automatically replan to try to find a plan that satisfies specified conditions.

Above the COA-planning layer is a further layer that supports interaction with human users on the World-Wide Web. Requests are sent to O-Plan in the form of URLs and values from HTML forms, and the results are returned in HTML or, in some cases, PostScript. The central page in this interaction is a matrix, written as an HTML table, that shows evaluation results for a set of COAs. Early versions of this Web layer used the Web's Common Gateway Interface (CGI) to process each request, with O-Plan being run anew each time. Since this made it effectively impossible to use options, in later versions O-Plan continued to run throughout a planning session. This was accomplished by writing an HTTP server in Common Lisp to run together with O-Plan, with the CGI method being used only to start the server.

This approach could easily be adapted to allow O-Plan to be used on the Internet in other ways, for instance by software agents other than browsers. Results could be returned in HTML, as now, or in other forms.

4.5 Handling the Authority to Plan

There are three reasons a task-assignment agent might want to control what a planner is authorized to do. First, the agent might not want a full plan. It would therefore tell the planner to plan only to a certain level of detail, or do develop only certain "phases" of the plan. Second, the task-assignment agent might want to control *how* the planner goes about finding a plan. For instance, it might want to develop a high-level plan and then block backtracking before allowing the planner to add lower-level actions. That way of preserving partial results can be done by using authority together with options. Finally, when working in a mixed-initiative fashion, the task-assigner may want to grant the planner the authority to make certain decisions, but not others.

The current O-Plan allows the task-assignment agent to control the level of detail to which O-Plan can expand actions in the plan, specified in terms of level numbers determined by the TF compiler, and whether the planner has the authority to make certain decisions, such which method (schema) is used to expand an action and which object to take as the value of a variable. All of these authority features are exploited in the Web-based matrix interface.

The implementation of authority required that the O-Plan controller adopt a more complex view of the agenda of "issues" to be addressed in the plan. In the past, once an issue had been "triggered", and hence could be selected for processing, it never lost that status and to become untriggered again. Now, when the planner's authority is changed during planning, the status of issues might change as well. This was handled by reexamining the agenda after each authority change.

4.6 Handling Plan Options

Options provide a way to refer to designated plan states and can also help to control how planning is carried out. The planning process can be seen as the construction of a tree of plan states (partial plans), as the planner tries to reach states in which all "issues" have been resolved. (It's best to think of the tree as growing down.) At any point, an option can be created as a way to refer to the current plan state. The options also form a tree, with each option having a single parent and zero or more children. There is always one option that is the "current option", and planning is confined to the subtree that descends from the current option. When the planner reconsiders a choice made earlier, it goes to a higher point in the tree and creates a new branch. The rule that restricts this to the subtree below the current option gives the task-assigner some control over which decisions can be reconsidered.

Top-level options typically correspond to variations on the task given to O-Plan. (When an action, or some other constraint, is added to the plan, this happens within the current option.) Descendents of those options are used to refer to plans, to add new constraints to plans, and to perform "what if" explorations.

The principal option-related messages that can be sent by the task-assigner are:

make-option Construct an option based on the current plan state and with the current option as its parent.

get-option Get the name of the current option.

set-option Make a designated option be the current option.

twin-option Make a "twin" of the current option as it was when first created.

clear-option Change the current option to be equivalent to how it was when first created.

Twinning is useful when trying variations, since it's possible to add different constraints, or plan to different levels, in each twin, without having to anticipate this in advance. An option named *option-1* is automatically created to capture the plan state immediately after all constraints in the task schema have been added to the plan. The task can therefore be varied by making a twin of *option-1* and then adding constraints.

4.7 Plug-in Constraint Managers

This work makes progress towards two goals. One is to make it easier to change the constraint managers (CMs) in O-Plan: to add managers for new types of constraints, to replace existing managers by more capable ones, and to omit managers when working in domains where they are not needed. The other and perhaps, in the long term, more important aim is to identify the responsibilities of constraint managers and then to define interfaces and protocols that would allow those roles to be filled in plug-in fashion.

Here, "plug-in" means that it is possible to add a constraint manager to O-Plan without editing any existing source code. Instead, a manager is added by defining new classes, methods, and table entries. In this approach, all code for handling particular types of constraints must be provided by the managers for those types, and there must be a well-defined interface for the managers to plug into.

O-Plan began moving towards the plug-in model in version 2.3 (delivered in July 1995), where the Constraint Associator [5, 20] [**see Appendix C**] supported constraint-addition in plug-in fashion, and the TF compiler allowed a CM to define its own parser for constraint specifications. A CM could also define a method for checking constraints, to be called if O-Plan was asked to check the plan for errors.[5] A new CM was written to test this framework, but the existing managers were not converted.

Further progress was made in O-Plan 3.1. The constraint-addition protocol was extended to support more sophisticated CMs and hence a greater range of constraint types. CMs that could take advantage of seeing several constraints at once would be able to receive constraints in blocks. There was also explicit support for relatively simple CMs that processed constraints one-at-a-time or that could not handle variables. (A number of the existing CMs were of those sorts.) A new CM was defined for collecting constraints that did not otherwise have a manager. It became possible to define a constraint syntax without defining a CM. Finally, the process of identifying other CM responsibilities continued, and, as time permitted, those activities were brought into the plug-in framework by converting the existing managers.

Moreover, the simple object-oriented approach of version 2.3 was replaced by a more flexible, CLOS-based framework which included class and method definitions that supported by inheritance several different types of CMs, such as the "relatively simple" ones mentioned above.

[5] It is often possible to check constraints in ways that don't just call the same code that adds constraints to the plan, thus making it possible to find bugs.

4.8 O-Plan Availability and Use

O-Plan release materials, documents and demonstrations are available via the O-Plan web site:

```
http://www.aiai.ed.ac.uk/~oplan
```

The source code and documentation for O-Plan version 3.2 (plus earlier versions and any subsequent releases) is available via:

```
http://www.aiai.ed.ac.uk/~oplan/release
```

The O-Plan Web demonstration is available to anyone with access to the Internet via any browser. Since a continuous connection is made to a system at AIAI (which runs the O-Plan plan server), a password is needed to make the connection. This is available on request (see Section 4.9). The Web demonstration can be found by loading the following URL and following the link to "Pacifica COA Matrix server":

```
http://www.aiai.ed.ac.uk/~oplan/web-demo
```

Because of the need to make a continuous connection to the plan server, users of this demonstration are asked to close the connection explicitly by selecting `Exit` from the top bar of the Task Assigner's interface. If there is no activity within a time limit, the connection is closed automatically. This time limit is currently set to one hour.

The Java code and documentation for the Air Campaign Planning Process Panel (ACP^3) is also available:

```
http://www.aiai.ed.ac.uk/~arpi/ACP3
```

At the time of writing, ACP^3 is at version 1.1, released in September 1998. This version is fully functional and was demonstrated successfully as part of the TIE 97-1 demonstration at EFX'98.

4.9 Contacting the O-Plan Team

Queries on any aspect of O-Plan, including installation and use can be sent to the O-Plan team at the following e-mail address:

```
oplan@ed.ac.uk
```

5 Demonstration Scenario

We have used a crisis operations domain based on the Pacifica scenarios [16, 29] that we call "Go Places and Do Things" (GPDT). This is a domain modelled on three levels which closely follows what we observe in large real domain models. The top level is mostly about setting objectives (i.e. COA requirements). The second level is the real planning level and where technological interactions, such as allocating limited resources, need to be resolved. The third level is needed to add detail to the skeleton plans that have been selected.

This domain is a natural extension of our earlier work in the Pacifica Non-combative Evacuation Operations (NEO) domain. In the earlier work, people are evacuated (following some crisis) from a small island using trucks and helicopters. In the new domain, the main goal is to avert a developing crisis in one of the cities on the island, using various vehicles, pieces of equipment and specialist teams. In the crisis domain, unlike previous Pacifica scenarios, the tasks to be performed are complex and may involve plans consisting of hundreds of individual actions.

This domain has been chosen for our current work to demonstrate that O-Plan is sufficiently powerful to be able to cope with these complicated and dynamic planning problems and also to provide the O-Plan team with a problem domain which is general enough to allow expansion and experimentation as our ideas develop.

5.1 Scenario

The action takes place somewhere in a network of cities on the island of Pacifica (see figure 14). A number of crisis situations can arise in the cities and on the roads joining them, such as power stations becoming inoperative or people needing medical treatment. The goal of the commander (i.e. the Task Assigner agent) is to respond effectively to the situation so that the immediate crisis situation is dealt with and appropriate repairs are made to restore the status quo.

5.2 World Description

The following types of objects exist in this domain:

Cities: these can contain other objects, such as teams, people and equipment.

Roads: these connect some of the cities. They may become blocked to certain classes of vehicle due to weather conditions or landslides. Some may be permanently blocked to certain classes of vehicle (e.g. mud tracks).

Vehicles: these are used to carry equipment, teams and people between cities. There are various types of vehicle which have very different capabilities, such as fast air vehicles of low carrying capacity and slow ground transports capable of carrying large pieces of equipment.

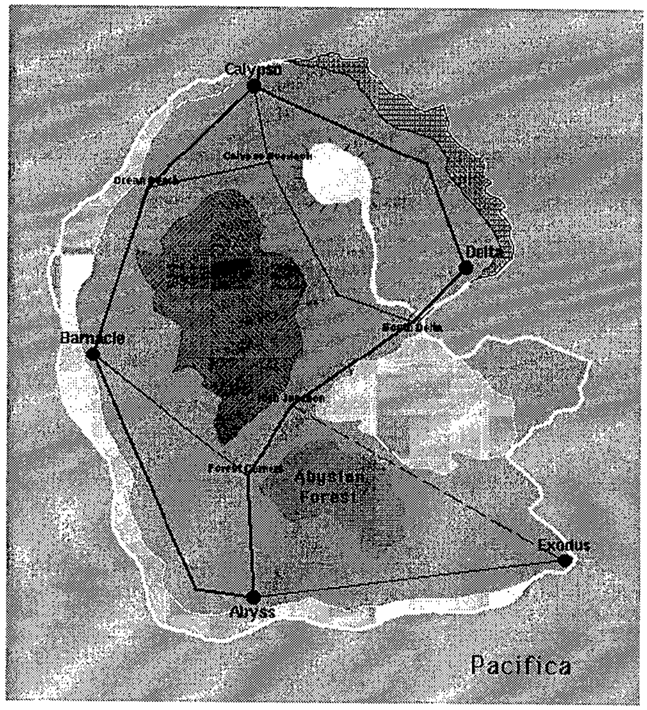


Figure 14: The Island of Pacifica

Equipment: there are various pieces of specialist equipment located in the network of cities. These are needed to perform certain tasks, such as repairs at a power station or emergency medical treatment.

Teams: there are also various specialist teams of people located in the cities. These teams perform specialist tasks, such as fast evacuation or building emergency housing.

People: people are located at cities and may need medical treatment or evacuation. As a simplification, we treat people as a single entity to be treated or moved around, rather than counting a specific number.

Weather: the weather may restrict the options available to the planner, such as not allowing helicopters to fly in thunderstorms.

The world state can be described by giving the locations and contents of the vehicles, the locations of the people, teams and pieces of equipment, and the status of the roads, people and weather.

5.3 Actions and Plans

The GPDT domain is modelled at three levels:

Level 1: the task assignment level. A collection of tasks at level 1, together with any constraints on the plan is a *COA requirement.* There are 12 possible tasks at level 1: evacuate injured, evacuate population, evacuate with medical team, send medical supplies, send emergency food, send medical team, repair gas leak, defuse terrorist bomb, build emergency housing, repair power station turbine, provide immediate assistance, and shut down power station. Each of these tasks takes a single argument: the name of the city where the task is to be performed (e.g. Abyss).

Level 2: the main resource allocation level. The top level tasks specified at level 1 say nothing about which resources should be used in the plan. When the plan is expanded to level 2, the actions require that specific vehicles, teams and equipment are allocated to the actions. If this leads to conflicts in the plan (e.g. two separate actions using the same vehicle at the same time) then an issue will be added to the agenda and O-Plan will solve it in subsequent process by, for example, putting the actions in sequence rather than in parallel. There are 18 different possible actions at level 2, which are combined in partially ordered task networks to form the expansion patterns for the level 1 tasks.

Level 3: adding the detail. A plan at level 2, while having its resources specified, is not sufficiently detailed to be executable. The individual steps involved in carrying out an action at level 2 is specified by a partially ordered task network of actions at level 3. There are 50 different possible tasks at level 3.

Most of the interesting interactions between tasks happen at level 2, since this is where limited resources have to be distributed among the tasks an conflicts resolved. For example, the teams, equipment and people can be moved around using a TRANSPORT[6] action which is modelled at Level 2:

> TRANSPORT cargo ITEM using VEHICLE from CITY to CITY
> where ITEM is an object of type team, vehicle, equipment or people.

The result of the action is that the cargo moves from the source to the destination. This Level 2 schema will expand to give a number of individual actions at Level 3. For example, "TRANSPORT cargo MT1 using GT1 from Abyss to Barnacle" might expand to the following sequence: fuel GT1 at Delta, drive GT1 to Abyss, load MT1 at Abyss, drive GT2 to Barnacle and unload MT1 at Barnacle. If another TRANSPORT action is placed in parallel with this which also uses GT1, then O-Plan will have to resolve the conflict.

Typically, an entire plan specified at level 2 will consist of a number of TRANSPORT operations (of various specific types) to bring the necessary teams and equipment together, followed by the main tasks. The TRANSPORT operations and main tasks may overlap, as in the example where it is necessary to send someone to perform emergency operations while the main equipment and teams are arriving using slower vehicles of high carrying capacity. Once the plan has been specified successfully at level 2, the detail of the plans can be added by refining them to level 3, which is a straightforward process of schema expansion.

[6]The GPDT implementation used contains various specialisations of this action, such as transport-by-road.

5.4 Current Status

The current O-Plan Task Formalism (TF) file implements the crisis operations domain for the island of Pacifica and its five cities (Abyss, Barnacle, Calypso, Delta and Exodus) using 12 top level tasks. A Course of Action consisting of 5 tasks at the top level expands to give approximately 30 actions at the second level and 150 tasks at the third level. The exact numbers will depend on the particular Level 1 tasks selected for the Course of Action.

The version of the TF file used for the final demonstration has been expanded from the version used in an earlier version [30] [**see Appendix L**], in order to provide some interesting schema choice options at level 2. This then allows scenarios where the Planner User can guide O-Plan to producing a good plan which takes the Task Assigner's requirements into account. The level 2 choices implemented in the current TF file are concerned with vehicle choice – a given TRANSPORT action can be performed using either a helicopter, a fast ground transport, a slow ground transport or a fleet of trucks, but there are certain constraints on the choice (e.g. the cargo must be able to be carried by the vehicle, injured people cannot be carried in trucks, certain combinations of vehicle are disallowed for tasks involving more than one vehicle).

5.5 Demonstration Storyboard

The following storyboard illustrates how we envisage the system being used. This scenario can be used in actual demonstrations of this work where two users are working together using two different screens. The two users may be sitting next to each other or connected by telephone. A version of the scenario for a single user acting as both Task Assigner and Planner User is given in [14].

Initial situation: as shown in figure 15 the action takes place on the island of Pacifica, with emergencies being planned for at the cities of Abyss, Barnacle and Calypso. The task assigner (TA) is told to deal with injured civilians at Abyss, Barnacle and Calypso within the next 18 hours. Plans are only acceptable if their effectiveness is rated (on a combination of evaluation factors) as 75% or greater. The weather forecast gives a 50% chance of a storm within the next 24 hours.

Initial preparations: the TA hits "select evaluations" and turns off everything apart from "minimum duration" and "effectiveness". This shows how critical elements of evaluation may be selected. The TA then sets up the default situation, setting the time limit to 18 hrs. The weather and road situations are left with their default values pending more accurate reports. This shows how environmental data can be recorded for subsequent planning.

COA-1: The TA first explores the option of evacuating the injured from all three cities in clear weather. The COA requirements are passed directly to the planner user. A plan is generated which executes in 12 hrs and has an effectiveness of 77%, which is acceptable. The plan has 3 issues outstanding, which are shown and addressed. The plan is then returned to the TA. This shows how the TA sets up tasks and assumptions, how the two users communicate and how different users can view appropriate evaluations of the plan.

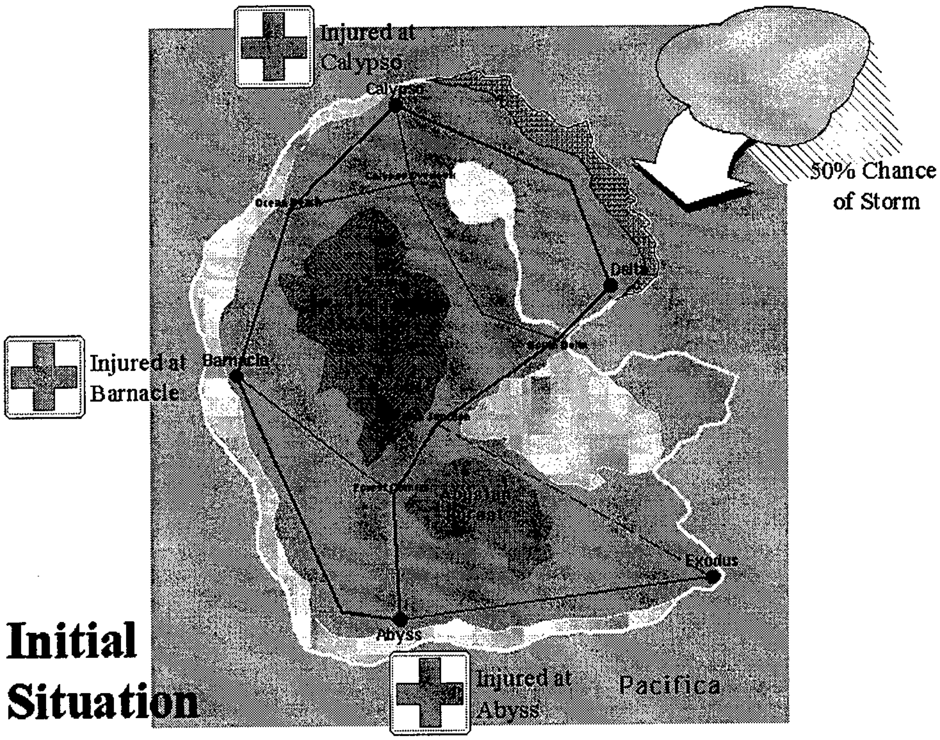


Figure 15: Pacifica - Initial Situation

COA-2: The TA then sets up a second COA with the same evacuation tasks but this time assuming stormy weather, to check for all eventualities. [The TA's authority screen could be shown and explained at this point to show that it is possible to develop plans at an appropriate level.] This new set of COA requirements is passed to the planner user. The first plan generated takes 21hrs and has an effectiveness of 61%, both of which are unacceptable. The planner asks the O-Plan planner for an alternative plan by pressing the "replan" button. The new plan (COA-2.2) executes in 16 hrs and has an effectiveness of 75%, both of which are acceptable. The planner user selects COA-2.2 for return and deletes COA-2.1 and then selects "return plans". This shows how the planner user can generate alternative plans for the same COA requirements and select which ones to return to the TA. At this point, the TA has an acceptable plan for both clear and stormy conditions.

Developing situation: however, as shown in figure 16, the TA is now interrupted by a call from the Barnacle field station. Reports are coming in of an explosion at the power station, causing a gas leak. It is thought that this is due to a terrorist bomb, so it seems wise to fix the gas leak and send a bomb squad to defuse any remaining bombs. Meanwhile, the latest weather report indicates that a storm is brewing and has a 95% chance of hitting the island.

COA-2.2.2: to deal with this turn of events, the TA splits COA-2.2 (the realistic weather assumption) into two sub-options and adds two new tasks to COA-2.2.2, to repair the gas leak

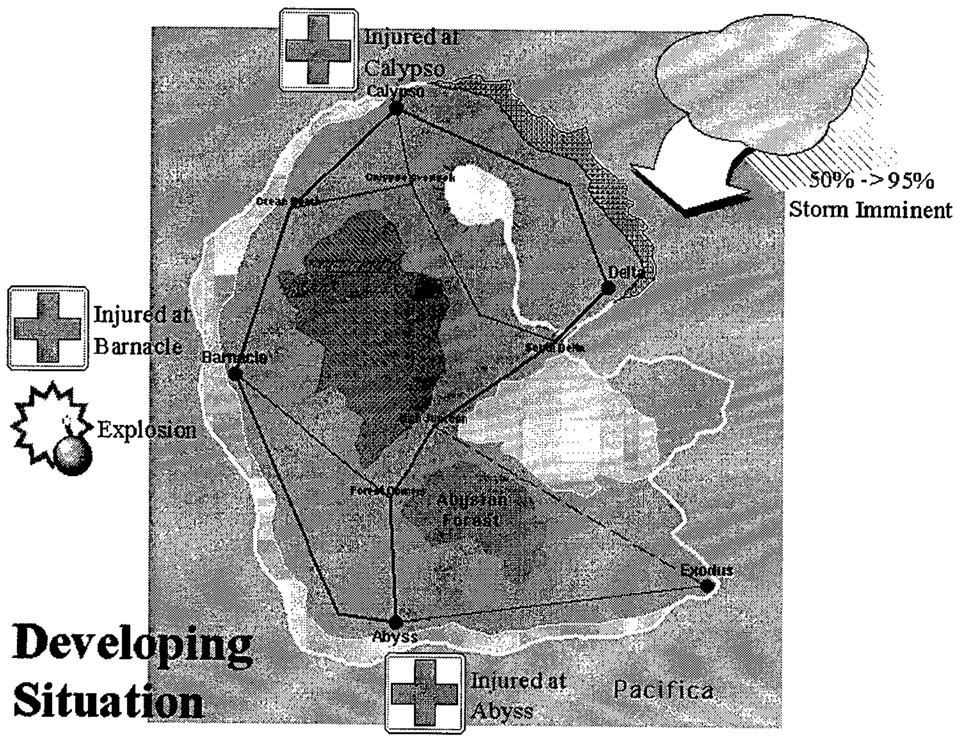


Figure 16: Pacifica - Developing Situation

at Barnacle and send a bomb squad to Barnacle. This shows how the TA can split COAs into sub-options and add further tasks to existing COAs.

COA-2.2.2 is now passed to the planner user. Since the original COA-2.2 took 16 hrs, the planner user selects "Auth" (automated planner authority setting) and switches schema choice to "ask user", to have fine control of the addition of the two new tasks to the existing plan. The planner user is given the option of using fast or slow vehicles for the two tasks and chooses fast vehicles (the first option in each case). This demonstrates schema choice by the planner user.

However, this plan takes 22 hrs and has an effectiveness of 63%. The planner user replans and chooses a mixture of fast and slow vehicles for the "repair gas leak" task and a fast vehicle for the "defuse terrorist bomb" task. While better, the new plan takes 19 hrs and has an effectiveness of only 68%.

The TA is getting impatient and tells the planner user "this is taking too long. Just give me the best one so far." The planner user returns COA-2.2.2.2, keeping COA-2.2.2.1 for further back office work. This shows how the planner user can return some plans to the TA and keep others for further planning.

COA-3: The TA decides to try to deal with this problem by sending medical teams to the three cities to deal with the injured civilians rather than evacuating them, and after updating the default situation to reflect the weather report, starts to set up COA-3 with these tasks, and so begins to define the requirements on the screen.

COA-2.2.2.3: [Note: this part is to be shown in parallel with COA-3 above.] Meanwhile, the planner user has continued to explore the possibilities for COA-2.2.2. The plan was improved when the planner user used some slow vehicles in the plan, so it seems likely that this is because the limited number of fast vehicles are being used repeatedly, resulting in a longer (i.e. more linear) plan. The planner user presses "replan" and chooses to use a slow vehicle in the "defuse terrorist bomb" task (rather than the default fast vehicle). [Note: a choice is not offered for the "repair gas leak task" – slow vehicles are selected automatically because the other 2 possibilities have already been tried.] The planner user was right – the resulting plan executes in 16 hrs and has an effectiveness of 80%. Viewing the plan output (postscript file) at level 2 displays that this plan has good parallelism. This shows how the planner user can use experience of previous cases and human judgement to guide O-Plan at choice points in the planning process.

The planner user now addresses the issues raised by COA-2.2.2.3 and then returns this plan to the TA, saying "I think I've fixed the problem with COA-2.2.2." This shows the asynchronous and collaborative nature of the interactions and in this case shows the initiative being taken by the planner user working in parallel with the TA. The TA presses reload at a convenient time. [Note: for demonstration purposes, the TA should reload before COA-3 is passed over to the planner user.]

Back to COA-3: The TA sees the new plan. "That looks good, now see what you can do with COA-3 as an alternative". The planner user (still in "ask user" schema selection mode) selects the fast vehicle option for 4 of the tasks, but selects a slow vehicle for the "defuse terrorist bomb" task, since this spreads tasks between the fast and slow vehicles and brings more vehicles into play. The resulting plan executes in 12 hrs and has an effectiveness of 79%. This shows how the TA can explore different tasking level options to address the same initial situation using a different strategy.

The TA now has a choice between COA-2.2.2.3 and COA-3. While COA-3 takes 4 hrs less, it is slightly less effective, and more importantly, it only sends medical teams to the three cities rather than evacuating the injured people. The TA could now examine other details of the two plans, using the plan views and the other elements of evaluation, in order to make an informed choice between the two or plan further.

COA-4: the TA decides to try a combination of the two approaches before proceeding to a briefing. COA-4 is set up with the injured being evacuated from Barnacle (because of the increased risk there) and medical teams being sent to Abyss and Calypso. There is also still the requirement for the gas leak to be repaired at Barnacle and a bomb squad sent. The TA asks the planner user to generate a wide range of options and return the best one. The planner user sets schema choice back to "automatic" and does a number of successive replans – at least 4 are required for an effective demonstration. The replans can be done manually or by using the "automatic replanning" facility offered by the authority screen (set number of replans to 5 and effectiveness to be at least 80%). COA-4.4 (or COA-4 if done by automatic replanning) executes in 10 hrs and has an effectiveness of 84%, so this is returned to the TA.

Briefing: the TA clicks on "select COAs" and deletes COA-1, COA-2.2 and COA-2.2.2.2, leaving COA-2.2.2.3, COA-3 and COA-4 as candidates. After viewing the plan, the TA decides to brief on COA-4.4 because it is the shortest, has the highest effectiveness rating and it has the advantage of evacuating the injured civilians from Barnacle.

6 Evaluation

This section evaluates this work. Six different evaluation criteria have been used: meeting our stated vision from the project proposal (both in terms of initial aims and proposed storyboard), an evaluation matrix of domain features and planning technology elements, a consideration of good/bad domains for O-Plan, a critical evaluation of the two-user Web demonstration, a set of scaling experiments carried out on the Task Formalism file for the crisis operations domain, and an assessment of the impact of O-P^3 technology.

6.1 Meeting our Stated Vision – Initial Aims

The initial aims of the project, taken directly from the project "Quad chart" (see figure 2 in Section 2.1), were as follows:

- Generation of multiple qualitatively distinct alternative COAs dependent upon alternative assumptions concerning the emerging crisis.
- Support for mixed-initiative incremental plan development, manipulation and use.
- Promotion of intelligent process management and workflow concepts.
- Integration framework for large-scale modular planning systems.
- Contribution to shared plan representations.

The O-Plan Web demonstration (and associated plan representation work described in Section 3.5) addresses and achieves all of these initial aims. The two-user Web demonstration uses planning technology elements which have been developed over the last 3 years of work and which have been incorporated into O-Plan Version 3.1 since its release in January 1997 (and in the new Version 3.2 which was released in October 1998). These include:

- Mixed-initiative interaction: multiple users and software agents working together in designated roles;
- Multiple option management: exploration of separate options and sub-options.
- Multiple initial conditions: exploration of different initial assumptions about the domain.
- Incremental tasking: adding further constraints to a plan after an initial phase of planning.
- Authority to plan: authorities can be set for any COA investigated allowing for incremental plan refinement alongside user addition of constraints.
- Plan analysis: facilities for plan analysis/evaluation can be installed which have both brief and longer analysis results to present to the user.
- Evaluation selection: the evaluations presented can be selected to show the ones which are critical.

- Issue maintenance: the analysis or planning can leave outstanding issues to be addressed, which are summarised and collected to help with planning and coordination workflow.
- User interaction: an intuitive user interface based on a COA evaluation matrix.
- Status indication: traffic lights are used (as in other ARPI plan visualisation work) to indicate that a chosen plan for a COA is complete (green), has warnings or notes to read (orange) or have issues that need attention (red).

In terms of the technology used, the O-Plan Web demonstration shows the two human users working together with the O-Plan plan server in their designated user roles of Task Assigner, Planner User and software planning agent. The demonstration also uses O-Plan as a true Web-based plan server and exploits the options and authority features of the O-Plan API. It significantly has extended our COA evaluation matrix style of user interface based on "Open Planning Process Panels" (O-P^3) [31] [see **Appendix K**].

6.2 Meeting our Stated Vision – Storyboard

The script on the following page was included in the original project proposal to give a flavour of the type of interaction which we intended to support between the Task Assigner agent and the Planner User agent. It is instructive to compare this script with the demonstration storyboard given in Section 5.5 and with the capabilities provided by the current O-Plan technology.

The demonstration we have provided covers virtually all of this projected storyboard. The interaction between the Task Assigner and the Planner User is very close to that described in the demonstration storyboard. In addition, our demonstration scenario shows how the Planner User, while acting under the authority of the Task Assigner, can take the initiative and develop new plan options within the authority that is given. This was not explicitly predicted in the proposed storyboard.

There are a two minor items in the proposed storyboard which were not explicitly addressed: explicit handling of mission critical milestones and construction of plans which do not meet all specified objectives. However, it seems plausible that both of these could be added to the current demonstration within the framework presented.

	Task Assigner	*Planner User*
1.	Task Assigner asks for the development of 3 to 4 COAs to a given level of detail for different levels of response to the emerging crisis, with initial objectives identified, and with a safe (low) estimate of available assets.	Planner User generates initial COA using the outline plan, objectives and constraints available in the plan library for each level of response indicated.
2.	Task Assigner evaluates the plans related to various elements of evaluation and in particular identifies the potential date at which various mission critical milestones occur.	Planner User improves the evaluation criteria to be used when selecting between alternatives considered in its search for COAs.
3.	For certain COAs the Task Assigner adds in specific plan milestones requirements or new objectives.	Planner User working with system support refines the plan to account for the additional requirements.
4.	Emerging intelligence about the developing crisis provides a better estimate of the date at which certain critical milestones must occur, military objectives to be achieved, and availability/location information for given assets. This invalidates one or more COAs or requires extensions to given COAs. For each level of response, the Task Assigner seeks to maintain an alternative valid COA to a given level of detail by adding new requirements and constraints to those COAs affected and by assigning more assets if necessary.	Planner User develops new COAs as necessary, using the tasking information provided or refines the previously developed COAs to cope with the added requirements.
5.	Following further plan analysis and presentation of options to the relevant authorities, more detailed plan information is sought by the Task Assigner.	A chosen COA is refined to a greater level of detail for nominated phases to establish detailed plan feasibility.
6.	The Task Assigner asks for a number of specific changes to the nominated detailed COA to deal with issues raised during briefings to the authorities.	Further plan development and planner user/system dialogue takes place leading to the development of a number of alternative detailed plans. Some may meet the new objectives but introduce negative factors. Others may not meet all objectives, but could remain within other tasking requirements.
7.	Following further briefings supported by the plan representations and analysis information now available, a particular COA is selected.	Planner User revises the chosen COA to reflect updated asset location information and crisis intelligence.

6.3 The Evaluation Matrix

The work carried out in this part of the evaluation work package consists of two parts: the creation of an evaluation matrix and a set of matrix cell experiments.

In the E-1 project deliverable [8], an evaluation matrix was proposed in which the rows are generic domain features and the columns are generic planner technology and plan representation features. The individual cells of this matrix would be populated by performing cell experiments. The completed matrix would show which technology and plan representation features were needed to support a particular domain feature. An initial list of domain features and technology features was proposed.

In the E-2 deliverable, a collection of 11 cell experiments which fit into the matrix is presented [9]. The aim of this work is to provide details of a number of evaluation experiments which have been carried out on the O-Plan project to date. During each of the three phases of the O-Plan project a great deal of experimentation has taken place to verify and validate the O-Plan model of planning and the components of the O-Plan system. However, most of the experimentation has taken place in the absence of a project evaluation framework in which the results could be classified and analysed. It was found that the existing experiments are tightly clustered in the matrix and are concerned with showing that the search mechanism used in O-Plan can cope with various aspects of the domains under consideration, such as the need to create solutions in the minimum amount of time or the need for mixed initiative exploration of the search space.

In seeking to broaden the scope of the experiments and address some of the other areas of the evaluation matrix, we realised that the set of domain features needed some revision in order to allow meaningful experiments to be designed. The 11 experiments listed in the E-2 deliverable served as useful input to other parts of the evaluation work and demonstrated the basic capability of O-Plan in the domain areas addressed by the E-2 report.

6.4 Good and Bad Domains for O-Plan

This section uses the current implementation of the GPDT domain as a case study in our attempt to characterise good domains for O-Plan – those that O-Plan seemed most suited to.

6.4.1 Expansion-based Plans

O-Plan is a hierarchical task network (HTN) planner, which means that it is well suited towards plans in which the planning task consists of expanding high-level actions into networks of lower-level actions and then resolving any conflicts that occur (e.g. linearising two actions in parallel because they use the same resource). O-Plan does well with domains in which there are distinct levels and where the planning consists of expansion, with appropriate adjustments to the plan for the state of the world being considered.

The GPDT domain fits very well into this picture of things. There are three levels and much of the planning consists of expansion.

6.4.2 Solution Density

O-Plan does well when the solution density is low. This means that coming up with one solution is a good result. In contrast, it is possible to pose planning problems that have a great many solutions where coming up with some solution (as opposed to the best solution) is both easy and not very useful. O-Plan can do well with problems that have a moderate solution density, so long as the solutions are interestingly different (e.g. use a helicopter rather than a truck) rather than being uninteresting permutations of each other (e.g. use the green truck rather than the blue truck).

The GPDT domain does moderately well here. There are some duplicate resources in the domain (e.g. two helicopters) but in general the solution density is quite low and where there are multiple solutions to a sub-task, the various solutions do tend to use different resources, take different amounts of time to complete, and so on.

6.4.3 Optimisation

O-Plan can do well in domains where optimisation by a single criterion is not possible. It may be necessary to compromise on the time taken, the resources used, the robustness of the plan, and so on. It may also be necessary to make sure that multiple different initial assumptions are catered for, rather than fixing a single set of assumptions in advance.

The GPDT domain does well here, because of the uncertainty involved in trying to prepare a response to a developing crisis situation. It would not be possible for a planner to come up with "the best plan" – instead, the planner and task assigner need to work together to explore a range of options based on different assumptions, so see that the chosen plan is the best one as evaluated by multiple criteria.

6.4.4 Cost/Benefit Analysis

For any domain, we need to count the cost of modelling the domain, writing the TF file [32] [**see Appendix J**], writing any plan evaluation functions needed and providing appropriate constraint managers [20] [**see Appendix C**] and special purpose visualisation tools.

For the GPDT domain, the cost of creating the TF file was relatively low. The evaluation functions currently used are meant as examples only and so in a real domain, some effort would be required here. No additional constraint managers are used in this domain at present. The visualisation tools used at the moment are relatively simple (e.g. a postscript viewer is used to examine the plan). Better domain-independent visualisation tools are being developed at present, which would improve the cost/benefit trade-off in this area.

On the benefits side, this fairly low-cost domain modelling effort has created a system which can be used to explore multiple plan options from multiple requirements and assumptions. The cost/benefit trade-off for the GPDT looks favourable because it is both a good fit for O-Plan's abilities and, when mechanised using O-Plan, a good level of benefit is achieved.

6.5 Evaluation of the O-Plan Web Demonstration

This section gives a brief evaluation of the O-Plan demonstration in the context of its intended use as a plan server working with a Planner User and a Task Assigner to explore a space of plan options.

On the positive side, it has been commented both by people within ARPI and outside that the COA evaluation matrix acts as an intuitive and appealing interface. It allows the Task Assigner to set up the top-level tasks for various possible courses of action and then directly visualise various properties of those courses of action via the elements of evaluation. The course of action can be split into sub-options, with the possibility of adding further top-level actions to the plan. The Task Assigner can also explore various plans based on different initial assumptions about the weather and the condition of the roads, in order (for example) to make sure that all possibilities are accounted for. The Planner User can incrementally develop multiple COAs for a given COA requirement and then choose the best COAs to return to the Task Assigner.

Another positive aspect of the Web-based implementation is the people can run the system and collaborate using different machines and different Web browsers. In addition, the system has reached a good level of robustness and has been demonstrated live at various ARPI workshops without problem.

There are a still a number of deficiencies in the TF file which we may aim to address in any future versions:

- The status of the roads is not currently taken into account, even though it is possible to change this from the "COA N definition" screen.
- The transportation is done in a very simple way, without doing any route planning. Also, all trips by a given class of vehicle take the same amount of time no matter what the distance is.
- At the end of a COA, the equipment, teams and vehicles are scattered around the map. It would be useful to be able to "bring everything home" after the main task of the COA are complete.

A version of the TF file has been defined which corrects the first two points (using methods which could also correct the third), but it was found in practice that the backtracking involved in using `achieve` conditions interacted badly with the mechanism for allowing schema choice by the Planner User. It is possible that this could be corrected by making schema choice persistent or by moving the `achieve` conditions to a lower level than the schema choice. Our aim in selecting the TF forms used has been to produce a demonstration domain in which responses from O-Plan take only a few seconds.

6.6 Scaling Experiments

A set of scaling experiments were planned and carried out as part of the progress towards our final deliverable. We used the Task Formalism definition of the crisis operations domain and saw how varying the number of certain types of plan entity changes the nature of the planning process (e.g. to see how doubling the number of cities affects the time taken to find a plan). We found that the expansion-based GPDT domain was unaffected by the number of cities, vehicles and other entities. We then extended GPDT with some precondition-achievement style planning for route finding and found that this was more sensitive to the number of plan entities. We also found that certain styles of achieve condition produced tractable results whereas others caused O-Plan to search hard to find a result.

The final result of these experiments was as reported above – a version of the TF file was produced which used an appropriate style of `achieve` conditions to give route planning. This was not chosen for the final demonstration for the reason given at the end of Section 6.5: persistent used-directed schema choice is required when `achieve` conditions lead to backtracking.

However, as a result of this work, it is possible to say that the GPDT domain is capable of being scaled up to realistic levels, both in its expansion-oriented form (where increasing the number of plan entities has no effect on the time taken to find a plan) and in the extended form with precondition-achievement constructs in the TF file (so long as care is taken with the knowledge engineering so that increasing the number of plan entities has a low impact on the time taken to find a plan).

6.7 Possible Impact of O-P^3 Technology

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 [18].
- 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 system 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 on distinct parts of the plan synthesis in parallel. 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.

7 Conclusions

Five concepts are being used as the basis for exploring multi-agent and mixed-initiative planning involving users and systems: Together these provide for a *shared* model of what each agent can and is authorised to do and what those agents can act upon.

1. *Shared Plan Model* – a rich plan representation using a common constraint model of activity (<I-N-OVA>).
2. *Shared Task Model* – Mixed initiative model of "mutually constraining the space of behaviour".
3. *Shared Space of Options* – explicit option management.
4. *Shared Model of Agent Processing* – handlers for issues, functional capabilities and constraint managers.
5. *Shared Understanding of Authority* – management of the authority to plan (to handle issues) and which may take into account options, phases and levels.

Using these shared views of the roles and function of various users and systems involved in a command, planning and control environment, we have demonstrated a planning agent being used to support mixed initiative task specification and plan refinement over the World Wide Web. It has been applied to the generation of multiple qualitatively different courses of action based on emerging requirements and assumptions. The demonstration takes place in a realistic crisis management domain.

O-Plan technology has been developed in this project which offers:

- Mixed-initiative incremental plan development, manipulation and use for multiple options in a dynamically emerging situation.
- Improved communication between users and system agents acting in various roles (e.g. Task Assigner/Commander and Planner User).
- A workflow approach to enacting the planning process.
- Contribution to shared plan, process and activity representations.
- A "plug and plan" framework for systems integration of planners, schedulers, plan analysers and simulators.
- An AI planning agent which:
  - incrementally refines hierarchically structured plans;
  - provides a constraint manager interface to allow for plan analysis and feasibility checks;

  - maintains a set of outstanding plan "issues" which direct planning process workflow.

- An intuitive user interface using a COA evaluation matrix.

The O-Plan system is provided as a service over the World Wide Web and is accessible on any machine via any browser.

References

[1] Allen, J.F., Ferguson, G.M. and Schubert, L.K., Planning in Complex Worlds via Mixed-Initiative Interaction. In Advanced Planning Technology, 53–60, (Tate, A., ed.), AAAI Press, 1996.

[2] AFSC, The Joint Staff Officer's Guide 1993, AFSC Pub 1, Armed Forces Staff College, Norfolk, VA, USA, US Government Printing Office.

[3] Beck, H. and Tate, A., Open Planning, Scheduling and Constraint Management Architectures for Virtual Manufacturing. Proceedings of the Intelligent Manufacturing Systems (IMS) Workshop at IJCAI-95, Montreal, Canada, August 1995, AAAI Press. Attached to this report as Appendix B.

[4] Currie, K.W. and Tate, A., O-Plan: the Open Planning Architecture. Artificial Intelligence, 51(1), Autumn 1991, North-Holland.

[5] Dalton J., Drabble B. and Tate, A., The O-Plan Constraint Associator, O-Plan Technical Paper ARPA-RL/O-Plan/TP/14, September 1994.

[6] Drabble, B., Dalton, J. and Tate, A., Repairing Plans on the Fly, Proceedings of the NASA Workshop on Planning and Scheduling for Space, Oxnard CA, USA, October 1997, NASA Jet Propulsion Laboratory. Attached to this report as Appendix H.

[7] Drabble, B., Tate, A. and Dalton, J., Applying O-Plan to the NEO Scenarios, in An Engineer's Approach to the Application of Knowledge-based Planning and Scheduling Techniques to Logistics. Appendix O, USAF Rome Laboratory Technical Report RL-TR-95-235, December 1995.

[8] Drabble, B. and Tate, A., O-Plan Evaluation Methodology and Experiments, O-Plan Technical Report ARPA-RL/O-Plan/TR/27 version 1, May 1996.

[9] Drabble, B., Lydiard, T. and Tate, A., O-Plan Project Evaluation Experiments, O-Plan Technical Report Supplement ARPA-RL/O-Plan/TR/33 version 1, March 1997.

[10] Ferguson, G.M., Allen, J.F. and Miller, B.W., TRAINS-95: Towards a Mixed-Initiative Planning Assistant. Proceedings of the Third International Conference on AI Planning Systems (AIPS-96), 70–77, (Drabble, B., ed.), AAAI Press, 1996.

[11] Fowler, N., Garvey, T.D., Cross, S.E., and Hoffman, M., Overview: ARPA-Rome Laboratory Knowledge-Based Planning and Scheduling Initiative (ARPI). In Advanced Planning Technology, 3–9, (Tate, A., ed.), AAAI Press, 1996.

[12] Fraser, J. and Tate, A., The Enterprise Tool Set – An Open Enterprise Architecture. Proceedings of the Workshop on Intelligent Manufacturing Systems, International Joint Conference on Artificial Intelligence (IJCAI-95), Montreal, Canada, August 1995.

[13] Gil, Y., Tate, A. and Hoffman, M., Domain-Specific Criteria to Direct and Evaluate Planning Systems, Proceedings of the ARPA/Rome Laboratory Planning Initiative Workshop, (ed. Burstein, M.), Morgan-Kaufmann, 1994.

[14] Levine, J., Dalton, J. and Tate, A., Mixed-initiative Multi-agent Planning: the O-Plan A-4 Demonstration, O-Plan Technical Report DARPA-AFRL/O-Plan/TR/39, November 1998.

[15] Mayer, R., Cullinane, T., deWitte, P., Knappenberger, W., Perakath, B. and Wells, S., Information Integration for Concurrent Engineering (IICE): IDEF3 Process Description Capture Method Report, Technical Report AL-TR-1992-0057, Armstrong Laboratory, Wright-Patterson Air Force Base, Ohio, May 1992.

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

[17] Stillman J. and Bonissone, P., Technology Development in the ARPA/RL Planning Initiative. In Advanced Planning Technology, 10-23, (Tate, A., ed.), AAAI Press, 1996.

[18] Tate, A., 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.

[19] Tate, A., Mixed Initiative Planning in O-Plan2. Proceedings of the ARPA/Rome Laboratory Planning Initiative Workshop, 512–516, (Burstein, M., ed.), Tucson, Arizona, USA, Morgan Kaufmann, 1994.

[20] Tate, A., Integrating Constraint Management into an AI Planner, Journal of Artificial Intelligence in Engineering, Vol. 9, No. 3, 221–228, Elsevier Applied Science, 1995. Attached to this report as Appendix C.

[21] Tate, A. (ed.), Advanced Planning Technology. AAAI Press, 1996.

[22] Tate, A., Towards a Plan Ontology, AI*IA Notiziqe (Quarterly Publication of the Associazione Italiana per l'Intelligenza Artificiale), Special Issue on "Aspects of Planning Research", Vol. 9. No. 1, 19–26 - March 1996. Attached to this report as Appendix D.

[23] Tate, A., 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, (Drabble, B., ed.) Edinburgh, Scotland, AAAI Press, 1996. Attached to this report as Appendix E.

[24] Tate, A., Mixed Initiative Interaction in O-Plan. Proceedings of AAAI Spring 1997 Symposium on Computational Models for Mixed Initiative Interaction, Stanford University, March 1997.

[25] Tate, A., Multi-agent Planning via Mutually Constraining the Space of Behaviour, Proceedings of the AAAI-97 Workshop on Constraints and Agents, Providence, Rhode Island, USA, July 1997. Attached to this report as Appendix G.

[26] Tate, A., A Planning Agent on the World Wide Web, Seminar on Agents in Information Systems, Heathrow, London, UK, 9th October 1997, Unicom Seminars Ltd., Uxbridge, Middlesex, UK. Attached to this report as Appendix I.

[27] Tate, A., Roots of SPAR - Shared Planning and Activity Representation, The Knowledge Engineering Review Vol 13(1), 121–128, Cambridge University Press, 1998. Attached to this report as Appendix F.

[28] Tate, A., Drabble, B. and Kirby, R., O-Plan2: an Open Architecture for Command, Planning and Control. In Intelligent Scheduling, (eds, M.Zweben and M.S.Fox), Morgan Kaufmann, 1994.

[29] Tate, A., Drabble, B. and Dalton, J., O-Plan: a Knowledge-Based Planner and its Application to Logistics. In Advanced Planning Technology, 259–266, (Tate, A., ed.), AAAI Press, 1996. Attached to this report as Appendix A.

[30] Tate, A., Dalton, J. and Levine, J., Generation of Multiple Qualitatively Different Plan Options. *Proceedings of Fourth International Conference on AI Planning Systems (AIPS-98)*, Pittsburgh, USA, June 1998. Attached to this report as Appendix L.

[31] Tate, A., Levine, J., Dalton, J. and Aitken, S., O-P^3: Open Planning Process Panels, ARPI Workshop, Washington DC, October 1998. Attached to this report as Appendix K.

[32] Tate, A., Polyak, S. and Jarvis, P., TF Method: An Initial Framework for Modelling and Analysing Planning Domains, Workshop on Knowledge Engineering and Acquisition, AIPS-98, Pittsburgh, PA, USA, AAAI Press, 1998. Attached to this report as Appendix J.

Appendices

Starting Point - Applying O-Plan

APPENDIX A: Tate, A., Drabble, B. and Dalton, J., O-Plan: a Knowledge-Based Planner and its Application to Logistics. In Tate, A. (ed.), Advanced Planning Technology, 259–266, AAAI Press, May 1996.

Provides a description of O-Plan at the beginning of the project at which time realistic applications were being attempted, and feedback from the use of O-Plan was generated to drive subsequent developments.

Open Planning Architecture

APPENDIX B: Beck, H. and Tate, A., Open Planning, Scheduling and Constraint Management Architectures for Virtual Manufacturing. Proceedings of the Intelligent Manufacturing Systems (IMS) Workshop at IJCAI-95, Montreal, Canada, August 1995, AAAI Press.

Describes the core architecture of O-Plan and its use in areas as diverse as crisis planning, manufacturing scheduling and enterprise process support.

The following paper is an extended version of the work reported in Appendix B.

Beck, H. and Tate, A., Open Planning, Scheduling and Constraint Management Architectures, British Telecommunication's Technical Journal, Special Issue on Resource Management, Vol. 13, No. 1, pp. 95-101, January 1995, BT Laboratories, Martlesham, UK.

APPENDIX C: Tate, A., Integrating Constraint Management into an AI Planner, Journal of Artificial Intelligence in Engineering, Vol. 9, No. 3, 221–228, Elsevier Applied Science, 1995.

Describes the way in which rich constraint representation and handling can be plugged into O-Plan via the O-Plan Constraint Associator.

Plan and Process Representation

APPENDIX D: Tate, A., Towards a Plan Ontology, AI*IA Notiziqe (Quarterly Publication of the Associazione Italiana per l'Intelligenza Artificiale), Special Issue on "Aspects of Planning Research", Vol. 9. No. 1, 19–26 - March 1996.

Describes the activity, process and plan ontology upon which the project has provided input to a number of international standards efforts.

APPENDIX E: Tate, A., 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, Edinburgh, May 1996, AAAI Press.

Describes a unifying constraint-based framework for representing, reasoning about and communicating activity, process and plan information between human and system agents.

APPENDIX F: Tate, A., Roots of SPAR - Shared Planning and Activity Representation, The Knowledge Engineering Review Vol 13(1), 121–128, Cambridge University Press, 1998.

Provides a historical survey and extensive bibliographic source for work on activity, process and plan representations, and shows how they have been used to design a shared planning and activity representation for use in US military programs.

Mixed Initiative Planning

APPENDIX G: Tate, A., Multi-agent Planning via Mutually Constraining the Space of Behaviour, Proceedings of the AAAI-97 Workshop on Constraints and Agents, Providence, Rhode Island, USA, July 1997.

Describes the central approach to multi-agent and mixed initiative planning in O-Plan.

The following two papers describe related work to that reported in Appendix G.

Tate, A., Using Constraints for Task-oriented Communication, Planning and Control, Proceedings of the Workshop on "Theories of Action, Planning and Control: Bridging the Gap", at the National Conference on Artificial Intelligence (AAAI-96) - Portland, Oregon, USA, August 1996, AAAI Technical Report WS-98-03, AAAI Press.

Tate, A., Mixed Initiative Interaction in O-Plan, Proceedings of the AAAI Spring Symposium on Computational Models for Mixed Initiative Interaction, Stanford, California, USA, March 1997, AAAI Press.

Dynamic Manipulation of Plans

APPENDIX H: Drabble, B., Dalton, J. and Tate, A., Repairing Plans on the Fly, Proceedings of the NASA Workshop on Planning and Scheduling for Space, Oxnard CA, USA, October 1997, NASA Jet Propulsion Laboratory.

Planning takes place in a dynamic environment where tasks, assumptions and information from the environment itself may all be changing rapidly. This paper described the algorithms used in O-Plan to allow plans to be altered to respond to such changes.

A Planning Service and Web Delivery

APPENDIX I: Tate, A., A Planning Agent on the World Wide Web, Seminar on Agents in Information Systems, Heathrow, London, UK, 9th October 1997, Unicom Seminars Ltd., Uxbridge, Middlesex, UK.

Describes the way in which O-Plan has been re-engineered to act as a service to other systems or to act as a server over the World Wide Web.

Knowledge Engineering of Planning Domains

APPENDIX J: Tate, A., Polyak, S. and Jarvis, P., TF Method: An Initial Framework for Modelling and Analysing Planning Domains, Workshop on Knowledge Engineering and Acquisition, AIPS-98, Pittsburgh, PA, USA, AAAI Press, 1998.

It is vital to be effective in capturing a model of the domain in which planning takes place, and to ensure that the model can be maintained. Initial work on a methodology and toolset for applying domain modelling, software engineering, issue-based reasoning, requirements capture and knowledge engineering principles to planning domain acquisition are described in this paper.

Planning User Interfaces

APPENDIX K: Tate, A., Levine, J., Dalton, J. and Aitken, S., O-P^3: Open Planning Process Panels, ARPI Workshop, Washington DC, October 1998.

Provides a description of "Planning Process Panels" used to provide an intuitive interface to display status and allow for control of the planning process when multiple plan options are being generated by a number of planning agents who may be geographically separated.

Putting it all Together

APPENDIX L: Tate, A., Dalton, J. and Levine, J., Generation of Multiple Qualitatively Different Plan Options, Proceedings of Fourth International Conference on Artificial Intelligence Planning Systems (AIPS-98), 27–34, Pittsburgh PA, USA, June 1998, AAAI Press.

Provides an overview of the results of the project and describes the demonstration scenario. The paper thus acts as a short version of the overall final report of the project.

Appendix A:

O-Plan: a Knowledge-Based Planner and its Application to Logistics

Austin Tate, Brian Drabble and Jeff Dalton

Citation:
Tate, A., Drabble, B. and Dalton, J., O-Plan: a Knowledge-Based Planner and its Application to Logistics. In Tate, A. (ed.), Advanced Planning Technology, 259–266, AAAI Press, May 1996.

Purpose:
Provides a description of O-Plan at the beginning of the project at which time realistic applications were being attempted, and feedback from the use of O-Plan was generated to drive subsequent developments.

Abstract:
O-Plan is a command, planning and control architecture with an open modular structure intended to allow experimentation on, or replacement of, various components. The research is seeking to determine which functions are generally required in a number of application areas and across a number of different command, planning, scheduling and control systems.

O-Plan aims to demonstrate how a planner, situated in a task assignment and plan execution (command and control) environment, and using extensive domain knowledge, can allow for flexible, distributed, collaborative, and mixed-initiative planning. The research is seeking to verify this total systems approach by studying a simplified three-level model with separable task assignment, plan generation and plan execution agents.

O-Plan has been applied to logistics tasks that require flexible response in changing situations.

1 Summary

The O-Plan research and development project is seeking to identify re-usable modules and interfaces within planning systems which will enable such systems to be tailored or extended quickly to meet new requirements. A common framework for representing and reasoning about plans based on the manipulation of constraints underlies the model used by the architecture. Within this framework, rich models of an application domain can be provided to inform the planner when creating or adapting plans for actual use.

A number of important foundations have been laid for flexible planning work in the future. They are:

- A view of the planner as *situated* in the context of task assignment, plan execution and change.
- A simple abstract architecture based on an agenda of "issues" from which items can be selected for processing. The processing takes place on an available computational platform (human or machine), with the appropriate functional capabilities described as knowledge sources.

  This architecture allows for independent progress to be made in a number of important areas for successful planning systems, including search control and opportunism, planner capability description, and system resource scheduling.
- A structure that allows separate (often specialised) handlers for different types of constraint to be included, so that the results provide effective overall constraints on the operation of a planner.
- Ways to use domain knowledge, where possible, to constrain the search of a planner.
- The common model of activity, tasks and plans based on a set of constraints – the <I-N-OVA> constraint model. A common model can in turn support systems integration and open up collaboration and distribution opportunities.
- Symmetric interaction by system components and users. Both are seen as manipulating the same set of constraints.
- An approach to the user interface of a planner, based on Plan and World Views.

The O-Plan planner is general purpose and applies to a wide variety of important application areas. Its current application to military logistics planning tasks is described.

2 O-Plan – the Open Planning Architecture

The O-Plan Project at the Artificial Intelligence Applications Institute of the University of Edinburgh is exploring a practical computer-based environment that provides for the specification, generation and execution of activity plans, and for interaction with such plans. O-Plan is intended to be a domain-independent general planning and control framework with the ability to employ detailed knowledge of the domain. See [Allen et. al. 90] for background reading on AI planning systems. See [Currie & Tate] for details of the first version of the O-Plan planner which introduced an agenda-based architecture and the main system components. That paper also includes a chart showing how O-Plan relates to other planning systems. The second version of the O-Plan system adopted a multi-agent approach and situated the planner in a task requirement and plan execution setting [Drabble & Tate 95]. The multi-agent approach taken is described in greater detail in [Tate et. al. 94b].

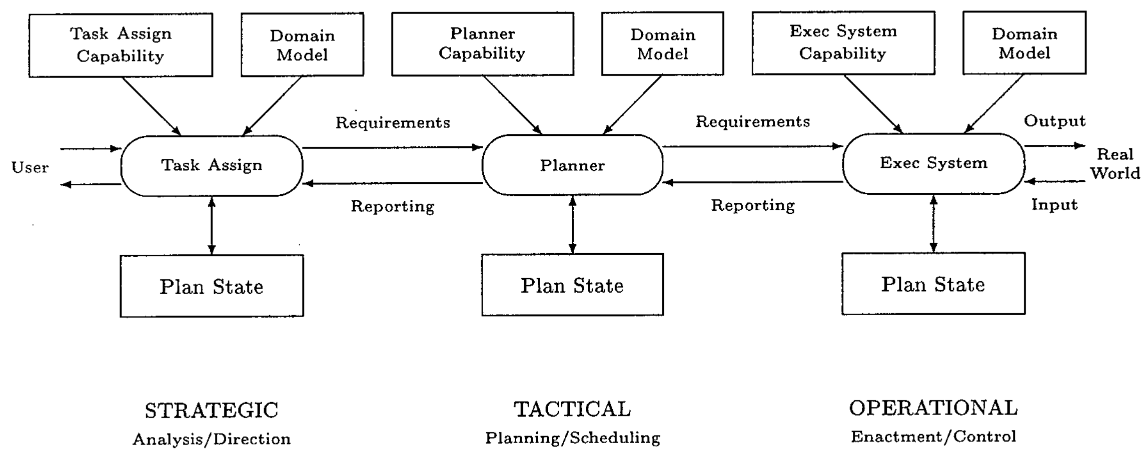


Figure 1: Communication between Strategic, Tactical and Operational Agents

Figure 1 shows the communications between the 3 agents in the O-Plan architecture[1]. A user specifies a task that is to be performed through some suitable interface. We call this process *task assignment.* A *planner* plans to perform the task specified. The *execution system* seeks to carry out the detailed actions specified by the planner while working with a more detailed model of the execution environment. The activities of the three agents may be more or less concurrent.

The O-Plan approach to command, planning, scheduling and control can be characterised as follows:

- successive refinement/repair of a complete plan or schedule which contains an agenda of outstanding issues;
- a least commitment approach;
- opportunistic selection of the focus of attention on each problem-solving cycle;

[1]This simplified view of the environment within which a planner operates helps to clarify the O-Plan research objectives. It is sufficient to ensure that the tasking and execution environments are represented.

- incremental tightening of constraints on the plan, performed by "constraint managers", e.g.,
  - time point network manager,
  - object/variable manager,
  - effect/condition manager,
  - resource utilisation manager;
- localised search to explore alternatives where advisable;
- global alternative re-orientation where necessary.

The O-Plan project has sought to identify modular components within an AI command, planning and control system and to provide clearly defined interfaces to these components. The background to this work is provided in [Tate 93b]. The various components plug into "sockets" within the architectural framework. The sockets are specialised to ease the integration of particular types of component. See figure 2.

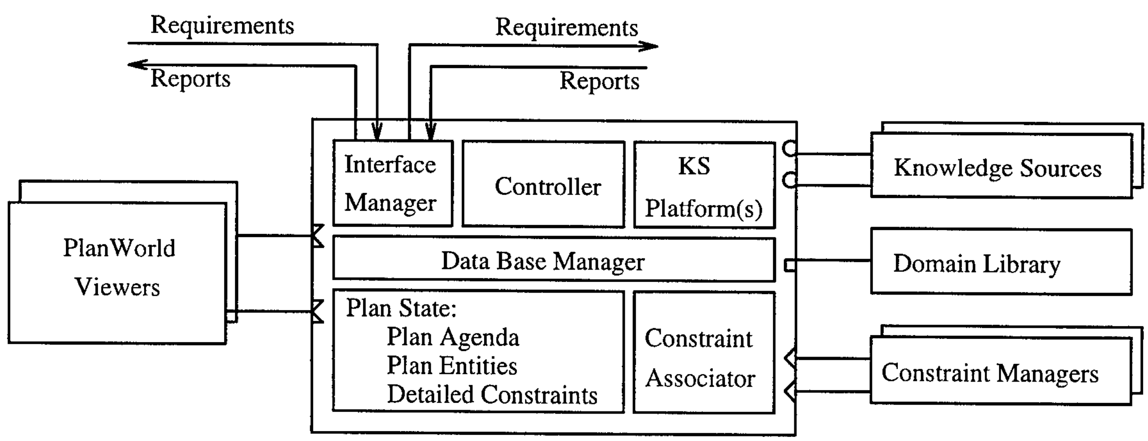


Figure 2: O-Plan Agent Architecture

The components that plug into the O-Plan agent architecture are:

PlanWorld Viewers – User interface, visualisation and presentation viewers for the plan – usually differentiated into technically oriented *plan* views (charts, structure diagrams, etc.) and domain oriented *world* views (simulations, animations, etc.).

Knowledge Sources – Functional components which can analyse, synthesise or modify plans. They provide the *capabilities* of the agent.

Domain Library – A model of the domain, including a library of possible actions. Different models or levels of detail of the model are possible within different agents.

Constraint Managers – Components which manage detailed constraints within a plan and seek to maintain as accurate a picture as possible of the feasibility of the current plan with respect to the domain model.

These plug-in components are orchestrated by an O-Plan agent kernel which carries out the tasks assigned to it via appropriate use of the Knowledge Sources and manages options being maintained within the agent's *Plan State.* The roles of the components are as follows:

Interface Manager – Handles external events (requirements or reports) and, if they can be processed by the agent, posts them on the agent *Agenda.*

Controller – Chooses Agenda entries for processing by suitable Knowledge Sources.

Knowledge Source Platform(s) – Chosen Knowledge Sources are run on an available and suitable Knowledge Source Platform.

Data Base Manager – Maintains the Plan State and provides services to the Interface Manager, Controller and Knowledge Sources.

Constraint Associator Acts as a mediator between changes to the Plan State made by the Data Base Manager and the activities of the various Constraint Managers that are installed in the agent. It eases the management of interrelationships between the main plan entities and detailed constraints [Tate et. al. 94c].

3 A Situated Planner – Coordinating Task Assignment, Planning and Plan Execution

The O-Plan project has identified the need for AI planners to be viewed as situated in an environment where planning is one of a number of tasks involved in dealing with the whole problem of task assignment, planning, execution and control. While the planner deals with the plan generation aspect of the problem, other agents may deal with task elicitation, plan analysis, reactive execution, plan repair, etc. Each of these systems has its own perspective on the planning problem and each is capable of communicating in a way which allows other systems to assimilate new information into their perspective of the problem. This view of planners introduces a number of new issues: the role of authority, determining the quality of the plans being generated by other systems and controlling the execution of plans within other situated agents.

The activities of the various agents need to be coordinated, and authority management is viewed as one way in which this can be done [Tate 93a]. For example, in plan generation, it may be necessary to be given authority to work on certain options and to have direction on the level of detail to which a plan should be developed. In plan enactment, it is important to identify (and possibly name) which phases of the plans can be executed and which parts should be held back for further approval.

Current AI planners can generate a solution that satisfies the requirements they are given. Some planners provide facilities to control the quality of the solution to be returned, by using evaluation functions or search-control rules. However, they do not usually integrate plan quality considerations across several plans. In addition, their plan representations may not reflect the plan quality criteria that are necessary in practice. To date the O-Plan system is able to generate plans and communicate them to the EXPECT [Gil 94],[Gil et. al. 94] system for evaluation. Work is continuing to expand the interface between EXPECT and O-Plan to strengthen the support for users in specifying, comparing and refining the constraints on a range of different plan options, at the task assignment level of a planning support environment, and to allow this information to be used directly by O-Plan in guiding it in its search for a good solution.

The O-Plan architecture has been designed to support the creation of agents which are situated in an environment involving communication with other agents, and work to date has concentrated on building generative planning agents and execution agents, with links between them. The results of this research have been used in a number of systems that have drawn on the O-Plan work. For example, the Optimum-AIV [Aarup et. al. 94] system, developed for Assembly, Integration and Verification of spacecraft at the European Space Agency, and now in use for Ariane Launcher preparations, uses concepts from O-Plan's plan representation to support the repair of plans to deal with test failures. As part of the O-Plan research, an associated Ph.D student project explored the creation of a reactive execution agent within the O-Plan agent architecture [Reece 94]. This work also showed the value of using the plan intentions captured in Goal Structure to support effective reactive execution and re-planning [Reece & Tate 94].

4 Using Domain Knowledge in Planning

O-Plan provides the ability to use domain knowledge about time constraints, resource requirements and other features to restrict the range of plans being considered as feasible solutions to the tasks specified. The O-Plan research programme has studied a number of mechanisms for using such knowledge to prune or prioritise search. These include using temporal constraints [Bell & Tate 85],[Drabble & Kirby 91], resource constraints [Drabble & Tate 94], temporal coherence of conditions [Drummond & Currie 89], and Goal Structure condition type information [Tate 75],[Tate 77].

- **Temporal Constraints** – Each time point referred to in a plan is constrained to have an upper and lower bound on its temporal distance from other time points and from time "zero". The time points held in the Time Point Network (TPN) are indirectly linked to actions and events in a plan - which we refer to as the Associated Data Structure (ADS) [Drabble & Kirby 91]. This ensures that the TPN and entities represented in the ADS can both be independently changed. In addition, the functional interface to the TPN does not reveal the underlying representation, so that a different way of handling time constraints could be substituted.

- **Object/Variable Constraints** – O-Plan uses a rich model of constraints to handle the interactions and dependencies among the different objects and variables, including co-designation (equality), non-codesignation (inequality), scalar (set membership), and numeric range constraints.

- **Resource Constraints** – O-Plan uses a rich model to manage the detailed resource constraints within a plan. The Resource Utilisation Manager (RUM) [Drabble & Tate 94] can handle a number of different resource types and can reason about how resource levels change during the generation of a plan. There are two major resource types supported by the RUM: consumable resources and reusable resources. Each of these can be further subdivided to model the resources of the domain.

- **Goal Structure and Condition Types** – One powerful means of using domain knowledge to restrict and guide search in a planner is to recognise explicit precondition types, as introduced into Interplan [Tate 75] and Nonlin [Tate 77] and subsequently used in other systems such as Deviser [Vere 81], SIPE–2 [Wilkins 88], and O-Plan [Currie & Tate],[Tate et. al. 94b]. O-Plan and Nonlin Task Formalism (TF) extends the notion of a precondition on an action and mates it with a "process-oriented" view of action descriptions. A TF schema description specifies a method by which some higher level action can be performed (or higher level goal achieved). A detailed description of the use of condition types to inform search in an AI planner is provided in [Tate et. al. 94a]. That paper also compares the use of condition types in O-Plan with a number of other planners.

5 <I-N-OVA> – Manipulating Plans as a Set of Constraints

The <I-N-OVA>[2] (*Issues – Nodes – Orderings/Vari- ables/Auxiliary*) Model is a means to represent plans as a set of constraints [Tate 95],[Tate 96]. By having a clear description of the different components within a plan, the model allows for plans to be manipulated and used separately to the environments in which they are generated.

Our aim is to characterise the plan representation used within O-Plan [Currie & Tate],[Tate et. al. 94b] and to relate this work to emerging formal analyses of plans and planning. This synergy of practical and formal approaches can stretch the formal methods to cover realistic plan representations, as needed for real problem solving, and can improve the analysis that is possible for production planning systems.

A plan is represented as a set of constraints which together limit the behaviour that is desired when the plan is executed. Work on O-Plan and other practical planners has identified different entities in the plan which are conveniently grouped into three types of constraint. The set of constraints describes the possible plan elaborations that can be reached or generated as shown in figure 3.

The three types of constraint in a plan are:

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

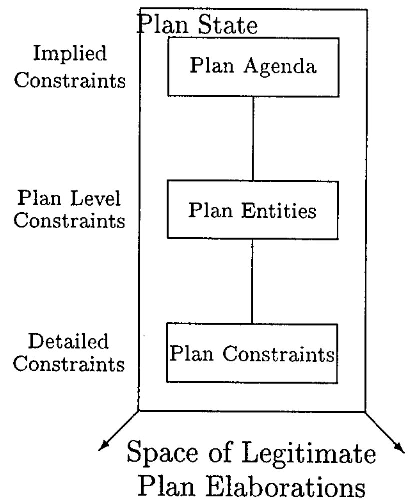


Figure 3: Plan Constraints Define Space of Plan Elaborations

1. Implied Constraints or "Issues" – the pending or future constraints that will be added to the plan as a result of handling unsatisfied requirements, dealing with aspects of plan analysis and critiquing, etc. The implied constraints are the issues to be addressed, i.e., the "to-do" list or agenda which can be used to decide what plan modifications should be made to a plan by a planner (user or system).

2. Plan Entities or Plan Node Constraints – the main plan entities related to external communication of a plan. They describe a set of external names associated to time points. In an activity planner, the nodes are usually the actions in the plan associated with their begin and end time points. In a resource-centred scheduler, nodes may be the resource reservations made against the available resources with a begin and end time point for the reservation period.

3. Detailed Constraints – specialised constraints on the plan associated with plan entities. Empirical work on the O-Plan planner has identified the desirability of distinguishing two special types of detailed constraint: Ordering or Temporal Constraints (such as temporal relationships between the nodes or metric time properties); and Variable Constraints (co-designation and non-co-designation constraints on plan objects in particular). Other Detailed Constraints relate to input (pre-) and output (post-) and protection conditions, resources, authority requirements, spatial constraints, etc. These are referred to as *Auxiliary Constraints.*

6 Abstract View of the O-Plan Control Flow

O-Plan operates on a workflow principle, being driven by an agenda of "issues". It is useful to present a simple abstraction of the workflow within such systems.

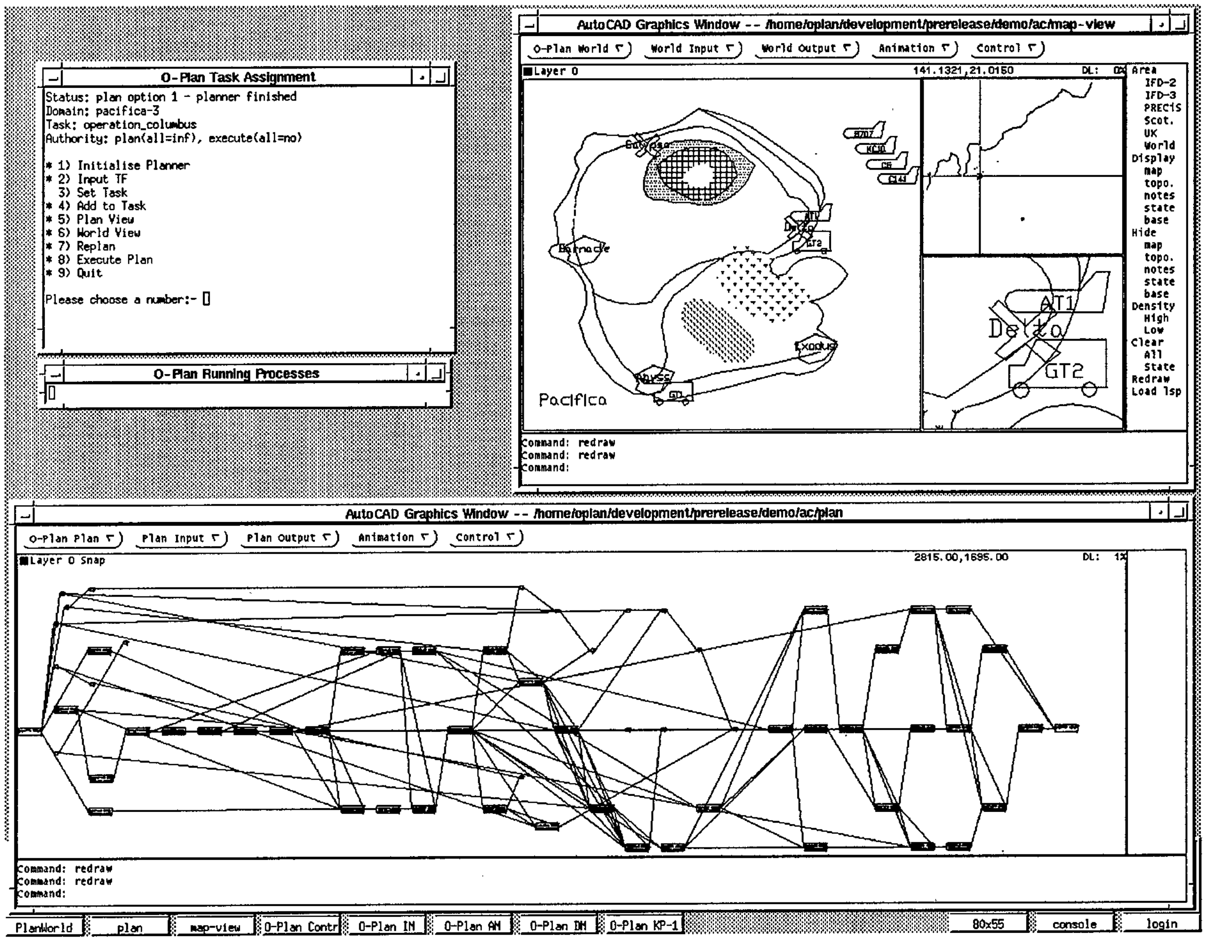


Figure 4: Example Output of the PlanWorld Viewer User Interface

O-Plan refines a "current state". It maintains one or more *options* within the state for alternative decisions about how to restrict the space of state elaborations which can be reached[3]. The system needs to know what outstanding processing requirements exist in the state – the *Agenda of Issues*. These represent the implied constraints on valid future states. One (normally) of these outstanding processing requirements is chosen to be worked upon next (by the *Controller*). This calls up processing capabilities (*Knowledge Sources* or *Issue Handlers*) within the system which can make decisions and modify the State. The modifications can be in terms of definite changes to entities in the state or by noting further processing requirements (as a result of state analysis and critiquing, etc.) on the agenda.

We have found it useful to separate the entities representing the decisions already made during processing into a high level (representing the main entities shared across all planning

[3]State constraint relaxation is also possible to increase the space of state elaborations in some systems.

system components and known to various parts of the system), and more detailed specialised entities (which form a specialised area of the representation of the plan state). These lower level, more compartmentalised, parts can represent specialised constraints within the plan state such as time, resource, spatial and other features. This separation can assist in the identification of opportunities for modularity within the system.

7 Working with the User

O-Plan is implemented in Common Lisp on Unix Workstations with an X-Windows interface. It is designed to be able to exploit distributed and multi-processor delivery systems in future. An interface to AutoCAD has been built to show the type of User Interface we envisage (see Figure 5). This is called the PlanWorld Viewer Interface [Tate & Drabble 95]. The window in the top left corner shows the Task Assignment menu and supports the management of authority [Tate 93a] to plan and execute plans for a given task. The lower window shows a *Plan View* (such as a graph or a gantt chart), and the upper right window shows a *World View* (for visualisation or simulations of the state of the world at points in the plan). The particular plan viewer and world viewer provided are declared to the system and the interfaces between these and the planner uses a defined interface to which various implementations can conform. O-Plan has been interfaced to a number of Plan and World Viewers including process modelling tools, map-based interfaces and tools that create animation sequences of possible plan execution. The developer interface to O-Plan is not shown to the normal planner user.

Recent work on O-Plan has focussed on the representation and management of constraints in planning, particularly in order to simplify some aspects of the architecture and to act as a mechanism for user/system mixed-initiative planning [Tate 94].

8 Target Applications for O-Plan

O-Plan is aimed at the following types of problems:

- project management, systems engineering, construction, process flow, integration and verification, etc.
- planning and control of supply and distribution logistics.
- mission sequencing and control of space probes and satellites such as VOYAGER, ERS-1, etc.

These applications fit midway between the large-scale manufacturing scheduling problems found in some industries (where there are often few inter-operation constraints) and the complex *puzzles* dealt with by very flexible logic-based tools. However, the problems of the target type represent an important class of industrial, scientific and engineering relevance.

The architecture itself has wider applicability. For example, it has been used as the basis for the design of the TOSCA manufacturing scheduler in a project for Hitachi [Beck 93].

9 Crisis Action Planning

The application emphasis of the O-Plan project has been to aid in the definition, generation and enactment of Courses of Action (COAs) within the military crisis action planning process. There are six phases identified in reponding to a crisis are shown in the table.

Phase 1	Situation Development
Phase 2	Crisis Assessment
Phase 3	COA Development: O-Plan provides support in the development of COAs and in estimating the feasibility of the generated COAs. This is the main contribution of the project.
Phase 4	COA Selection: O-Plan provides support in the refinement and presentation of COAs.
Phase 5	Execution Planning
Phase 6	Execution

The O-Plan research principally addresses phases three through six. AIAI has also worked with a number of groups on the representations of plans which can be used to communicate across the different phases and agents involved in the crisis planning process.

Crisis action planning has provided the focus for recent O-Plan applications with problems being tested in the PRECiS domain [Reece et. al. 93] and a simplified version of Integrated Feasibility Demonstration scenario number 2 (IFD-2) from the ARPA/Rome Laboratory Planning Initiative [Fowler et. al. 95]. These test domains allow for realistic, and military-relevant, scenarios and issues to be addressed in a setting suitable for research and development. Crisis action planning calls for plans to be developed which are flexible, robust and responsive to changing task requirements and changes in the operational situation. Current planning aids are too inflexible.

Current military planning systems usually allow only one COA to be fully thought through, and any alternatives are seen as poor relations. This is due to the fixed-step nature of the process: it is not viewed as an iterative process in which several sources of knowledge and techniques (e.g., tasking, planning, scheduling, resourcing and repairing) can be brought in as and when required. A more flexible planning framework may allow military planners to be freed from a step-by-step approach to consider more options and constraints where appropriate within the planning process.

9.1 PRECiS/Pacifica Domain

The principal development of O-Plan has been motivated by applications related to logistics, transportation planning/scheduling problems and Non-combatant Evacuation Operations (NEOs). The testbed is provided by the PRECiS (Planning, Reactive Execution and Constraint

Satisfaction) environment. It defines the data and hypothetical background for logistics planning and reacting scenarios which can be used for demonstration and evaluation purposes.

The definition of the PRECiS environment has drawn on work by several people: Brown at Mitre Corporation to describe a realistic NEO scenario for the Planning Initiative's Integrated Feasibility Demonstration Number 3 (IFD-3); Reece and Tate to define an openly accessible fictional environment based on the island of Pacifica [Reece & Tate 93] suitable for enabling technology researchers interested in planning and reactive execution of plans; and Hoffman and Burnard at ISX Corporation to produce a cut-down demonstration scenario suitable for transportation scheduling research experiments within the ARPA/Rome Laboratory Planning and Scheduling Initiative. The results have been provided in a publicly available document [Reece et. al. 93] and other materials.

Four primary needs of the ARPA/Rome Laboratory Planning and Scheduling Initiative are met by the PRECiS environment.

1. Realistic scenarios can be explored from the data provided in the environment for COA generative planning, case based reasoning, transportation scheduling and the reactive execution of plans.
2. Requirements of "tier-1" enabling researchers are sufficiently met by the data in order for them to pursue their individual research programmes.
3. Entities in the environment are hypothetical and do not reflect actual peoples and locations, yet are realistic in the types of data that would normally be available.
4. The scenario and domain descriptions are not confidential or military critical. They can be openly demonstrated and publications can be based upon them. This is important for enabling researchers.

Work on the PRECiS environment and the Pacifica island model has continued. Map viewers and simulators are now available for demonstration and evaluation purposes. O-Plan has been demonstrated developing Non-combatant Evacuation Operation (NEO) plans in this environment and a reactive execution agent (REA) based on the O-Plan architecture has been used to reactively modify plans to respond to operational demands in a simulation of the Pacifica island in the context of a NEO.

Acknowledgements

The O-Plan project is sponsored by the Advanced Research Projects Agency (ARPA) and Rome Laboratory, Air Force Materiel Command, USAF, under grant number F30602-95-1-0022. The O-Plan project is monitored by Dr. Northrup Fowler III at the USAF Rome Laboratory. The U.S. Government is authorised to reproduce and distribute reprints for Governmental 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 ARPA, Rome Laboratory or the U.S. Government.

References

[Aarup et. al. 94] Aarup, M., Arentoft, M.M., Parrod, Y., Stokes. I., Vadon, H. and Stader, J. *Optimum-AIV: A Knowledge-Based Planning and Scheduling System for Spacecraft AIV*, in Intelligent Scheduling (eds. Zweben, M. and Fox, M.S.), Morgan Kaufmann, San Francisco, 1994.

[Allen et. al. 90] Allen, J., Hendler, J. and Tate, A., *Readings in Planning*, Morgan Kaufmann, Palo Alto, 1990.

[Beck 93] Beck, H., TOSCA: A Novel Approach to the Management of Job-shop Scheduling Constraints, Realising CIM's Industrial Potential: Proceedings of the Ninth CIM-Europe Annual Conference, pages 138-149, (eds. Kooij, C., MacConaill, P.A., and Bastos, J.), 1993.

[Bell & Tate 85] Bell, C.E. and Tate, A., Using Temporal Constraints to Restrict Search in a Planner, Paper presented to the Third UK Planning SIG Workshop, Sunningdale, Oxon, UK. Proceedings published by the Institution of Electrical Engineers, London, January 1985.

[Currie & Tate 91] Currie, K.W. and Tate, A., O-Plan: the Open Planning Architecture, *Artificial Intelligence* 52(1), pp. 49-86, North-Holland, 1991.

[Drabble & Kirby 91] Drabble, B. and Kirby, R., Associating A.I. Planner Entities with an Underlying Time Point Network, Proceedings of the First European Workshop on Planning (EWSP-91), Springer-Verlag Lecture Notes in Artificial Intelligence No 522, 1991.

[Drabble & Tate 94] Drabble, B. and Tate, A., The Use of Optimistic and Pessimistic Resource Profiles to Inform Search in an Activity Based Planner, Proceedings of the Second International Conference on AI Planning Systems (AIPS-94), AAAI Press, Chicago, USA, June 1994.

[Drabble & Tate 95] Drabble, B. and Tate, A., O-Plan: A Situated Planning Agent, Proceedings of the Third European Workshop on Planning (EWSP'95), Assisi, Italy, September, 1995. In *New Directions in Planning*, (eds. Ghallab. M. and Milani, A.), Frontiers in AI and Applications Series, No. 31, IOS Press, Amsterdam, 1995.

[Drummond & Currie 89] Drummond, M. and Currie, K. Exploiting Temporal Coherence in Nonlinear Plan Construction, in Proceedings of the International Joint Conference on Artificial Intelligence IJCAI-89, Detroit, USA, 1989.

[Fowler et. al 95] Fowler, N., Cross, S.E. and Owens, C. The ARPA-Rome Knowledge-Based Planning and Scheduling Initiative, IEEE Expert: Intelligent Systems and their Applications, Vol. 10, No. 1, pp. 4-9, February 1995, IEEE Computer Society.

[Gil 94] Gil, Y. *Knowledge Refinement in a Reflective Architecture*, in the proceedings of the Twelfth National Conference on Artificial Intelligence, Seattle, WA, USA. August 1994. Published by AAAI Press/ The MIT Press Menlo Park, CA, USA.

[Gil et. al. 94] Gil, Y., Tate, A. and Hoffman, M., Domain-Specific Criteria to Direct and Evaluate Planning Systems, Proceedings of the ARPA/Rome Laboratory Planning Initiative Workshop, (ed. Burstein, M.), Morgan Kaufmann, 1994.

[Reece 94] Reece, G.A., Characterization and Design of Competent Rational Execution Agents for Use in Dynamic Environments, Ph.D Thesis, Department of Artificial Intelligence, University of Edinburgh, November 1994.

[Reece & Tate 93] Reece, G.A. and Tate, A. The Pacifica NEO Scenario, Technical Paper ARPA-RL/O-Plan/TP/3, March 1993.

[Reece et. al. 93] Reece, G.A., Tate, A., Brown, D. and Hoffman, M., The PRECiS Environment, Paper presented at the ARPA-RL Planning Initiative Workshop at AAAI-93, Washington D.C., July 1993. Also available as University of Edinburgh, Artificial Intelligence Applications Institute Technical Report AIAI-TR-140.

[Reece & Tate 94] Reece, G.A. and Tate, A., Synthesizing Protection Monitors from Causal Structure, Proceedings of the Second International Conference on AI Planning Systems (AIPS-94), AAAI Press, Chicago, USA, June 1994.

[Tate 75] Tate, A., Using Goal Structure to Direct Search in a Problem Solver. Ph.D. Thesis, University of Edinburgh, 1975.

[Tate 77] Tate, A., Generating Project Networks, Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-77), Cambridge, Mass., USA, 1977.

[Tate 93a] Tate, A., 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 93b] Tate, A., The Emergence of "Standard" Planning and Scheduling System Components, in *Current Trends in AI Planning*, (eds. Backström, C. & Sandewall, E.), IOS Press, 1993.

[Tate 94] Tate, A., Mixed Initiative Planning in O-Plan2, Proceedings of the ARPA/Rome Laboratory Planning Initiative Workshop, (ed. Burstein, M.), Tucson, Arizona, USA, Morgan Kaufmann, 1994.

[Tate 95] Tate, A. Characterising Plans as a Set of Constraints – the <I-N-OVA> Model - a Framework for Comparative Analysis, Special Issue on "Evaluation of Plans, Planners, and Planning Agents", ACM SIGART Bulletin Vol. 6 No. 1, January 1995.

[Tate 96] Tate, A. 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), Edinburgh, UK, AAAI Press, May 1996.

[Tate & Drabble 95] Tate, A. and Drabble, B., PlanWorld Viewers, Proceedings of the 14th Workshop of the UK Planning and Scheduling Special Interest Group, Colchester, UK, November 1995.

[Tate et. al. 94a] Tate, A., Drabble, B. and Dalton, J., "The Use of Condition Types to Restrict Search in an AI Planner" Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94), Seattle, USA, August 1994.

[Tate et. al. 94b] Tate, A., Drabble, B. and Kirby, R.B., O-Plan2: an Open Architecture for Command, Planning and Control, in *Intelligent Scheduling* (eds. Zweben, M. and Fox, M.S.), Morgan Kaufmann, San Francisco, 1994.

[Tate et. al. 94c] Tate, A., Drabble, B. and Dalton, J. Reasoning with Constraints within O-Plan2, Proceedings of the ARPA/Rome Laboratory Planning Initiative Workshop, (ed. Burstein, M.), Tucson, Arizona, USA, Morgan Kaufmann, 1994.

[Vere 81] Vere, S. Planning in Time: Windows and Durations for Activities and Goals, *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 5, 1981.

[Wilkins 88] Wilkins, D. *Practical Planning*, Morgan Kaufmann, Palo Alto, 1988.

Appendix B:

Open Planning, Scheduling and Constraint Management Architectures

Howard Beck and Austin Tate

Citation:
Beck, H. and Tate, A., Open Planning, Scheduling and Constraint Management Architectures for Virtual Manufacturing. Proceedings of the Intelligent Manufacturing Systems (IMS) Workshop at IJCAI-95, Montreal, Canada, August 1995, AAAI Press.

Purpose:
Describes the core architecture of O-Plan and its use in areas as diverse as crisis planning, manufacturing scheduling and enterprise process support.

Abstract:
The development of open planning and scheduling systems seeks to (i) support incremental extension and change, and (ii) facilitate communication between processing agents (both computer and human). This paper presents the open planning and scheduling approach adopted in the O-Plan and TOSCA systems at the Artificial Intelligence Applications Institute (AIAI) in Edinburgh. The purpose is to bring together a description of the concepts developed at AIAI and to relate them in a common framework. References to more detailed descriptions are provided. The paper describes:

1. the key characteristics of the open planning and scheduling systems developed at AIAI;
2. the basis of the separation of the constraint elements in planning and scheduling tasks, distinguishing a high-level model of what remains to be done, a user level view of the plan/schedule entities and the low-level detailed constraints;
3. generic constraint managers for planning and scheduling including time constraint, plan state variables and resource constraints managers. An example using the time point network within an activity or resource reservation framework is provided.

1 Introduction

Historically, planning and scheduling tasks have been treated as static problems; now, it is generally appreciated that these tasks need to be viewed as part of a dynamic process which is subject to external impacts, be they a consequence of concurrent activities (e.g. engineering design, quality etc) or unforeseen events. In many aspects of planning and scheduling (especially in response to change), the role of the human scheduler/system operator is crucially important. To support the user's ongoing decision making, planning and scheduling systems need to be able to communicate in an understandable and useful form. The development of open planning and scheduling systems seeks to (i) support incremental extension and change, and (ii) facilitate communication between processing agents (both computer and human). The motivation for establishing an architecture which supports incremental extendability and modifiability is to enhance flexibility and provide a basis for modular system building. A generic framework and re-usable components would allow system developers the opportunity to exploit the considerable commonality in component functionality typically found in the construction of application systems. The need to support inter-process communication has become apparent from practical experience, especially in the context of increasing enterprise integration.

This paper presents the open planning and scheduling approach adopted in the O-Plan and TOSCA systems at the Artificial Intelligence Applications Institute (AIAI) in Edinburgh. The purpose is to bring together a description of the concepts developed at AIAI and to relate them in a common framework. References to more detailed descriptions are provided.

2 Open Planning and Scheduling at AIAI

O-Plan (The Open Planning Archicture) [7] and TOSCA (The Open Scheduling Architecture) [3] are systems being developed at AIAI. Their approaches to planning, scheduling and control can be characterised as follows:

- open interfaces and communications protocols
- successive refinement/repair of a complete but flawed plan or schedule
- least commitment approach
- using opportunistic selection of the focus of attention on each problem solving cycle
- building information incrementally in "constraint managers", e.g.,
  - time point network manager
  - object/variable manager
  - resource utilisation manager
- using localised search to explore alternatives where advisable

- global alternative re-orientation where necessary.

The open planning and scheduling approach grew out of the experiences of other research in AI planning, particularly with Nonlin [16] and "blackboard" systems [12]. Some of the primary influences include: hierarchical planning [13], the notion of plan state (similar to the work of [11]), constraint posting and least commitment [15], and temporal and resource constraint handling [20]. The *Readings in Planning* volume [1] includes a taxonomy of earlier planning systems that places O-Plan in relation to the influences on its design. The TOSCA scheduling system is heavily influenced by O-Plan and the micro-opportunistic approach to scheduling [14].

O-Plan and TOSCA have been designed as generic planning and scheduling tools applying component technologies. O-Plan has been applied to the following types of problems: (i) mission sequencing and control of space probes such as Voyager, (ii) project management, and (iii) planning and control of supply and distribution logistics. TOSCA has been applied to factory scheduling. Whereas O-Plan is concerned with the detailed construction of activity plans to achieve specific goals and handles problems of moderate scale, TOSCA is concerned with the allocation of activities to resources and start times and, comparatively speaking, handles problems of very large scale.

3 Components of the AIAI Open Planning and Scheduling Systems

O-Plan and TOSCA have as a fundamental design goal the clear separation of system components. There are two broad motivations for this design goal: (i) increased modularity can lead to re-usability, embedability and improved implementation, and (ii) the decomposition of planning and scheduling systems promotes the understandability of the working of the system. This is important for the theoretical exploration of models of the planning and scheduling tasks.

In order to benefit from advances in various technologies and to allow improved implementation of components to be used, it it necessary that the separable functions and capabilities of planners and schedulers be recognised. By separating the processing capabilities at the *architecture* level of a planner or scheduler from the *plan* or *schedule representation*, it becomes possible to address modularity issues of this kind. This separation underpins the generic planning and scheduling model being developed at the AIAI. The architecture and the basic processing cycle is shown in Figure 1.

The architecture of O-Plan and TOSCA has the following primary components:

- domain information
- plan/schedule states
- knowledge sources

- controller
- support modules

The processing cycle is driven by the outstanding or critical issues which need to be addressed. The Controller selects a particular issue and a Knowledge Source to address the issue. When the Knowledge Source is applied, the plan or schedule state is modified. The resulting updates to the plan or schedule state is supported by the Constraint Managers and other Support Modules.

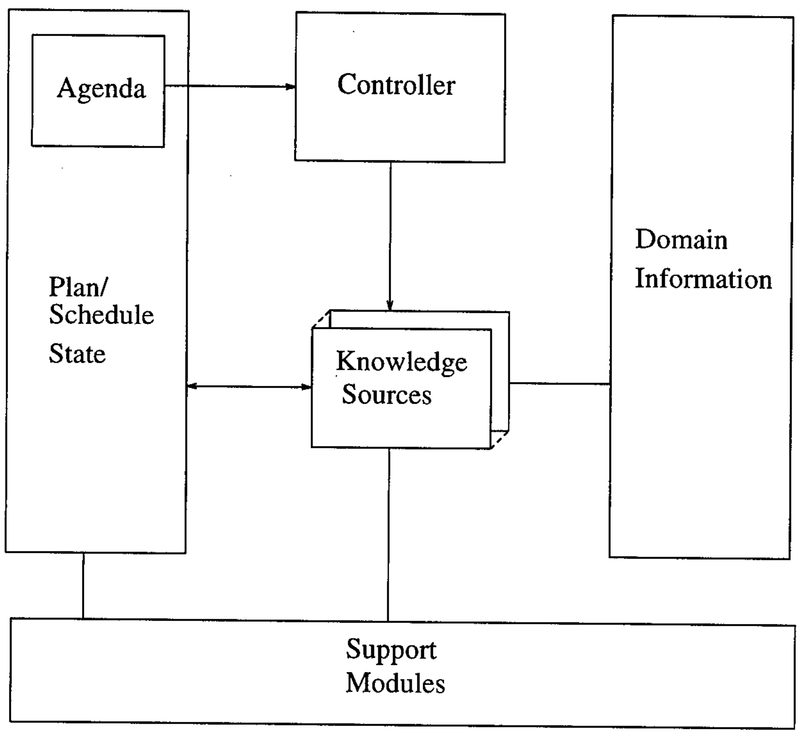


Figure 1: Open Planning and Scheduling Architecture

3.1 Domain Information

Domain descriptions are supplied to O-Plan in a language called Task Formalism (TF). This is compiled into the internal data structures to be used during planning. TF is the means through which a domain expert or domain writer can supply the domain specific information to the O-Plan system.

The domain information describes a model of an application and the tasks to be undertaken. In TOSCA, the model describes the factory, its methods of production and specific production requirements over a given scheduling period [4]. The key elements are:

Production: the manufacturing process concerned with the transformation of materials into end-products. Associated with each product is a set of process plans. Each process plan describes a method of production (*i.e.*, a set of temporally ordered operation types).

Demand for Production: imposed by the orders accepted and predicted by the manufacturing system. Demand is a description of the obligations for production that the manufacturing system has undertaken.

Capacity to Produce: the factory resources and production plans. The capacity of the factory resources are described by their capabilities, corresponding to the various operation types which they can process, and their speed of processing.

Production Constraints: conditions which must be satisfied for a schedule to be valid. Overall schedule objectives (*e.g.*, minimise Work-in-Process) are a special type of constraint in that they apply across the entire schedule.

The domain model is read in from files and converted into internal data structures. A domain description language (DDL) for factory scheduling has been formulated which serves as the basis for the generic specification of discrete factory scheduling problems [4].

3.2 Plan/schedule States

Planning and scheduling *states* can be thought of as snapshots taken during the problem solving process. Each state is associated with: the plan/schedule agenda, the plan/schedule entities and the plan/schedule constraints.

A plan/schedule state may be represented as a set of constraints which together define the range of possible plans or schedules which can be elaborated. Work on O-Plan and other practical planners has identified different entities in the plan which may be grouped into three constraint types which correspond to the high level description above. These are shown below in Figure 2.

The types of constraints are:

- Implied constraints or "Issues" — representing the pending or future constraints that will be added to the plan or schedule as a result of handling outstanding requirements, dealing with aspects of plan/schedule analysis. The implied constraints are the issues to be addressed, *i.e.*, the 'to-do list' or agenda.
- Plan/schedule entities or node constraints — the main plan entities related to external communication of a plan or schedule. They describe a set of external names associated with time points. In an activity planner, the nodes are usually the actions in the plan associated with their begin and end time points. In a resource centred scheduler, nodes are usually the resource reservations made against the available resources with a begin and end time point for the reservation period.

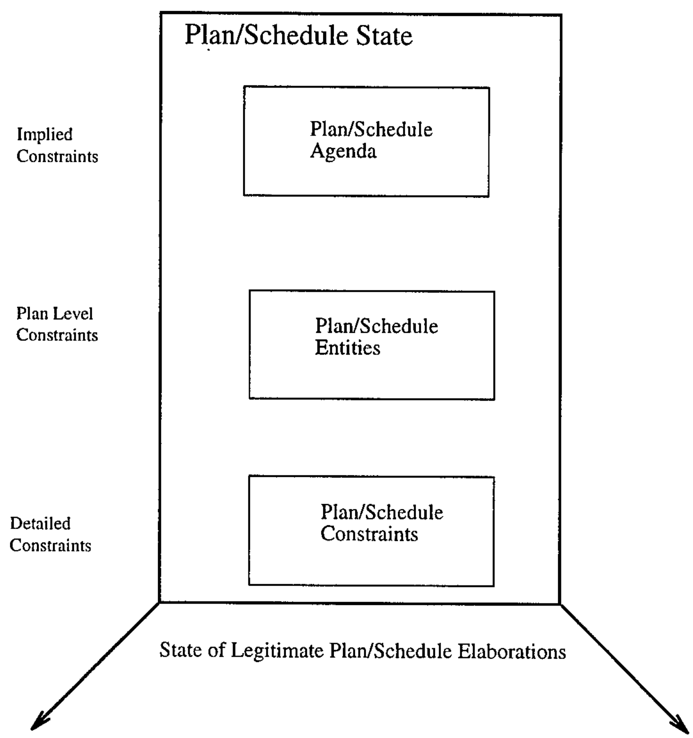


Figure 2: Space of plan/schedule states

- Detailed constraints — associated with plan/schedule entities and representing specialised constraints on the plan or schedule. These are subdivided into three constraint subtypes: Ordering constraints, Variable Constraints and Auxiliary Constraints.

This constraint based description of a plan/schedule state space is described as the <I-N-OVA> (Issues - Nodes - Orderings/Variables/Auxiliary constraints) Model [18].

3.3 Knowledge Sources

Knowledge Sources are defined to address specific plan/schedule requirements through the application of various state modification operators. In O-Plan these include: Expand action, Choose action to satisfy required condition and Select instantiations of object variables. In TOSCA these include: Merge operations, Drop resourcing option, Restrict time window and Allocate start time. Below is a brief description of the state modification operators used in TOSCA:

1. **Merge operations**: a decision to process two or more operations consecutively on the same resource.

   Merging operations reduces the total number of setups and also alters the distribution of demand for setups over time. It is used to manage the constraint regarding the maximum number of setups at a resource and workcentre, and could also be applied to save resource processing time.

2. **Drop resourcing option**: a decision to restrict the resourcing options of an operation.

   Dropping a resourcing option redistributes the demand for capacity and demand for setups *between resources.* It is used to manage both time and setup constraints. The 'drop resourcing option' operator is iteratively applied and a resource allocation is made when the number of resourcing options is reduced to one.

3. **Restrict time window**: a decision to reduce the time window of an operation.

   Restrict an operation time window redistributes the demand for capacity and demand for setups *over time.* It is used to manage both time and setup constraints. The 'restrict time window' operator is iteratively applied and a high-level temporal allocation (*i.e.*, operation to start during a particular time period) is made when the number of high-level time period options is reduced to one.

4. **Allocate start time point**: a decision to allocate a specific time to an operation which has already been allocated a resource.

   Allocating a start time point reserves a specific start time point for an operation taking into account all constraints including those which are not monitored. It is used to check and avoid constraint violations.

3.4 Controller

Throughout the plan generation process, O-Plan identifies outstanding issues to address and these issues are posted on an agenda list. The controller computes the context-dependent priority of the agenda items and selects an item for processing. This provides the fundamental opportunism inherent in the system.

Control in TOSCA can be viewed at two levels: one based on a coarse level of problem decomposition, the other based on a much finer granularity of subproblem. At the coarse level, the current implementation of TOSCA adopts a simple linear flow from phase to phase; at the finer level (within phases), there is highly opportunistic control, corresponding closely to what Sadeh has described as a 'micro-opportunistic' approach to scheduling [14]. The phases at the top-level are:

- *Pre-scheduling* provides an opportunity to analyse the job-shop scheduling problem with the purpose of identifying infeasible constraints. When there is an excessive demand for setups, demand is reduced by merging operations.
- *High-level scheduling* deals with the monitored constraints (*i.e.*, temporal-capacity constraints including temporal preferences). During high-level scheduling, resources are allocated and the possible start times of operations restricted to a time period, the granularity of which is defined by the user.
- *Low-level scheduling* allocates operations to a specific start time.

3.5 Support Modules

In order to efficiently support the planning and scheduling functionality in O-Plan and TOSCA, a number of support modules have been separated out from the core decision making capabilities. These modules have carefully designed functional interfaces to allow for piecewise system engineering and experimentation.

Support modules are distinguished from the other processing capability of the system in that they do not take decisions themselves, but serve to provide efficient support to the higher level Knowledge Sources where decisions are taken. The support modules include database management and retrieval facilities, context layered access to the plan/schedule state, instrumentation and diagnostics as well as the constraint managers, some of which are described in Section 4. Other support modules not included here are described in [19].

4 Constraint Management in Planning and Scheduling

Both O-Plan and TOSCA use a number of *constraint managers* to maintain information about a plan while it is being generated. The information can then be utilised to prune search (where plans are found to be invalid as a result of propagating the constraints managed by these managers) or to order search alternatives according to some heuristic priority.

To improve the modularity of our planning and scheduling systems, we have separated the management of detailed constraints from the explicit manipulation of planning and scheduling entities. This is done via the Associated Data Structure (ADS) abstraction. Data maintained by constraint managers is indirectly linked to the activities, resources, events *etc.* through the ADS level.

Below is a description of *temporal*, *variable* and *resource* constraint managers.

4.1 Management of Temporal Constraints

O-Plan and TOSCA use a point based temporal representation with range constraints between time points and with the possibility of specifying range constraints relative to a fixed time point [5]. This provides the capability of specifying relative and metric time constraints on time points. The functional interface to the Time Point Network (TPN), as seen by the Associated Data Structure (ADS) has no dependence on a particular representation of the plan or schedule [8]. For example, rather than the simple 'before' relationship used in the O-Plan planner's plan state representation, a parallel project exploring temporal logics, reasoning mechanisms and representations for planning has investigated alternative higher level Associated Data Structure time relationships.

The Time Point Network is the lowest level of temporal data structure and consists of a set of points (and associated time constraints between two points) each of which has an upper and lower bound on its temporal distance from:

1. other points in the network
2. a (user defined absolute) start time reference point

This is strong enough for both representing metric and relative time constraints between time points. The points are numbered to give an index with a constant retrieval time for any number of points. This structure allows points to be retrieved and compared through a suitable module interface and with a minimum of overhead. The interface is important and reflects the *functionality* required of the TPN, and hides the detail. This ensures that we have no absolute reliance on points as a necessary underlying representation. The TPN is maintained by the Time Point Network Manager (TPNM). Through application in TOSCA, the current TPNM has been proven on large resource allocation scheduling problems where the number of time points was in excess of 5000 and the number of temporal constraints exceeded 3000.

Figure 3 and Figure 4 show the use of the TPN to underpin two different styles of ADS. Figure 3 is an application involving task planning and Figure 4 is an application where the ADS represents resource allocation.

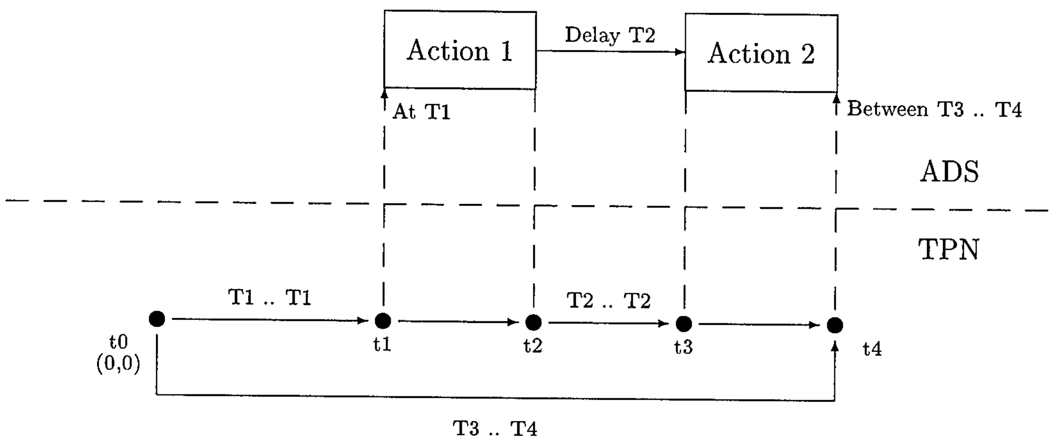


Figure 3: Example of activity planning at ADS using TPN

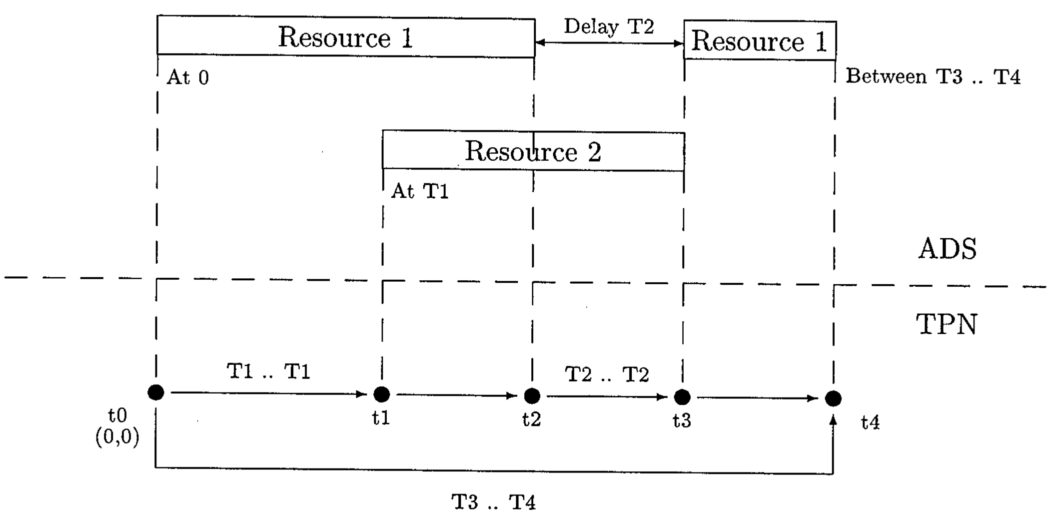


Figure 4: Example of resource allocation at ADS using TPN

4.2 Management of Plan State Variables

In the O-Plan system, the Plan State Variable Manager is responsible for maintaining the consistency of restrictions on plan objects during plan generation. O-Plan adopts a least commitment approach to object handling in that variables are only bound as and when necessary. The constraints are specified as:

- **Sames:** This specifies that this plan state variable should be the same as another plan state variable
- **Not-Sames:** This specifies that this plan state variable should not be the same as another plan state variable
- **Constraint-list:** This specifies a list of attributes which the value to which the plan state variable is bound must have.

4.3 Management of Resource Constraints

O-Plan and TOSCA employ different mechanisms for tracking resource demand and availability: O-Plan uses a simple Resource Utilisation Manager (RUM) [9]; TOSCA uses a more comprehensive model based on habographs [2].

The Resource Utilisation Manager monitors resource levels and utilisation. Resources are divided into different types such as:

1. Consumable: these are resources which are "consumed" by actions within the plan. For example: bricks, petrol, money, etc.
2. Re-usable: these are resources which are used and then returned to a common "pool". For example, robots, workmen, lorries, etc.

Consumable resources can be subcategorised as *strictly consumed* or may be *producible* in some way. Substitutability of resources one for the other is also possible. Some may have a single way mapping such as money for petrol and some can be two way mappings such as money for travellers' cheques. Producible and substitutable resources are difficult to deal with because they *increase* the amount of choice available within a plan and thus *open up* the search space.

The current implementation uses the same mechanism for maintaining resource constraints as did the original O-Plan system [7]. A new scheme is however under study which is based on the maintenance of optimistic and pessimistic resource profiles with resource usage events and activities tied to changes in the profiles [9].

The TOSCA system is particularly concerned with managing high resource contention, and provides mechanisms referred to as habographs to deal more precisely with demand from multiple sources which need to be considered as an aggregate. The aims are: (i) identify constraint violations as early as possible, and (ii) monitor threats of possible constraint violations. The basic insight underlying the habographs representation is described in [2]. The fundamental distinction between habographs and other temporal-capacity constraint representations [10, 14] is in the way that the demand imposed by an operation over time is estimated — specifically, in terms of the assumptions underlying the estimations. Most systems *assume* a demand profile for each operation. Any resource demand profile based upon an aggregation of operation demands is also subject to those assumptions, and as a result, are unable to distinguish between constraint threats and constraint violations. Habographs, by not making this assumption, are able to distinguish constraint violations from threats and more accurately identify constraint threats.

The identification of constraint violations and the monitoring of constraint threats plays a central role in schedule generation both in terms of (i) directing the scheduling process and (ii) informing scheduling decisions.

Habographs extend on similar resource profiling methods [6, 10, 14] in a number of ways. As well as monitoring *temporal* demand on resources, they also provide:

1. a lookahead for and *setup capacity* constraints to allow setup constraints to be dynamically maintained throughout schedule generation.
2. a *hierarchical model* of strategic knowledge constraints to support the analysis of demand for resources over time. This allows compound constraints applying to *machine groups* with overlapping capabilities to be analysed.
3. a separate representation reflecting *temporal preferences*. By distinguishing the temporal preference from the temporal range (limits) of valid allocations of each operation a mechanism is provided for exploring optimality without introducing infeasibility.

In important respects, these extensions support the scheduling of more complex factory domains: *viz.*, the management of setups, the allocation of resources of overlapping capabilities, and the management of the trade-off between hard and preference temporal constraints.

5 Conclusion

This paper has described the open planning and scheduling approach adopted in the O-Plan and TOSCA systems at the Artificial Intelligence Applications Institute in Edinburgh. One particular area highlighted has been the interface between planning systems, scheduling systems and constraint managers responsible for certain specialised aspects of planning and scheduling states. An interface to such constraints managers has been developed to show how improved packaging can be beneficial for the re-use of components. We view this work as a necessary development of recent attempts to re-use components of planning and scheduling systems, particularly specialist constraint managers [17].

References

[1] James Allen, James Hendler, and Austin Tate. *Readings in Planning.* Morgan Kaufmann, 1990.

[2] H. Beck. Constraint Monitoring in TOSCA. In A.Tate M.E.Drummond, M.Fox and M.Zweben, editors, *Working Notes from the AAAI Spring Symposium on Practical Approaches to Planning and Scheduling*, 1992. Also available as Technical Report AIAI-TR-121.

[3] H. Beck. TOSCA: A novel approach to the management of job-shop scheduling constraints. In C. Kooij, P.A. MacConaill, and J. Bastos, editors, *Realising CIM's Industrial Potential: Proceedings of the Ninth CIM-Europe Annual Conference*, pages 138–149, Amsterdam, 12-14 May 1993.

[4] H. Beck, K. Currie, and A. Tate. A Domain Description for Job-Shop Scheduling. Technical Report AIAI-TR-137, A.I.A.I., 1993.

[5] C.E. Bell and A. Tate. Using Temporal Constraints to Restrict Search in a Planner. Technical Report AIAI-TR-5, A.I.A.I., 1986.

[6] P. Berry. *A Predictive Model for Satisfying Conflicting Objectives in Scheduling.* PhD thesis, Dept. of Computer Science, Strathclyde University, Glasgow, 1991.

[7] K.W. Currie and A Tate. O-Plan: the Open Planning Architecture. *Artificial Intelligence*, 52(1), 1991.

[8] B. Drabble and R. Kirby. Associating A.I. Planner Entities with an Underlying Time Point Network. In *European Workshop on Planning (EWSP '91).* Springer-Verlag Lecture Notes in Artificial Intelligence, 1991.

[9] B. Drabble and A. Tate. The use of optimistic and pessimistic resource profiles to inform search in an activity based planner. In Chris Hammond, editor, *Proceedings of the Second International Conference on AI Planning Systems*, University of Chicago, Chicago, Illinois, June 1994.

[10] B. Liu. Scheduling via reinforcement. *Journal of AI in Engineering*, 3(2), 1988.

[11] D.V. McDermott. A Temporal Logic for Reasoning about Processes and Plans. *Cognitive Science*, 6:101–155, 1991.

[12] H.P. Nii. Blackboard systems: the blackboard model of problem solving and the evolution of blackboard architectures. *AI magazine.*, 7(2):38–53., 1986.

[13] E.D. Sacerdoti. *A Structure for Plans and Behaviour.* Artificial Intelligence Series. North Holland, 1977.

[14] N. Sadeh. *Look-ahead Techniques for Micro-oportunistic job shop scheduling.* PhD thesis, School of Computer Science, Carnegie Mellon University, 1991. Technical Report CMU-CS-91-102.

[15] M. Stefik. Planning with constraints. *Artificial Intelligence*, 16:111–140, 1981.

[16] A. Tate. Generating project networks. In *Proceedings 5th IJCAI*, pages 888–893, 1977.

[17] A. Tate. The Emergence of 'Standard' Planning and Scheduling System Components — Open Planning and Scheduling Architectures. In *European Workshop on Planning (EWSP '93)*, 1993.

[18] A. Tate. Characterising Plans as a Set of Constraints — the <I-N-OVA> Model — A Framework for Comparative Analysis. *To appear in ACM SIGART Bulletin*, 6(1), January 1995.

[19] A. Tate, B. Drabble, and R. Kirby. O-Plan2: An Open Architecture for Command, Planning and Control. In M. Fox and M. Zweben, editors, *Knowledge Based Scheduling*, pages 213–239. Morgan Kaufmann., Palo Alto, California, 94303, USA, 1994.

[20] S.A. Vere. Planning in Time: Windows and Durations for Activities and Goals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5(3):246–267, 1981.

Appendix C:

Integrating Constraint Management into an AI Planner

Austin Tate

Citation:
Tate, A., Integrating Constraint Management into an AI Planner, Journal of Artificial Intelligence in Engineering, Vol. 9, No. 3, 221–228, Elsevier Applied Science, 1995.

Purpose:
Describes the way in which rich constraint representation and handling can be plugged into O-Plan via the O-Plan Constraint Associator.

Abstract:
O-Plan is a command, planning and control architecture which has an open modular structure intended to allow experimentation on or replacement of various components. The research is seeking to isolate functionality that may be generally required in a number of applications and across a number of different planning, scheduling and control systems.

This paper describes the way in which plan constraints are represented and handled in the O-Plan architecture. It gives details of a rational reconstruction of the constraint management interfaces now being used as a design principle within the latest version of O-Plan.

The cooperative manipulation of constraints on plans by a user and by the capabilities provided in computer systems provides a useful and natural paradigm for effective planning and scheduling support systems. The provision of powerful computer based constraint management languages and tools could lead to a rapid expansion of the benefits to be gained by identifying more standard ways in which constraints can be handled in future planning and scheduling systems.

1 O-Plan – the Open Planning Architecture

The O-Plan Project at the Artificial Intelligence Applications Institute of the University of Edinburgh is exploring a practical computer based environment to provide for specification, generation, interaction with, and execution of activity plans. O-Plan is intended to be a domain-independent general planning and control framework with the ability to embed detailed knowledge of the domain. See [1] for background reading on planning systems. See [4] for details of the first version of the O-Plan planner which introduced an agenda-based architecture and the main system components. That paper also includes a chart showing how O-Plan relates to other planning systems. The second version of the O-Plan system adopted a multi-agent approach and situated the planner in a task requirement and plan execution setting. The multi-agent approach taken is described in greater detail in [21].

The O-Plan system combines a number of techniques:

- A multi-agent approach to strategic task assignment, tactical planning elaboration, and operational plan execution support.
- A control architecture within each agent in which each control cycle can post further processing steps on an agenda which are then picked out and processed by appropriate handlers (Knowledge Sources).
- The uniform treatment of the user (in the role of planner) and computer based planning capabilities as Knowledge Sources.
- The notion of a "Plan State" which is the data structure containing the emerging plan, the "issues" remaining on its agenda, and the information used in building the plan.
- A hierarchical planning system which can produce plans as partial orders on actions.
- Constraint posting and least commitment on object variables.
- Temporal and resource constraint handling using incremental algorithms which are sensitively applied only when constraints alter.
- O-Plan is derived from the earlier Nonlin planner [15] from which it takes and extends the ideas of Goal Structure, Question Answering (Truth Criterion) and typed conditions.
- We have extended Nonlin's style of domain description language – Task Formalism (TF).

O-Plan is aimed to be relevant to the following types of problems:

- project management for product introduction, systems engineering, construction, process flow for assembly, integration and verification, etc.
- planning and control of supply and distribution logistics.
- mission sequencing and control of space probes and satellites such as VOYAGER, ERS-1, etc.

A user specifies a task that is to be performed through some suitable interface. We call this process *task assignment.* A *planner* plans to perform the task specified. The *execution system* seeks to carry out the detailed actions specified by the planner while working with a more detailed model of the execution environment.

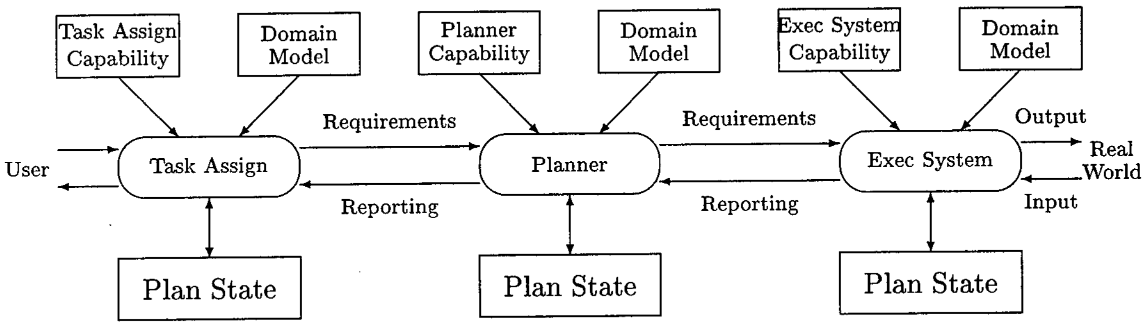


Figure 1: Communication between Strategic, Tactical and Operational Agents

Figure 1 shows the communications between the 3 agents in the O-Plan architecture. The current O-Plan system has a comprehensive planner agent and a simple execution agent [21]. A comprehensive reactive execution agent has also been built in the O-Plan architecture [11]. The task assignment function is provided by a separate process which has a simple menu interface and is not currently in the form of an O-Plan agent.

The O-Plan project has sought to identify modular components within an AI command, planning and control system and to provide clearly defined interfaces to these components and modules.

The main components within a single O-Plan agent are:

1. Domain Information – the information which describes an application domain and tasks in that domain to the planner.
2. Plan State – the emerging plan to carry out identified tasks.
3. Knowledge Sources – the processing capabilities of the planner (also referred to as *Plan Modification Operators* – PMOs).
4. Constraint Managers and Support Modules – functions which support the processing capabilities of the planner and its components.
5. Controller – the decision maker on the *order* in which processing is done.

The agent components as they appear within the O-Plan planner agent are shown in Figure 2.

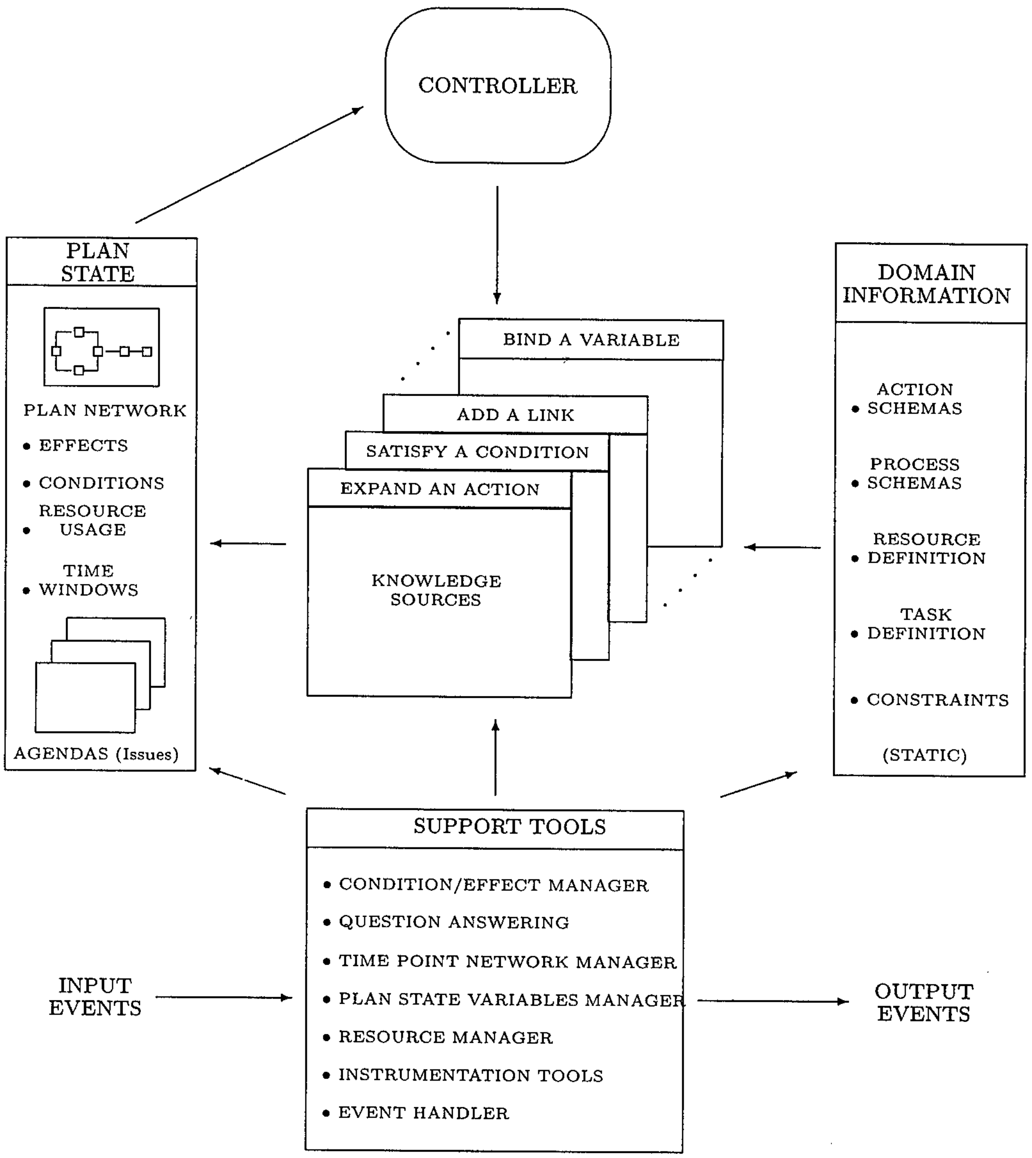


Figure 2: O-Plan Planner Agent Components

O-Plan is implemented in Common Lisp on Unix Workstations with an X-Windows interface. It is designed to be able to exploit distributed and multi-processor delivery systems in future. An interface to AutoCAD has been built to show the type of User Interface we envisage (see Figure 3). The window in the top left corner shows the Task Assignment menu and supports the management of authority [18] to plan and execute plans for a given task. The lower window shows a *Plan View* (such as showing the plan as a graph or as gantt charts), and the upper right window shows a *World View* for visualisation or simulations of the state of the world at points in the plan. The particular plan viewer and world viewer provided are declared to the system and the interfaces between these and the planner uses a defined interface to which various implementations can conform. O-Plan has been interfaced to a number of Plan and World Viewers including process modelling tools, map-based interfaces and tools to create animation sequences of possible plan execution. The developer interface to O-Plan is not shown to the normal user. In figure 3, developer window icons appear along the bottom edge of the screen.

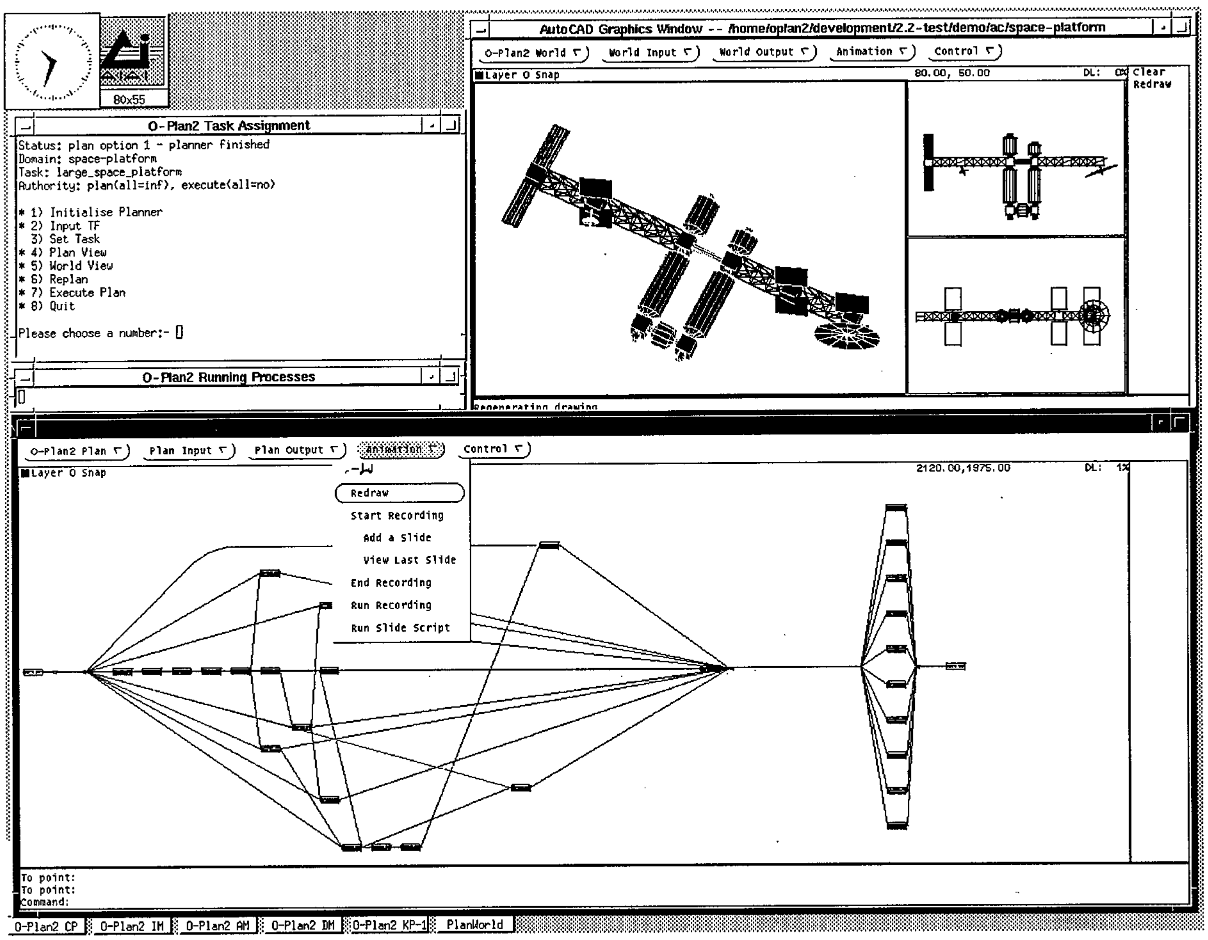


Figure 3: Example Output of the AutoCAD-based User Interface

Recent work on O-Plan has focussed on the representation and management of constraints in planning, particularly in order to simplify some aspects of the architecture (the subject of this paper) and to act as a mechanism for user/system mixed initiative planning [19].

2 Plans Represented as Constraints on Plan Elaborations

It is useful to present a simple abstraction of how a planner or scheduler operates. Figure 4 shows such an abstraction that will be useful in this paper.

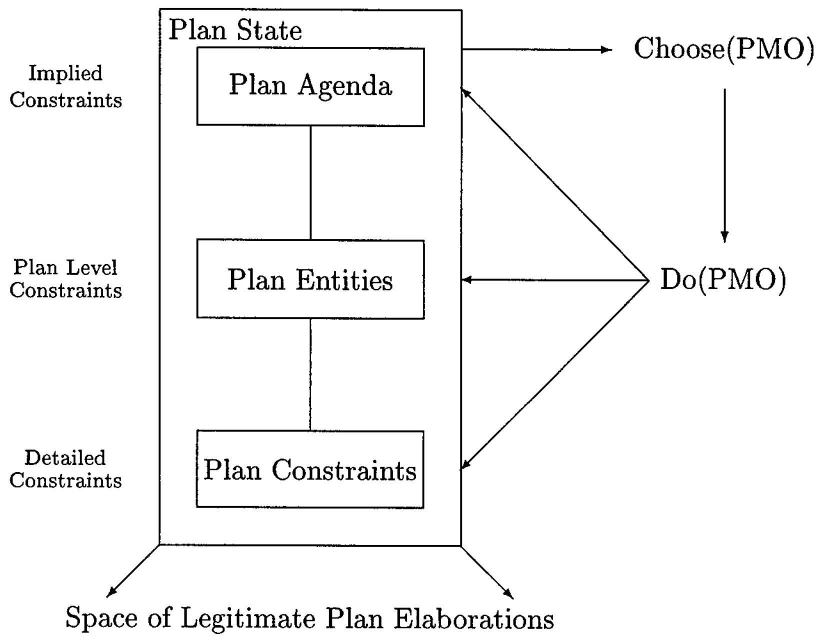


Figure 4: A Framework of Components in a Planning/Scheduling System

Many planners and schedulers work by refining a "current" plan (shown in figure 4 as the *Plan State*). They maintain one or more *partial plans* in this Plan State in which the previous decisions taken during the planning process restrict the space of plan elaborations which can be reached from that point.[1] The planner or scheduler needs to know what outstanding processing requirements exist in the plan (shown in figure 4 as the *Agenda*). These represent the implied constraints on valid plan solutions. One (normally) of these outstanding processing requirements is chosen to be worked upon next. This calls up processing capabilities within the planner which can make decisions and modify the Plan State - these are sometimes called *Plan Modification Operators*. The modifications can be in terms of definite plan structure in the Plan State or by noting further processing requirements (as a result of Plan State critiquing, etc).

We have found it to be useful to separate the plan entities representing the decisions already made during planning into a high level representing the main plan entities shared across all planning system components and known to various parts of the systems, and more detailed specialised plan entities which form a specialised area of the representation of the plan. These lower level more compartmentalised parts can represent specialised constraints within the plan

[1] Plan constraint relaxation is also possible to increase the space of plan elaborations in some systems.

such as time, resource, spatial and other constraints. This separation can assist in the identification of modularity within planning and scheduling systems.

O-Plan has an *Associated Data Structure* (ADS) level of representation [7] which holds the main plan entities (such as actions). The lower level constraints, such as those on on time points and resources in the plan, are managed separately. These lower level constraints are tied to the higher ADS level entities via associations. The TOSCA manufacturing scheduling system [2] which was based on the O-Plan architecture makes use of quite a different ADS level based on resource reservations, but shares the same time point constraint management code at the lower level.

3 Benefits of "Standardising" Constraint Management in Planners

Moves to provide powerful constraint management languages and tools could lead to a rapid expansion of the benefits to be gained by identifying more standard components that can be combined and re-used in planning and scheduling systems. This can allow time network management, management of the persistence of facts across time, resource management, spatial constraint management and other such constraints to be managed by separate components provided by someone other than the original developer or integrator and possibly using more efficient algorithms.

As one example, consider support for the management of temporal relationships in a planner. All modern planners embed some degree of time management for temporal relationships between time points or across time intervals and may provide support for metric (definite) time "stamps" on time points. Many planners also relate their time management to the management of the persistence of facts or propositions across time. This allows planners to reason about whether some required condition is satisfied at a given time. The Time Map Management concepts, clearly described in [5] and used in the FORBIN planner [6], are a good example of the approach. The management of effect and condition (Goal Structure) tables in Nonlin [15] uses a similar approach.

This type of packaging has led to separate study of the support for time management and fact persistence management in planners at various research centres. O-Plan has a Time Point Network Manager [7]. A commercial Time Map Manager (TMM) is available from Honeywell based on the concepts described in [5]. More powerful temporal relationships are managed by the General Electric TACHYON temporal system [13]. In some cases, it has already proved possible to replace some simpler level of time constraint management in a planner with a better packaged and more powerful capability. One example of this has been the combining of the SRI SIPE-2 planner with the GE TACHYON temporal system. Other studies have indicated that the O-Plan Time Point Network Manager can be replaced quite straightforwardly with the Honeywell TMM.

Studies at Edinburgh [8] relating to Resource Management have shown how progressively more capable resource management systems can be incorporated into O-Plan to replace the simple consumable resource handler in the system at present. These studies have developed a

Resource Criterion interface to a Resource Utilisation Manager for the O-Plan planner which has many similarities to the interface used for the Truth Criterion/QA algorithm used in our systems [15]. This framework could incorporate resource handling by mechanisms as powerful as those based on the Habographs [2] constraint management mechanism incorporated in the Edinburgh TOSCA manufacturing scheduler.

Spatial constraint management, which is not currently provided inside O-Plan, has also been explored in the same framework. We believe that clear modular interfaces can allow even such a "foreign" type of constraint management not understood by the core system to be be added reasonably straightforwardly to O-Plan.

4 Constraint Managers in the O-Plan Architecture

O-Plan uses a number of *Constraint Managers* to maintain information about a plan while it is being generated. The information can then be used to prune search (where plans are found to be invalid as a result of propagating the constraints managed by these managers) or to order search alternatives according to some heuristic priority. It is intended that some of these Constraint Managers could be replaced by more efficient or more capable systems in future. This section considers the interfaces between the O-Plan architecture components and Constraint Managers to help others consider packaging and integration issues.

Our experience with earlier AI planners such as Nonlin and the early versions of O-Plan was that a large proportion of the processing time of a planner could be spent in performing basic tasks on the plan network (such as deciding which nodes are ordered with respect to others) and in reasoning about how to satisfy or preserve conditions within the plan. Such functions have been modularised and provided in later versions of O-Plan as Constraint Managers (such as a Time Point Network Manager, an Effect/Condition Manager and a Resource Utilisation Manager), and Support Routines (such as a Graph Operations Processor) to allow for future improvements and replacement by more efficient versions.

Constraint Managers are intended to provide efficient support to a higher level of the planner where decisions are taken. They do not take any decision themselves. They are intended to provide maintain all the information about the constraints they are managing and to respond to questions being asked of them by the decision making level. Examples of Constraint Managers in O-Plan include:

- Time Point Network Manager.
- Effect/Condition Manager and the related Question Answerer.
- Resource Utilisation Manager.
- Object Instantiation (Plan State Variables) Manager.

A guideline for the provision of a good Constraint Manager in O-Plan is the ability to specify the calling requirements for the module in a precise way (i.e., the *sensitivity rules* under which the Constraint Manager should be called by a knowledge source or from another component of the architecture).

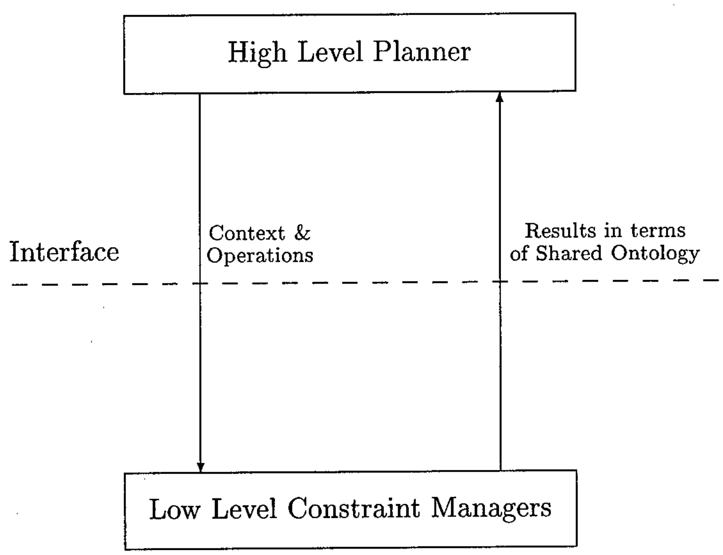


Figure 5: The Interface to Constraint Managers

The following sections explore the definition of an interface between the higher level decision making part of a planning or scheduling system and a lower level constraint manager. Figure 5 shows an overview of the interface.

4.1 Constraint Manager Procedural Interface

A Constraint Manager is a part of the Database Manager component in an O-Plan agent which looks after the Plan State and all of its alternatives (if any). A Constraint Manager may look after a specialised aspect of the Plan State on behalf of the O-Plan Database Manager.

The O-Plan design is being rationalised so that a Constraint Manager has the following generic procedural interface:

1. initialise Constraint Manager and name base context with a given tag[2].
2. terminate Constraint Manager.
3. push context and name new context with a given tag.

[2]Contexts specify alternative views of a Plan State. A tree of such contexts is manipulated by O-Plan.

4. pop context to parent of current context.
5. restore a previously created context which has the tag specified.
6. open update transaction, and within this allow:
   - allow changes to managed entities.
   - queries can be made inside an open transaction. Any query reflects the changes made within the transaction to date.
   - nested open update transactions are not allowed (in O-Plan at present).
7. commit changes made within the update transaction.
8. abort changes made within the update transaction.

Some of the above routines may be inoperative or null for specific managers. In particular, context management as specified above is not needed for any Constraint Manager which chooses to make use of the O-Plan/O-Base context managed structures – since the implementation of the Associated Data Structure layer in O-Plan guarantees that Constraint Managers will only ever be called when the contexts being referred to are preset within the O-Plan planner.

4.2 Shared Plan Ontology between O-Plan and Constraint Managers

There are specialised update and query routines supported by each constraint Manager. These share a common plan entity model within the planner and its Associated Data Structure layer. The design intention has been to keep this minimal, including only those elements that allow relevant communication between higher level planning decisions and lower level constraint management. This model includes *only*:

- a directed acyclic graph of time points.
- ability to map a plan activity node end to a unique time point and a time point to all associated node ends.
- time points as plan entities.
- an ordering relation on two time points – before(tp1,tp2).
- context <tag>s to represent alternative Plan States.
- An understanding of the meaning of a Plan State Variable[3].

These entities allow for information about constraints and options for correcting constraint violations to be communicated in terms of the shared model. All other more specific entities may be unique to a specific Constraint Manager or shared only between pairs of caller and manager.

[3] Currently we represent equality (variable codesignation), inequality (non-codesignation) and other restriction (range or property) constraints on the variable.

4.3 The New O-Plan "Standard" Interface for Constraint Managers

The aim in O-Plan is to provide a standardised interface between each Constraint Manager and the rest of the planner. For this we are seeking to employ a very similar interface to that used by the Nonlin or O-Plan style Condition Question Answerer (QA) or Truth Criterion [15].

A Constraint Manager cannot take any decisions and cannot change parts of the Plan State not under its immediate management. It must return all legitimate answers for the query it is given or must undertake reliably the task it is given. One focus of the O-Plan research has been to build a *planning ontology* which describes those concepts which are shared between constraint managers and those parts of the Plan State which are private to the relevant manager.

A Constraint Manager's primary function is to manage the current set of constraints relevant to that manager (time, resource, spatial, objects, etc) which are part of the Plan State. It must signal to the caller when there is an inconsistent set of such constraints.

The interface allows for a constraint entry to be tested against existing managed constraints to see what the impact of making the entry would be, and then a commit or abort can be done to add it or not (either the commit or the abort could be active – the caller not being able to tell).

All Constraint Manager update routines return one of three results:

- **yes** – constraint is now under management (to be confirmed later by a caller using a commit update transaction).
- **no** – constraint cannot be added within the capabilities of the Constraint Manager and its communications capability to the caller (in terms of the shared ontology of entities).
- **maybe** – constraint can be added if plan entities are altered as specified in terms of the shared entity model. This normally means returning a standard O-Plan "or-tree"[4] of *all* (for search space completeness) the legal ways in which the Plan State can be altered (sets of Plan State Variable restrictions and ordering constraints between time points) to maintain consistency.

  The constraint is *not* added after this maybe response. However, from an implementation perspective, an "actually add constraint" routine may be provided to more cheaply add the constraint immediately following a query which returned "maybe". This would follow action by the caller to ensure at least one of the relevant binding constraints and/or time point orderings options were either dealt with or noted as necessary in the Plan State - thus the caller takes responsibility for resolving inconsistencies (*not* the Constraint Manager).

It is hoped to be able to take the result or-trees generated by the various Constraint Managers in O-Plan (Condition/Effect manager, Resource Utilisation Manager, Plan State Variables

[4] a data structure representing the alternative ways in which the Plan State may be altered in terms of the shared plan ontology.

Manager and the Time Point Network Manager) and merge them into a consistent or-tree which would represent an efficiently ordered set of possibilities – thus reducing the size of the search space.

5 The Constraint "Associator"

To improve the separation of functionality with respect to constraint management in O-Plan, we wish to localise the interactions between changes in one type of constraint that can lead to changes in other types of constraint. In particular, changes in constraints on time points and changes to constraints on plan state variables can have implications for most other constraints being managed (such as effects/conditions, resources, etc.). The detection and cross-relating of such mutual constraints has been problematic in O-Plan to date. Previously, Knowledge Sources had to be written such that any change in one constraint type that could influence another was programmed in. This was a source of complexity and dependency in teh design that we wish to avoid.

The clarification of the constraint manager interface for O-Plan as described in this paper has made us realise the special requirements for the handling of time point constraints and variable constraints in the architecture[5]. These form the core elements in the shared ontology in which communication occurs between the plan entity (ADS) layer and the constraint managers in O-Plan. By recognising that there is a normal constraint management function for time points and variable, but also an *additional* function of association and mutual constraints with other constraint types, we can design better and more modular support for constraints handling in O-Plan and simplify the writing of Knowledge Sources.

Accordingly, the O-Plan agent architecture design in future will allow for an "Associator" component as part of the data base manager which looks after plan states. The Associator mediates between the decisions made by Knowledge Sources and the underlying constraint managers (see figure 6). The function of detecting mutual constraints in which changes to time and/or variable constraints may affect other constraints which themselves refer to the affected time points or variables is localised in the Constraint Associator.

A number of constraint managers can be "installed" into an O-Plan agent. As a minimum, each agent will have a time point manager and a variables manager installed into the Associator. Any number of other constraint managers may then be added depending on the requirements. To give the functionality of the current O-Plan planner this will include the effect/condition manager, the resource utilisation manager, and an "other constraints" manager to keep annotations of other requirements on a plan state (beyond those managed actively by the currently installed managers). In other applications it may be necessary to include spatial constraint managers, etc.

[5]Other evidence from formal studies is also highlighting the value of separating the constraints on time and the variable codesignation/non-codesignation constraints from other aspects of plan representation (e.g., in [9]). We are developing a description of plans as a set of constraints differentiated into *Issues – Nodes – Orderings/Variables/Auxiliary* that we refer to as the <I-N-OVA> model [20] to act as a framework for further study and comparison.

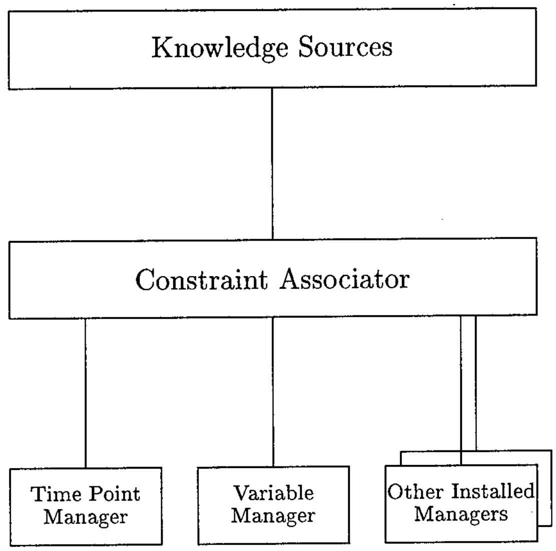


Figure 6: Associator to mediate between Knowledge Sources and Constraint Managers

We believe that this style of interface between the higher level decision making level of the planner and the various Constraint Managers could improve modularity in planning systems[6].

6 Summary

This paper was intended to further discussions on the identification of suitable "standard" re-usable components in planning and scheduling systems.

This paper has presented an overview of the O-Plan system under development at the Artificial Intelligence Applications Institute of the University of Edinburgh. Aspects of the system concerned with separation of functionality within the system, internal and external interfaces have been addressed. The O-Plan system is starting to address the issue of what support is required to build an evolving and flexible architecture to support command, planning and control tasks.

One particular area highlighted has been the interface between planning systems and Constraint Managers able to look after certain specialised aspects of parts of a plan on behalf

[6]Recent work by others (e.g., [10]) is also recognising the practical benefits of being able to isolate the work done for parts of a planning problem into well defined managers which can use specialised algorithms. By not relying on a general search mechanism for all aspects of planning, more realistic tasks can be handled without combinatorial search problems becoming a problem too quickly.

of the overall planning system. An interface to such Constraint Managers has been developed to show how improved packaging can be beneficial to re-use of components. The value of the type of interface developed for the Condition Question Answering procedure in planners (the Truth Criterion) to act as a general interface to a number of different Constraint Managers has been explored.

Acknowledgements

O-Plan is an on-going project at Edinburgh. Current O-Plan work is supported by the US Advanced Research Projects Agency (ARPA) and the US Air Force Rome Laboratory acting through the Air Force Office of Scientific Research (AFSC) under contract F49620-92-C-0042. The United States Government is authorised to reproduce and distribute reprints for government purposes notwithstanding any copyright notation hereon.

Parts of this paper were previously presented at the European Workshop on Planning Systems 1993 (EWSP-93), December 1993, Linkoping, Sweden. Thanks to my colleagues on the O-Plan project, Brian Drabble and Jeff Dalton for the discussions we have held to adopt the architecture discussed here.

References

[1] Allen, J., Hendler, J. & Tate, A., *Readings in Planning*, Morgan-Kaufmann, 1990.

[2] Beck, H., TOSCA: A Novel Approach to the Management of Job-shop Scheduling Constraints, Realising CIM's Industrial Potential: Proceedings of the Ninth CIM-Europe Annual Conference, pages 138-149, (eds. Kooij, C., MacConaill, P.A., and Bastos, J.), 1993.

[3] Chapman, D. Planning for Conjunctive Goals. *Artificial Intelligence*, 32:333-377, 1991.

[4] Currie, K.W. & Tate, A., O-Plan: the Open Planning Architecture, *Artificial Intelligence* 52(1), pp. 49-86, Autumn 1991, North-Holland.

[5] Dean, T. and McDermott, D., Temporal Database Management, *Artificial Intelligence* 32(1):1-56, 1987.

[6] Dean, T., Firby, J. and McDermott, D., Hierarchical Planning Involving Deadlines, Travel Time and Resources, *Computational Intelligence*, 6(1), 1990.

[7] Drabble, B. and Kirby, R.B., Associating A.I. Planner Entities with an Underlying Time Point Network, European Workshop on Planning (EWSP) 1991, Springer-Verlag Lecture Notes in Artificial Intelligence.

[8] Drabble, B. and Tate, A., The Use of Opportunistic and Pessimistic Resource Profiles to Inform Search in an AI Planner, Proceedings of the Second International Conference on AI Planning Systems (AIPS-94), AAAI Press, Chicago, USA, 1994.

[9] Kambhampati, S., Design Tradeoffs in Partial Order Planning, Proceedings of the Second International Conference on AI Planning Systems (AIPS-94), Chicago, IL., USA, 1994.

[10] Penberthy, J.S. and Weld, D.S., Temporal Planning with Continuous Change, Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94), pp. 1010-1015, AAAI Press, 1994.

[11] Reece, G.A. and Tate, A., Synthesizing Protection Monitors from Causal Structure, Proceedings of the Second International Conference on AI Planning Systems (AIPS-94), AAAI Press, Chicago, USA, 1994.

[12] Sacerdoti, E., *A Structure for Plans and Behaviours*, Artificial Intelligence Series, North Holland, 1977.

[13] Stillman, J., Arthur, R. and Deitsch, A., Tachyon: A Constraint-based Temporal Reasoning Model and its Implementation, *SIGART Bulletin*, 4:3, July 1993.

[14] Sussman, G.J., A Computational Model of Skill Acquisition, MIT AI Laboratory Technical Report TR-297, 1973.

[15] Tate, A., Generating Project Networks, Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-77), Cambridge, Mass., USA, 1977.

[16] Tate, A., Planning and Condition Monitoring in a FMS, Proceedings of the International Conference on Flexible Automation Systems, Institute of Electrical Engineers, London, UK, 1984.

[17] Tate, A., Goal Structure, Holding Periods and "Clouds", Proceedings of the Reasoning about Actions and Plans Workshop, Timberline Lodge, Oregon, USA, (eds, Georgeff, M.P. and Lansky, A.) Morgan Kaufmann, 1986.

[18] Tate, A., 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.

[19] Tate, A., Mixed Initiative Planning in O-Plan2, Proceedings of the ARPA/Rome Laboratory Planning Initiative Workshop at Tucson, Arizona, USA, (ed. M. Burstein), Morgan-Kaufmann, 1994.

[20] Tate, A. Characterising Plans as a Set of Constraints - the <I-N-OVA> Model - a Framework for Comparative Analysis, to appear in Special Issue on "Evaluation of Plans, Planners, and Planning Agents", ACM SIGART Bulletin Vol. 6 No. 1, January 1995.

[21] Tate, A., Drabble, B. and R.B.Kirby, R.B., O-Plan2: an Open Architecture for Command, Planning and Control, in *Intelligent Scheduling* (eds. M.Fox and M.Zweben), Morgan Kaufmann, 1994.

[22] Wilkins, D., *Practical Planning*, Morgan Kaufmann, 1988.

Appendix D:

Towards a Plan Ontology

Austin Tate

Citation:

Tate, A., Towards a Plan Ontology, AI*IA Notiziqe (Quarterly Publication of the Associazione Italiana per l'Intelligenza Artificiale), Special Issue on "Aspects of Planning Research", Vol. 9. No. 1, 19–26 - March 1996.

Purpose:

Describes the activity, process and plan ontology upon which the project has provided input to a number of international standards efforts.

Abstract:

This paper describes inputs to various international standardisation efforts for process and plan interchange. Our approach takes a *top down* perspective. It seeks to add the small but *vital* overview that can sit above the detailed representations or ontologies already available. It seeks to provide a framework within which alternative detailed ontologies can be created and evaluated in use.

The contribution of this paper is to propose a structure for a plan ontology which is intended to allow for the progressive definition of the various components in a way which should increase the prospect of achieving a smooth fit of the various components into the whole.

1 Background

It is important that information about processes and activities are sharable within and across organisations. Cooperation and coordination of the planning, monitoring and workflows of the organisations can be assisted by having a clear shared model of what comprises plans, processes and activities.

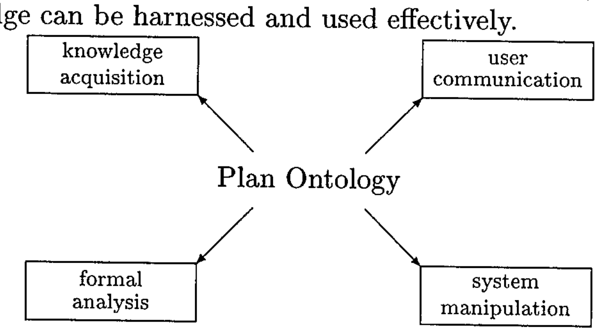
The AI planning community has used explicit domain description languages and plan definitions for more than 25 years. There is a wealth of experience of defining plan representations for both theoretical studies and practical planning. More recently, there have been a number of initiatives to standardise terminology related to processes in PIF (the Process Interchange Format [8]); workflow (the International Workflow Management Coalition [16]); and in the US military planning research community.

In 1992, under the ARPA/Rome Laboratory Planning Initiative (ARPI) [5], a number of participants created the KRSL plan language [9]. Although this has been used for some transfers of information between planning components within the ARPI [1] it has not had the widespread impact desired. Its structure is too rigid and KRSL excludes much that is already being done within planners. In 1994, a group was formed to approach the creation of an ontology for plans using new insights gained over the last few years in the knowledge-sharing community in the US and Europe.

The current document describes a framework for a plan or activity ontology and shows the basis of inputs given to a number of standards activities that relate to plan and process interchange.

2 Purpose of the Plan Ontology

The plan ontology is intended to contribute to a range of purposes including domain modelling, plan capture, plan generation, plan analysis, plan communication, behaviour modelling, etc. By having a shared model of what constitutes a plan, process or activity, organisational knowledge can be harnessed and used effectively.



For example, the Edinburgh plan/activity ontology work has provided input for the following:

1. The ontology for the Enterprise Toolkit on the UK Enterprise Project (partners AIAI, Lloyds Register, Logica, IBM UK and Unilever) [6].

2. To rationalise the O-Plan Task Formalism (Domain Description Language) on the ARPA/Rome Laboratory Planning Initiative project [14].

3. To provide a target representation for a Plan Knowledge Capture Tool on the UK Defence Research Agency project "Acquiring and Using Planning Knowledge for Search and Rescue" [2].

4. To provide a relationship to work on Structured Analysis and Design Techniques (e.g., SADT), Issue-Based Design Methods (e.g., IBIS), Process Management Models and Methods (e.g., IDEF), Entity-Relationship Modelling, Object-Role Modelling (e.g., NIAM), Process Workflow Support, etc.

5. Input to the ARPA/Rome Laboratory Planning Initiative KRSL [9] follow on efforts and the ARPI Plan Ontology Construction Group.

6. Input to discussions and workshops organised by ARPA into ontologies for knowledge sharing, such as the Workshop on Ontology Development and Use, November '94, La Jolla, CA.

7. Input to the Process Interchange Format (PIF) standard being worked on by a number of projects interested in exchanging process information [8]. In particular to move to a more robust basis for version 1.1 of this standard.

8. To relate to the International Workflow Management Coalition work in standardising workflow systems and process terminology via their Glossary of Workflow terms [16].

3 Ontology Structure

The following is the proposed structure of a Plan Ontology document. The structure of the ontology itself and the document that describes it are intended to increase the prospects of achieving integration of the various parts and extensions into the whole.

Meta-ontology Fundamental ontological elements used to describe the ontology itself and the assumptions behind the description.

Top Level Ontology The minimal ontology used as a framework for detailed sections of the ontology. The detailed sections then refine this top level definition.

Library of Shared Ontological Elements Ontological elements which are shared across the detailed sections but which are not necessary for the description of the top level ontology. These are introduced to ensure that detailed ontology sections are more easily integrated into the whole and shared aspects are standardised across the detailed ontologies. This is similar to and shares the objectives of the "Partial Shared View Mechanism" adopted in the Process Interchange Format (PIF) documents [8].

Detailed Ontology Sections The specific section headings for the detail of the ontology reflects experience in the field. They also may reflect a division of responsibility for some aspects of the ontology. Alternative section groupings are admitted. These detailed ontology sections refine the top level ontology and are, where appropriate, encouraged to make use of components from the library of shared ontological elements.

The detailed ontology will include:

Agent
Issue
Activity
Time
Variable
Auxiliary Constraint
Preference
Documentation and Annotation

The core activity model within this ontology draws on the <I-N-OVA> (Issues – Nodes – Orderings/Variables/Auxiliary) constraint model of plans [13] proposed recently to integrate a number of perspectives on plan and process representation.

To give detail to the various detailed sections of the plan ontology, current best practice may be derived from the ontologies in the current KRSL 2.0.2 [9], SRI's ACT language [15], O-Plan's Task Formalism [12], Toronto's TOVE [7], etc.

In a complete document describing the Plan Ontology, encodings of the ontology may also be given in a language which expresses the ontological entities and relationships in symbols. KIF, Conceptual Graphs, LOOM or other representations of the ontology are possible. Experience of using the ontology should also be brought together in some form such as a collection of papers relating experience in using, adapting or extending the ontology.

The rest of this paper gives a complete top level description of a plan ontology within the structure proposed above. It is the basis on which inputs to the various process and plan standardisation efforts and contributions to a number of collaborative projects involving plan interchange have been made.

4 Meta-ontology

The Plan Ontology is composed of a set of ENTITIES and a set of RELATIONSHIPS between ENTITIES.

A RELATIONSHIP is itself an ENTITY that can participate in further RELATIONSHIPS.

ENTITY is a fundamental thing in the domain being modelled. An ENTITY may participate in RELATIONSHIPs with other entities.

RELATIONSHIP is an association between two or more entities[1].

[1] Some means to regularise the terminology used to associate functional or truth values with some relationships is advisable and included in our full proposals.

5 Plan Ontology

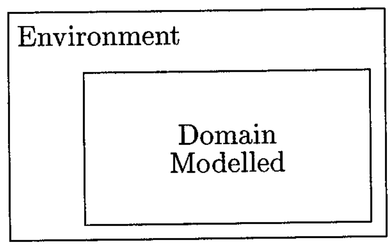
5.1 Informal Context

A Plan is a Specialised Type of Design.

Design for some artifact is a set of constraints on the relationships between the entities involved in the artifact.

Plan is a set of constraints on the relationships between agents, their purposes and their behaviour.

The ontology defines a domain model within which some agents may have purposes and some agents may be capable of performing behaviour. A plan is related to agent purposes and behaviour. Purposes are expressed as constraints on the plan.



The domain modelled sits within an outer environment which may also contain agents whose behaviour is not directly specifiable.

5.2 Principal Definition of a Plan

PLAN is a SPECIFICATION of BEHAVIOUR for some PURPOSE(s). A PLAN may or may not be EXECUTABLE.

BEHAVIOUR is something that one or more AGENTs PERFORM.

AGENT is an entity that can do one or both of the following:

- PERFORM [, or participate in the PERFORMance of,] BEHAVIOUR. It can be a supplier of force behind BEHAVIOUR.
- HOLD some PURPOSE(s).

EXECUTABLE means a PLAN can be PERFORMed by some AGENT(s).

PURPOSE is a CONSTRAINT which is HELD by one or more AGENT(s).

CONSTRAINT is a RELATIONSHIP. It expresses an assertion that can be evaluated with respect to a given PLAN as "something that may hold" and can be elaborated in some language.

SPECIFICATION is a set of CONSTRAINTs.

5.3 Agent to Constraint Relationships

There is a need to differentiate constraints associated with a plan which are hard (environmental and set) requirements and those soft constraints or desirable features. There is also a need to recognise the agent (or computer process) that adds specific constraints during the planning process. It is likely that this information will be needed in the core ontology rather than being left to the detailed ontologies. The following is one suggestion for this.

INTEND, DESIRE, ENFORCE, SYNTHESIZE An AGENT may INTEND, DESIRE, ENFORCE or SYNTHESIZE a CONSTRAINT.

INTENDED CONSTRAINT is a CONSTRAINT, INTENDED by some AGENT, which, when satisfied, supports the RELEVANCE of a PLAN.

DESIRED CONSTRAINT is a CONSTRAINT, DESIRED by some AGENT, which, when satisfied, [supports or increases] the EFFECTIVENESS of a PLAN. It may be a DOMAIN OBJECTIVE CRITERION in domains for which such criteria have have defined.

AGENT HELD CONSTRAINT is an INTENDED CONSTRAINT or a DESIRED CONSTRAINT. I.e., PURPOSE = CONSTRAINT which is HELD by an AGENT = AGENT HELD CONSTRAINT.

ENFORCED CONSTRAINT is a CONSTRAINT, ENFORCED by some AGENT, which, when satisfied, supports the EXECUTABILITY of a PLAN. [The AGENT is often the "ENVIRONMENT" but can also be some other agent outside of the modelled agents (e.g., regulatory authorities if these are not modelled).]

SYNTHESIZED CONSTRAINT is a CONSTRAINT, SYNTHESIZED by some AGENT, which is added to a PLAN as part of the planning process. [The AGENT is often a computer system assisting with planning.]

6 Library of Shared Ontological Elements

The library of shared ontological elements contains elements which are shared across the detailed sections but which are not necessary for the description of the top level ontology. These are introduced to ensure that detailed ontology sections are more easily integrated into the whole and minimum shared aspects are standardised across the detailed ontologies.

This library can be viewed as having two parts:

1. a minimum set of shared elements common to many of the ways in which detailed ontology sections are provided within the ontology. These are provided as a way to ease the integration of the detailed ontology sections into the whole ontology. The minimal set of shared ontological elements is likely to be quite small.

2. convenient extensions shared across two or more detailed sections. We can thus view the library as making available a range of already defined ontological elements which we can draw on to define the detailed ontological sections. Existing ontologies for relevant or commonly used elements can thus be made available.

Only two entities and one relationship are proposed for inclusion in the minimum set – TIME POINT, ENTITY VARIABLE and TEMPORAL CONSTRAINT.

Since the subject of the ontology is activity plans which are modelled with a temporal aspect, a single shared ontological entity related to time is provided to assist in defining detailed ontologies for time itself and for other related detailed ontological components.

TIME POINT is an ENTITY that represents a specific, instantaneous, point along a time line which is an infinite sequence of time points.

TEMPORAL CONSTRAINT is a RELATIONSHIP between a CONSTRAINT and one or more TIME POINTs.

A detailed ontology of time defines the relationships possible between time points (e.g., a TIME INTERVAL may be defined as a RELATIONSHIP between two TIME POINTs.

ENTITY VARIABLE allows reference to an entity without naming the specific entity. An ENTITY VARIABLE is a virtual entity which anticipates a deferred real entity.

It is often necessary to defer the naming of an entity within a plan or an activity – much in the same way that natural language provides pronouns. A single shared ontological entity is provided to assist in defining the detailed ontologies.

The detailed definition for ENTITY VARIABLE is given in the detailed ontology for variables.

7 Agent

Detailed ontology for Agent.

AGENT to PLAN RELATIONSHIPS are certainly important to model the notion of "having a plan" (as described by Martha Pollack in her thesis [10]). These relationships can also capture the notion of commitment to plans, plan purpose relationships, etc.

AGENT to AGENT RELATIONSHIPS can express authority, delegation, contracts, organisational relationships etc.

Predefined Constants

ENVIRONMENT – There is a predefined AGENT called the "environment". It can only establish ENFORCED CONSTRAINTS and cannot participate in INTENTED, DESIRED or SYNTHESIZED CONSTRAINT relationships. It may be used to describe all BEHAVIOUR which is not EXECUTABLE by specifically modelled AGENTs.

8 Issue

ISSUE is an implied or pending constraint on a plan. Issues or requirements remaining to be addressed in the plan. These can be used to hold outstanding requirements, the results of plan analysis (e.g., critics) which need attention, etc.

The ontology for issues is likely to be the subject of active research. Discussion of the granularity level of issues is also likely. One source of the types of Issues used in planning is from the ontology used on the PLANIT project [4].

An open ended framework for issues should be provided.

9 Activity

9.1 Principal Definition of Activity

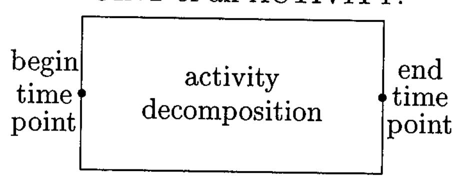
ACTIVITY is a BEHAVIOUR.

ACTIVITY is PERFORMed by one or more AGENTs.

BEGIN TIME POINT, END TIME POINT An activity has a BEGIN TIME POINT and an END TIME POINT.

The CONSTRAINT BEFORE(BEGIN TIME POINT,END TIME POINT) holds.

TEMPORAL CONSTRAINTS may be stated with respect to the BEGIN TIME POINT and/or END TIME POINT of an ACTIVITY.



An activity may optionally have one or more ACTIVITY DECOMPOSITIONs. These provide encapsulation of the detailed descriptions of activities.

Abstraction level modelling may or may not be used within such an encapsulation. Abstraction is an orthogonal issue which can be addressed in a detailed ontology.

Note that an activity may be an action, a resource usage period or some external (to the model) event at this level of the ontology, as no ontological commitment to an action based representation is made at this level.

9.2 Actions and Events

ACTION is an ACTIVITY done by a known (modelled) AGENT.

EVENT is an ACTIVITY done by an unknown (or unmodelled) agent (conventionally referred to as the "environment").

9.3 Activity Decomposition

ACTIVITY DECOMPOSITION is the set of SUB-ACTIVITIES and/or SUB-ACTIVITY CONSTRAINTS.

In general there may be multiple ways in which an activity can be decomposed.

SUB-ACTIVITIES is a set of ACTIVITIES.

SUB-ACTIVITY CONSTRAINTS is a set of CONSTRAINTS.

Predefined Constants

SELF – Within an activity decomposition, the activity itself can be referred to as "SELF" (if necessary).

START, FINISH may be defined to assist in the definition of activity decompositions for a top level activity which serves to specify a PLAN.

10 Time

TIME POINT – elaboration of minimal shared ontology entity.

TIME INTERVAL is a specific TEMPORAL CONSTRAINT that is usefully defined in the detailed time ontology. It is a RELATIONSHIP between two TIME POINTS.

DURATION – an absolute distance between two time points measured in some units (e.g., years, weeks, etc.).

Further details can be included from, e.g., the KRSL 2.0.2 ontology section 2 [9].

11 Variable

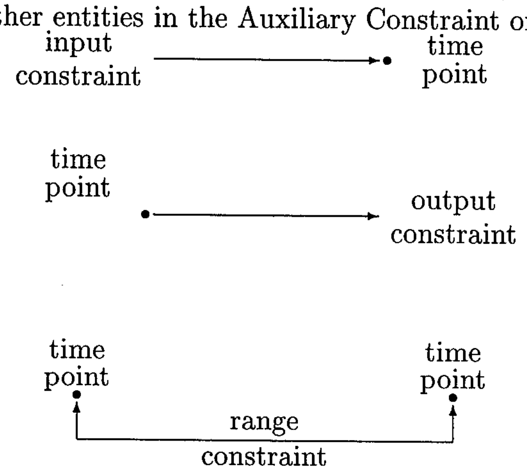
ENTITY VARIABLE – an elaboration of the minimal shared ontology entity is possible.

ENTITY VARIABLE CONSTRAINT allows RELATIONSHIPS such as co-designation (equality) between variables, non-co-designation (in-equality) between variables, and possibly other constraints such as type membership, general restriction facilities, ranges, etc.

12 Auxiliary Constraint

12.1 Constraints involving Time Points

Three types of TEMPORAL CONSTRAINT are usefully defined – input, output and range constraints. They are not the only types of constraint which can be stated in the ontology (as any relationship between two or more entities can be a constraint). However, they are used frequently in describing other entities in the Auxiliary Constraint ontology.



INPUT CONSTRAINT is a TEMPORAL CONSTRAINT between a CONSTRAINT and a TIME POINT that may or may not be satisfied immediately before the given time point. It is evaluated with respect to that time point.

OUTPUT CONSTRAINT is a TEMPORAL CONSTRAINT between a CONSTRAINT and a TIME POINT that may or may not be satisfied immediately after the given time point. It is evaluated with respect to that time point.

RANGE CONSTRAINT is a TEMPORAL CONSTRAINT between a CONSTRAINT and two TIME POINTs that may or may not be satisfied at all times between the two given time points.

12.2 Details of Auxiliary Constraints

This is likely to be the subject of active research, so a general framework and extension facilities should be provided. The following is the framework adopted in the O-Plan <I-N-OVA> ontology [13] and as a basis for the O-Plan Task Formalism language [12]. This framework deliberately seeks to ensure overlap with activity and process representations in workflow and software engineering work.

AUTHORITY CONSTRAINTS are AGENT to AGENT RELATIONSHIPS. Possibly based on the ORDIT ontology [3]. Also see O-Plan TF Authority Statements [11].

STATE CONSTRAINTS express domain statements with respect to time. A Synonym for State Constraint might be World Condition. Possibly based upon SRI's ACT [15] and O-Plan TF condition/effect ontologies [12].

There are three purposes for state constraints:

1. context or environment constraints (filter conditions).
2. value added input/output chain.
3. setup conditions and/or side-effects.

RESOURCE CONSTRAINTS Possibly based on Toronto TOVE resource ontology [7]. See also KRSL [9], O-Plan TF [12]and SRI's ACT [15].

OTHER CONSTRAINTS Open ended framework (e.g., for spatial constraints and research opportunities). E.g., see O-Plan TF "other constraints" statement [12]

13 Preference

DESIRED CONSTRAINTS relate individual AGENT DESIRES for some CONSTRAINT within a plan. An ability to describe the relationship between different agent's preferences and to provide facilities to allow a pairwise comparison of two plans with respect to these preferences should be provided in a detailed ontology.

14 Documentation and Annotation

Although not part of the ontology, any supporting language in which the ontology can be expressed is required to provide documentation and annotation facilities. An ability to name and give a version number or revision date to an ontology section, or to an ontological element in a library of such elements is to be provided. An ability to note which other ontology sections or library elements are used as a basis for any given section is to be provided.

Acknowledgements

The present state of this ontology of plans has benefited from detailed comment and input from Mike Uschold and members of the Enterprise, Defence Research Agency Search and Rescue, and O-Plan project teams. The details have been refined in discussion with members of the KRSL planning ontology development group formed under the ARPA/Rome Laboratory Planning Initiative (ARPI) and the ARPI Plan Ontology Construction Group.

Effort sponsored by the Advanced Projects Research Agency (ARPA) and Rome Laboratory, Air Force Materiel Command, USAF, under grant number F30602-95-1-0022. The U.S. Government is authorised to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon.

The views and conclusions contained herein are those of the author and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of ARPA, Rome Laboratory or the U.S. Government.

References

[1] Burstein, M.H., Schantz, R., Bienkowski, M.A., desJardins, M.E. and Smith, S.F., The Common Prototyping Environment – A Framework for Software Technology Integration, Evaluation and Transition, *IEEE Expert*, Vol. 10, No. 1, pp. 17-26, February 1995, IEEE Comp. Soc.

[2] Cottam, H., Shadbolt, N., Kingston, J., Beck H.A. and Tate, A., Knowledge Level Planning in the Search and Rescue Domain. Proceedings of Expert Systems 95 Conference, Cambridge, 1995.

[3] Dobson, J. and Strens, R., Organisational Requirements Definition for Information Technology Systems, in Proceedings of the Conference on the Theory, Use and Integrative Aspects of IS Methodologies, pp 295-308, British Computer Society, 1993.

[4] Drummond, M.E. and Tate, A. PLANIT Interactive Planners' Assistant – Rationale and Future Directions. Reprints of working papers to the Alvey Programme PLANIT Community Club distributed in 1986-7. Available as AIAI-TR-108, AIAI, University of Edinburgh.

[5] Fowler, N., Cross, S.E. and Owens, C. The ARPA-Rome Knowledge-Based Planning and Scheduling Initiative, *IEEE Expert*, Vol. 10, No. 1, pp. 4-9, February 1995, IEEE Comp. Soc.

[6] Fraser, J. and Tate, A., The Enterprise Tool Set – An Open Enterprise Architecture, Proceedings of the Workshop on Intelligent Manufacturing Systems, International Joint Conference on Artificial Intelligence (IJCAI-95), Montreal, Canada, August 1995.

[7] Fox, M.S., Chionglo, J.F. and Fadel, F.G., A Common-Sense Model of the Enterprise, Proceedings of the Second IERC. Department of Industrial Engineering, University of Toronto.

[8] Lee, J., Yost, G. and the PIF Group, Process Interchange Format and Framework, Version 1.0, MIT Center for Coordination Science, Working Paper No. 180, December 22, 1994. `http://www-sloan.mit.edu/CCS/pifmail.html`

[9] Lehrer, N. (ed.), ARPI KRSL Reference Manual 2.0.2, February, 1993. ISX Corporation.

[10] Pollack, M. Inferring Domain Plans in Question Answering, Ph.D. Thesis, Department of Computer and Information Science, University of Pennsylvania, May 1986.

[11] Tate, A., 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.

[12] Tate, A., O-Plan Task Formalism Manual, Version 2.2, July 5, 1994. Artificial Intelligence Applications Institute, University of Edinburgh.

[13] Tate, A., Characterising Plans as a Set of Constraints – the <I-N-OVA> Model – a Framework for Comparative Analysis, to appear in Special Issue on "Evaluation of Plans, Planners, and Planning Agents", ACM SIGART Bulletin Vol. 6 No. 1, January 1995.

[14] Tate, A., Drabble, B. and Kirby, R.B., O-Plan2: an Open Architecture for Command, Planning and Control, in *Intelligent Scheduling* (eds. Fox, M. and Zweben, M.), Morgan Kaufmann, 1994.

[15] Wilkins, D.E. and Myers, K.L., A Common Knowledge Representation for Plan Generation and Reactive Execution, SRI International Artificial Intelligence Center. This paper has been accepted to the Journal of Logic and Computation, and should appear in late 1994 or 1995.

[16] Workflow Management Coalition Glossary – A Workflow Management Coalition Specification, November 1994.
`http://www.aiai.ed.ac.uk/WfMC/`

Attachment: KRSL Plan Ontology Working Group

During 1994, the ARPA/Rome Laboratory Planning Initiative (ARPI) Plan Ontology Construction Group decided to discuss a follow on to the previous KRSL version 2.0.2 used within the ARPI. The plan ontology structure described in this paper was provided as input to these deliberations.

What is a Plan?

Following some preparatory electronic discussions, at the 12th October 1994 meeting they agreed 4 sentences to define what a plan is and how the principal entities relate to a plan. The definition was:

- A PLAN is a SPECIFICATION of BEHAVIOUR for some PURPOSE(s).
- BEHAVIOUR is something that one or more AGENTs PERFORM.
- An AGENT is an entity that PERFORMs BEHAVIOUR and/or can have PURPOSE(s).
- A PURPOSE is an EFFECT that is [INTENDED or DESIRED] by an AGENT.

KRSL-Plans Ontology for Activity

Over the following months a working group[2,3] worked on the next level of the ontology and agreed the next level of definition (draft of 2nd February 1995 with minor later lexical edits).

ACTIVITY is an important building block in the Plan Ontology. A Plan is itself a description of activity but with the additional relationship of the activity to purpose (and the agents which have the purpose).

An Activity can relate directly to an action that is performed in a discrete fashion, or may relate to the period of usage of resources. This can allow the ontological entity of activity to merge both action planning and resource scheduling perspectives.

BEHAVIOUR is the performance of one or more ACTIVITIES (a non-empty set of activities).

An ACTIVITY takes place over a TIME INTERVAL.

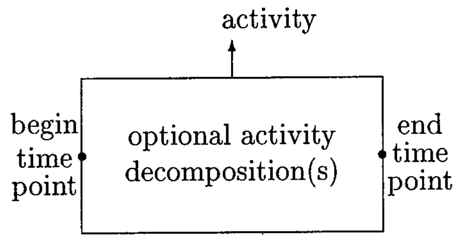
The TIME INTERVAL for an ACTIVITY is identified by its two ends, the BEGIN TIME POINT and the END TIME POINT.

An ACTIVITY may optionally have CONSTRAINTS associated with it or with its TIME INTERVAL.

An ACTIVITY may bring about certain STATES OF AFFAIRS.

[2] Austin Tate (chair), David Wilkins (SRI), Steve Smith (CMU) and Bill Swartout (USC/ISI).

[3] A more detailed level of activity model in the ontology was proposed but is not reproduced here – see `http://www.aiai.ed.ac.uk/~bat/krsl-plans.html`.



Optionally, an ACTIVITY may be decomposed into one or more SUB-ACTIVITIES to provide more detail. There can be several alternative such ACTIVITY DECOMPOSITIONS.

SUB-ACTIVITY: Sub-activities are the constituent activities designated in any ACTIVITY DECOMPOSITION.

Notes: Referring to an activity as a sub-activity refers to the role of an ACTIVITY in a relationship with another ACTIVITY such that performance of the SUB-ACTIVITY is considered to be part of the performance of the other ACTIVITY.

ACTIVITY DECOMPOSITION: The specification of how an ACTIVITY is decomposed into one or more SUB-ACTIVITIES; this may include the specification of constraints on and between the SUB-ACTIVITIES.

Notes: The constraints can be sub-activity orderings, world conditions, effects, resource requirements, organisational permissions, etc.

Notes: Activity decomposition does not necessarily imply that a different level of abstraction to that used in the main activity is used in the description of the sub-activities and the constraints on them. For example, it is possible to provide an activity decompositions which uses recursion by including the parent activity type as a sub-activity. Model Abstraction level is orthogonal to structural activity decomposition level.

PRIMITIVE ACTIVITY is an ACTIVITY with no (further) ACTIVITY DECOMPOSITION.

STATES OF AFFAIRS - broadly defined to mean things we can evaluate as holding or not in the (model of the) world. They can refer to an individual world state (such as NOW), or may refer to world histories, changes between world states, etc.

An ACTIVITY may change the STATE-OF-AFFAIRS during its performance.

CONSTRAINTS can be stated with respect to none, one or more than one time point. They express things which are required to hold. They are evaluable with respect to a specific PLAN as holding or not holding.

Such constraints may refer to world statements (conditions and effects), resource requirements and usage, authority requirements or provision, etc.

Appendix E:

Representing Plans as a Set of Constraints - the <I-N-OVA> Model

Austin Tate

Citation:
Tate, A., 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, Edinburgh, May 1996, AAAI Press.

Purpose:
Describes a unifying constraint-based framework for representing, reasoning abouyt and communicating activity, process and plan information between human and system agents.

Abstract:
This paper presents an approach to representing and manipulating plans based on a model of plans as a set of constraints. The <I-N-OVA>[1] (*Issues – Nodes – Orderings/Variables/Auxiliary*) model is used to characterise the plan representation used within O-Plan and to relate this work to emerging formal analyses of plans and planning. This synergy of practical and formal approaches can stretch the formal methods to cover realistic plan representations, as needed for real problem solving, and can improve the analysis that is possible for production planning systems.

<I-N-OVA> is 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 is intended to support new work on automatic manipulation of plans, human communication about plans, principled and reliable acquisition of plan information, and formal reasoning about plans.

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

1 Motivation

The <I-N-OVA> (*Issues – Nodes – Orderings/Vari- ables/Auxiliary*) Model is a means to represent plans 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 plans can be represented by a set of constraints on the behaviours possible in the domain being modelled and that plan communication can take place through the interchange of such constraint information.

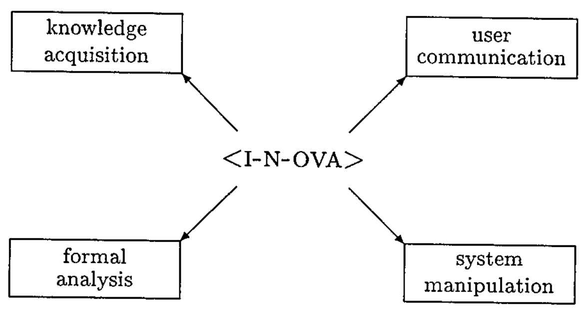


Figure 1: <I-N-OVA> Supports Various Requirements

As shown in figure 1, the <I-N-OVA> constraint model underlying plans is intended to support a number of different uses of plan representations:

- for automatic manipulation of plans and to act as an ontology to underpin such use;
- a common basis for human communication about plans;
- a target for principled and reliable acquisition of plan information;
- formal reasoning about plans.

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

Our aim is to characterise the plan representation used within O-Plan [Currie & Tate 91],[Tate et. al. 94c], to link this to emerging work on process modelling in the workflow community, and to more closely relate this work to emerging formal analyses of plans and planning. This synergy of practical and formal approaches can stretch the formal methods to cover realistic plan representations as needed for real problem solving, and can improve the analysis that is possible for production planning systems.

2 Representing Plans as a Set of Constraints

A plan is represented as a set of constraints which together limit the behaviour that is desired when the plan is executed. Work on O-Plan [Currie & Tate 91],[Tate et. al. 94c] and other practical planners [Allen et. al. 90] has identified different entities in the plan which are conveniently grouped into three types of constraint. The set of constraints describes the possible plan elaborations that can be reached or generated as shown in figure 2.

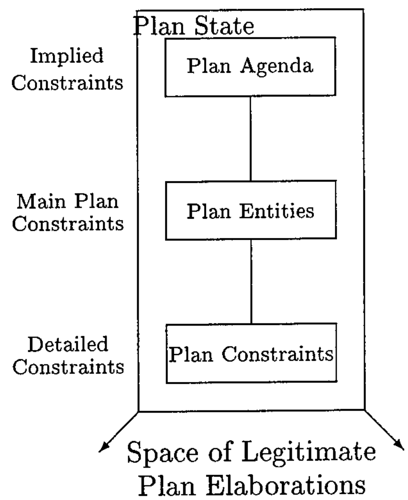


Figure 2: Constraints Define the Space of Plan Elaborations

The three types of constraint in a plan are:

1. Implied Constraints or "Issues"[2] – representing the pending or future constraints that will be added to the plan as a result of handling unsatisfied requirements, dealing with aspects of plan analysis and critiquing, etc. The implied constraints are the issues to be addressed, i.e., the "to-do" list or agenda which can be used to decide what plan modifications should be made to a plan by a planner (user or system).

2. Plan Entities or Plan Node constraints – the main plan entities related to external communication of a plan. They describe a set of external names associated with time points. In an activity planner, the nodes are usually the actions in the plan associated with their begin and end time points. In a resource centred scheduler, nodes may be the resource reservations made against the available resources with a begin and end time point for the reservation period.

[2]We have previously used a variety of different names for these constraints: *Agenda Entries* reflecting the chosen method of representation in O-Plan; *Flaws* as suggested by Sam Steel of Essex University in the mid 1980s and reflecting the original concentration of representing the outcome of plan critics which found interactions in the teleological structure that had to be corrected; *To-do list entries* reflecting common usage in business; *Pending Processing Requirements* reflecting the notion that they implied future plan manipulation or constraints; and others. We have settled on *Issues* suggested by Craig Wier of ARPA in 1994 as being an easily understood term that reflects both the need to handle problems and the positive opportunities that present themselves.

3. Detailed Constraints – associated with plan entities and representing specialised constraints on the plan. Empirical work on the O-Plan planner has identified the desirability of distinguishing two special types of detailed constraint:

   - Ordering or Temporal Constraints (such as temporal relationships between the nodes or metric time properties).
   - Variable Constraints (especially co-designation and non-co-designation constraints on plan objects).

   These two constraint types are highlighted since they may form part of other constraints within a temporal reasoning domain such as occurs in planning and scheduling problems. Knowing that these constraints have such "cross-associations" has been found to simplify the design of constraint handling mechanisms and ease implementation issues [Tate 93b],[Tate et. al. 94d].

   Other Detailed Constraints relate to input (pre-) and output (post-) and protection conditions, resources, authority or control requirements, spatial constraints, etc. These are referred to as:

   - Auxiliary Constraints

   Auxiliary Constraints may be expressed as occurring at a time point (referred to as "point constraints") or across a range of the plan (referred to as "range constraints"). Point constraints can be used to express input and output constraints on nodes or for other constraints that can be expressed at a single time point. Range constraints relate to two or more time points and can be used to express protection intervals, etc.

3 The <I-N-OVA> Model

A plan is represented as a set of constraints of three principal types. To reflect the three main types of constraint identified and their differentiation in the model, the constraint set for a plan is written as <I-N-OVA> (*Issues – Nodes – Orderings/Variables/Auxiliary*). I stands for the the issues agenda or implied constraints, N for the node or plan entity constraints, and OVA for the detailed constraints held as three types (O for ordering constraints, V for variable constraints, and A for the other auxiliary constraints).

The auxiliary constraints are given 4 sub-types: Authority, Conditions, Resources and Other and all may be stated as point (related to a single time point), range (related to two time points) or multi-point constraints. Further sub-types are possible for any of the Auxiliary Constraints and the nature of these reflects on-going work on knowledge modelling for planning, scheduling and process modelling domains (e.g., [Tate 93a], [Tate et. al. 94b], [Uschold et. el. 95]).

The <I-N-OVA> constraint model for plans contains a hierarchy of constraint types and sub-types as follows:

```
Plan Constraints
  I   - Implied Constraints
  N   - Node Constraints
  OVA - Detailed Constraints
        O - Ordering Constraints
        V - Variable Constraints
        A - Auxiliary Constraints
            - Authority Constraints
              - subtypes
            - Condition Constraints
              - subtypes
            - Resource Constraints
              - subtypes
            - Other Constraints
              - subtypes
```

The node constraints in the <I-N-OVA> model set the space within which a plan may be further constrained. The issues and OVA constraints restrict the plans within that space which are valid.

The <I-N-OVA> model currently assumes that it is sufficiently general for each node (referred to as N constraints) to be associated with just two time points, one representing the begin of the node and the other representing the end of the node. Further research may indicate that a more general, multiple time point association of nodes to time points may be necessary.

Hierarchical or abstraction level modelling is possible for all constraint types within the <I-N-OVA> model. To reflect this possibility, an <I-N-OVA> model which is described hierarchically or with levels of abstraction will be referred to as a Hierarchical <I-N-OVA> model. This will be written as Δ-<I-N-OVA>.

The Δ is a triangle pictogram used to represent hierarchical expansion. It can be written in an alternate all character version as H-<I-N-OVA>.

4 The Triangle Model of Activity

The <I-N-OVA> auxiliary constraints incorporate details from the Triangle Model of Activity used to underpin the Task Formalism (TF) domain description language [Tate et. al. 94a] used for O-Plan [Currie & Tate 91],[Tate et. al. 94c]. The Triangle Model seeks to give a clear description of activities, tasks and plans in a common framework that allows for hierarchical decomposition and time relationships along with authority, pre- and post-conditions, resources and other constraints. The Triangle Model can be used as a basis for planning domain modelling and for supportive task description interfaces.

The aim in the Triangle Model is to simplify some of the notions for expressive plan and activity representations from AI planning. It seeks to relate these notions to existing systems-engineering requirements capture and modelling languages and methods (like SADT [Ross 85], IDEF [Mayer & Painter 91], CORE [Curwen 90], HOOD [HOOD 89], etc.), and to recent work on Process Interchange Format (PIF) [PIF 94], workflow standards [WfMC 94] and enterprise modelling frameworks [Uschold et. al. 1995].

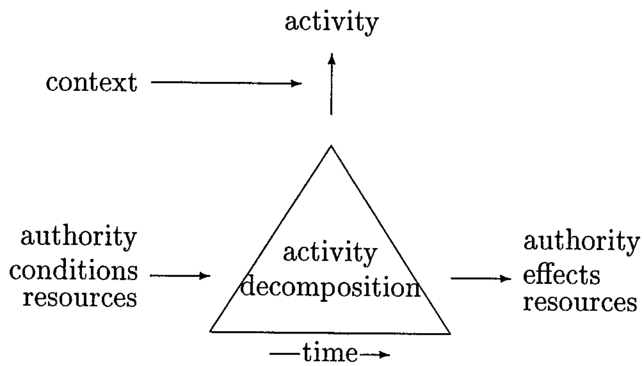


Figure 3: O-Plan Triangle model of Activity

Figure 3 shows the Triangle Model of Activity. The vertical dimension reflects activity decomposition, the horizontal dimension reflects time. A context allows for the relevance of a particular decomposition to be made to depend on the situation in which it may be used. Inputs and outputs are split into three principal categories (authority, conditions/effects and resources). Arbitrarily complex modelling is possible in all dimensions. Types and sub-types are used to further differentiate the inputs and outputs, and their semantics.

"Entry" to the model can be from any of the three points in the triangle: it can be used from the top vertex to ask for activity expansions or decompositions, or from the right side to ask for activities satisfying or providing the output requirement (authority, goal or resource). These two points are used mostly by AI planners to date. The third point from the left side can reflect non-intended triggering conditions for an action and will be needed when improved independent processes are modelled within planers as in the EXCALIBUR [Drabble 93] extension to Nonlin [Tate 77].

The activity decompositions shows the expansion of the activity to a greater level of detail if that is modelled. It can include details of protection conditions that span points within a decomposition.

Variables may appear in an activity description. Differentiation between those variables used in the external specification (outside the triangle) and those only used within the activity decomposition (internal to the triangle) is possible.

The O-Plan time model defines a set of time points which can be related to an absolute start of time (for metric time statements) or which can be related to one another (for relative time relationships). Temporal relationships between an activity (referred to as *self*) and the sub-activities within a decomposition may be stated with reference to the two "ends" of any activity. Arbitrarily complex temporal relationships (e.g., [Allen & Koomen 93]) are possible in the general Triangle Model.

The "intentions" or "rationale" behind the use of a particular activity can be related to the features of this Triangle Model. Causality or teleology modelled via activity pre-conditions/post-conditions has been used in AI planners for many years to record the plan rationale (e.g., in Nonlin [Tate 77]). In the richer model now in use in O-Plan, rationale in terms of resource usage and supply, authority requirements or delegation may also be stated. This makes it possible to use a uniform approach to the modelling of authority, product flow and resource requirements.

5 Relationship of Triangle Model to O-Plan TF Schemas

The Triangle Model of activity maps directly to an O-Plan Task Formalism (TF) schema. TF is the domain description language for O-Plan. The following shows the components of a simplified schema. "..." indicates repetition of the previous component. Further detail is available in [Tate et. al. 94a].

```
schema <schema_name>;
   ;;; public information
   vars     <var> = <var_restriction>, ... ;
   expands <pattern> ;
   only_use_for_authority <authority_statement>,...;
   only_use_for_effects   <effect_statement>,...;
   only_use_for_resources <resource_statement>,...;

   ;;; private information
   local_vars      <var> = <var_restriction>,...;
   vars_relations <var> <relation> <var>,...;
   nodes           <node_number> <node_form>,...;
   orderings       <node_end> ---> <node_end>,...;
   time_windows    <time_window_spec>,...;
   authority       <authority_statement>,...;
   conditions      <condition_statement>,...;
   effects         <effect_statement>,...;
   resources       <resource_statement>,...;
   other_constraints <constraint_statement>,...;
end_schema;
```

6 Domain Operators, Tasks and Plans

Figure 4 illustrates the dependency relationships between domain, task and plan knowledge. Tasks and Plans are both based upon the entities in the Domain model. Plans also are elaborations of a specific Task.

- *Domain* knowledge describes "fixed" things like facilities, organisational relationships, procedures, systems, products and the types of resource available. This knowledge is likely to be highly reusable for many different requirements.

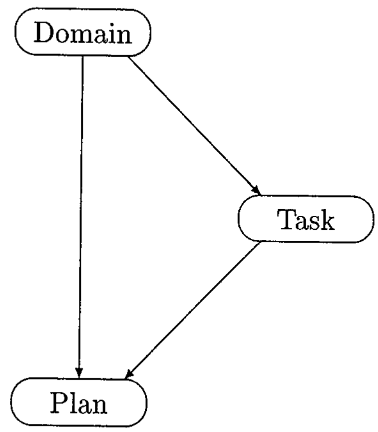


Figure 4: Dependencies between Domain, Task and Plan Knowledge Partitions

- *Task* knowledge describes the objectives such as the goal or goals which the plan is designed to achieve, the activity to be carried out, the actual resources available, the time available, etc.
- *Plan* knowledge describes a particular way (currently under exploration) in which the specified task objectives can be achieved in the current domain.

<I-N-OVA> is intended to underpin domain, task and plan modelling needs in a planning system whether human, computer or mixed agents are involved. Communication between planning agents in O-Plan takes place via Plan Patches [Tate 89] which are also based on the Triangle Model of Activity and the <I-N-OVA> constraint components.

7 Relationship of <I-N-OVA> to Work in Systems Engineering

There is a deliberate and direct mapping from the O-Plan Triangle Model of Activity and the <I-N-OVA> Constraint Model of Plans to existing structured analysis and diagraming methods such as IDEF and R-Charts. Other researchers have recognised the value of merging AI representation concepts with structured analysis and diagramming techniques for systems requirements modelling [Borgida et. al. 85],[Ramesh & Dhar 94] and the earlier work on the Programmer's Apprentice [Rich & Waters 88].

7.1 Modelling Processes and Activities

IDEF0 [Mayer 92] is a functional modelling method and diagraming notation that has been used for modelling processes[3]. Figure 5 shows the basic component.

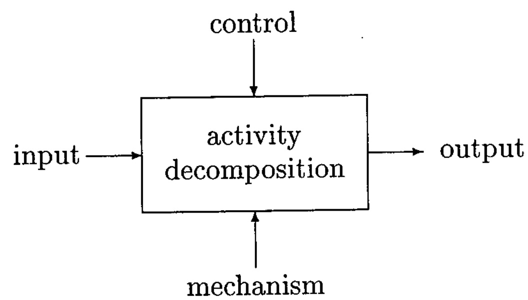


Figure 5: IDEF0 model

IDEF modellers usually use "control" for authority-related triggers and "mechanism" to reflect resource availability. A criticism of IDEF is the lack of direct support for modelling the different types of output and their intended destination. Experienced IDEF modellers use the arc labels, naming conventions and the "notes" system in an IDEF support "kit" to encode this information.

R-Charts [Ushakiv & Velbitskiy 93] are one of the ISO approved diagraming conventions for program constructs (ISO/IEC 8631 [ISO/IEC 89]). Figure 6 shows the basic component which explicitly acknowledges the importance of control (or authority) related outputs.

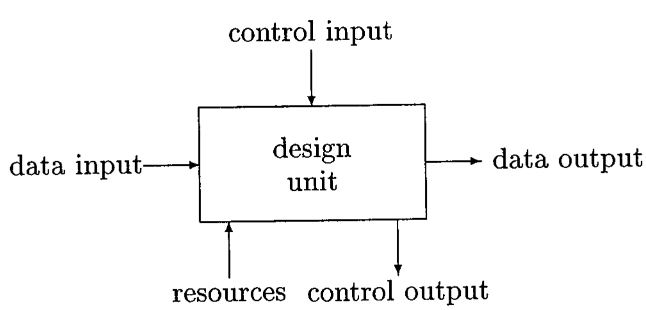


Figure 6: R-Chart Model

The O-Plan Triangle Model represents all three types of input and output more uniformly and

[3] IDEF3 [Mayer & Painter 91] is a later, more comprehensive IDEF method specifically targeted at the modelling of processes.

directly and will allow for improved support tools.

7.2 Capturing Design Rationale in Systems Development

Work in systems enginering and other fields is addressing the need to capture and make use of the rationale behind designs, decisions or regulations. An example is the Remap (for "Representation and maintenance of processes knowledge") system [Ramesh & Dhar 94] which uses the IBIS (Issue-based Information System) concepts. The issues are explicitly maintained as in the <I-N-OVA> model, and the Remap system allows for the ways in which the issues are resolved to be recorded and used.

8 Relationship to Other Work

A general approach to designing AI-based planning and scheduling systems based on partial plan or partial schedule representations is to have an architecture in which a plan or schedule is critiqued to produce a list of issues or agenda entries which is then used to drive a workflow-style processing cycle of choosing a "plan modification operator" and then executing it to modify the plan state. Figure 7 shows this graphically.

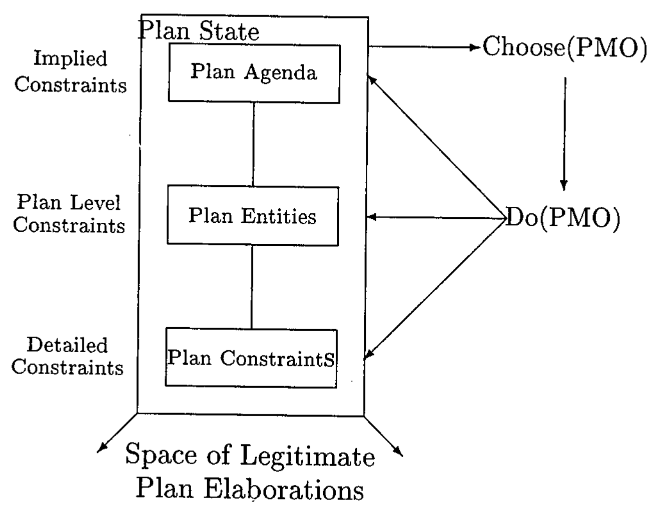


Figure 7: A Framework of Components in a Planning/Scheduling System

This approach is taken in systems like O-Plan [Currie & Tate 91],[Tate et. al. 94c], RT-1 [D'Ambrosio et. al. 87], OPIS [Smith 94], DIPART [Pollack 94], TOSCA [Beck 93], etc. The approach fits well with the concept of treating plans as a set of constraints which can be refined as planning progresses. Some such systems can act in a non-monotonic fashion by relaxing constraints in certain ways.

Having the implied constraints or "agenda" as a formal part of the plan provides an ability to separate the plan that is being generated or manipulated from the planning system itself. The benefits were first noted by McDermott [McDermott 78] and are used as a core part of the O-Plan design.

A recently described approach to Mixed Initiative Planning in O-Plan [Tate 94] proposes to improve the coordination of planning with user interaction by employing a clearer shared model of the plan as a set of constraints at various levels that can be jointly and explicitly discussed between and manipulated by user or system in a cooperative fashion.

9 Relationship to Formal Studies of Plans and Planners

The Nonlin QA Algorithm [Tate 77] establishes the modifications that are needed in terms of plan step ordering and variable binding to ensure that a given statement has a required value at a given point in a partially ordered network of nodes. This has been a basis for the formal work by Chapman [Chapman 91] on the Modal Truth Criterion. However, the MTC uses a simplification of the plans being represented in practical planners such as Nonlin [Tate 77], O-Plan [Currie & Tate 91],[Tate et. al. 94c] and SIPE-2 [Wilkins 88]. It took a non-hierarchical view and ignored specialised domain knowledge of activity condition types and constraints. Many of these were those very features that allowed planners like Nonlin and SIPE-2 to solve problems at a scale that was beyond the more theoretically based planners. Drummond [Drummond 93] explains that formal approaches have concentrated on goal achievement aspects of planners in a simplified environment that is not representative of the approaches actually taken in practical planners.

Recently however, formal representations have begun to address issues of realistic plan representations and to model hierarchical planning [Barrett & Weld 94],[Kambhampati & Hendler 91],[Penberthy & Weld 90], [Yang 90]. In particular, Kambhampati has described a formal truth criterion for plans which are represented with greater levels of realism. He describes plans as a 5 tuple <S, O, B, ST, L> [Kambhampati 94a] where:

```
S     a set of plan steps or nodes

O     a partial ordering over S

B     a set of variable binding co-designation
      and non-co-designation constraints

ST    a symbol table mapping each plan step or
      node to a domain operator

L     a set of auxiliary constraints (mainly
      intended for pre- and post-conditions)
```

This representation can be related directly to the N (incorporating the S and ST parts) and OVA (incorporating the O, B and L parts) of the <I-N-OVA> model[4].

[4]The use of the term "Auxiliary Constraints" in <I-N-OVA> was adopted as a means to relate to this formal

Hendler and Kambhampati are also studying hierarchical approaches to formal methods in planning [Kambhampati 94b],[Kambhampati & Hendler 91]. Work is underway by Kambhampati and by Young [Young et. al. 94] to understand aspects of the use of "condition types" [Tate et. al. 94b] used to provide domain semantic information to Nonlin, O-Plan and other practical planners.

The <I-N-OVA> model also has a direct relationship to the *plan recipes* described by Traum and Allen [Traum & Allen 94]. They view plans as a set of actions (*c.f.* N) and a set of constraints relating various properties of these actions (*c.f.* OVA). The issues element (I) of <I-N-OVA> is not directly modelled.

10 A Framework for Further Study

To provide a framework for further study, the following classification of models related to <I-N-OVA> is provided.

	partial plan	partial plan with issues
single level model	<N-OVA>	<I-N-OVA>
hierarchical model	Δ-<N-OVA>	Δ-<I-N-OVA>

A base model <N-OVA> is used to represent a basic plan without hierarchy or abstraction modelling and not including implied constraints (the issues agenda). The other models extend this basic model along these two dimensions[5]. They are all supersets of <N-OVA>, and are collectively termed *Super-*<N-OVA> models.

The <N-OVA> element most closely relates to the model being studied by Kambhampati today [Kambhampati 94a]. The Δ-<I-N-OVA> element is the closest to the plan representation used within O-Plan today.

11 Summary

The <I-N-OVA> Constraint Model of Plans and its relationship to the O-Plan Triangle Model of Activity has been described to assist in more closely relating new work in formal

work. In fact the <S, O, B, ST, L> constraint set acts as a refinement filter on all possible plans, whereas <I-N-OVA> also defines the candidate set from which the solutions may come (through the N component). This needs further study to relate the two approaches.

[5]Non-determinism is a property of the system (human or computer based) which manipulates the plans and is not necessarily represented in the constraint model. However, it is usual to include explicit dependency information in a plan via constraints to support non-monotonic planners. This may indicate that it would be useful to define a third dimension to this framework for further study.

descriptions of plans and planners to practical work on realistic planning systems. <I-N-OVA> is intended to act as a bridge to improve dialogue between the communities working in these two areas and potentially to support work on automatic manipulation of plans, human communication about plans, principled and reliable acquisition of plan information, and formal reasoning about plans.

Acknowledgements

The O-Plan project is sponsored by the Advanced Research Projects Agency (ARPA) and Rome Laboratory, Air Force Materiel Command, USAF, under grant number F30602-95-1-0022. The O-Plan project is monitored by Dr. Northrup Fowler III at the USAF Rome Laboratory. The U.S. Government is authorised to reproduce and distribute reprints for Governmental 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 ARPA, Rome Laboratory or the U.S. Government.

The work has also benefited from discussions concerning the use of knowledge-rich plan and process representations in commercial applications though collaboration in the Enterprise project [Fraser & Tate 95]. Enterprise is a consortium including AIAI, IBM, Unilever, Logica and Lloyds' Register and is supported by the U.K. Government's Intelligent Systems Integration Programme.

My thanks to the researchers on the O-Plan and Enterprise projects and for discussions with researchers elsewhere which have helped formulate the <I-N-OVA> model.

References

[Allen et. al. 90] Allen, J.F., Hendler, J. and Tate, A., Readings in Planning, Morgan Kaufmann, Palo Alto, CA., 1990.

[Allen & Koomen 93] Allen, J.F. and Koomen, J.A., Planning Using a Temporal World Model, Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-83), Karlsruhe, Germany, 1993.

[D'Ambrosio et. al. 87] D'Ambrosio, B., Raulefs, P., Fehling, M.R., and Forrest, S., Real-time Process Management for Materials Composition in Chemical Manufacturing, Technical Report, Technowledge Inc, 1850 Embarcadero Road, Palo Alto, CA 94303 and FMC Corporation, AI Center, Central Engineering Laboratories, Box 580, 1205 Coleman Avenue, Santa Clara, CA 95052, USA, 1987.

[Barrett & Weld 94] Barrett, A. and Weld, D.S., Task-Decomposition via Plan Parsing, Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94), Seattle, USA, 1994.

[Beck 93] Heck, H., TOSCA: A Novel Approach to the Management of Job-shop Scheduling Constraints, Realising CIM's Industrial Potential: Proceedings of the Ninth CIM-Europe

Annual Conference, pages 138-149, (eds. Kooij, C., MacConaill, P.A., and Bastos, J.), 1993. Also available as AIAI Technical Report AIAI-TR-121.

[Borgida et. al. 85] Borgida, A., Greenspan, S. and Mylopoulos, J., Knowledge Representation as the Basis for Requirements Specifications, IEEE Computer Magazine, Special Issue on Requirements Engineering Environments, April 1985.

[Chapman 91] Chapman, D., Planning for Conjunctive Goals. *Artificial Intelligence*, 32:333-377, 1991.

[Currie & Tate 91] Currie, K.W. and Tate, A., O-Plan: the Open Planning Architecture, *Artificial Intelligence* 52(1), Autumn 1991, North-Holland.

[Curwen 90] Curwen, P., System Development Using the CORE Method, British Aerospace Technical Report BAe/WIT/ML/GEN/SWE/1227, 1990.

[Drabble 93] Drabble, B., Excalibur: A Program for Planning and Reasoning with Processes, *Artificial Intelligence*, Vol. 62 No. 1, pp. 1-40, 1993.

[Drummond 93] Drummond, M.E., On Precondition Achievements and the Computational Economics of Automatic Planning, in Current Trends in AI Planning (eds. C.Backstrom and E.Sandewall) IOS Press, Sweden, 1993.

[Fraser & Tate 95] The Enterprise Toolset – An Open Enterprise Architecture, Proceedings of the Workshop on Intelligent Manufacturing Systems, pp. 95-103, (ed. Sadeh, N.M.), Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95), Montreal, Canada, August 1995.

[HOOD 89] HOOD Working Group, HOOD Reference Manual, Issue 3.0 European Space Agency, Noordwijk, Netherlands, 1989.

[ISO/IEC 89] ISO/IEC 8631-1989 Information Technology - Program Constructs and Conventions for their Representation, second edition, ISO/IEC, 1989.

[Kambhampati 94a] Kambhampati, S., Design Tradeoffs in Partial Order Planning, Proceedings of the Second International Conference on AI Planning Systems (AIPS-94), Chicago, IL., USA, 1994.

[Kambhampati 94b] Kambhampati, S., Comparing Partial Order Planning and Task Reduction Planning: a Preliminary Report, Proceedings of the Workshop on Comparative Analysis of AI Planning Systems, AAAI-94, Seattle, USA, 1994.

[Kambhampati & Hendler 91] Kambhampati, S. and Hendler, J., A Validation-Structure-Based Theory of Plan Modification and Reuse, *Artificial Intelligence*, May, 1992.

[Mayer 92] Mayer, R.J. (editor), IDEF0 Functional Modeling: A Reconstruction of the Original Air Force Wright Aeronautical Laboratory Technical Report AFWAL-TR-81-4023 (The IDEF0 Yellow Book), Knowledge Based Systems Inc., College Station, TX, 1992.

[Mayer & Painter 91] Mayer, R.J. and Painter, M., IDEF Family of Methods, Technical Report, Knowledge Based Systems Inc., College Station, TX, 1991.

[McDermott 78] McDermott, D.V. A Temporal Logic for Reasoning about Processes and Plans In *Cognitive Science, 6, pp. 101-155, 1978.*

[Penberthy & Weld 90] Penberthy, J.S. and Weld, D.S., UCPOP: A Sound, Complete, Partial Order Planner for ADL, Proceedings of the Third International Conference on Knowledge Representation and Reasoning, 1990.

[PIF 94] Process Interchange Format Working Group, The PIF Process Interchange and Framework, MIT Center for Coordination Science Working Paper No. 180, MIT, Boston, December 1994.

[Pollack 94] Pollack, M., DIPART Architecture, Technical Report, Department of Computer Science, University of Pittsburgh, PA 15213, USA, 1994.

[Ramesh & Dhar 94] Ramesh, B. and Dhar, V., Representing and Maintaining Process Knowledge for Large-Scale Systems Development, IEEE Expert, pp. 54-59, April 1994.

[Rich & Waters 88] Rich, C. and Waters, R.C., The Programmer's Apprentice: A Research Overview, *Computer*, Vol. 21, No. 11, pp. 11-25, November 1988.

[Ross 85] Ross, D.T., Applications and Extensions of SADT, IEEE Computer Magazine, Special Issue on Requirements Engineering Environments, April 1985.

[Smith 94] Smith, S., OPIS: A Methodology and Architecture for Reactive Scheduling, in Intelligent Scheduling, (eds, Zweben, M. and Fox, M.S.), Morgan Kaufmann, Palo Alto, CA., USA, 1994.

[Tate 77] Tate, A., Generating Project Networks, Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-77), Cambridge, Mass., USA, 1977.

[Tate 89] Tate, A., Coordinating the Activities of a Planner and an Execution Agent, Proceedings of the Second NASA Conference on Space Telerobotics, (eds. G.Rodriguez and H.Seraji), JPL Publication 89-7 Vol. 1 pp. 385-393, Jet Propulsion Laboratory, February 1989.

[Tate 93a] Tate, A., 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 93b] Tate, A., The Emergence of "Standard" Planning and Scheduling System Components, in Current Trends in AI Planning (eds. C.Backstrom and E.Sandewall) IOS Press, Sweden, 1993.

[Tate 94] Tate, A., Mixed Initiative Planning in O-Plan2, Proceedings of the ARPA/Rome Laboratory Planning Initiative Workshop, (ed. Burstein, M.), Tucson, Arizona, USA, Morgan Kaufmann, Palo Alto, 1994.

[Tate et. al. 94a] Tate, A., Drabble, B. and Dalton, J., O-Plan Version 2.2 Task Formalism Manual, O-Plan Project Documentation, AIAI, University of Edinburgh, 1994.

[Tate et. al. 94b] Tate, A., Drabble, B. and Dalton, J., The Use of Condition Types to Restrict Search in an AI Planner, Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94), Seattle, USA, 1994.

[Tate et. al. 94c] Tate, A., Drabble, B. and Kirby, R., O-Plan2: an Open Architecture for Command, Planning and Control, in Intelligent Scheduling, (eds, Zweben, M. and Fox, M.S.), Morgan Kaufmann, Palo Alto, 1994.

[Tate et. al. 94d] Tate, A., Drabble, B. and Dalton, J. Reasoning with Constraints within O-Plan2, Proceedings of the ARPA/Rome Laboratory Planning Initiative Workshop, (ed. Burstein, M.), Tucson, Arizona, USA, Morgan Kaufmann, Palo Alto, 1994.

[Traum & Allen 94] Traum, D.R. and Allen, J.F., Towards a Formal Theory of Repair in Plan Execution and Plan Recognition, Proceedings of the Thirteenth UK Planning and Scheduling Special Interest group, Glasgow, UK, September 1994.

[Ushakiv & Velbitskiy 93] Ushakov, I. and Velbitskiy, I., Visual Programming in R-technology: Concepts, Systems and Perspectives, Proceedings of the Third East-West International Conference on Human Computer Interaction, Moscow, Russia, 1993.

[Uschold et. al. 95] Uschold, M., King, M., Moralee, S. and Zorgios, Y., The Enterprise Ontology, Enterprise Project Report, Artificial Intelligence Applications Institute, University of Edinburgh, Edinburgh, UK, July 1995.

[WfMC 94] Workflow Management Coalition, Glossary – A Workflow Management Coalition Specification, Workflow Management Coalition, Avenue Marcel Thiry 204, 120 Brussels, Belgium, November 1994

[Wilkins 88] Wilkins, D., *Practical Planning*, Morgan Kaufmann, Palo Alto, 1988.

[Yang 90] Yang, Q. Formalizing Planning Knowledge for Hierarchical Planning, *Computational Intelligence*, Vol. 6, No. 1, pp. 12-24, 1990.

[Young et. al. 94] Young, R.M., Pollack, M.E. and Moore, J.D., Decomposition and Causality in Partial-Order Planning, Proceedings of the Second International Conference on AI Planning Systems (AIPS-94), Chicago, IL, USA, 1994.

Appendix F:

Roots of SPAR - Shared Planning and Activity Representation

Austin Tate

Citation:
Tate, A., Roots of SPAR - Shared Planning and Activity Representation, The Knowledge Engineering Review Vol 13(1), 121–128, Cambridge University Press, 1998.

Purpose:
Provides a historical survey and extensive bibliographic source for work on activity, process and plan representations, and shows how they have been used to design a shared planning and activity representation for use in US military programs.

Abstract:
The Defense Advanced Research Projects Agency (DARPA) and US Air Force Research Laboratory Planning Initiative (ARPI) has initiated a project to draw on the range of previous work in planning and activity ontologies to create a practically useful "Shared Planning and Activity Representation" - SPAR - for use in technology and applications projects within their communities.

This article describes the previous work which has been used to create the initial SPAR representation. Key examples of the work drawn upon are published in the Knowledge Engineering Review Special Issue on Ontologies [Uschold & Tate, 1998]. The paper provides a comprehensive bibliography and related world wide web resources for work in the area of plan, process and activity representation.

SPAR is now being subjected to refinement during several review cycles by a number of expert and user panels.

1 Aims for SPAR

It is important that information about processes, plans and activities are sharable within and across organisations. Cooperation and coordination of the planning, monitoring and workflows of the organisations can be assisted by having a clear shared model of what comprises plans, processes and activities. The Shared Planning and Activity Representation (SPAR) is intended to contribute to a range of purposes including domain modelling, plan generation, plan analysis, plan case capture, plan communication, behaviour modelling, etc. By having a shared model of what constitutes a plan, process or activity, organisational knowledge can be harnessed and used effectively.

The design of SPAR provides structure where there is a consensus on the key entities and relationships amongst those creating and using planning and activity representations. It specifies the structure to a level of detail judged to relate to the needs of the majority of potential users of the representation. However, planning and activity representations are the subject of active research. Some of the current approaches are conceptually simpler or more uniform than SPAR is intended to be - e.g., using pure logic or all constraint [Joslin, 1996; Tate, 1996a] bases. Even where the structure of SPAR itself is not suitable as a basis for novel research or applications, the intention is that the semantics of the SPAR Representation should be clearly defined such that it can be translated to alternative representations. This also provides the important capability that SPAR-represented information will be able to be communicated to future representational frameworks as and when those are adopted in a widespread way.

2 Scope

The principal scope of SPAR is to represent past, present and possible future activity and the command, planning and control processes that create and execute plans meant to guide or constrain future activity. It can be used descriptively for past and present activity and prescriptively for possible future activity.

Within the SPAR structure it is possible to specialise the representation through the provision of application, domain or technology specific extensions via Plug-in Ontologies or Grammars with their associated Lexicons of the terms used. This is the level at which current and novel representations of activity and the constraints on activity will be attached. The plug in ontologies or grammars may draw on standard representations being adopted in the AI planning research community such as PDDL [McDermott et. al., 1997] or more constrained grammars may be specified for practical use in today's applications.

Where further shared structure can be agreed in future within a sufficiently broad community, it could be included in a future revision of SPAR. Where more limited communities representing vendors, specific sector users or research interest groups agree on a shared representation, it may be possible to create an extension used within that community using a mechanism such as the PIF "Partially Shared Views" (PSVs) [Lee & Malone, 1990].

Any practical use of a planning and activity representation naturally will relate to more detailed models of the objects involved or the organisational relationships between the people or agents included. It is not intended that SPAR itself prescribes structure for detailed object models or for detailed organisation or agent relationship models. SPAR can co-exist with one or more such models which can therefore be chosen to reflect standards established elsewhere. An example of a detailed object description standard is that established by International Standards Organisation's STEP [ISO, 1995] or EXPRESS [Express, 1995] for product interchange in manufacturing. Examples of organisational and agent relationships models can be found in the Enterprise Models [Fraser & Tate, 1995; Fox et. al., 1993] and the ORDIT Organisational Model [Blyth et. al., 1993].

SPAR may be expressed or represented in a wide variety of ways. It is intended that reference designs and implementations for a number of those which will be mostly commonly useful will be provided. KIF [Genesereth & Fikes, 1992], CommonKADS Conceptual Modelling Language [Schreiber et. al., 1994], Conceptual Graphs [Sowa, 1984], LOOM [Brill, 1993], CDIF [Ernst, 1997] or other representations of SPAR are possible.

3 History

The AI planning community has used explicit domain description languages and plan definitions for more than 25 years [Tate et. al., 1990; Allen et. al., 1990]. As long ago as the late 1960s, work on the STRIPS operator representation of actions [Fikes & Nilsson, 1971] was used to practical effect for planning and control of the SRI Shakey robot. This early application was based upon more theoretical roots in the QA3 theorem prover [Green, 1969] and situation calculus [McCarthy & Hayes, 1969]. There is now a wealth of experience of defining plan representations from both theoretical studies and practical planning. In 1992, under the DARPA/Air Force Research Laboratory (Rome) Planning Initiative (ARPI) [Fowler et. al., 1995], a number of participants created the KRSL plan language [Lehrer, 1993]. Although this has been used for some transfers of information between planning components within the ARPI (in particular an Integrated Feasibility Demonstration called IFD-2 [Burstein et. al., 1995]), it has not had the widespread impact desired. Its structure was too rigid and KRSL excluded much that was already being done within AI planning research. However, it did establish a range of entities which needed to be in a plan representation and was an influence on subsequent work.

In 1994, a group was formed to create an ontology for plans, using new insights gained over the last few years in the knowledge-sharing community in the US [Neches et. al., 1991; Genesereth & Fikes, 1992; Gruber, 1993] and Europe [Uschold, 1998; Breuker & van de Velde, 1994]. This led to an outline plan model called KRSL-Plans [Tate, 1996b]. However, this work was not brought to a conclusion though it did feed into subsequent work.

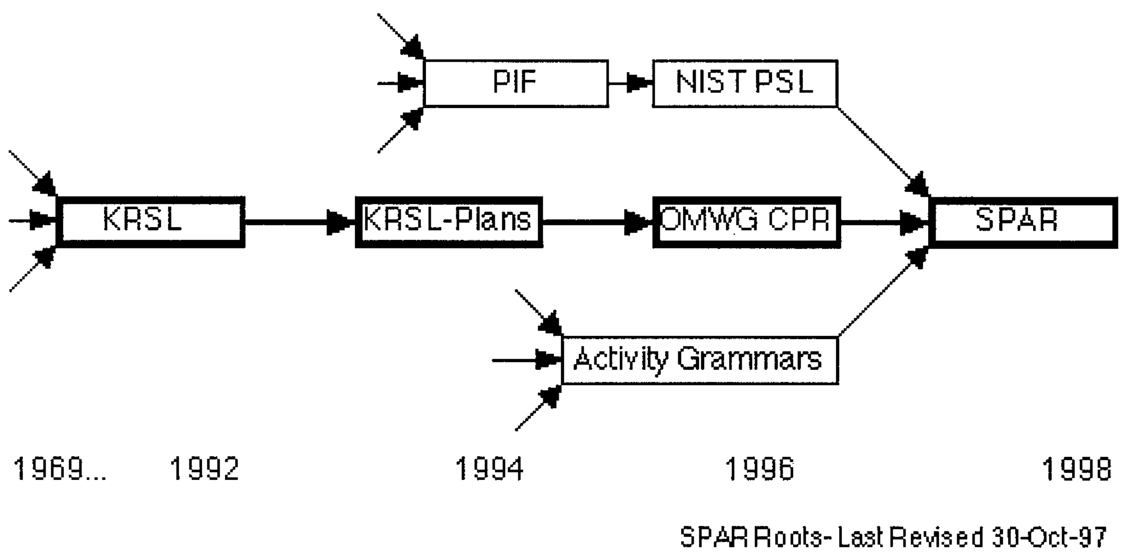
Since 1995, there have been a number of initiatives to standardise terminology in the subject area of activities and plans. These include enterprise processes in PIF (the Process Interchange Format [Lee et. al., 1996]); workflow (International Workflow Management Coalition [WfMC, 1994]); CASE systems data modelling exchange in CDIF [Ernst, 1997; Navarro, 1996]; manufacturing processes (NIST's Process Specification Language [Schlenoff et.

al., 1996]); and the Object Model Working Group's CPR (Core Plan Representation - [Pease & Carrico, 1997]). These initiatives have involved academic, government and industry participants.

In the US military planning research community and beyond, there has been work to use verb/noun phrase grammars to represent plan objectives and activities [Valente et al., 1996; Hess, 1996; Kingston et. al., 1997; Drabble et. al., 1997].

In August 1997, DARPA and the Air Force Research Laboratory (Rome) Programme Managers for ARPI proposed a renewed effort to tap into this accumulated expertise, and to create a shared plan representation suitable for practical use in ARPI and on applied research programmes in their communities. The representation is expected to be considerably more detailed than the more conceptual ontology efforts which have gone before it.

SPAR is drawing on this wide range of previous work.



Plan ontologies and representations created by participants of ARPI-related projects include:

1. ARPI KRSL 2.0.2 [Lehrer, 1993] as noted above.
2. ARPI KRSL-Plans ontology [Tate, 1996b] as noted above.
3. SRI International ACT language from the Cypress project [Wilkins & Myers, 1995] and SIPE's domain description language [Wilkins, 1988].
4. Edinburgh O-Plan Task Formalism [Tate et. al., 1994; Tate, 1995] and the related <I-N-OVA> constraint model of activity [Tate, 1994; Tate, 1996a].
5. CMU OZONE scheduling ontology [Smith & Becker, 1997].
6. CMU Prodigy [Carbonell et. al., 1992] work on decisions made in planning.

7. ISTI IDEON Object-Oriented Enterprise Ontology [Madni & Mi, 1997].

8. The Planning Domain Definition Language (PDDL) [McDermott et. al., 1997] created by a community of researchers wanting to exchange planning challenge problems. PDDL draws on work on expressive planner operator languages in ADL [Pednault, 1989] and hierarchical task network planning [Erol et. al., 1994].

9. Process Interchange Format (PIF) standard being worked on by a number of projects interested in exchanging process information [Lee et. al., 1996].

10. USC/ISI work on EXPECT regarding the representation of goals and tasks [Swartout and Gil, 1995], its recent application to structured representations of air campaign objectives [Valente et al., 1996] as well as USC/ISI work on the SENSUS ontology and lexicon [Knight & Luk, 1994].

11. Edinburgh and ISX Corporation work on process models and grammars for describing the actions and products flowing for US Air Campaign Planning [Kingston et. al., 1997].

SPAR has drawn on the ontologies created on collaborative projects related to Enterprise Modelling and Integration including:

1. Enterprise Ontology (Edinburgh, Lloyds Register, Logica, IBM UK and Unilever) [Fraser & Tate, 1995; Uschold et. al., 1998] and the report of the Enterprise Project workshop on "Content, Form and Methods for Ontologies" in May 1994 [Moralee, 1994]. that were provided to the KRSL-Plans, PIF and other communities.

2. TOronto Virtual Enterprise (TOVE) ontology [Fox et. al., 1993].

In addition, there is relevant work on Structured Analysis and Design Techniques (e.g., SADT [Marca & McGowan, 1988]), Issue-Based Design Methods (e.g., IBIS [Conklin & Burgess-Yakamovic, 1995]), Process Management Models and Methods (e.g., IDEF [NIST, 1993]), Entity-Relationship Modelling [Chen, 1976], Object-Role Modelling (e.g., Nijssen's Information Analysis Method (NIAM) [Nijssen, 1989]), Process Workflow Support, etc.

Since 1994, work within ARPI on plan representations has proceeded in parallel with pre-standards work on process interchange and representation. ARPI researchers have been involved in these activities, and others have supported the continuing ARPI work - including helping in the development and review of SPAR. The relevant standards activities that have been jointly pursued are:

1. Object Model Working Group (OMWG) Core Plan Representation (CPR) [Pease & Carrico, 1997].

2. National Institute of Standards and Technology (NIST) Process Specification Language (PSL) [Schlenoff et. al., 1996].

3. Workflow Management Coalition work in standardising workflow systems and process terminology via their Glossary of Workflow terms [WfMC, 1994] and their interface 1 - the Workflow Process Definition Language (WPDL).

4. CASE Data Interchange Format (CDIF [Ernst, 1997]) group of the Electrical Industries Association who are standardising data model exchanges between CASE systems in a number of areas including the Project Management, Planning and Scheduling Subject Area [Navarro, 1996].

A common model of processes and activity has emerged from this work and has been used as the basis for the initial version of SPAR.

4 Development, Refinement and Review Process

SPAR is being developed in the following way:

1. The SPAR Core Group has merged existing plan ontology work into a solid core representation as a starting point.

2. Via three panels, the Core Group is seeking input to the SPAR work, reviews of the representation proposed, and issues raised by it. The panels are: o User Requirements Panel. o Specialism Experts Panel. o Formalization Review Panel.

3. The Core Group will take the lessons learned in 2 to refine 1, and repeat the process.

4. The Core Group and others will communicate the work in progress and then promote the availability of the representation to the potential user, technical and standards communities.

5. The research community will collect experience in the use of SPAR and use lessons learned to refine the representation.

6. The research community will publish a report on the representation and experience gained.

5 SPAR Model

A set of statements called the KRSL-Plans description (version dated 20-Sep-96) was used as a starting point for the SPAR model of planning and activity. These were created by the Plan Ontology Construction Group within ARPI. These statements were a refinement of an earlier version dated 2-Feb-95 (published in [Tate, 1996b]). The later version of these statements was also used to provide ARPI participants input to the development of OMWG's Core Plan Representation (CPR). [square brackets are used to indicate phases or options that were not fully agreed]. The top level statements are:

1. A PLAN is a SPECIFICATION of ACTIVITY to meet one or more OBJECTIVES.
2. A SPECIFICATION of ACTIVITY denotes or describes one or more ACTIVITIES.
3. An ACTIVITY may change the STATES-OF-AFFAIRS.
4. STATES-OF-AFFAIRS is something that can be evaluated as holding or not. [They can refer to an individual world state (such as NOW), or may refer to world histories, changes between world states, etc.]
5. An AGENT can perform ACTIVITIES and/or hold OBJECTIVES.
6. An OBJECTIVE may have one or more EVALUATION-CRITERIA.

Then at a second level of detail, the statements are:

7. An EVALUATION-CRITERIA is an ASPECT of [past, present or possible future] STATES-OF-AFFAIRS or an ASPECT of [one or more] PLANS.
8. An EVALUATION is a predicate (holds/does not hold) or a preference ranking on [one or more] EVALUATION-CRITERIA.
9. An ACTIVITY takes place over a TIME-INTERVAL identified by its two ends, the BEGIN-TIME-POINT and the END-TIME-POINT. The BEGIN-TIME-POINT is temporally before the END-TIME-POINT.
10. An ACTIVITY-SPECIFICATION may have CONSTRAINTS associated with it [and its TIME-INTERVAL].
11. An ACTIVITY-SPECIFICATION may be decomposed into one or more ACTIVITY-DECOMPOSITIONS.
12. ACTIVITY-DECOMPOSITION: The specification of how an ACTIVITY is decomposed into one or more SUB-ACTIVITIES; this may include the specification of constraints on and between the SUB-ACTIVITIES [and their TIME-INTERVALs].
13. SUB-ACTIVITY: Sub-activities are the constituent activities designated in any ACTIVITY-DECOMPOSITION.
14. PRIMITIVE-ACTIVITY is an ACTIVITY with no (further) ACTIVITY-DECOMPOSITION.
15. CONSTRAINTS can be stated with respect to none, one or more than one time point. They express things which are required to hold. They are evaluable with respect to a specific PLAN as holding or not holding. Such constraints may refer to world statements (conditions and effects), resource requirements and usage, authority requirements or provision, etc.

The model is expected to develop from this base.

6 Status

A project has been started within the ARPI community to develop a Shared Planning and Activity Representation - SPAR - for practical use in technology and applications projects within US government research and development communities and beyond.

An initial version of SPAR has been produced utilising the wide range of previous work on plan, process and activity ontologies and representations. This is being subjected to review and refinement through a number of Request For Comment documents involving technical specialists and application-oriented user panels. Following these reviews, the intention is to publish a first version in mid 1998 and then to collect experience in the application of the representation.

It is intended that a process for sharing experiences of using SPAR will be established and continuing design issues tracked. A collected volume of papers describing SPAR and relating experience in using, adapting or extending the representation is planned in the medium term.

7 Acknowledgements

This work is sponsored by the Defense Advanced Research Projects Agency (DARPA) and the Air Force Research Laboratory (Rome), Air Force Materiel Command, USAF, under grant numbers F30602-95-1-0022 and F30602-97-C-0275. The US Government is authorised to reproduce and distribute reprints for Governmental 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, AFRL, the US Government, members of the SPAR Core Group or other SPAR participants.

Thanks to the many people who have provided input to the planning, process and activity ontologies which have provided the roots for the work on SPAR and enabled rapid progress to be made in its initial development. Special thanks to Steve Polyak for helping to complete the extensive set of references in this paper. Thanks to Paul Krause, Simon Parsons, Steve Polyak and Mike Uschold for reviewing the text.

8 References

Other World Wide Web accessible references and resources can be accessed through the SPAR Web Area at http://www.aiai.ed.ac.uk/~arpi/spar

Allen, J.F., 1984. "Towards a General Theory of Action and Time" Artificial Intelligence 23 123-154.

Allen, J.F., Hendler, J. and Tate, A. (eds.), 1990. Readings in Planning Morgan Kaufmann, Palo Alto, CA.

Blyth A.J.C., Chudge, J., Dobson, J.E. and Strens, M.R., 1993. "ORDIT: A new methodology to assist in the process of eliciting and modelling organisational requirements" In Proceedings on the Conference on Organisational Computing Systems, November, San Jose, CA, USA. http://www.twente.research.ec.org/esp-syn/text/2301.html

Breuker, J., van de Velde, W., 1994. The CommonKADS Library for Expertise Modelling: Reusable Components for Artificial Problem Solving, IOS Press. http://www.swi.psy.uva.nl/projects/CommonKADS/home.html.

Brill, D., 1993. "LOOM Reference Manual v.2.0", Information Sciences Institute, University of Southern California. http://www.isi.edu/isd/LOOM/LOOM-HOME.html

Burstein, M.H., Schantz, R., Bienkowski, M.A., desJardins, M.E. and Smith, S.F., 1995. "The Common Prototyping Environment – A Framework for Software Technology Integration, Evaluation and Transition" IEEE Expert 10(1) February 17-26, IEEE Comp. Soc. http://arpi.isx.com

Carbonell, J.G., Blythe, J., Etzioni, O., Gil, Y., Joseph, R., Kahn, D., Knoblock, C., Minton, S., Perez, A., Reilly, S., Veloso, M., and Wang, X., 1992. "PRODIGY4.0: The Manual and Tutorial", Technical Report CMU-CS-92-150, Department of Computer Science, Carnegie Mellon University.

Chen, P.P., 1976. "The Entity-Relationship Model - Toward a Unified View of Data" ACM Transactions on Database Systems 1(1) March 9-36.

Conklin, J. and Burgess-Yakamovic, K., 1995. "A Process-Oriented Approach to Design Rationale" In T. Moran and J. Carroll (eds.) Design Rationale Concepts, Techniques, and Use, Lawrence Erlbaum Associates, Mahwah, N.J., pp. 393-428.

Drabble, B., Lydiard, T. and Tate, A., 1997. "Process Steps, Process Products and System Capabilities", Artificial Intelligence Applications Institute, University of Edinburgh, Technical Report ISAT-AIAI/TR/4 Version 2, April 14, 1997. http://www.aiai.ed.ac.uk/~arpi/ACP-MODELS/ACP-PROCESS/97-APR/TEX-PS/isat-report4.ps

Ernst, J., 1997. Introduction to CDIF, CASE Data Interchange Format Division Electronic Industries Association. http://www.cdif.org

Erol, K., Hendler, J. and Nau, D., 1994. "UMCP: A Sound and Complete Procedure for Hierarchical Task Network Planning" In Proceedings of the Second International Conference on AI Planning Systems (AIPS-94), Chicago, IL, USA, pp. 249-254.

EXPRESS Information Modelling Language, 1995. ISO WD 10303-11. http://www.eurpc2.demon.co.uk/part11.html

Fikes, R.E. and Nilsson, N., 1971. "STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving" Artificial Intelligence 5(2).

Fowler, N., Cross, S.E. and Owens, C., 1995. "The ARPA-Rome Knowledge-Based Planning and Scheduling Initiative" IEEE Expert, 10(1) February 4-9, IEEE Comp. Soc. http://arpi.isx.com

Fraser, J. and Tate, A., 1995. "The Enterprise Tool Set – An Open Enterprise Architecture" In Proceedings of the Workshop on Intelligent Manufacturing Systems, International Joint

Conference on Artificial Intelligence (IJCAI-95), Montreal, Canada. http://www.aiai.ed.ac.uk/~entprise/enterprise/ontology.html

Fox, M.S., Chionglo, J.F. and Fadel, F.G., 1993. "A Common-Sense Model of the Enterprise" In Proceedings of the Second Industrial Engineers Research Conference (IERC) Norcross, GA, Institute for Industrial Engineers. http://www.ie.utoronto.ca/EIL/tove/toveont.html

Genesereth, M.R. and Fikes, R.E., 1992. "Knowledge Interchange Format, Version 3.0 Reference Manual", Knowledge Systems Laboratory, KSL-92-86. http://ksl-web.stanford.edu/KSL_Abstracts/KSL-92-86.html.

Green, C., 1969. "Applications of Theorem Proving to Problem Solving" In Proceedings of the First International Joint Conference on Artificial Intelligence (IJCAI-69), Morgan Kaufmann, pp. 741-747.

Gruber, T., 1993. "Ontolingua: A Translation Approach to Providing Portable Ontology Specifications" Knowledge Acquisition 5(2) 199-200. http://ksl-web.stanford.edu/KSL_Abstracts/KSL-92-71.html.

Hess, D. (editor), 1996, Theater Battle Management Command, Control, Communications, Computer and Intelligence Architecture (TBM C4I) Air Operations Center (AOC), Report prepared by Government Personnel and Mitre Corporation under Contract for Project 6970.

ISO, 1995. "Product Data Representation and Exchange: Part 49: Integrated Generic Resources: Process Structure and Properties", ISO Standard 10303-49, International Standards Organization. http://www.nist.gov/sc4/www/stepdocs.htm

Joslin, D., 1996. "Planner/Scheduler Interface Proposal", Draft, CIRL, University of Oregon, 17-May-96. http://www.cirl.uoregon.edu/joslin/Papers/psi.ps.

Khambhampati, S, 1997. "Refinement Planning" AI Magazine, 18(2), pp. 67-97, Summer 1997.

Kingston, J.K., Griffiths, A. and Lydiard, T.J., 1997. "Multi-Perspective Modelling of the Air Campaign Planning Process" In Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97), Nagoya, Japan. http://www.aiai.ed.ac.uk/~arpi

Knight, K. and S. Luk., 1994. "Building a Large Knowledge Base for Machine Translation". In Proceedings of the American Association of Artificial Intelligence Conference (AAAI-94), July 31, August 3, 1994, Seattle, WA. http://www.isi.edu/natural-language/resources/sensus.html#pubs

Lee, J. and Malone, T., 1990. "Partially Shared Views: A Scheme for Communicating between Groups Using Different Type Hierarchies" ACM Transactions on Information Systems 8(1) 1-26.

Lee, J. (ed.), Gruninger, M., Jin, Y., Malone, T., Tate, A., Yost. G., and other members of the PIF Working Group, 1996. "Process Interchange Format and Framework, Version 1.1", MIT Center for Coordination Science, Working Paper No. 194. http://ccs.mit.edu/pif

Lehrer, N. (ed.), 1993. "ARPI KRSL Reference Manual 2.0.2", ISX Corporation. http://isx.com/pub/ARPI/ARPI-pub/krsl/krsl-info.html

McCarthy, J. and Hayes, P.J., 1969. "Some Philosophical Problems from the Standpoint of Artificial Intelligence" In B. Meltzer and D. Michie (eds.) Machine Intelligence 4, Edinburgh University Press, 463-502.

McDermott, D., 1997. "PDDL - The Plan Domain Definition Language", Computer Science, Yale University. http://www.cs.yale.edu/HTML/YALE/CS/HyPlans/mcdermott.html

Moralee, S., 1994. "Notes from Enterprise Project Workshop on Content, Form and Methods for Ontologies", IBM(UK) Offices, Nottingham, UK, Dated 25-Nov-94. http://www.aiai.ed.ac.uk/~bat/ontology-may94.html.

Navarro, T., 1996. "CDIF - Integrated Meta-model - Project Management Planning and Scheduling Subject Area", Report EIA-PN3239, CDIF-DRAFT-PMPS-V04, CASE Data Interchange Format Division, Electronic Industries Association. http://www.cdif.org/overview/ProjectManagement.html

Neches, R., Fikes, R., Finin, T., Gruber, T.R., Patil, R., Senator, T. and Swartout, W.R., 1991. "Enabling Technology for Knowledge Sharing" AI Magazine 12(3) 36-56. http://ksl-web.stanford.edu/KSL_Abstracts/KSL-93-23.html.

Nijssen, G.M. and Halpin, T.A., 1989. Conceptual Schema and Relational Database Design: A Fact-Based Approach, Prentice Hall.

NIST, 1993. "Integrated Definition for Function Modeling (IDEF0)", Federal Information Processing Standards (FIPS) Publication 183, National Institute of Standards and Technology, December 21, 1993. http://www.idef.com

Pease, R.A. and Carrico, T.M., 1997. "Object Model Working Group (OMWG) Core Plan Representation - Request for Comment", version 2, 24 January 1997, Defense Advanced Research Projects Agency. http://www.teknowledge.com/CPR2/

Pednault, E, 1989. "ADL: Exploring the Middle Ground between STRIPS and the Situation Calculus" In Proceedings of the First International Conference on the Principles of Knowledge Representation and Reasoning, pp. 324-332.

Polyak, S. and Tate, A., 1998. "Rationale in Planning: Causality, Dependencies and Decisions" Knowledge Engineering Review, To appear, Cambridge University Press.

Schlenoff, C. (ed.), Knutilla, A., and Ray, S., 1996. "Unified Process Specification Language: Functional Requirements for Modeling Processes", National Institute of Standards and Technology, Gaithersburg, Maryland. http://www.nist.gov/psl/

Madni, A. and Mi, P., 1997. "IDEON Specification in CORBA IDL", Technical Memo ISTI-TM-7/97, July 17, 1997, Intelligent Systems Technology, Inc., Santa Monica, CA. http://www.intelsystech.com

Marca, D.A. and McGowan C.L., 1988. SADT: Structured Analysis and Design Techniques, McGraw-Hill, New York.

Schreiber, G., Weilinga, B., Akkermans, H. and Van de Velde, W., 1994. "CML: The CommonKADS Conceptual Modelling Language" In L. Steels, G. Schreiber and W. Van de Velde (eds.) A future for knowledge acquisition: Proceedings of EKAW-94, Hoegaarden, Belgium, Springer-Verlag. http://www.swi.psy.uva.nl/projects/Kactus/toolkit/cml.html

Smith, S.F. and Becker, M., 1997. "An Ontology for Constructing Scheduling Systems" In Working Notes from 1997 AAAI Spring Symposium on Ontological Engineering, Stanford, CA, AAAI Press.
http://www.cs.cmu.edu/afs/cs/project/ozone/www/AAAI_Symp_On_Ontol_97/abstract.html

Sowa, J.F., 1984. Conceptual Structures: Information Processing In Mind and Machine, Reading, MA., Addison-Wesley.

Swartout, B. and Gil, Y., 1995. "EXPECT: Explicit Representations for Flexible Acquisition" In Proceedings of the Ninth Knowledge Acquisition for Knowledge-Based Systems Workshop, February 26-March 3, 1995. http://www.isi.edu/expect/expect-homepage.html

Tate, A., 1994. "A Plan Ontology - a Working Document - October 31, 1994" In Proceedings of the Workshop on Ontology Development and Use, 2nd - 4th November, La Jolle, CA. ftp://ftp.aiai.ed.ac.uk/pub/documents/1994/94-ont-plan-ontology.ps

Tate, A., 1995. "O-Plan Task Formalism Manual", Version 2.3, July 12, 1995. Artificial Intelligence Applications Institute, University of Edinburgh.
ftp://ftp.aiai.ed.ac.uk/pub/documents/ANY/oplan-tf-manual.ps

Tate, A., 1996a. "Representing Plans as a Set of Constraints – the <I-N-OVA> Model" In Proceedings of the Third International Conference on Planning Systems (AIPS-96), Edinburgh, Scotland, May 1996, AAAI Press, pp. 221-228.
http://www.aiai.ed.ac.uk/~oplan/inova.html

Tate, A. (ed.), 1996b. "KRSL-Plans", Appendix of Tate, A, "Towards a Plan Ontology" AI*IA Notizie (Quarterly Publication of the Associazione Italiana per l'Intelligenza Artificiale), Special Issue on "Aspects of Planning Research" 9(1), pp. 19-26. A shorter version of [Tate, 1994]. ftp://ftp.aiai.ed.ac.uk/pub/documents/1996/96-aiia-plan-ontology.ps

Tate, A (reporter) and the PIF Working Group, 1996. "PIF and the Workflow Management Coalition WPDL, Report of Session at the PIF Workshop", Stanford University, 11-Jul-96. http://ccs.mit.edu/pif

Tate, A., Drabble, B. and Kirby, R.B., 1994. "O-Plan2: an Open Architecture for Command, Planning and Control" In M. Fox and M. Zweben (eds.) Intelligent Scheduling, Morgan Kaufmann. http://www.aiai.ed.ac.uk/~oplan/

Tate, A., Hendler, J. and Drummond, M., 1990. A Review of AI Planning Techniques, In J. Allen, J. Hendler and A. Tate (eds.) Readings in Planning Morgan Kaufmann, pp. 26-49.

Uschold, M.F. and Tate, A., 1998. "Putting Ontologies to Use" Knowledge Engineering Review, Special Issue on Ontologies, To appear, Cambridge University Press.

Uschold, M.F., 1998. "Concepts and Terminology for Knowledge Sharing" Knowledge Engineering Review, Special Issue on Ontologies, To appear, Cambridge University Press. http://www.aiai.ed.ac.uk/~euroknow

Uschold, M.F., Moralee, S., King, M. and Ziorgios, Y., 1998. "The Enterprise Ontology" Knowledge Engineering Review, Special Issue on Ontologies, To appear, Cambridge

University Press. http://www.aiai.ed.ac.uk/~entprise/enterprise/ontology.html

Valente, A., Swartout, W. R. and Gil, Y., 1996. "A Representation and Library for Objectives in Air Campaign Plans". http://www.isi.edu/expect/inspect.html

Wilkins, D.E., 1988. Practical Planning Morgan Kaufmann.http://www.ai.sri.com/~sipe

Wilkins, D.E. and Myers, K.L., 1995. "A Common Knowledge Representation for Plan Generation and Reactive Execution" Journal of Logic and Computation 5(6) 731-761. http://www.ai.sri.com/~act/act-formalism.html

World Wide Web Consortium, RDF Metadata Format, Draft 3-Oct-97. http://www.w3.org/Press/RDF

Workflow Management Coalition, 1994. "Workflow Management Coalition Glossary", A Workflow Management Coalition Specification, November. http://www.wfmc.org

Appendix G:

Multi-agent Planning via Mutually Constraining the Space of Behaviour

Austin Tate

Citation:
Tate, A., Multi-agent Planning via Mutually Constraining the Space of Behaviour, Proceedings of the AAAI-97 Workshop on Constraints and Agents, Providence, Rhode Island, USA, July 1997.

Purpose:
Describes the central approach to multi-agent and mixed initiative planning in O-Plan.

Abstract:
Work is described which seeks to support multi-agent mixed initiative interaction between a "task assignment" or "command" agent and a planning agent[1]. Each agent maintains an agenda of outstanding tasks it is engaged in and uses a common representation of tasks, plans, processes and activities based on the notion that these are all "constraints on behaviour". Interaction between the agents uses explicit task and option management information. This framework can form a basis for mixed initiative user/system agents working together to mutually constrain task descriptions and plans and to coordinate the task-oriented generation, refinement and enactment of those plans.

[1]This paper is based on material presented at the AAAI-97 Spring Symposium on Mixed Initiative Interaction, March 1997, Stanford University.

1 Introduction

Under the O-Plan Project (Currie and Tate, 1991; Tate, Drabble and Kirby, 1994) at the University of Edinburgh, which is part of the DARPA/Rome Laboratory Planning Initiative (Tate, 1996a), we are exploring mixed initiative planning methods and their application to realistic problems in logistics, air campaign planning and crisis action response (Tate, Drabble and Dalton, 1996). In preparatory work, O-Plan has been demonstrated operating in a range of mixed initiative modes on a Non-Combatant Evacuation Operation (NEO) problem (Tate, 1994; Drabble, Tate and Dalton, 1995). A number of "user roles" were identified to help clarify some of the types of interaction involved and to assist in the provision of suitable support to the various roles (Tate, 1994)

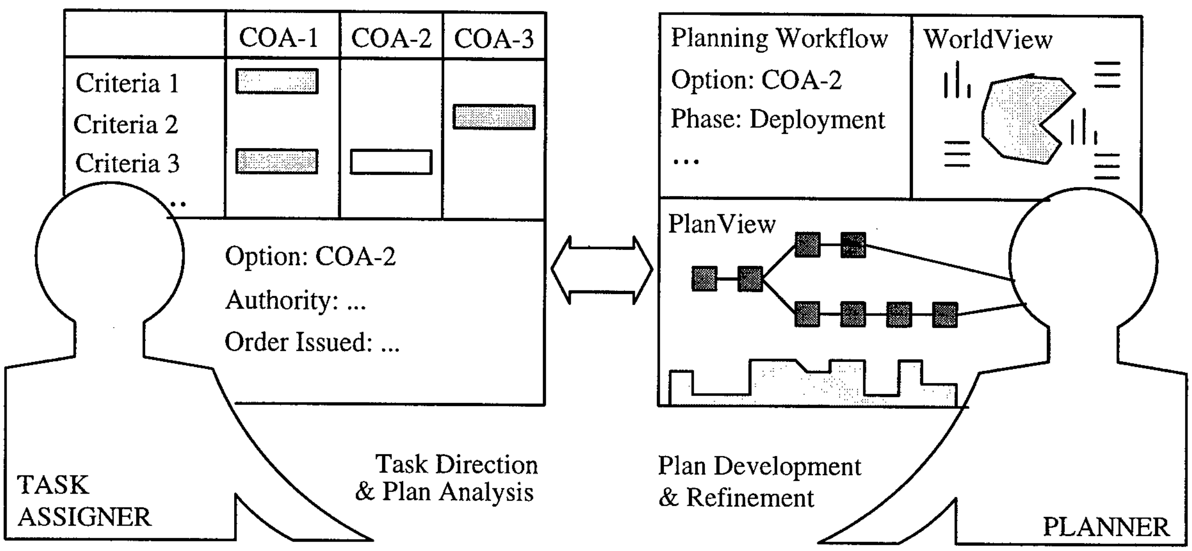


Figure 1: Communication between Task Assigner and Planner

New work started in 1995 is exploring the links between key user roles in the planning process and automated planning support aids – see figure 1. Research is exploring a planning workflow control model using:

- the <I-N-OVA> constraint model of activity as the basis for communication;
- explicit management between agents of the tasks and options being considered;
- agent agendas and agenda issue handlers.

2 Representing Plans as a Set of Constraints on Behaviour

The <I-N-OVA>[2] (*Issues – Nodes – Orderings / Variables / Auxiliary*) Model is a means to represent and manipulate plans 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 to the environments in which they are generated.

In Tate (1996), the <I-N-OVA> model is used to characterise the plan representation used within O-Plan and is related to the plan refinement planning method used in O-Plan. The <I-N-OVA> work is related to emerging formal analyses of plans and planning. This synergy of practical and formal approaches can stretch the formal methods to cover realistic plan representations as needed for real problem solving, and can improve the analysis that is possible for production planning systems.

<I-N-OVA> is 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 is intended to support new work on automatic manipulation of plans, human communication about plans, principled and reliable acquisition of plan information, and formal reasoning about plans.

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.

```
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

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

[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

constraint which will often refer to plan objects/variables and to time points or ranges.

3 Task and Option Management

3.1 O-Plan Architecture

Task and option management facilities are provided by the *Controller* in O-Plan. The O-Plan Controller takes its tasks from an agenda which indicates the outstanding processing required and handles these with its *Knowledge Sources.* The components of a single O-Plan agent are shown in figure 3.

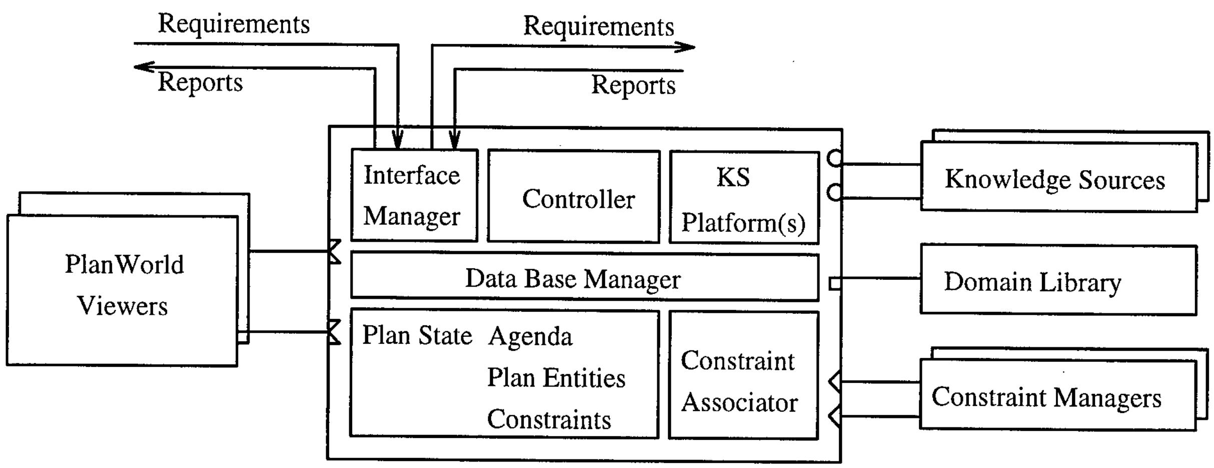


Figure 3: O-Plan Agent Architecture

O-Plan has explicit facilities for managing a number of different options which it is considering. O-Plan has an agent level agenda, and agendas which relate to each option it is considering (in fact these are part of the plan representation for these options - the I part of <I-N-OVA>). Many of these options are internal to the planning agent, and are generated during search for a solution. Others are important for the interaction between the planner and a user acting as a task assigner.

3.2 Abstract Model of Planning Workflow – Plan Modification Operators

A general approach to designing AI-based planning and scheduling systems based on partial plan or partial schedule representations is to have an architecture in which a plan or schedule is critiqued to produce a list of issues or agenda entries which is then used to drive a workflow-style processing cycle of choosing a "plan modification operator" (PMO) to handle one or more agenda issues and then executing the PMO to modify the plan state. Figure 4 shows this graphically.

cross-constraints in some other domain may be some other constraint type.

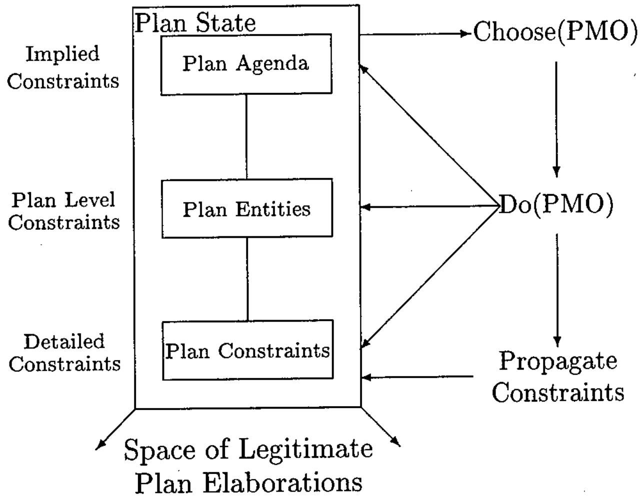


Figure 4: Planning Workflow - Using PMOs to Handle Agenda Issues

This approach is taken in O-Plan. The approach fits well with the concept of treating plans as a set of constraints which can be refined as planning progresses. Some such systems can act in a non-monotonic fashion by relaxing constraints in certain ways. Having the implied constraints or "agenda" as a formal part of the plan provides an ability to separate the plan that is being generated or manipulated from the planning system itself.

3.3 Generic Systems Integration Architecture

The O-Plan agent architecture has been generalised into the generic systems integration architecture shown in figure 5. This general structure has been adopted on a number of AIAI projects (Fraser and Tate, 1995).

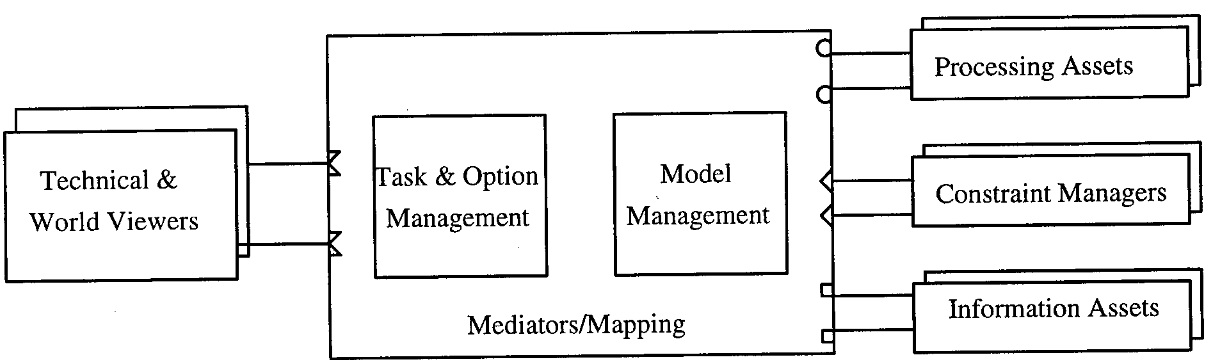


Figure 5: Generic Systems Integration Architecture

The various components "plug" into "sockets" within the architectural framework. The sockets are specialised to ease the integration of particular types of component.

The components are as follows:

Viewers – User interface, visualisation and presentation viewers for the model - sometimes differentiated into *technical* model views (charts, structure diagrams, etc.) and *world* model views (simulations, animations, etc.)

Task and Option Management – The capability to support user tasks via appropriate use of the processing and information assets and to assist the user in managing options being used within the model.

Model Management – coordination of the capabilities/assets to represent, store, retrieve, merge, translate, compare, correct, analyse, synthesise and modify models.

Mediators – Intermediaries or converters between the features of the model and the interfaces of active components of the architecture (such as viewers, processing assets, constraint managers and information assets).

Processing Assets – Functional components (model analysis, synthesis or modification).

Constraint Managers – Components which assist in the maintenance of the consistency of the model.

Information Assets – Information storage and retrieval components.

3.4 Communicating Plan Information Between the Task Assignment and Planning Agents

The <I-N-OVA> constraint model of activity allows planning process state as well as the current state of the plan generated to be communicated between agents involved in the planning process. This is done via the Issues part of <I-N-OVA> - which can be used to amend the task and option specific agenda which a planning agent is using for its problem solving. Ways to authorise agents to take initiative in the problem solving process are being explored. This can be done by communicating the types of agenda entry or issue which the planning agent may handle and giving limitations on which types of constraint that may be manipulated and the extent to which they may be manipulated while problem solving.

This involves improving the workflow controller at the heart of the O-Plan planner agent. This will allow dialogue between users and automated planners as the problem solving takes place. Methods to allow for coordination of task and option management between users and the automated planner are being added to O-Plan.

4 Authority to Plan

At the moment the Task Assignment agent tells the O-Plan planner when it can create a plan for a nominated task. This is done through a simple menu interface today. As described in Tate (1993) it is intended that O-Plan will support authority management in a more comprehensive and principled way in future. This section describes the way in which this is being done. O-Plan will support:

- the notion of separate *plan options* which are individually specified task requirements, plan environments and plan elaborations. The Task Assignment agent can create as many as required. The plan options may contain the same task[4] with different search options or may contain a different task and environmental assumptions. It is possible to have only one plan option as the minimum[5]. *Sub-options* may be established between the task assignment and planner agents to give some structure to the ways in which the space of such options and sub-options is explored between the two agents.
- the notion of plan *phases*. These are individually provided actions or events stated explicitly in the top level task description given by the Task Assignment agent. Greater precision of authority management is possible by specifying more explicit phases at the task level. It is possible to have only one "phase" in the task as the minimum[6].
- the notion of plan *levels*. Greater precision of authority management is possible by specifying more explicit levels in the domain Task Formalism (TF). It is possible to have only one "level" in the domain as the minimum.
- for each "phase", planning will only be done down to an authorised "level" at which point planning will suspend leaving appropriate agenda entries until deeper planning authorisation is given.
- execution will be separately authorised for each "phase".

Domain related names that are meaningful to the user may be associated with these options, sub-options, phases and levels through the Task Assignment agent.

Changes of authority are possible via Task Assignment agent communication to the Planner agent. This may be in the context of a current plan option and task provided previously or it is possible to give defaults which apply to all future processing by the planner agent.

5 Mutually Constraining Plans for Mixed Initiative Planning and Control

Our approach to Mixed Initiative Planning in O-Plan proposes to improve the coordination of planning with user interaction by employing a clearer shared model of the plan as a set of constraints at various levels that can be jointly and explicitly discussed between and manipulated by the user or system in a cooperative fashion.

[4]Multiple conjunctive tasks in one scenario is also possible.

[5]Plan options may be established and explicitly switched between by the Task Assignment agent.

[6]In fact any sub-component of any task schema or other schema included by task expansion in a plan can be referred to as a "phase" within the O-Plan planner agent. This can be done by referring to its node number.

The model of Mixed Initiative Planning that can be supported by the approach is *the mutual constraining of behaviour* by refining a set of alternative partial plans. Users and systems can work in harmony though employing a common view of their roles as being to constrain the space of admitted behaviour. Further detail is given in Tate (1994).

Workflow ordering and priorities can be applied to impose specific styles of authority to plan within the system. One extreme of user driven plan expansion followed by system "filling-in" of details, or the opposite extreme of fully automatic system driven planning (with perhaps occasional appeals to an user to take predefined decisions) are possible. In more practical use, we envisage a mixed initiative form of interaction in which users and systems proceed by mutually constraining the plan using their own areas of strength.

Coordination of problem solving must take place between users and the automated components of a planning system. In joint research with the University of Rochester (whose work is described in Allen, Ferguson and Schubert, 1996) we are exploring ways in which the O-Plan controller can be given specific limitations on what plan modifications it can perform, and the specific plan options or sub-options it is working on can be coordinated with those being explored by a user supported by a suitable interface.

6 Summary

Five concepts are being used as the basis for exploring multi-agent and mixed-initiative planning involving users and systems:

1. a rich plan representation using a common constraint model of activity (<I-N-OVA>).
2. mixed initiative model of "mutually constraining the space of behaviour".
3. explicit task and option management – via a tasking interace which can share options and sub-options between agents.
4. abstract model of the planning agent having handlers for issues, functional capabilities and constraint managers.
5. management of the authority to plan (to handle issues) which may be given in advance or may be stated with the task specification and which may take into account options, phases and levels.

Together these provide for a *shared* model of what each agent can and is authorised to do and what those agents can act upon.

Acknowledgements

The O-Plan project is sponsored by the Defense Advanced Research Projects Agency (DARPA) and Rome Laboratory, Air Force Materiel Command, USAF, under grant number

F30602-95-1-0022. The O-Plan project is monitored by Dr. Northrup Fowler III at the USAF Rome Laboratory. The U.S. Government is authorised to reproduce and distribute reprints for Governmental 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, Rome Laboratory or the U.S. Government.

References

Allen, J.F., Ferguson, G.M. and Schubert, L.K. (1996), Planning in Complex Worlds via Mixed-Initiative Interaction, in Advanced Planning Technology, pp. 53-60, (Tate, A., ed.), AAAI Press.

Currie, K.W. and Tate, A. (1991), O-Plan: the Open Planning Architecture, Artificial Intelligence, 51(1), Autumn 1991, North-Holland. Information available at `http://www.aiai.ed.ac.uk/~oplan/`

Drabble, B., Tate, A. and Dalton, J. (1995) Applying O-Plan to the NEO Scenarios, in An Engineer's Approach to the Application of Knowledge-based Planning and Scheduling Techniques to Logistics, Appendix O, USAF Rome Laboratory Technical Report RL-TR-95-235, December 1995. Expanded version available as Drabble, B., Tate, A. and Dalton, J. (1995) O-Plan Project Evaluation Experiments and Results, O-Plan Technical Report ARPA-RL/O-Plan/23 Version 2, 10-Nov-95. Available at `ftp://ftp.aiai.ed.ac.uk/pub/projects/oplan/ documents/95-tr-23-experiments.ps`

Fraser, J. and Tate, A. (1995), The Enterprise Tool Set – An Open Enterprise Architecture, Proceedings of the Workshop on Intelligent Manufacturing Systems, International Joint Conference on Artificial Intelligence (IJCAI-95), Montreal, Canada, August 1995. Available at `ftp://ftp.aiai.ed.ac.uk/pub/documents/1995/ 95-ims-ijcai95-ent-toolset.ps`

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. Available at `ftp://ftp.aiai.ed.ac.uk/pub/documents/1993/ 93-ijcai-authority.ps`

Tate, A. (1994), Mixed Initiative Planning in O-Plan2, Proceedings of the ARPA/Rome Laboratory Planning Initiative Workshop, pp. 512-516, (Burstein, M., ed.), Tucson, Arizona, USA, Morgan Kaufmann. Available at `ftp://ftp.aiai.ed.ac.uk/pub/documents/1994/ 94-arpi-mixed-initiative.ps`

Tate, A. (1996a) (ed.), Advanced Planning Technology, AAAI Press.

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), pp. 221-228, (Drabble, B., ed.) Edinburgh, Scotland, AAAI Press. Available at `ftp://ftp.aiai.ed.ac.uk/pub/documents/1996/ 96-aips-inova.ps`. Further information available at `http://www.aiai.ed.ac.uk/~bat/inova.html`

Tate, A., Drabble, B. and Kirby, R. (1994), O-Plan2: an Open Architecture for Command, Planning and Control, in Intelligent Scheduling, (eds, M.Zweben and M.S.Fox), Morgan Kaufmann. Available at `ftp://ftp.aiai.ed.ac.uk/pub/documents/1994/94-is-oplan2.ps`

Tate, A., Drabble, B. and Dalton, J. (1996), A Knowledge-Based Planner and its Application to Logistics, in Advanced Planning Technology, pp. 259-266, (Tate, A., ed.), AAAI Press. Available at `ftp://ftp.aiai.ed.ac.uk/pub/documents/1996/96-arpi-oplan-and-logistics.ps`

Appendix H:

Repairing Plans on the Fly

Brian Drabble, Jeff Dalton and Austin Tate

Citation:
Drabble, B., Dalton, J. and Tate, A., Repairing Plans on the Fly, Proceedings of the NASA Workshop on Planning and Scheduling for Space, Oxnard CA, USA, October 1997, NASA Jet Propulsion Laboratory.

Purpose:
Planning takes place in a dynamic environment where tasks, assumptions and information from the environment itself may all be changing rapidly. This paper described the algorithms used in O-Plan to allow plans to be altered to respond to such changes.

Abstract:
Even with the most careful advance preparation, and even with inbuilt allowance for some degree of contingency, plans need to be altered to take into account execution circumstances and changes of requirements. We have developed methods for repairing plans to account for execution failures and changes in the execution situation. We first developed these methods for the Optimum-AIV planner designed to support spacecraft assembly, integration and verification at ESA, and later deployed for Ariane IV payload bay AIV. This system was itself based on our Nonlin and O-Plan planning algorithms and plan representation. We subsequently refined the methods for the O-Plan planner and incorporated plan repair methods into the system. This paper describes the algorithms used for plan repair in O-Plan and gives an example of their use.[1]

[1]Brian Drabble is now a member of the Computational Intelligence Research Laboratory, University of Oregon.

1 Introduction

Even with the most careful advance preparation and even allowing for some degree of contingency pre-built into the plans, any plan being executed in the real world will have to be adapted to take into account execution circumstances and changes of requirements. For example, a deep space probe may require to adapt to new science experiments as new information leads to further experiments. Alternatively, cases such as Galileo have shown that failures in the spacecraft's hardware may need to be overcome by altering the current set of tasks and plans.

One of the aims of the O-Plan project during Phase II of the DARPA/Rome Laboratory Planning Initiative (Tate, 1996a) was to develop techniques to allow plans to be changed to take into account modifications in the task requirements and in the execution environment. The techniques allowed a failure to be identified and repaired with minimum impact on the rest of the plan.

The basis for the techniques was first developed for the Optimum-AIV planner designed for spacecraft assembly, integration and verification support at ESA and later deployed for Ariane IV payload bay AIV.

This paper will briefly describe some of the background work on O-Plan and Optimum-AIV, and then describe the algorithms used for plan repair in O-Plan. The paper describes a demonstration was which conducted in a command, planning, and control environment of the US air force. The task was to evacuate a number of foreign nationals from the fictional island of Pacifica (Reece et.al. 1993) and to transport them to safety. While the example is not directly related to the space domain, the demonstration does show how new requirements and changes in the environment can be integrated into an ongoing and executing plan and would be of use in the solving problems such as AIV, control of autonomous spacecraft, and lander missions.

2 Optimum-AIV – Assembly, Integration and Verification Planning

Planning is a key issue in the management of the assembly, integration and verification (AIV) activities of a space project. Not only must technological requirements be met, but cost and time are critical. There are costly testing facilities which must be shared with other projects, and there is a need to plan the coordination between a number of participants (agencies, contractors, launcher authorities, users). A delay caused by one participant normally leads to serious problems for others. Managers at all levels of a space project are concerned with planning, and they control closely the progress of the work. However, it has been difficult to find computer-based planning aids which meet the needs of this application. General purpose project management software cannot represent the wide range of factors to be taken into account, and is too complex to be used to interactively modify plans during project execution (Parrod et. al. 1993). For this reason, the European Space Agency commissioned the Optimum-AIV system which utilizes AI planning representations and techniques (Aarup et. al.

1995; Tate, 1996b).

The system which was developed was based on the earlier Nonlin (Tate, 1977) and O-Plan (Currie and Tate, 1991; Tate et.al., 1994) planning algorithms and plan representation. The following techniques are used in Optimum-AIV

- Optimum-AIV adopts a partially-ordered plan representation, which supports causally independent activities that can be executed concurrently.
- It searches through a space of partial plans, modifying them until a valid plan/schedule is found.
- The system employs hierarchical planning. The term hierarchical refers to both the representation of the plan at different levels, and also the control of the planning process at progressively more detailed levels.
- During plan specification and generation, the system operates on explicit preconditions and effects of activities that specify the applicability and purpose of the activity within the plans. With this knowledge, it is possible to check whether the current structure of the plan introduces any conflicts between actual spacecraft system states, computed by the system, and activity preconditions, which have been specified by the user. Such conflicts would arise if one activity deletes the effect of another, thus removing its contribution to the success of a further activity. The facility for checking the consistency of the plan logic, by dependency recording, is not possible within existing project management tools, which assume that the user must get this right.
- Detailed constraints are associated with the plan. These represent resource and temporal constraints on the activities in the plan as well as a more general class of global activity constraints. The scheduling task in Optimum-AIV is considered as a constraint satisfaction problem solved by constraint-based reasoning. The constraints are propagated throughout the plan, gradually transforming it into a realizable schedule. Invariably not all of the constraints can be met, such that some have to be relaxed via user intervention.
- During planning, the system records the rationale behind the plan structure; that is, user decisions on alternatives are registered. This is used to assist during plan repair where the user tries to restore consistency. Information can then be derived about alternative activities, soft constraints that may be relaxed, and potential activities that may be performed in advance.
- Test Failure Recovery Plans are available as plan fixes to enable the plan to be brought back on track after the failure of a test during the assembly and integration process. The same AI planning methods used to generate a plan are also used to assist in fixing such problems. Optimum-AIV assists the user in plan repair in an interactive way rather than performing the repair itself.

Following an evaluation of Optimum-AIV at ESA, it has been reported (Parrod et. al, 1993) that the system is in use for planning the production of the vehicle equipment bays of the

European Ariane IV launcher. It was reported that the system was chosen by the Araine IV project team due to the following:

- the wealth of information which can be provided to and used by the tool to describe the constraints inherent in the AIV activity.
- the quality of support provided by the tool to allow resource conflicts to be resolved.
- the clear representation of information and the interactive capabilities which enables engineering management to access several planning scenarios on-line.
- the fact that Optimum-AIV provides a single solution to problems of managing the plan, schedule, and allocation of resources amongst competing vehicle equipment bays which are concurrently being assembled.

Optimum-AIV provides a rich plan representation and aids to allow for the editing of AIV planning information and a wide range of constraints on the process. This information forms a basis for plan generation, checking of plan logic, and analysis of plans. Facilities are available to allow for the interactive repair of executing plans when tests indicate failures of components under assembly and integration. Optimum-AIV is an example of a deployed application of a number of AI planning techniques.

3 O-Plan Demonstration and Scenario Description

A demonstration experiment was performed which showed O-Plan solving a number of tasks from an integrated command, planning and control scenario related to the performance of Non-combatant Evacuation Operations (NEOs) on the fictional island of Pacifica (Reece et.al. 1993). The aims of the demonstration were to show:

- O-Plan reacting to changes in the environment and identifying those parts of the plan which were now threatened by these changes.
- O-Plan reacting to changes in the overall task by integrating new plan requirements into the plan.

The types of plan repairs explored in this demonstration include responses to failures of trucks due to blown engines and tyres and the inclusion of new task objectives, such as to pick up an extra group of evacuees. The Pacifica scenario used for the demonstration is a simplification of a real logistics problem of interest to the DARPA/Rome Laboratory Planning Initiative (Tate, 1996a). The plan schema library for this domain contained 12 schemas which defined alternative evacuation methods: trucks or helicopters, fuel supplies, transport aircraft, etc. The plans generated contained an average of 20 actions and were developed in approximately 40-60 seconds. Four different repair plans were used in the demonstration:

- Three cases of a blown engine on a ground transport:

- The engine can only be fixed by a repair crew which is dispatched from the Pacifica airport at Delta with a tow truck. The ground transport is then towed to Delta for repairs. The evacuees remain with the ground transport while it is being towed.
- The failure of the transport occurs in a time critical situation and there is insufficient time to tow the broken transport to Delta. The evacuees are moved from the broken ground transport by helicopter to Delta and the transport is abandoned.
- The failure of the transport occurs in a time critical situation, and the evacuees are moved by another ground transport instead of by helicopter.

• One case of a blown tyre on a ground transport:
  - The driver of the ground transport can fix the tyre by the side of the road. The effect of the repair action is to delay the ground transport by a fixed amount of time.

In addition, a closely allied Ph.D student project by Glen Reece developed a more comprehensive reactive execution agent (Reece, 1994; Reece and Tate, 1994) based on the O-Plan architecture. It has been used to reactively modify plans in response to operational demands in a simulation of the Pacifica island in the context of a NEO.

4 O-Plan Plan Repair Algorithms

The plan-repair mechanisms allow O-Plan to integrate a number of pre-assembled repair plans—e.g., to repair a blown engine, or to repair a flat tyre—into an ongoing and executing plan. Although the integration was performed by the planning agent, the techniques and methods could easily have been added to the capabilities of a separate execution agent.

O-Plan's plan representation contains two tables used by the plan repair algorithms to determine the consequences of failures: the **Table of Multiple Effects** (TOME) and the **Goal Structure Table** (GOST). Plans contain actions (nodes), and actions can have effects. Effects can take place at either end of an action: at the start (`begin_of`) or finish (`end_of`). Each effect is recorded in the TOME by an entry of the form ⟨pattern⟩ `=` ⟨value⟩ `at` ⟨node-end⟩. For example, `(colour_of ball) = green at end_of node-1`.

When an action depends on an effect asserted earlier, that is recorded in the GOST by an entry of the form ⟨condition-type⟩ ⟨pattern⟩ `=` ⟨value⟩ `at` ⟨condition-node-end⟩ `from` ⟨contributor-node-ends⟩. This specifies a protected range: ⟨pattern⟩ = ⟨value⟩ is asserted at one of the contributor-node-ends and is required at the condition-node-end. For example, `unsupervised (colour_of ball) = green at begin_of node-1-2 from end_of node-1`.

These tables are maintained by the O-Plan TOME and GOST Manager (TGM). A plan repair is required when one or more of the GOST entries are broken—i.e. a contributor of a GOST entry is not asserted as expected, or an external world event occurs and asserts extra effects into the plan that break a protected range by undoing a required effect.

Plan repairs are dealt with by a number of knowledge sources—pieces of code which deal with a specific aspect of the planning problem. The knowledge sources are responsible for determining the consequences of unexpected events, or of actions that do not execute as intended, for deciding what action to take when a problem is detected, and for making repairs to the effected plan.

O-Plan maintains an agenda of "issues" that need to be resolved in the plan. For each type of issue, there is a corresponding knowledge source; and the top-level control structure in O-Plan is a loop that repeatedly selects an issue from the agenda and calls the appropriate knowledge source. When describing algorithms below, we will therefore sometimes speak of "posting" and agenda entry, where the issue type is represented by the knowledge source name (KS-CONTINUE-EXECUTION, KS-FIX, etc.)

The two types of problems that dealt with by the repair mechanisms can now be described in more detail:

- **Execution Failure**:
  An execution failure occurs when one or more of the expected effects at a node-end fail to be asserted. For example, the node-end corresponding to the end of the action `Check_out_ground_transport` should assert that the status of the engine and tyres is fine: `(engine_status gt1) = working` and `(tyre_status gt1) = working`. This may not in fact be so if the action has not executed correctly. This type of failure may cause problems if the expected effects of the action are needed to satisfy the preconditions of a later action. For example, the evacuation of people from an outlying city can only precede if the tyres and engine of the ground transport continue to function correctly.
- **Unexpected World Event**:
  Unexpected events cause effects in the world which can make planned actions fail. For example, a landslide event may have the effect `(road_status Abyss_to_Barnacle) = closed` and this would interfere with any action requiring the road to be open.

The description of the algorithms of the execution and plan repair system is divided into three main sections. The first describes how the system maintains an execution fringe of the node-ends awaiting execution; the second describes how the system deals with plan failures; and the third describes how it handles unexpected world events.

4.1 Maintaining the Execution Fringe and "Necking" the Plan

An activity is represented in an O-Plan plan as a node with two ends. Conditions and effects can be attached to either end of a node and are monitored by the execution system. The system reasons purely in terms of node-ends and not in terms of activities or events.

The "execution fringe" is the list of node-ends currently ready for execution. A node-end is ready when all node-ends that must execute before it in the partially ordered plan have completed execution.[2] When ready, it can be dispatched for execution. That involves sending

[2] This check considers both links explicitly in the plan and temporal constraints maintained by a Time-Point

a message to an execution agent, which in turn sends messages to a world simulator that maintains a model of the world in which execution is taking place. As actions begin and end in the world, the simulator reports back to the execution agent, resulting in success and failure messages about the corresponding node-ends being sent to the planner. When the planner receives a success or failure message about a node-end, it marks the end as having completed execution; and that may lead to further node-ends being considered ready.[3]

By keeping track of which node-ends have finished execution, the system maintains a content within which replanning for plan repair can take place and can establish a focus point when considering where to insert repair actions—after all node-ends which have executed and before any node-ends waiting to execute. This point is known as the plan's *neck point* and a single dummy node can be added to the plan by the repair algorithm to *neck* the plan at that point, when necessary.

Note that the "ready to execute" check for a node-end E considers only whether all the node-ends that must execute before E have been executed, regardless of whether the execution was successful. It assumes that any problems due to execution failures or world events have been fixed, and it is the responsibility of other parts of the system to ensure that this is so.

A node-end that is ready can have its status set back to not-ready after a plan repair, because the repair may introduce new actions that must execute first.

4.2 Dealing with Execution Failures

When an execution failure occurs at a particular node-end, some of the expected effects may not occur. They are returned from the execution monitoring system to the planning agent as a list of failed-effects. The task of the planning system is to fix the plan so that any condition that needed one of the failed effects as a contributor is satisfied in some other way. The fix can be relatively simple if there is already another contributor in the GOST entry or if there is a suitable alternative contributor already present in the plan. If these simple fixes cannot be applied, then the system will attempt to add a new action to the plan. However, if nothing requires the failed effects, then the execution "failure" can be ignored.

The main algorithm used by the system to track execution and initiate repairs is as follows:

- Mark the node-end as having been executed.
- If there are no failed effects, then a repair is not needed.
- If there are failed effects then remove the TOME entries that correspond to them
- Determine which GOST entries are affected by the failed (removed) effects. If there are none, then a repair is not needed.
- At this point there is a failure that must be repaired.

Network Manager (TPNM).

[3]It is assumed that execution is not so rapid relative the the planner's ability to respond that the planner's model becomes significantly out of date.

- Search through the affected GOST entries in turn. If a GOST entry has more than one contributor, check if any are still valid. If so, reduce the contributor list; otherwise record the GOST entry as truly broken.
  - If no GOST entries are truly broken, then the repair is complete.

- At this point, some GOST entries are truly broken and result in "issues" that must be resolved. For each of the broken GOST entries, post a KS-FIX agenda entry. When that agenda entry is processed, the KS-FIX knowledge source will be invoked, and it will consider two repair methods for satisfying the condition in the broken GOST entry:[4]
  - Find an existing alternative contributor in the plan.
  - Bring in additional actions (a repair plan) which assert the appropriate effect. Any new nodes will be linked after the *neck point* described above.

- Post a KS-CONTINUE-EXECUTION to continue execution after the fixes have been made.

Certain details of the repair depend on the type of the condition recorded in the broken GOST entry. In particular, a `supervised` condition is unlike all other types because it requires that a ⟨pattern⟩ = ⟨value⟩ assignment be true *across a range*, rather than only at a single point.

Suppose a broken GOST entry g has the form `supervised` $p = v$ `at` e `from` c. Then c is a node end that asserts $p = v$, and $p = v$ must be so not only at node-end e (which is all that other condition types would require) but also at node ends between c and e that are *spanned by the condition*. These are the siblings of c and e that are explicitly linked between c and e, or the descendants of such siblings, where two node-ends are siblings if they were introduced as sub-actions of the same action.

Broken `supervised` conditions are handled as follows:

- Create a new dummy node d to act as the "delivery point".

- Link d after the neck point, before e, and before all node-ends that are spanned by the condition and have not yet been executed.

- Change the GOST entry to list d as the contributing node-end, and give d $p = v$ as an effect in the TOME.

- Post a KS-FIX to re-establish $p = v$ at d.

The system must be consistent in its use of the "ends" (`begin` and `end`) of d to avoid "gaps" in the goal structure which would effect the meaning of the plan.

[4]The "fix" issue is in effect a condition of type `achieve` as described in (Tate et.al., 1994).

4.3 Dealing with Unexpected World Events

When a significant event that is not in the plan occurs in the world, it is reported to the planner as a time, an event pattern, and a list of effects (⟨pattern⟩ = ⟨value⟩ pairs). For instance, the occurrence of a landslide might be reported as:

```
event {landslide} with effects
   {status road-a} = blocked,
   {status road-b} = blocked;
```

Events are treated the same way as plan activities except they are not placed in the plan until they have occurred. The effects may break GOST ranges in the plan and if so, the planner must try to satisfy those conditions some other way. However, even if no GOST entries are broken, the planner needs to add a node to represent the world event. This is because, even if the event's effects don't make any difference now, they may matter later on.

The new event node represents something that has definitely and already happened. So it must be linked after all node-ends that have already been executed and before all node-ends that have not yet been executed.

The algorithm for dealing with unexpected world events is as follows:

- Add an event node, E, to represent the world event. Link it as described above. Mark E as having already been executed.
- Edit the GOST to remove any contributors that can no longer contribute, and get a list of the truly broken GOST entries. A contributor is removed when
  - the condition is at a node-end that has not been executed,
  - the contributor is a node-end that has been executed, and
  - the unexpected world-event has a conflicting effect.
- For each truly broken GOST entry g, post a KS-FIX agenda entry as in the case of an execution failure, using `end_of` E as a neck point.
- Add the world event's effects at `end_of` E.
- If there were no truly broken GOST entries, then we are finished. Otherwise, Post a KS-CONTINUE-EXECUTION to continue execution after the fixes have been made. (The fixes will be made by processing the KS-FIX agenda entries.)

5 Conclusions

This paper has shown that current AI planning and scheduling techniques have reached the point where they can be deployed in real-world applications. Systems such as Optimum-AIV have shown that they provide valuable support to human users in identifying the point of failure in a plan and suggesting appropriate repairs. The techniques described in this paper to support plan repair are general enough to be applied in a wide variety of planning and scheduling applications.

References

Aarup, M., Arentoft, M.M., Parrod, Y., Stokes, I., Vadon, H. and Stader, J. (1994) Optimum-AIV: A Knowledge-Based Planning and Scheduling System for Spacecraft AIV, in Intelligent Scheduling (eds. Zweben, M. and Fox, M.S.), pp. 451-469, Morgan Kaufmann.

Currie, K. and Tate, A. (1991) O-Plan: the Open Planning Architecture, Artificial Intelligence Vol. 52, pp. 49-86, Elsevier.

Parrod, Y., Valera, S. (1993) Optimum-AIV, A Planning Tool for Spacecraft AIV, in Preparing for the Future, Vol. 3, No. 3, pp. 7-9, European Space Agency.

Reece, G.A., (1994) Characterization and Design of Competent Rational Execution Agents for Use in Dynamic Environments, Ph.D Thesis, Department of Artificial Intelligence, University of Edinburgh, November 1994.

Reece, G.A. and Tate, A. (1994) Synthesizing Protection Monitors from Causal Structure, Proceedings of the Second International Conference on AI Planning Systems (AIPS-94), AAAI Press, Chicago, USA, June 1994.

Reece, G.A., Tate, A., Brown, D. and Hoffman, (1993) M., The PRECiS Environment, Paper presented at the ARPA-RL Planning Initiative Workshop at AAAI-93, Washington D.C., July 1993. Also available as University of Edinburgh, Artificial Intelligence Applications Institute Technical Report AIAI-TR-140.

Tate, A. (1977) Generating Project Networks, Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-77), pp. 888-893, Cambridge, MA, USA, Morgan Kaufmann.

Tate, A. (1996a) Advanced Planning Technology, AAAI Press.

Tate, A. (1996b) Responsive Planning and Scheduling Using AI Planning Techniques, Trends and Controversies, IEEE Expert - Intelligent Systems and Their Applications, Winter 1996.

Tate, A., Drabble, B. and Dalton, J. (1994) The Use of Condition Types to Restrict Search in an AI Planner, Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94), pp. 1129-1134, Seattle, USA, August 1994.

Tate, A., Drabble, B. and Dalton, J. (1996), A Knowledge-Based Planner and its Application to Logistics, in Advanced Planning Technology, pp. 259-266, (Tate, A., ed.), AAAI Press.

Tate, A., Drabble, B. and Kirby, R. (1994), O-Plan2: an Open Architecture for Command, Planning and Control, in Intelligent Scheduling, (eds, M.Zweben and M.S.Fox), Morgan Kaufmann.

Appendix I:

A Planning Agent on the World Wide Web

Austin Tate

Citation:
Tate, A., A Planning Agent on the World Wide Web, Seminar on Agents in Information Systems, Heathrow, London, UK, 9th October 1997, Unicom Seminars Ltd., Uxbridge, Middlesex, UK.

Purpose:
Describes the way in which O-Plan has been re-engineered to act as a service to other systems or to act as a server over the World-Wide Web.

Abstract:
Work is described which seeks to support multi-agent mixed initiative interaction between a "task assignment" or "command" agent and a planning agent[1]. Each agent maintains an agenda of outstanding tasks it is engaged in and uses a common representation of tasks, plans, processes and activities based on the notion that these are all "constraints on behaviour". Interaction between the agents uses explicit task and option management information. This framework can form a basis for mixed initiative user/system agents working together to mutually constrain task descriptions and plans and to coordinate the task-oriented generation, refinement and enactment of those plans. The facilities have been provided as a planning support agent serving task assignment and planning users over the world wide web.

[1]Parts of this paper are based on a description of the O-Plan multi-agent system given at the AAAI-97 Workshop on "Constraints and Agents", Providence, RI, USA on 27th July 1997.

1 Introduction

Under the O-Plan Project (Currie and Tate, 1991; Tate, Drabble and Kirby, 1994) at the University of Edinburgh, which is part of the DARPA/Rome Laboratory Planning Initiative (Tate, 1996a), we are exploring mixed initiative planning methods and their application to realistic problems in logistics, air campaign planning and crisis action response (Tate, Drabble and Dalton, 1996). In preparatory work, O-Plan has been demonstrated operating in a range of mixed initiative modes on a Non-Combatant Evacuation Operation (NEO) problem (Tate, 1994; Drabble, Tate and Dalton, 1995). A number of "user roles" were identified to help clarify some of the types of interaction involved and to assist in the provision of suitable support to the various roles (Tate, 1994)

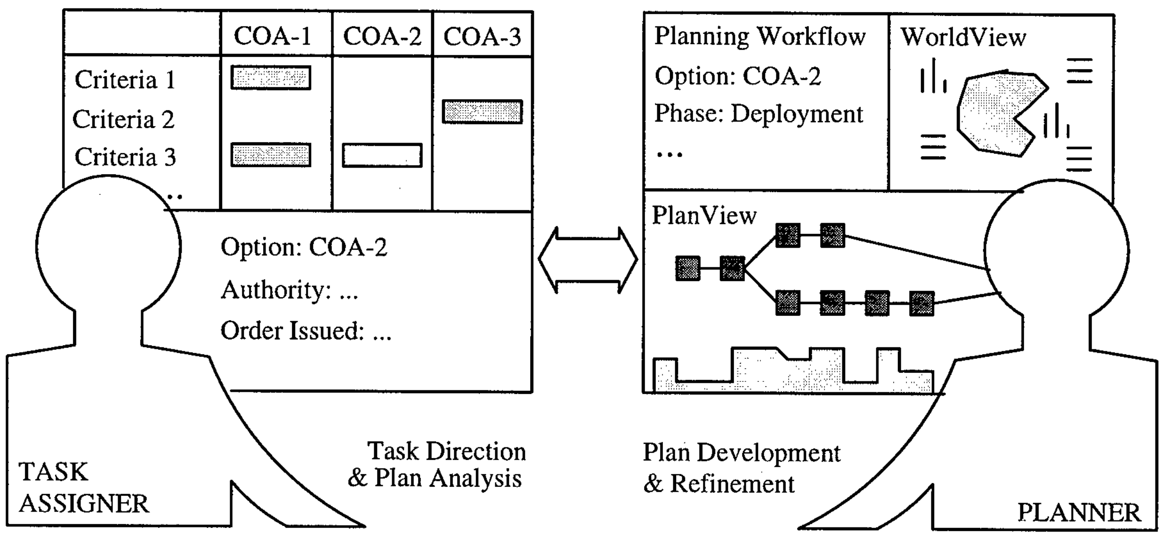


Figure 1: Communication between Task Assigner and Planner

New work started in 1995 is exploring the links between key user roles in the planning process and automated planning support aids – see figure 1. Research is exploring a planning workflow control framework and shared models using:

- the <I-N-OVA> constraint model of activity as the basis for communication;
- explicit management between agents of the tasks and options being considered;
- agent agendas and agenda issue handlers.

A demonstration environment has been created which uses the World Wide Web to allow users access from any web browser to an O-Plan planning agent[2].

[2]The demonstration is available through URL `http://www.aiai.ed.ac.uk/~oplan/` by following the link to the "Live Demonstrations" page entry for "Pacifica COA Matrix".

2 Generic Systems Integration Architecture

The O-Plan agent architecture to be described in the next section is a specific variant of a generalised systems integration architecture shown in figure 2. This general structure has been adopted on a number of AIAI projects (Fraser and Tate, 1995). The architecture is an example of a *Model/Viewer/Controller* arrangement.

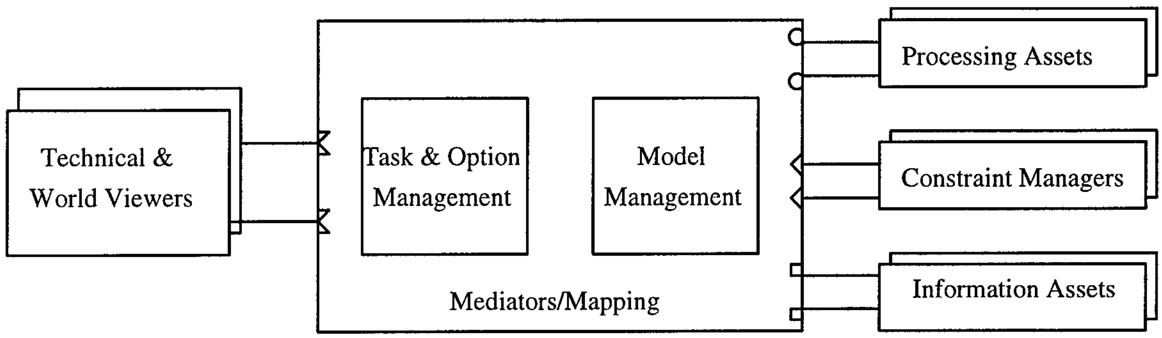


Figure 2: Generic Systems Integration Architecture

The various components "plug" into "sockets" within the architectural framework. The sockets are specialised to ease the integration of particular types of component.

The components are as follows:

Viewers – User interface, visualisation and presentation viewers for the model - sometimes differentiated into *technical* model views (charts, structure diagrams, etc.) and *world* model views (simulations, animations, etc.)

Task and Option Management – The capability to support user tasks via appropriate use of the processing and information assets and to assist the user in managing options being used within the model. This is sometimes referred to as the *Controller*.

Model Management – coordination of the capabilities/assets to represent, store, retrieve, merge, translate, compare, correct, analyse, synthesise and modify models.

Mediators – Intermediaries or converters between the features of the model and the interfaces of active components of the architecture (such as viewers, processing assets, constraint managers and information assets).

Processing Assets – Functional components (model analysis, synthesis or modification).

Constraint Managers – Components which assist in the maintenance of the consistency of the model.

Information Assets – Information storage and retrieval components.

3 O-Plan – the Open Planning Architecture

This section describes the O-Plan architecture and the structure of individual O-Plan agents. The components of a single O-Plan agent are shown in figure 3.

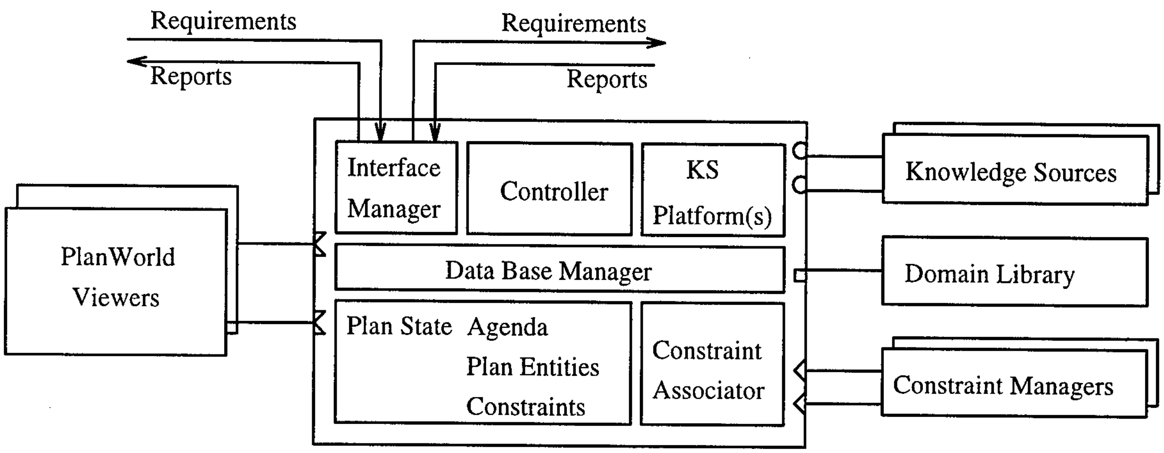


Figure 3: O-Plan Agent Architecture

3.1 Task and Option Management

Task and option management facilities are provided by the *Controller* in O-Plan. The O-Plan Controller takes its tasks from an agenda which indicates the outstanding processing required and handles these with its *Knowledge Sources.*

O-Plan has explicit facilities for managing a number of different options which it is considering. O-Plan has an agent level agenda, and agendas which relate to each option it is considering (in fact these are part of the plan representation for these options - the I part of <I-N-OVA>). Many of these options are internal to the planning agent, and are generated during search for a solution. Others are important for the interaction between the planner and a user acting as a task assigner.

3.2 Abstract Model of Planning Workflow – Plan Modification Operators

A general approach to designing AI-based planning and scheduling systems based on partial plan or partial schedule representations is to have an architecture in which a plan or schedule is critiqued to produce a list of issues or agenda entries which is then used to drive a workflow-style processing cycle of choosing a "plan modification operator" (PMO) to handle one or more agenda issues and then executing the PMO to modify the plan state. Figure 4 shows this graphically.

This approach is taken in O-Plan. The approach fits well with the concept of treating plans as a set of constraints which can be refined as planning progresses. Some such systems can act in a non-monotonic fashion by relaxing constraints in certain ways. Having the implied

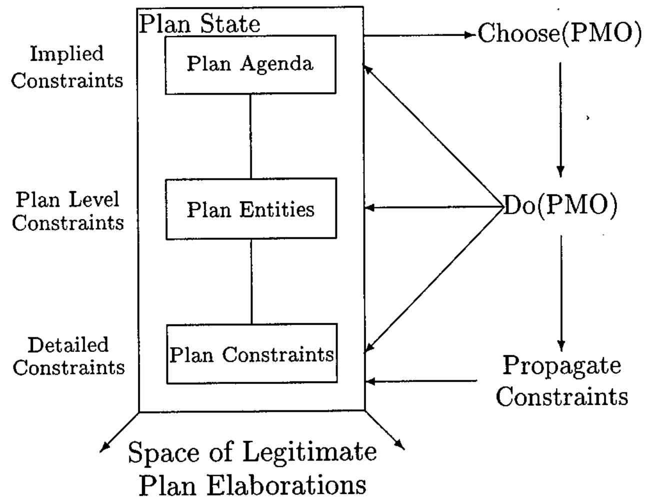


Figure 4: Planning Workflow - Using PMOs to Handle Agenda Issues

constraints or "agenda" as a formal part of the plan provides an ability to separate the plan that is being generated or manipulated from the planning system itself.

3.3 Representing Plans as a Set of Constraints on Behaviour

The <I-N-OVA>[3] (*Issues – Nodes – Orderings / Variables / Auxiliary*) Model is a means to represent and manipulate plans 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 to the environments in which they are generated.

In Tate (1996), the <I-N-OVA> model is used to characterise the plan representation used within O-Plan and is related to the plan refinement planning method used in O-Plan. The <I-N-OVA> work is related to emerging formal analyses of plans and planning. This ¡synergy of practical and formal approaches can stretch the formal methods to cover realistic plan representations as needed for real problem solving, and can improve the analysis that is possible for production planning systems.

<I-N-OVA> is 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 is intended to support new work on automatic manipulation of plans, human communication about plans, principled and reliable acquisition of plan information, and formal reasoning about plans.

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.

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

```
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 5: <I-N-OVA> Constraint Model of Activity

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*[4], 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.

3.4 Communicating Plan Information Between the Task Assignment and Planning Agents

The <I-N-OVA> constraint model of activity allows planning process state as well as the current state of the plan generated to be communicated between agents involved in the planning process. This is done via the Issues part of <I-N-OVA> - which can be used to amend the task and option specific agenda which a planning agent is using for its problem solving. Ways to authorise agents to take initiative in the problem solving process are being explored. This can be done by communicating the types of agenda entry or issue which the planning agent may handle and giving limitations on which types of constraint that may be manipulated and the extent to which they may be manipulated while problem solving.

This involves improving the workflow controller at the heart of the O-Plan planner agent. This will allow dialogue between users and automated planners as the problem solving takes place. Methods to allow for coordination of task and option management between users and the automated planner are being added to O-Plan.

[4]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.

3.5 Authority to Plan

At the moment the Task Assignment agent tells the O-Plan planner when it can create a plan for a nominated task. This is done through a simple mechanism today. As described in Tate (1993) it is intended that O-Plan will support authority management in a more comprehensive and principled way in future. *Changes* of authority are possible via Task Assignment agent communication to the Planner agent. This may be in the context of a current plan option and task provided previously or it is possible to give defaults which apply to all future processing by the planner agent. The authorities may use domain related names that are meaningful to the user and may be refer to the options, sub-options, phases and levels of tasks and plans known to O-Plan.

4 Mutually Constraining Plans for Mixed Initiative Planning and Control

Our approach to Mixed Initiative Planning in O-Plan proposes to improve the coordination of planning with user interaction by employing a clearer shared model of the plan as a set of constraints at various levels that can be jointly and explicitly discussed between and manipulated by the user or system in a cooperative fashion.

The model of Mixed Initiative Planning that can be supported by the approach is *the mutual constraining of behaviour* by refining a set of alternative partial plans. Users and systems can work in harmony though employing a common view of their roles as being to constrain the space of admitted behaviour. Further detail is given in Tate (1994).

Workflow ordering and priorities can be applied to impose specific styles of authority to plan within the system. One extreme of user driven plan expansion followed by system "filling-in" of details, or the opposite extreme of fully automatic system driven planning (with perhaps occasional appeals to an user to take predefined decisions) are possible. In more practical use, we envisage a mixed initiative form of interaction in which users and systems proceed by mutually constraining the plan using their own areas of strength.

Coordination of problem solving must take place between users and the automated components of a planning system. In joint research with the University of Rochester (whose work is described in Allen, Ferguson and Schubert, 1996) we are exploring ways in which the O-Plan controller can be given specific limitations on what plan modifications it can perform, and the specific plan options or sub-options it is working on can be coordinated with those being explored by a user supported by a suitable interface.

5 A Planning Agent on the WWW

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 in given roles to notionally populate a Course of Action (COA) versus Elements of Evaluation comparison matrix. This

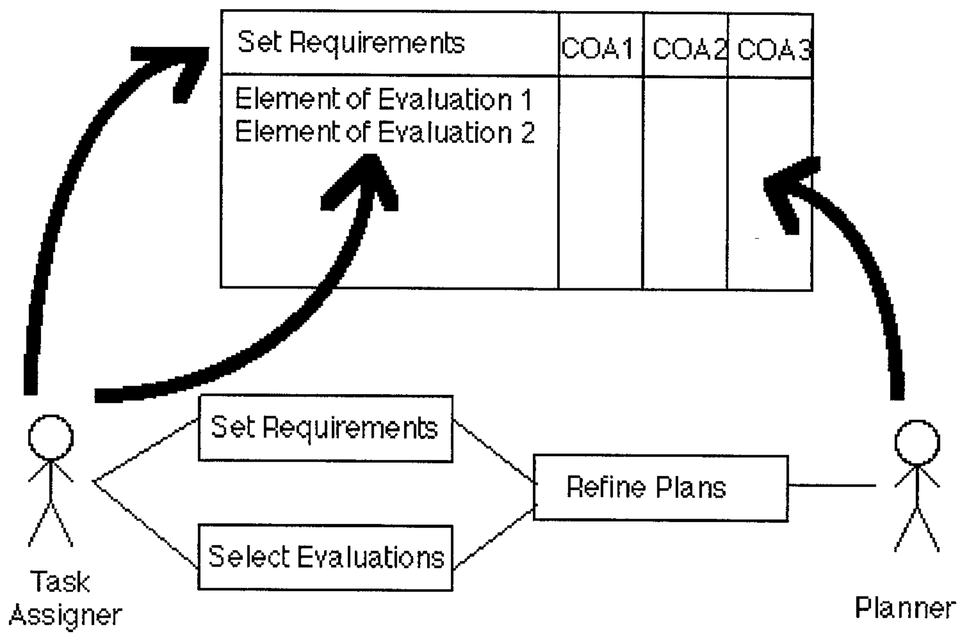


Figure 6: Roles of the Task Assigner and Planner Users

would be used to create a briefing about the alternative courses of action being proposed to meet some set of requirements together with appropriate and differentiating evaluations or advice about the options being proposed.

Figure 6 shows two human agents working together. The Task Assigner sets the requirements for a particular Course of Action (i.e., what top level tasks must be performed) and selects appropriate evaluation criteria (elements of evaluation) for the resulting plans. The Planner agent acts to refine the resulting plans by adding further constraints and splitting plans to explore two or more possible options for the same COA requirements.

The columns of the comparison matrix are alternative options being explored as a potential solution to a (possibly underspecified) problem and the rows are evaluations of the solution being considered and allow for "drilling down" into more detail of the evaluation information. The requirements, assumptions and constraints are all refined concurrently using the elements of evaluation. See the web display of the matrix in figure 7.

We have created a simple web-based demonstration which shows most aspects of the abstract framework described here[5]. The user is initially given a blank COA comparison matrix which is populated by the user and O-Plan during the course of a session (as in figure 7). The user acts in the role of the Task Assigner agent, setting the tasking level requirements for a Course of Action (see figure 8) and selecting elements of evaluation to include in the matrix.

[5]The demonstration is available through URL `http://www.aiai.ed.ac.uk/~oplan/` by following the link to the "Live Demonstrations" page entry for "Pacifica COA Matrix".

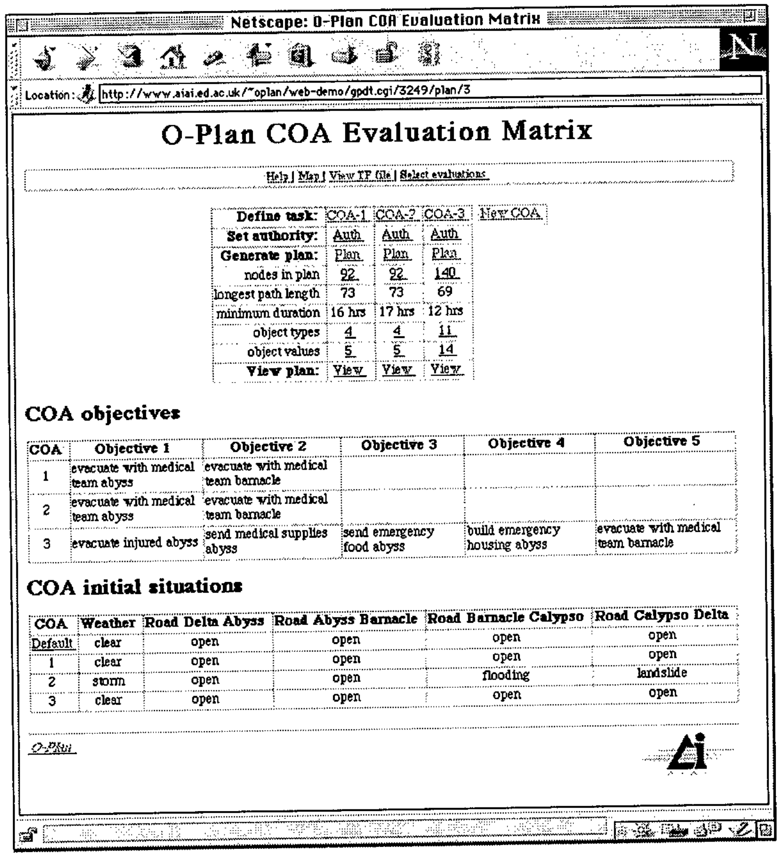


Figure 7: O-Plan running on the web and maintaining a matrix which compares alternative Courses of Action against a set of evaluation criteria

The COA matrix is an abstract underlying notion and may not appear in an actual user interface for a completed system. However, it is useful in this demonstration to show our ideas about what is being created and refined as mixed initiative problem solving takes place.

The two users involved will be collaborating via some suitable collaboration medium. This could be direct interaction if they are in the same room, but more likely will involve video teleconferencing, telephone or net-phone calls, shared displays such as text or whiteboard windows on their computers, or linked web browsers such as are provided in recent web browsers incorporating collaboration facilities. Figure 9 shows the arrangement.

The plan server itself is running on a host computer connected to the world wide web, and is accessed through Common Gateway Interface (CGI) scripts in its current version. Other means of serving commands from the web are available including specialised http servers.

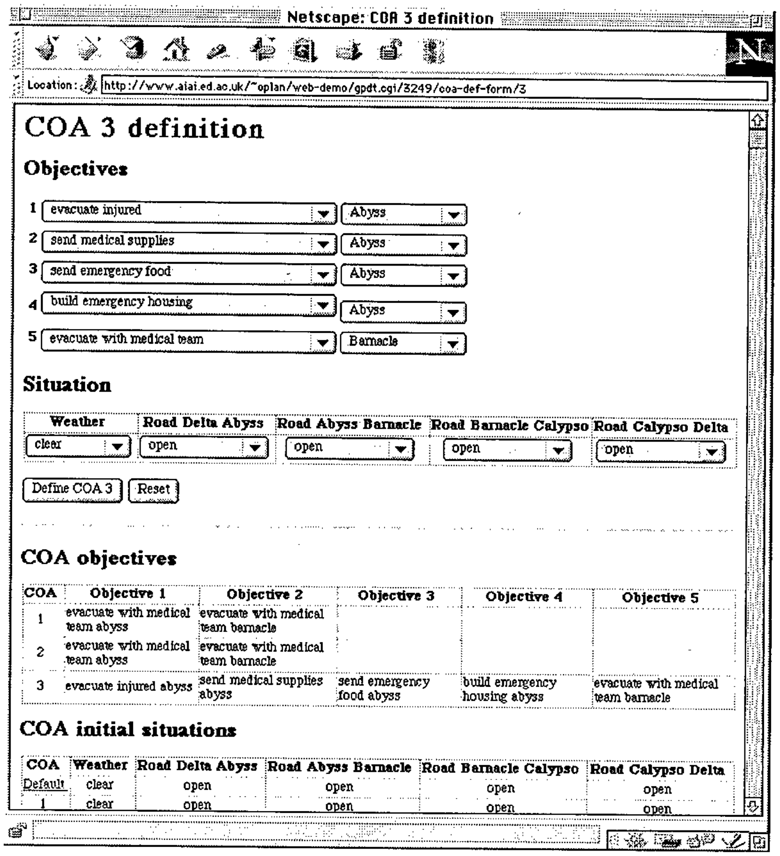


Figure 8: Using forms to set the objectives to O-Plan running on the web

6 Summary

Five concepts are being used as the basis for exploring multi-agent and mixed-initiative planning involving users and systems: Together these provide for a *shared* model of what each agent can and is authorised to do and what those agents can act upon.

1. *Shared Plan Model* – a rich plan representation using a common constraint model of activity (<I-N-OVA>).
2. *Shared Task Model* – Mixed initiative model of "mutually constraining the space of behaviour".
3. *Shared Space of Options* – explicit option management.
4. *Shared Model of Agent Processing* – handlers for issues, functional capabilities and constraint managers.
5. *Shared Understanding of Authority* – management of the authority to plan (to handle issues) and which may take into account options, phases and levels.

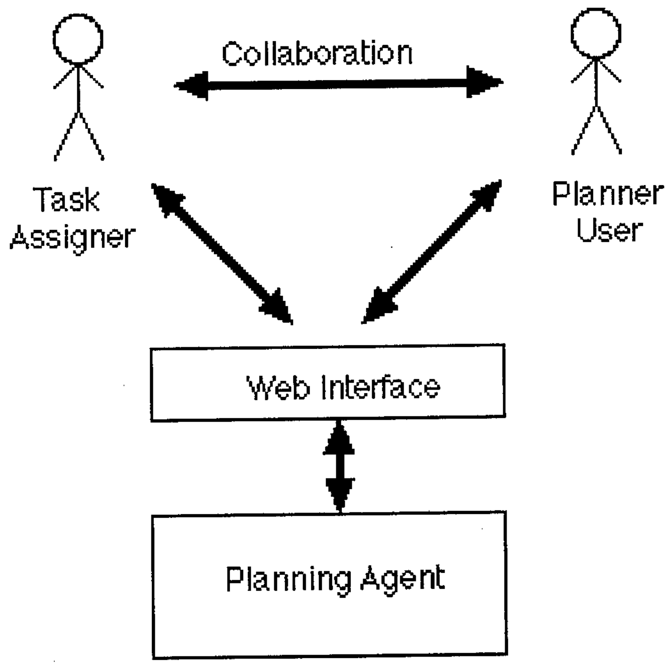


Figure 9: User Collaboration and Shared Use of the Plan Server

Using these shared views of the roles and function of various users and systems involved in a command, planning and control environment, we have demonstrated a planning agent being used to support mixed initiative task specification and plan refinement over the world wide web.

Acknowledgements

The O-Plan project is sponsored by the Defense Advanced Research Projects Agency (DARPA) and Rome Laboratory, Air Force Materiel Command, USAF, under grant number F30602-95-1-0022. The O-Plan project is monitored by Dr. Northrup Fowler III at the USAF Rome Laboratory. The U.S. Government is authorised to reproduce and distribute reprints for Governmental 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, Rome Laboratory or the U.S. Government.

References

Allen, J.F., Ferguson, G.M. and Schubert, L.K. (1996), Planning in Complex Worlds via Mixed-Initiative Interaction, in Advanced Planning Technology, pp. 53-60, (Tate, A., ed.), AAAI Press.

Currie, K.W. and Tate, A. (1991), O-Plan: the Open Planning Architecture, Artificial Intelligence, 51(1), Autumn 1991, North-Holland. Information available at `http://www.aiai.ed.ac.uk/~oplan/`

Drabble, B., Tate, A. and Dalton, J. (1995) Applying O-Plan to the NEO Scenarios, in An Engineer's Approach to the Application of Knowledge-based Planning and Scheduling Techniques to Logistics, Appendix O, USAF Rome Laboratory Technical Report RL-TR-95-235, December 1995. Expanded version available as Drabble, B., Tate, A. and Dalton, J. (1995) O-Plan Project Evaluation Experiments and Results, O-Plan Technical Report ARPA-RL/O-Plan/23 Version 2, 10-Nov-95. Available at `ftp://ftp.aiai.ed.ac.uk/pub/projects/oplan/documents/1995/95-tr-23-experiments.ps`

Fraser, J. and Tate, A. (1995), The Enterprise Tool Set – An Open Enterprise Architecture, Proceedings of the Workshop on Intelligent Manufacturing Systems, International Joint Conference on Artificial Intelligence (IJCAI-95), Montreal, Canada, August 1995. Available at `ftp://ftp.aiai.ed.ac.uk/pub/documents/1995/ 95-ims-ijcai95-ent-toolset.ps`

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. Available at `ftp://ftp.aiai.ed.ac.uk/pub/documents/1993/ 93-ijcai-authority.ps`

Tate, A. (1994), Mixed Initiative Planning in O-Plan2, Proceedings of the ARPA/Rome Laboratory Planning Initiative Workshop, pp. 512-516, (Burstein, M., ed.), Tucson, Arizona, USA, Morgan Kaufmann. Available at `ftp://ftp.aiai.ed.ac.uk/pub/documents/1994/ 94-arpi-mixed-initiative.ps`

Tate, A. (1996a) (ed.), Advanced Planning Technology, AAAI Press.

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), pp. 221-228, (Drabble, B., ed.) Edinburgh, Scotland, AAAI Press. Available at `ftp://ftp.aiai.ed.ac.uk/pub/documents/1996/ 96-aips-inova.ps`. Further information available at `http://www.aiai.ed.ac.uk/~bat/inova.html`

Tate, A., Drabble, B. and Kirby, R. (1994), O-Plan2: an Open Architecture for Command, Planning and Control, in Intelligent Scheduling, (eds, M.Zweben and M.S.Fox), Morgan Kaufmann. Available at `ftp://ftp.aiai.ed.ac.uk/pub/documents/1994/94-is-oplan2.ps`

Tate, A., Drabble, B. and Dalton, J. (1996), A Knowledge-Based Planner and its Application to Logistics, in Advanced Planning Technology, pp. 259-266, (Tate, A., ed.), AAAI Press. Available at `ftp://ftp.aiai.ed.ac.uk/pub/documents/1996/96-arpi-oplan-and-logistics.ps`

Appendix J:

TF Method: An Initial Framework for Modelling and Analysing Planning Domains

Austin Tate, Steve Polyak and Peter Jarvis

Citation:
Tate, A., Polyak, S. and Jarvis, P., TF Method: An Initial Framework for Modelling and Analysing Planning Domains, Workshop on Knowledge Engineering and Acquisition, AIPS-98, Pittsburgh, PA, USA, AAAI Press, 1998.

Purpose:
It is vital to be effective in capturing a model of the domain in which planning takes place, and to ensure that the model can be maintained. Initial work on a methodology and toolset for applying domain modelling, software enmgineering, issue-based reasoning, requirements capture and knowledge engineering principles to planning domain acquisition are described in this paper.

Abstract:
Early work on the NONLIN and O-Plan projects indicated a need for a defined methodology which would guide users performing various roles in the acquisition and analysis of domain requirements for planning. This work included links to a requirement analysis methodology, CORE (COntrolled Requirements Expression), tool support via an intelligent assistant as part of the Task Formalism (TF) Workstation and an initial collection of guidelines and checklists to aid in using the TF domain description language. This paper describes work underway to follow-on from this past research and to infuse it with knowledge gained from recent research related to planning domain development, knowledge modelling, design rationale and ontological and requirements engineering.

1 Introduction

The activities involved in discovering, engineering, documenting, and maintaining a set of domain constructs for most AI planning-based projects can be considered ad hoc and disorganised, at best. The current sources for advice on the process of writing AI planning domain descriptions have been summarised as

> "... it is the most neglected aspect of planning, and there is not an established software-engineering methodology to guide this job". [Erol, 1995]

Domain capture and modelling has been an issue in Edinburgh-based planning research as early as the work on the NONLIN [Tate, 1977] planner. In fact, the original O-Plan overall architecture and system design, which dates from 1983, outlined a need for a defined methodology which would guide users performing various roles in the acquisition and analysis of domain requirements for planning [Currie and Tate, 1991]. This planning life-cycle methodology was envisioned as encompassing a set of standardised activities and methods which had well-defined design criteria, techniques, and tools. This was proposed to assist in transforming planning domain development from a craft towards more of an engineering activity.

The domain description language used by both the NONLIN and O-Plan planners is the Task Formalism (TF) [Tate, 1977; Tate et. al. 1994]. Early prototyping efforts on a PERQ-based TF Workstation [Tate and Currie, 1984, 1985] demonstrated tool-support for the domain modellers (an expert providing the structure of the domain and specialists providing the details) and planners (acting in any one of a range of roles). This tool was designed to include an "intelligent assistant" which would interact with the user via a structured dialogue which was tied to a specific domain development methodology. Research was conducted into a requirements engineering methodology which could be adapted for use in this way. The Controlled Requirements Expression (CORE) [Mullery, 1979; Curwen, 1991] was proposed for structuring these domain management activities. It is hoped that an adaptation of this method, combined with experience in working with TF, would help to drive the development of planning domains in a more reliable fashion.

In this paper, we review past research efforts related to a move towards an overall TF Method framework. This includes a sampling of the guidelines and checklist included in the TF manual, advice on the use of TF condition types, work on prototype tool-support via the TF workstation and past research on links to the CORE methodology. These ideas form a base foundation upon which new efforts from the AI planning related domain modelling research community may be added.

In section 2, we present these components of the initial TF Method. Section 3 mainly reports on experience gained using this work in the development of domain models for the construction industry. A sampling of some of the existing research on domain development tools and approaches from the AI planning community is provided in section 4. Finally, in section 5, we widen the scope and discuss possible links to research in related fields.

2 Towards a TF Method

2.1 Guidelines for Writing TF

Initial work on pulling together the experience gained in coding specific domains in the Task Formalism domain description language resulted in a section on "guidelines for writing TF" which is part of the TF manual [Tate et. al., 1994]. This section provides advice on the use of various TF forms and elements, which can be seen as a start towards a general framework for the development of a methodology which would structure the domain design activities.

This advice is rooted in a project management perspective which describes the need for preparatory steps and uses role identification prior to engaging in the development-oriented activities. The central controlling role was identified as the **Domain Expert** who is in charge of managing the scope of the domain and introducing a "top level" description (e.g. in a house building domain this person might be an architect with overall responsibilities for a project). For large domain engineering efforts, a partitioning of the domain development responsibilities was recommended. These modular sections of the domain were viewed as "detailed" aspects of the top level descriptions which were provided by the domain expert. **Domain Specialists** would then be assigned to particular domain partitions and would have responsibility for providing specifications of activities, objects (resources), events and effects which were relevant to their particular needs. The specialists may be subject matter experts (e.g. in a house building domain they might represent a plumber, or electrician, etc.). More likely, the domain expert and specialists may be knowledge engineers who have performed the required knowledge elicitation and acquisition activities from those with knowledge of the domain.

The guidelines point toward necessary project management decisions such as the choice between one of two "main approaches" toward modelling a domain: hierarchical action expansion or goal achievement (conditions on world states). While these approaches can be mixed in the specification of the domain, experience had shown that it is useful, if not important, to specify what the main approach will be for a particular domain development process and to treat the other approach as secondary to it.

Another important management decision considers the selection of a method for structuring the domain specifications. A level-oriented approach to domain modelling is proposed in this work whereby actions, events, effects, and resources are all separated into a series of defined and increasingly detailed levels. This helps to avoid the commonly experienced problem of "hierarchical promiscuity" [Wilkins, 1988] or "level promiscuity" which is characterised by the inconsistent usage of various domain elements at varying areas in the overall domain description.

This level-oriented approach is further detailed via a checklist of activities which may suit either the action expansion approach or the goal achievement approach, depending on the ordering of the defined activities. This checklist includes the following activities:

- Identify the main actions (and events) that will appear at the top of a task or plan.
- Develop the detailed actions (and events) for lower action levels.

- Think about what world statements will be needed (effects) at which levels.
- Consider the conditions for actions. Ensure they are introduced at a level which is at or below the level at which the related effects are introduced.
- Add type information to restrict usage of conditions. Types are primarily used to differentiate what a condition means. This will lead to differences in which condition satisfaction methods apply. Consult the definitions of TF condition types (see section 2.2).
- Add resources at each level.
- Consider time restrictions and related information.

There are also notes on specific aspects/techniques

- Functional expression of properties
- Conditional actions
- Conditional effects
- Variable typing
- Modelling reusable, non-sharable resources (using conditions and effects)

2.2 TF Condition Types

The guidelines on the use of condition types described in the TF manual were detailed in [Tate, 1994b]. While the advice found in this work is oriented more towards the search effects in the planning system, it has also provided a useful perspective for domain modellers working with levels to constrain the use of condition types. Experience has shown that condition types, such as Supervised, Unsupervised, and Only_use_if map to domain expertise. A verification step, which would take the specific condition types into account, would help to ensure that the modelling levels are valid and that the modeller was not misusing conditions or unsuitable effects by specifying them at the wrong level. This would assist the domain modeller with a careful consideration of the reasons why effects were introduced and conditions placed at a particular level. A consistent, verified model, extracted from this step, would address a major part of the "hierarchical promiscuity" problem.

2.3 CORE (COntrolled Requirements Expression)

COntrolled Requirements Expression (CORE) was a method developed by British Aerospace (Warton) and systems designers in the late 70's [Mullery, 1979]. Over time, the method has evolved and CORE now provides techniques for requirements capture, analysis and specification [Curwen, 1991]. The method can be used to partition problems into manageable modules which can be assessed using CORE analytical techniques. This ensures that the requirements for a specification are complete and consistent. Some of the strengths of this methodology include decomposability of requirements and traceability mechanisms between different levels of requirements.

The CORE specifications are expressed in terms of graphics, structured text and mathematically based notations. These resultant requirements models start from operational requirements which influence functional requirements and, in turn, impact implementation requirements (with non-functional requirements acting as functional and implementation constraints). Viewpoints are used as logical partitionings of the system under consideration. These are divided into **bounding viewpoints**, which can be viewed from a planning context as providers of unsupervised conditions and **defining viewpoints** which are analogous to activities which can achieve supervised conditions. Viewpoint decompositions correspond to node expansions. The CORE notion of "scope" addresses choices between elements which may be included in the domain, and breaks them down into "local scopes" which designate responsibilities for domain specialists.

It is envisaged that an adaptation of the CORE methods can be used to structure the activities of users acting in particular roles throughout the life-cycle of a domain. For example, a domain expert divides a domain into a series of tasks to be completed by specialists. A domain specialist can list the assumptions he/she will be making (e.g. walls have been built and foundation laid). Specialists can retrieve previous plans to modify. For each plan, a viewpoint decomposition process is applied to it. This includes some checking based on CORE analysis techniques:

- Does every node have at least one precursor?
- For every node which has a precondition, is the precondition satisfied by the current network or by another node at the same level or higher?
- Do precursor and successor assignments match?

CORE provides specialised techniques for inspecting the evolving specification/domain. One example is the "viewpoint to viewpoint role-playing" technique. Using this approach, a structured document is produced which defines a particular perspective within the domain (e.g. between a builder and a floor installation procedure, or between a carpet layer and a floor installation procedure, etc.) Techniques such as this one aid in combining the viewpoints by showing where conflicting requirements are present. CORE has been used previously as the controlling methodology for an expert system-based requirements analysis tool [Stephens and Whitehead, 1984][1]. This tool utilised knowledge of CORE via stored relations, entities, rules, and could answer questions related to a requirement specification such as: how, why, and why not.

Future work will seek to adapt the CORE methodology and to provide tool-based support for it in the structuring of planning domain development activities.

[1] Joint work with the O-Plan team in the mid 1980s explored the use of O-Plan as a planning assistant within the Analyst Workbench

2.4 TF Workstation

The original O-Plan design described the development of an intelligent, graphical user interface between an AI planning system and its users. This tool was called the TF workstation [Tate and Currie, 1984][2]. The users of the TF workstation were separated into: those who describe the application domain; and those who require plans to operate within the domain.

During domain building, the workstation assisted in building up the details of the alternative actions possible in the domain, the resource or time constraints on the actions, and the ways in which actions can be combined, etc. In this role, it could communicate with a domain expert and possibly several domain specialists to elicit their knowledge about the applications domain.

The TF workstation also acted as the interface between a human planner and the AI planner. In this role, which can be thought of as a coordination activity, the workstation sought details of the task for which a plan is required. It checked that sufficient domain knowledge was available to enable a solution to be found (if necessary, the system pointed omissions out to the human planner, domain expert, or domain specialist) and acted as an intermediary to enable the human and AI planners to jointly generate a valid plan.

A hook for an expert system-style agent interface was provided to perform various services such as searching for close matches for terminological differences or incomplete information, etc. Preliminary work on the use of the CORE methodology within the TF workstation was performed [Wilson, 1984]. Unfortunately, this research was set aside once the initial prototype was completed. Research is currently underway to extend the original TF workstation/methodology ideas as part of the Common Process Editor (CPE) which is a component in a framework for applying AI planning to manufacturing, military and business process management[3].

2.5 TF Compiler

The O-Plan TF compiler converts the Task Formalism language (coming from a file or from a domain editing tool) into the internal domain information used by the O-Plan planner. The compiler can be run incrementally and will add to or modify the existing domain information available to the planner. It is anticipated that facilities to change specific aspects of forms previously submitted will be provided, along with the current facility of simply replacing old forms or adding new ones. There is an interaction between the facilities provided by the compiler and the possible activities performed in a domain life-cycle methodology. Future work on a richer interface to the TF compiler will facilitate steps in domain knowledge management which may overlap with planning, replanning, execution, etc.

[2] An example screen shot from the TF workstation is shown in appendix A

[3] An example screen shot from the Common Process Editor (CPE) is shown in appendix A

3 TF Method Experience

3.1 Domain Description Development for the Construction Industry

The initial TF Method components were used during a research project at The University of Brighton to guide the development of a TF encoding of planning knowledge elicited from the construction industry [Jarvis, 1997; Jarvis and Winstanley, 1996a, 1996b, 1998]. This section outlines this work to relate industrial experiences of the TF Method from individuals who were at the time independent of the O-Plan design team.

3.2 Planning the Development of a Domain Description

The first stage of the TF Method calls for a planned approach to the development of a domain description. It advises the identification of an overall domain expert to scope and structure the domain and a number of domain specialists to "fill-in" the structure with detailed knowledge. This approach worked well in the construction industry. The senior director used in the role of domain expert provided an overview of the planning process. Managers further down the organisation used in the role of domain specialist provided detailed knowledge about the areas in which they work and their interactions with other specialists.

The different views of the domain expert and domain specialists complemented one another. The expert understood the overall process and the relationships between each stage but not the detail of how each stage was performed. The specialists understood the detail of their area but not the complete context in which they worked. Reconciling these two views added a beneficial cross check to the modelling process. Mismatches were traced to one of two causes. Either the knowledge engineer had misunderstood a specialist's or expert's comments or an organisational problem had been encountered. In the former case, the mismatch provided a useful tool for prompting both specialist and expert to clarify their comments. In the latter case, the mismatch motivated the specialist and the expert to meet and clarify their perceptions of the actual planning process they engaged in.

3.3 Selecting between Action Expansion and Goal Achievement

The second stage of the TF Method recommends a conscious commitment to either action expansion or goal achievement as the primary modelling approach to a domain. Experimental modelling with both approaches was used to inform this decision. This experimentation categorised planning knowledge in the construction industry as being structured around the components of a building and the trades or specialists used to construct related groups of these components. A plumber, for example, is responsible for the installation of a building's bathroom fittings and a scaffolder is responsible for erecting the scaffolding that supports bricklayers in the task of constructing walls. This structure was readily mapped to the hierarchy of schemata inherent in the action expansion approach. Considering the earlier example, the overall task of installing a building's services was encapsulated within a single schema. This schema then refined to two schemas at a lower modelling level with the first

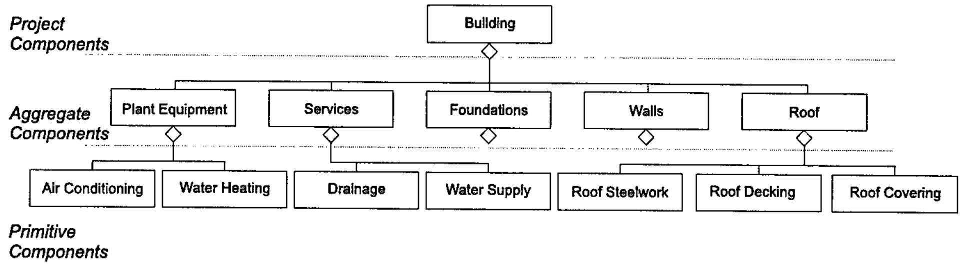


Figure 1: Construction Domain Model Partitioned into Modelling Levels

describing the work of the plumbing specialist and the second the electrician specialist.

This experience with the Task Formalism provides evidence to support Drummond's thesis [Drummond, 1994] that industrial planning problems are more readily addressed by action expansion than by goal achievement techniques.

3.4 Developing the TF Schemata

The third stage of the TF Method suggests that each schema expansion level should hold some meaning within the domain under consideration. Figure 1 shows a section of the building subcomponent hierarchy developed from the meetings with domain experts. In the figure, a building is shown as being decomposed into a number of subcomponents (Plant Equipment through to a Roof). These components are decomposed further until the level of detail required for producing a construction plan is reached. This structure allows experts to reason at different levels of abstraction. The assignment of the construction of the roof component to a contractor would, for example, assume that the contractor would then be responsible for the construction of all the roof's subcomponents (Roof Steelwork, Roof Decking and Roof Covering).

Part of the TF encoding of the components in figure 1 is shown in figure 2. Figure 1 is partitioned through the dashed horizontal lines into the modelling levels: project components, aggregate components, and primitive components. These modelling levels were used to guide the encoding process. Considering the schemata in figure 2, the building component at the project modelling level in figure 1 is encoded as the initial task plan_buildings_construction. The transition from the project modelling level to the aggregate modelling level via the subcomponent relationship is achieved in the TF encoding through the schema refinement mechanism. The schema build_building refines the initial task (plan_buildings_construction) and it introduces a node for each subcomponent of the building that resides within the aggregate modelling level. The transition from the aggregate modelling level to the primitive modelling level is also achieved through schema refinement as demonstrated by the schema erect_roof. The schema, which will be used to refine node 2 in the build_building schema, contains an action for each subcomponent of the Roof component.

The encoding shown in figure 2 preserves both the subcomponent structure and the modelling levels elicited from the domain. Figure 3 shows part of the subcomponent structure from figure 1 augmented with the scope of the schemata that describe that structure in the TF encoding. The dashed lines represent modelling levels and the dotted lines the scope of each schema.

```
task plan_buildings_construction; ;; modelling level project componets
  nodes 1 task {build ?building};
end_task;

schema build_building; ;; modelling level aggregate components
   expands {build ?building}
   nodes 1 action {lay ?foundations},
         2 action {erect ?roof},
         3 ...
end_schema;

schema erect_roof; ;; modelling level primitive components
   expands {erect ?roof};
   nodes 1 action {install ?roof_steelwork},
         2 action {lay ?roof_decking},
         3 action {lay ?roof_covering};
end_schema;
```

Figure 2: Schemas build_building and erect_roof

As advised by the TF Method, the effects produced by actions were considered before the conditions required by actions. Each component was considered to determine the effect(s) that would result from its construction. The components at the higher modelling levels produce effects that describe the overall result of constructing their subcomponents. Constructing The Foundations component, for example, adds the effect {State_Of Foundations} = laid. The components at the lower modelling levels produce effects that describe their own construction. Constructing the Roof Steelwork component, for example, adds the effect {State_Of _Roof_Steelwork} = erected. Figure 3 positions each effect within the same modelling level as the component which will produce it. Figure 4 shows how the schemata developed in figure 2 were modified to include these effects. The newly added TF elements in figure 4 are highlighted in bold.

Figure 3 contains both the effect levelling and schema scope information that is required to follow the guidelines on encoding action conditions described in [Tate, 1994b]. Consider the roof decking component in figure 3. This component requires the roof steelwork to be in place before its own construction is started as the roof steelwork supports it. The scope and levelling information in figure 3 informs us that both the roof steelwork and the roof covering are introduced by the same schema. An ordering constraint and a supervised condition may therefore be placed between the actions to describe this. This encoding is shown in figure 4 within the erect_roof schema as the ordering constraint "1 $\rightarrow$ 2" and the condition "supervised {State_of Roof_Steelwork} = installed at 2 from [1]". The levelling information shown in figure 3 informs us that the actions are at the same modelling level. This situation conforms to the guideline that a supervised condition must be placed at the same or at a lower modelling level than the effect that satisfies it.

Consider the arrow between the roof steelwork and pile components within figure 3. The arrow is depicting the knowledge that the roof steelwork is supported by the pile. Figure 3 shows that these components are described in different schemas. Hence, an unsupervised condition must be used to describe the relationship. This knowledge is encoded within the schema erect_roof as the condition "unsupervised {State_Of Pile} = laid at [1]".

```
schema build_building; ;; modelling level aggregate components
  expands {build ?building}
  nodes 1 action {lay ?foundations},
        2 action {erect ?roof},
        3 ...
  only_use_for_effects
        {state_of foundations} = laid at 1,
        {state_of roof} = erected at 2.
        ...
end_schema;

schema erect_roof; ;; modelling level primitive components
  expands {erect ?roof};
  nodes 1 action {install ?roof_steelwork},
        2 action {lay ?roof_decking},
        3 action {lay ?roof_covering};
  orderings 1-->2;
  condtions
        supervised {state_of roof_steelwork} = installed at 2 from [1],
        unsupervised {state_of pile} = laid at [1];
  only_use_for_effects
        {state_of roof_steelwork} = installed at 1,
        {state_of roof_decking} = installed at 2,
        {state_of roof_covering} = laid at 3;
end_schema;
```

Figure 4: Schemas build_building and erect_roof augmented with action conditions and effects

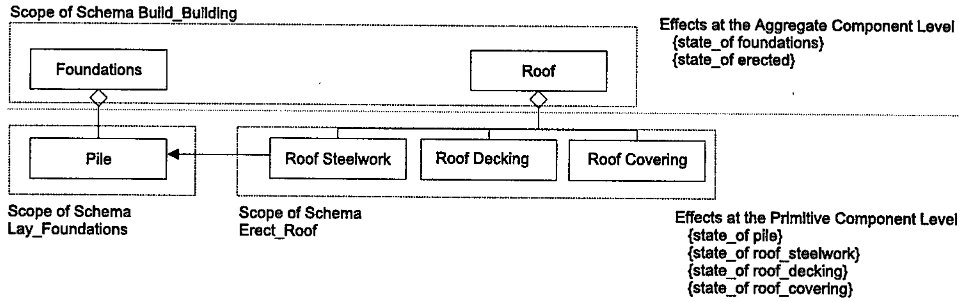


Figure 3: Schema Scope, Effects and Condition Types

3.5 Conclusions Drawn from this Experience

The TF Method provided a principled set of guidelines that aided the development of a TF representation of an aspect of the construction industry. The division of domain experts into the roles of expert and specialist mapped to the different views on planning knowledge that were held by people working in the domain. Reconciling these views provided a useful cross check that encouraged the knowledge engineer to clarify knowledge as it was elicited and domain experts to meet to clarify their own understandings of their domain.

The method made clear the importance of mapping schema expansions to modelling levels within the domain and it provided guidelines for ensuring the appropriate positioning of action conditions and effects within those levels. These guidelines assisted the development of a principled model of the domain.

The weakness of the method is the absence of tool support. The knowledge engineer must use pencil and paper to construct and maintain the figures shown in this section. Tools can be provided to automatically show the scope of schemas, highlight the levels of effects relative to a particular condition, and warn the knowledge engineer when the guidelines for relating condition types to modelling levels are violated.

3.6 Common Process Editor

Recent research on a Common Process Framework (CPF) is seeking to facilitate process management in a business and manufacturing application using AI planning representations. This framework includes tool support via a Common Process Editor (CPE) which acts as both the process visualisation and domain management tool for users. An example screen shot from the Common Process Editor (CPE) is shown in appendix A. Connection to an intelligent planning agent (e.g. O-Plan) allows for system-supported generation of business and manufacturing processes. A Common Process Assistant (CPA), which is also accessed as an agent, is used to perform analyses of the processes.

The language used to communicate between the CPE and the planning agent is currently the Task Formalism. This has provided us with insights into the use of TF as part of an integrated process management system. A defined TF Method may be adapted for use in structuring of activities related to the design, modelling, and maintenance of these processes. This tool-based assistance will help address the missing support mentioned in section 3.5.

4 Related Domain Research

A number of recent efforts in the AI planning research community have produced a variety of representations, approaches, tools, and architectures for working with AI planning domains. These range from machine learning approaches to user-based knowledge acquisition tools. This section samples some of the scope of ideas which may be utilised to provide a more effective methodology. We briefly present each approach in terms of its contribution and then discuss some of the possible issues.

4.1 A Formalisation of HTN Planning [Erol, 1995]

- *Contributions*: Formal representation of Hierarchical Task Network (HTN) planning that gives a clear understanding of what the different constructs and condition types mean. This gives the knowledge engineer a formal underpinning which they may consult to clarify precisely the operation of different facets of an HTN planner and how the constructs supported by HTN representational devices affect this operation. This work also presents a list of steps to follow when encoding a domain description.
- *Issues*: The work is only accessible to AI planning specialists and cannot be readily understood by domain experts. It does, however, provide a foundation for understanding HTN planning that planning specialists can use to guide them in the writing of user oriented methods like the guidelines in the TF manual.

4.2 An Object-centred Specification Approach [McCluskey and Porteous, 1997]

- *Contributions*: The authors seek to provide support for constructing planning domain descriptions by adapting methodological steps and notations of the object-oriented community. This approach utilises the notion of "lifting" domain representation from the level of the literal to the level of the object. Once a domain has been described in terms of a state transition graph, the author's algorithms compile the diagram into a STRIPS [Fikes and Nilsson, 1971] style action representation.
- *Issues*: This work assumes that a domain can be described as a state transition graph (STG). The technique cannot currently generate HTN representations. This might be possible if it is extended to include techniques which use hierarchies of STGs. However, there does not appear to be a mechanism for inferring condition types.

4.3 Domain Analysis Techniques and Tools [Chien, 1996]

- *Contributions*: Chien provides two types of tools for planning knowledge base development: static KB analysis techniques to detect certain classes of syntactic errors and completion analysis techniques to iteratively debug the planning knowledge base. This tool set supports typical user questions when investigating these types of error.
- *Issues*: The tool set can only be used after a significant proportion of a domain description has been elicited. It doesn't directly address how this initial description is to be constructed. Some AI planners may already perform such forms of domain checking during domain compilation.

4.4 Automatically Learning Operators [Wang, 1996]

- *Contribution*: Takes a set of example plans described in terms of the actions in each plan and the state of the world before and after each action. The system examines these examples and generates the preconditions and effects of operator descriptions.

- *Issues*: The technique assumes that the user can provide example plans described in terms of the state of the world before and after each action. It provides no assistance for the construction of these example plans. Again, the technique is only applicable to STRIPS style planning not HTN.

A number of other contributions from the AI planning community may be useful sources for the development of the TF Method as well. These works include architectures, such as the EXPECT knowledge acquisition architecture [Swartout and Gil, 1996] which dynamically forms expectations about the knowledge that needs to be acquired by the system and then uses these expectations to interactively guide the user through the knowledge acquisition process. There are are also specialised techniques, for example, knowledge acquisition on the fly (i.e. during planning) [desJardins, 1996] and tools for editing operators and domain knowledge (e.g. Act editor [Myers and Wilkins, 1997], Operator editor [desJardins, 1996], etc.).

5 Integrating with Other Research Areas

An increasing number of requirements are being placed on both domain representations and the processes in which these artifacts are created, maintained etc. as we forge ahead toward future implementations of artificial intelligence planning systems. Domain development methods require solid modelling techniques and well-defined, accepted concepts and terminology. Aspects of the domain may be linked to a specific set of possibly dynamic requirements. Modifications to the domain throughout its life-cycle may require contextual knowledge which expresses the rationale for particular domain design decisions.

Some of these issues facing applied planning efforts are being addressed by related research areas. These areas may provide sources of techniques, methods and guidelines which can be combined with AI domain development approaches to provide a more robust methodology. We briefly outline four possible research areas: knowledge modelling, ontology engineering, requirements engineering, and design rationale.

5.1 Knowledge Modelling

Several approaches have been developed to tackle AI planning problems [Allen et. al. 1990]. While the result is a rich corpus of techniques and methods, it is proving to be a very difficult task to compare and contrast each approach. Some researchers believe the best way is to chart these results with detailed algorithmic treatment [Kambhampati et. al., 1995]. Barros, Valente, and Benjamins present a differing perspective whereby the focus is on an abstract analysis which highlights the capabilities of the system and the way it represents and uses knowledge [Barros et. al. 1996].

This knowledge modelling research utilises the CommonKADS [Wielinga et. al. 1992; Brueker and van de Velde, 1994] methodology which outlines a set of detailed models to be created for an analysis. The AI planning community has gained a more informed perspective on the ways knowledge is used in various planning systems via the application of these methods [Jarvis, 1997; Barros et. al. 1996; Kingston et. al. 1996; Cottam et. al. 1995; Valente, 1995]. A hybrid domain life-cycle methodology that integrates these model-building techniques along with the current methods and guidelines from AI planning domain development could aid in lifting the domain engineers level of interaction with the domain and improve the overall construction process.

5.2 Ontology Engineering

Planning domain ontologies are specifications of the concepts, terms, relations, etc. that form the basic language used to describe a domain [Valente, 1995]. These specifications or definitions, expressed either informally or formally [Uschold and Grunninger, 1996], help to clarify the semantics of the planning domain concepts. Domain ontologies, along with domain-independent ontologies (cf. [Tate, 1996a, 1996b]), characterise elements in the planning world model separately from any particular system that is reasoned about (generative planning system, plan evaluation system, etc.). Shared domain ontologies (i.e. two or more systems/groups agree to defined terminology) assist in breaking down some of the arbitrary differences at the knowledge level and facilitate knowledge sharing [Neches et. al. 1991].

A methodology which seeks to address the construction of plan domain models in an environment where knowledge sharing is required must somehow be connected or combined with a methodology for building a shared domain ontology. Recent ontological engineering research has begun to address the design and development of such methodologies [Fernández et. al., 1997; Gómez–Pérez et. al., 1996; Mizoguch et. al., 1995]. For example, Gómez-Pérez et. al. propose the following set of phases [Gómez–Pérez et. al., 1996]

- Acquire Knowledge
- Build a requirements specification document
- Conceptualise the ontology
- Implement the ontology
- Evaluation during each phase
- Documentation after each phase

Some researchers propose general guidelines or techniques, such as a "middle-out" approach [Uschold and Grunninger, 1996] in which a glossary of terms is used to define an initial set of primitive concepts which, in turn, are used to define new ones. Other researchers propose more domain-specific approaches such as the ontology building process utilised for disturbance diagnosis and service recovery planning in electrical networks [Bernaras et. al., 1996]. Techniques developed in these projects may be candidates for integration into a planning domain development tool-box.

5.3 Requirements Engineering

Significant work in requirements engineering has been made since the early O-Plan research into adopting the CORE methodology for use in planning domain development. This includes work on viewpoint management and stake-holder analysis [Easterbrook and Nuseibeh, 1996; Kotonya and Sommerville, 1996; Finkelstein et. al., 1994], as well as work on various methodologies, techniques, and guidelines [Sommerville and Sawyer, 1997; van Lamsweerde and Letier, 1998] for eliciting, recording, and managing requirements.

Connecting domain aspects to their underlying requirements may assist in managing domain modifications which are the result of changing needs of an organisation. Clearly defined roles and responsibilities at the requirement level will help to organise the activities at the domain level. This will help to address one of the major impediments which has prevented the adoption of AI planning tools and techniques in applied settings: a lack of organisational context.

5.4 Design Rationale

A design rationale is a representation of the reasoning behind the design of a system [Shum, 1991]. It is essentially the explicit recording of the issues, alternatives and justifications that were relevant to elements in the design of an artifact. Examples of design rationale implementations include: QOC [MacLean et. al. 1991], DRL [Lee, 1990], gIBIS [Conklin and Begeman, 1988].

Large plan domains utilised within organisations can be viewed as complexly designed artifacts. These artifacts are managed, reviewed, and maintained just as information systems are. A methodology which encompasses the development of such artifacts may need to support the recording and replay of the rationale for the decisions taken during its design. In a recent review of planning rationale, we described a method for incorporating design rationale in planning [Polyak and Tate, 1998]. We believe that the benefits of a design rationale approach [Moran and Carroll, 1996], will aid in the reasoning, analysis and communication of planning domain knowledge.

6 Summary

This paper has presented perspectives on an initial framework which will assist in the process of modelling and analysing planning domains. These perspectives are based on past and present research efforts in: the TF guidelines; TF workstation; and integrating with a requirements engineering methodology, experience acquired in working with TF domains and insights gained through the other efforts of planning, knowledge modelling, ontology and requirements engineering, and design rationale research groups. We believe that a synthesis of the techniques and methods found in these works will be essential for improving the quality of AI planning domain management throughout its organisational life-cycle.

Acknowledgements

The O-Plan project is sponsored by the Defence Advanced Research Projects Agency (DARPA) and the U.S. Air Force Research Laboratory (AFRL), under grant number F30602-95-1-0022. The O-Plan project is monitored by Dr. Northrup Fowler III at AFRL (Rome). One author is sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant number F49620-96-1-0348 – an AASERT award monitored by Dr. Abe Waksman and associated with the O-Plan project. The U.S. Government is authorised to reproduce and distribute reprints for Governmental 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, AFRL or the U.S. Government.

Bibliography

Allen, J.; Hendler, J.; and Tate, A., eds. (1990) *Readings in Planning.* Palo Alto, CA: Morgan Kaufmann.

Barros, L.; Valente, A.; and Benjamins, R. (1996) Modeling planning tasks. In *Proceedings of the Third International Conference on Artificial Intelligence Planning Systems (AIPS-96)*, 11–18. Edinburgh, Scotland: Morgan Kaufmann.

Bernaras, A.; Laresgoiti, I.; and Corera, J. (1996) Building and reusing ontologies for electrical network applications. In *Proceedings of the European Conference on Artifical Intelligence (ECAI) '96*, 298–302.

Breuker, J., and van de Velde, W. (1994) *The CommonKADS Library for Expertise Modelling: reusable components for artificial problem solving.* Amsterdam, Tokyo: IOS Press.

Chien, S. (1996) Static and completion analysis for planning knowledge base development and verification. In [Drabble, 1996], 53–61.

Conklin, E., and Begeman, M.L. (1988) gIBIS: A hypertext tool for explanatory policy discussion. *ACM Transactions on Office Information Systems* 6:303–331.

Cottam, H.; Shadbolt, N.; Kingston, J.; Beck, H.; and Tate, A. (1995) Knowledge level planning in the search and rescue domain. In *Research and Development in Expert Systems XII, proceedings of BCS Expert Systems'95.*

Currie, K., and Tate, A. (1991) O-Plan: the open planning architecture. *Artificial Intelligence* 52:49–86.

Curwen, P. (1991) System development using the CORE method. Military Aircraft Ltd. BAe/WIT/ML/GEN/SWE/1227, British Aerospace, PLC, Warton Aerodrome, Preston, UK.

desJardins, M. (1996) Knowledge acquisition tools for planning systems. In [Tate, 1996], 53–61.

Drabble, B., ed. (1996) *Proceedings of the Third International Conference on Artificial Intelligence Planning Systems (AIPS-96).* Edinburgh, Scotland: Morgan Kaufmann.

Drummond, M. (1994) On precondition achievement and the computational economics of automated planning. In Backstrom, C., and Sandewall, E., eds., *Current Trends in AI Planning.* IOS Press. 6–13.

Easterbrook, S., and Nuseibeh, B. (1996) Using viewpoints for inconsistency management. *Soft. Engin. Journ.* January.

Erol, K. (1995) *Hierarchical Task Network Planning: Formalisation, Analysis, and Implementation.* Department of computer science, University of Maryland, College Park, USA.

Fernández, M.; Gómez-Pérez, A.; and Juristo, N. (1997) Methontology: From ontological art towards ontological engineering. In *Workshop on Ontological Engineering, Spring Symposium Series, AAAI97.*

Fikes, R., and Nilsson, N. (1971) STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence* 2:189–208.

Finkelstein, A.; Gabbay, D.; Hunter, A.; Kramer, J.; and Nuseibeh, B. (1994) Inconsistency handling in mulit-perspective specifications. *Trans Software Eng* 20(8):569–578.

Gómez-Pérez, A.; Fernández, M.; and Vicente, A. D. (1996) Towards a method to conceptualize domain ontologies. In *Workshop on Ontological Engineering, ECAI'96*, 41–51.

Jarvis, P., and Winstanley, G. (1996a) Dynamically assessed and reasoned task (DART) networks. In *Proceedings of the Sixteenth Annual Technical Conference of the British Computer Society Specialist Group on Expert Systems (ES-96), Cambridge, UK.*

Jarvis, P., and Winstanley, G. (1996b) Objects and objectives: the merging of object and planning technologies. In *Proceedings of the Fifteenth Workshop of the UK Planning and Scheduling Special Interest Group, Liverpool, UK.*

Jarvis, P., and Winstanley, G. (1998) Reducing the semantic gap between application domains and AI planning technology: a compilation based approach. In *Workshop on Knowledge Acquisition and Knowledge Elicitation, to be held within the Fourth International Conference on Artificial Intelligence Planning Systems, Pittsburgh, USA.*

Jarvis, P. (1997) *Integration of Classical and Model-Based Planning.* PhD thesis, School of Computing and Mathematical Sciences, University of Brighton, Sussex, UK.

Kambhampati, S.; Knoblock, C.; and Q., Y. (1995) Planning as refinement search: a unified framework for evaluating design tradeoffs in partial–order planning. *Artificial Intelligence* 76.

Kingston, J.; Shadbolt, N.; and Tate, A. (1996) CommonKADS models for knowledge based planning. Artificial Intelligence Application Institute AIAI-TR-199, University of Edinburgh, Edinburgh, Scotland.

Kotonya, G., and Somerville, I. (1996) Requirements engineering with viewpoints. *Soft. Engin. Journ.* 11(1).

Lee, J. (1990) SIBYL: A qualitative decision management system. In Winston, P., and Shellard, S., eds., *Artificial Intelligence at MIT: Expanding Frontiers.* MIT Press. 104–133.

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*, 720–730.

McCluskey, T., and Porteous, J. (1997) Engineering and compiling planning domain models to promote validity and efficiency. *Artificial Intelligence* 95(1):1–65.

Mizoguchi, R.; Vanwelkenhuysen, J.; and Ikeda, M. (1995) Task ontology for reuse of problem solving knowledge. In *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing.* IOS Press. 46–59.

Moran, T., and Carroll, J., eds. (1996) *Design Rationale: Concepts, Techniques, and Use.* Lawrence Erlbaum Associates.

Mullery, G. (1979) CORE: A method for controlled requirements specification. In *Proceedings of the 4th International Conference on Software Engineering.*

Myers, K., and Wilkins, D. (1997) The act-editor user's guide: A manual for version 2.2. SRI International Artificial Intelligence Center, Stanford University, Menlo Park, CA.

Neches, R.; Fikes, R.; Finin, T.; Gruber, T.; Patil, R.; Senator, T.; and Swartout, W. (1991) Enabling technology for knowledge sharing. *AI Magazine* Fall.

Polyak, S., and Tate, A. (1998) Rationale in planning: Causality, dependencies, and decisions. *Knowledge Engineering Review* 13(2):1–16.

Shum, S. (1991) Cognitive dimensions of design rationale. In Diaper, D., and Hammond, N., eds., *People and Computers VI*, 1–13. Cambridge: Cambridge University Press.

Sommerville, I., and Sawyer, P. (1997) *Requirements Engineering: A Good Practice Guide.* John Wiley and Sons.

Stephens, J., and Whitehead, R. (1984) The analyst – an expert system approach to requirements analysis. In *Proceeding of the third seminar on Application of Machine Intelligence to Defence Systems.*

Swartout, W., and Gil, Y. (1996) EXPECT: A user-centered environment for the development and adaptation of knowledge-based planning aids. In [Tate, 1996], 250–258.

Tate, A., and Currie, K. (1984) The O-Plan task formalism workstation. Artificial Intelligence Applications Institute (AIAI) AIAI-TR-7, University of Edinburgh.

Tate, A., and Currie, K. (1985) The O-Plan task formalism workstation. In *Proceedings of the Third Workshop of the UK Alvey Programme's Planning Special Interest Group.* London, UK: Institute of Electrical Engineers.

Tate, A.; Drabble, B.; and Dalton, J. (1994a) Task formalism manual. Artificial Intelligence Applications Institute AIAI-TF-Manual, University of Edinburgh, Edinburgh, UK ftp://ftp.aiai.ed.ac.uk/pub/documents/ANY/oplan-tf-manual.ps.gz.

Tate, A.; Drabble, B.; and Dalton, J. (1994b) The use of condition types to restrict search in an AI planner. In *Proceedings of Twelfth National Conference on AI (AAAI-94), Seattle.*

Tate, A. (1977) Generating project networks. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-77)*, 888–893.

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

Tate, A. (1996b) Representing plans as a set of constraints – the < I-N-OVA > model. In [Drabble, 1996], 221–228.

Tate, A. (1996c) Towards a plan ontology. *AI*IA Notiziqe (Publication of the Associazione Italiana per l'Intelligenza Artificiale), Special Issue on Aspects of Planning Research* 9(1):19–26.

Uschold, M., and Gruninger, M. (1996) Ontologies: Principles, methods and applications. *Knowledge Engineering Review* 11(2).

Valente, A. (1995) Knowledge-level analysis of planning systems. SIGART *Bulletin* 6(1).

van Lamsweerde, A., and Letier, E. (1998) Integrating obstacles in goal-driven requirements engineering. In *Proceedings (ICSE'98) - 20th International Conference on Software Engineering, (IEEE-ACM).*

Wang, X. (1996) Planning while learning operators. In [Drabble, 1996], 229–236.

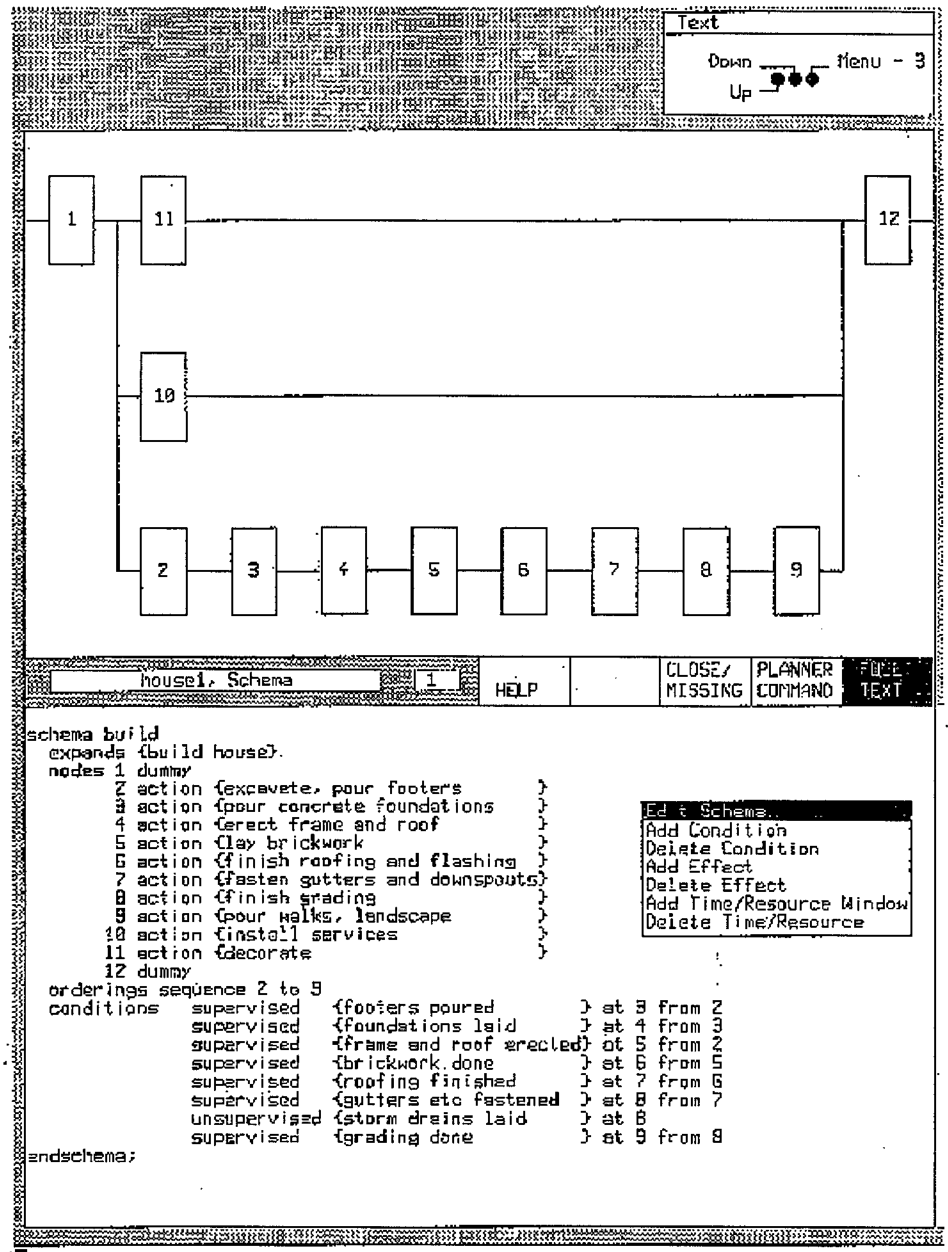
Wielinga, B.; van de Velde, W.; Schriber, G.; and Akkermans, H. (1992) The KADS knowledge modelling approach. In *Proceedings of the Japanese Knowledge Acquisition Workshop.*

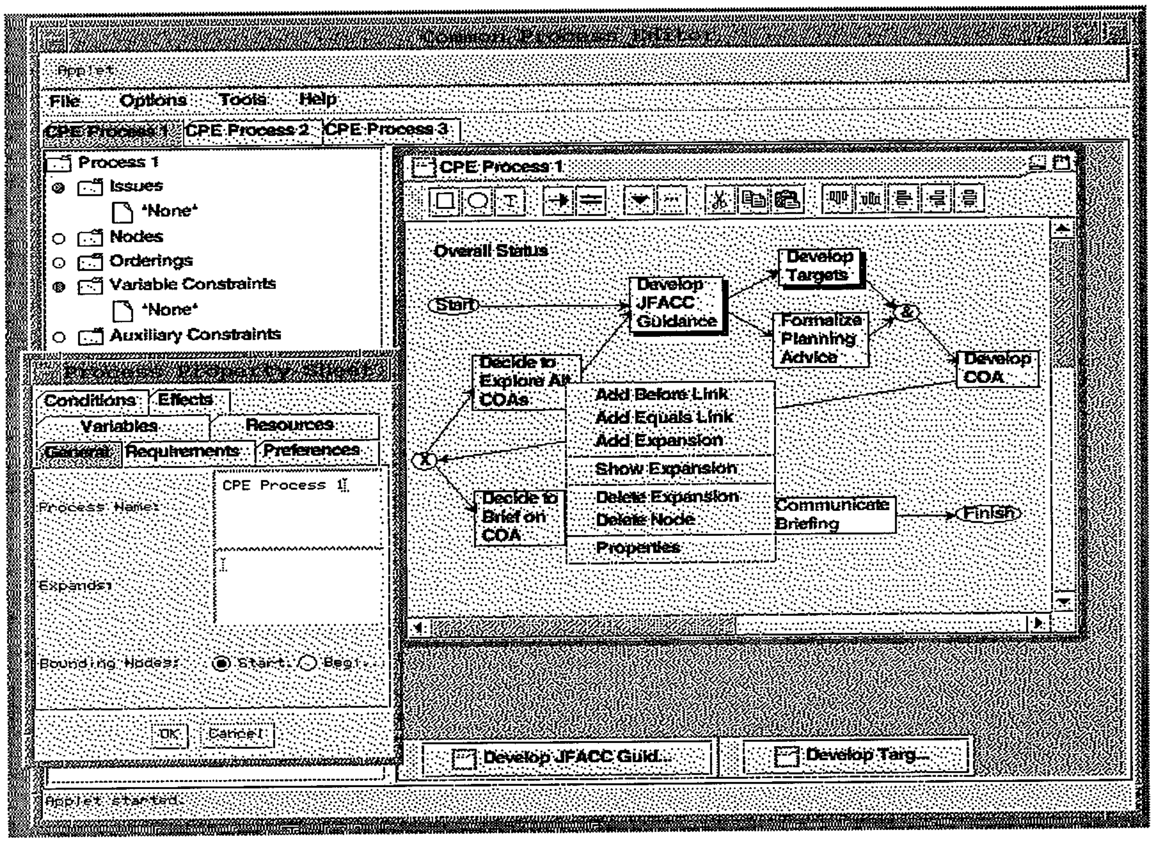
Wilkins, D. (1988) *Practical Planning: Extending the Classical AI Planning Paradigm.* Morgan Kaufmann.

Wilson, A. (1984) *Information for Planning.* M.Sc. Thesis, Department of Artificial Intelligence, University of Edinburgh, UK.

A Sample Graphical User Interface Screens

These two screen shots are examples from the TF workstation (top, 1984) and the Common Process Editor (CPE) (bottom, 1998) which provide tool-supported assistance for plan/process management.




Common Process Editor
Applet
File Options Tools Help
CPE Process 1
CPE Process 2
CPE Process 3
Process 1
Issues
None
Nodes
Orderings
Variable Constraints
None
Auxiliary Constraints
Process Property Sheet
Conditions
Effects
Variables
Resources
General
Requirements
Preferences
Process Name:
CPE Process 1
Expands:
Bounding Nodes:
Start
OK
Cancel
CPE Process 1
Overall Status
Start
Develop JFACC Guidance
Develop Targets
Formalize Planning Advice
Develop COA
Decide to Explore Alt COAs
Add Before Link
Add Equals Link
Add Expansion
Show Expansion
Delete Expansion
Delete Node
Properties
Decide to Brief on COA
Communicate Briefing
Finish
Develop JFACC Guid...
Develop Targ...
Applet started.

Appendix K:

O-P^3: Open Planning Process Panels

Austin Tate, John Levine, Jeff Dalton and Stuart Aitken

Citation:
Tate, A., Levine, J., Dalton, J. and Aitken, S., O-P^3: Open Planning Process Panels, ARPI Workshop, Washington DC, October 1998.

Purpose:
Provides a description of "Planning Process Panels" used to provide an intuitive interface to display status and allow for control of the planning process when multiple plan options are being generated by a number of planning agents who may be geographically separated.

Abstract:
This paper introduces Open Planning Process Panels (O-P^3). These panels are based on explicit models of the planning process and are used to coordinate the development and evaluation of multiple courses of action. We describe the generic ideas behind O-P^3 technology, a general methodology for building O-P^3 interfaces and two applications based on O-P^3 technology – the Air Campaign Planning Process Panel (ACP^3) and the O-Plan two-user mixed-initiative planning Web demonstration. This work has an impact on a number of important research areas outside planning, including Computer Supported Cooperative Work (CSCW) and workflow support.

1 Introduction

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 two real applications – the Air Campaign Planning Process Panel (ACP^3) and the O-Plan two-user mixed-initiative planning Web demonstration. In the former, O-P^3 is used to build a visualisation panel for a complex multi-agent planning and evaluation demonstration (TIE 97-1) which uses 11 different software components and involves several users. In the latter, 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.

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 system 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 on distinct parts of the plan synthesis in parallel. 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 paper 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. We then describe the two applications which have been based on O-P^3 – ACP^3 and the O-Plan Web demonstration. We conclude with a summary and future directions for O-P^3.

2 Generic O-P^3 Technology

The generic O-P^3 is based on an explicit model of the planning process, which would be encoded using an activity modelling language such as IDEF3. 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 nodes 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 IDEF3.
- 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 a O-P^3 interface which is tailored to the overall nature of their task.

Generic O-P^3 design rules are used to inform the construction of the O-P^3 interfaces:

- Each user role in the planning process is provided with a panel which is tailored to activities and needs of that role.
- Each user role is assigned a colour to distinguish between the roles. This is used, for example, as a background colour for the header of the panel. Since a given user may act in more than one distinct user role, this acts as a useful visual cue as to which user role is being enacted at any one time.
- The generic O-P^3 panel consists of three parts: a graph sub-panel (PERT chart), a matrix sub-panel (COA comparison matrix) and other sub-panels (e.g. information on assumed environmental conditions). The graph sub-panel and the other sub-panels are optional items (depending on how useful they are for a given application).
- The graph sub-panel contains a partially-ordered graph showing the activity model of the planning planning process. Since the activity model may be large and may apply for each COA being developed, it may not be possible to show the whole network, so some sort of navigation based on decompositions and switching between COAs may be needed.
- The actions shown in the graph sub-panel are annotated with colours to show their current status in the *state model* (see above). The colours used are adapted from other ARPI plan visualisation work (Stillman and Bonissone, 1996).
- The matrix sub-panel is a table which contains two types of rows and and two types of columns. The rows are process steps (verb phrases) and COA descriptors (noun phrases). The process steps labels are coloured with the user role background colour and the COA descriptors are white. The columns are the individual COAs being developed (labelled COA-N) and a column reflecting the overall workflow (labelled "Overall").

- The process steps in the matrix sub-panel are an appropriately flattened form of the activity model of the planning process. The status of the actions can be shown using the same colours as are used in the graph sub-panel. The currently active workflow fringe (i.e. what can be done next) is shown using active hyperlinks – clicking on a hyperlink initiates the action.
- The rows are arranged in three parts, running from top to bottom. The first section is concerned with process steps prior to plan synthesis, such as setting the COA requirements. The middle section consists of the COA descriptors and is filled out when a COA has been synthesised. The final section consists of process steps which come after plan synthesis, such as addressing any outstanding issues and viewing the resulting COA in various ways.
- The COA descriptors relate to the COA products produced by the steps of the planning process, such as the minimum duration of the plan and the effectiveness. These can be provided by separate plan evaluators, simulators, etc. The COA descriptors can be selected by the users to show only the critical elements of evaluation. Colours are used to show whether the result is acceptable and raises no issues (green), is possibly acceptable but has some issues to note (orange) or is not acceptable unless the user is prepared to relax the initial requirements (red).
- The other sub-panels can contain other useful information such as tables showing the COA objectives and assumed environmental conditions for each COA.

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 1.

3 Application 1 – ACP^3

The ARPI TIE 97-1 demonstration 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. 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.

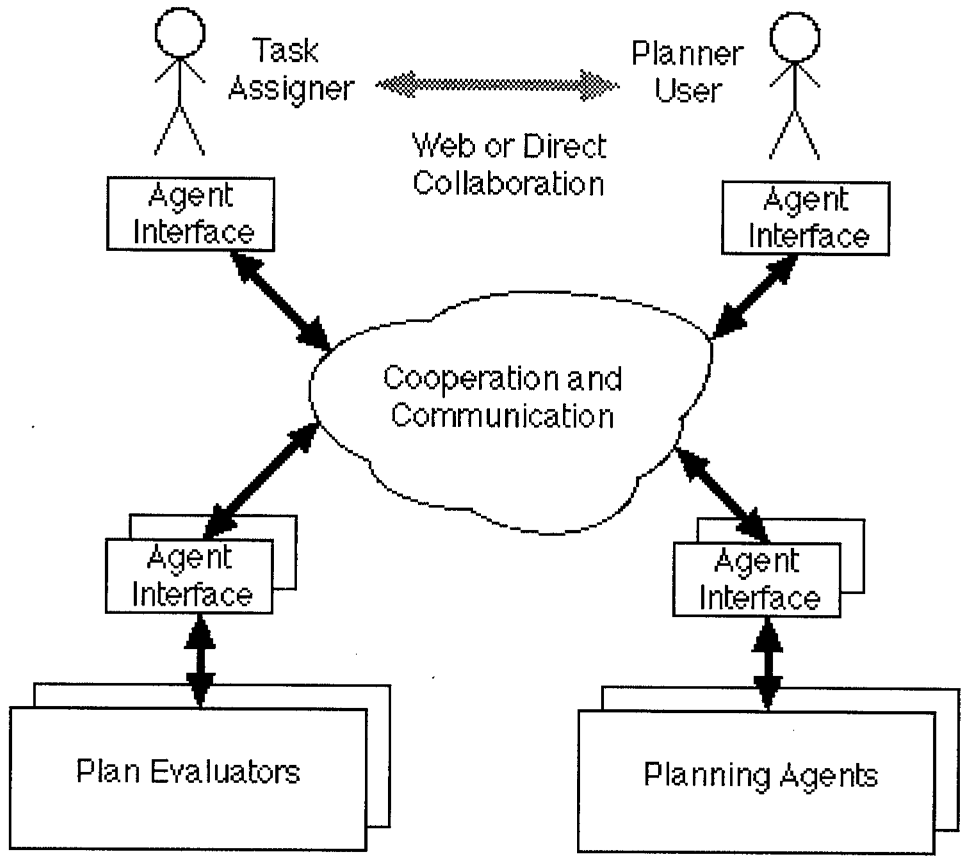


Figure 1: Using O-P^3 Interfaces

3.1 Modelling the Planning Process

The software components of TIE 97-1 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 TIE 97-1. The full models used for building ACP^3 are described in Aitken and Tate (1997).

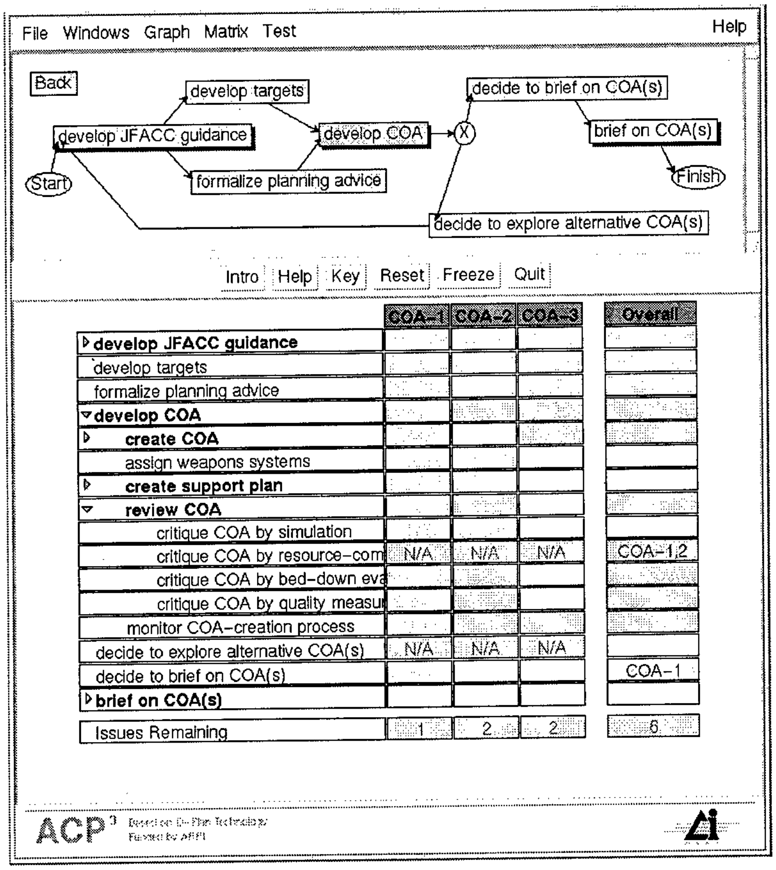


Figure 2: The ACP^3 Viewer

3.2 Building ACP^3

The ACP^3 viewer is shown in Figure 2. The purpose of ACP^3 is to track the overall planning process and display this to the viewers of the ARPI TIE 97-1 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 TIE 97-1 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 control panels.

4 Application 2 – O-Plan

The current O-Plan project (Tate, Drabble and Dalton, 1996; Tate, Dalton and Levine, 1998) is concerned with providing support for mixed-initiative planning. The current demonstration 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.

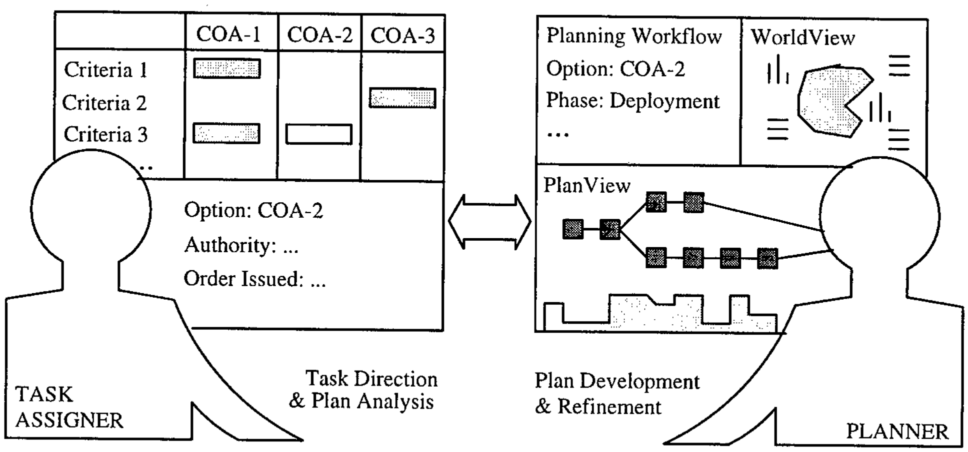


Figure 3: Communication between TA and Planner

As shown in Figure 3, 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 4 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 will be demonstrated in the example scenario.

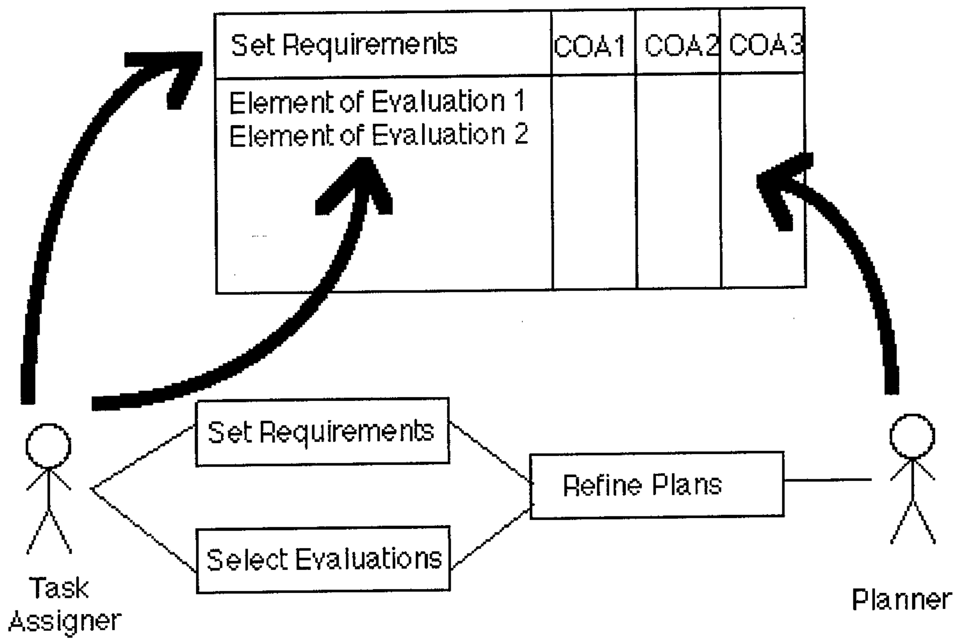


Figure 4: Roles of the Task Assigner and the Planner

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 TA 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 TA.

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 TA 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 TA has authority over the Planner User who in turn has authority over O-Plan. This means that the TA 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.

4.1 The COA Comparison Matrix

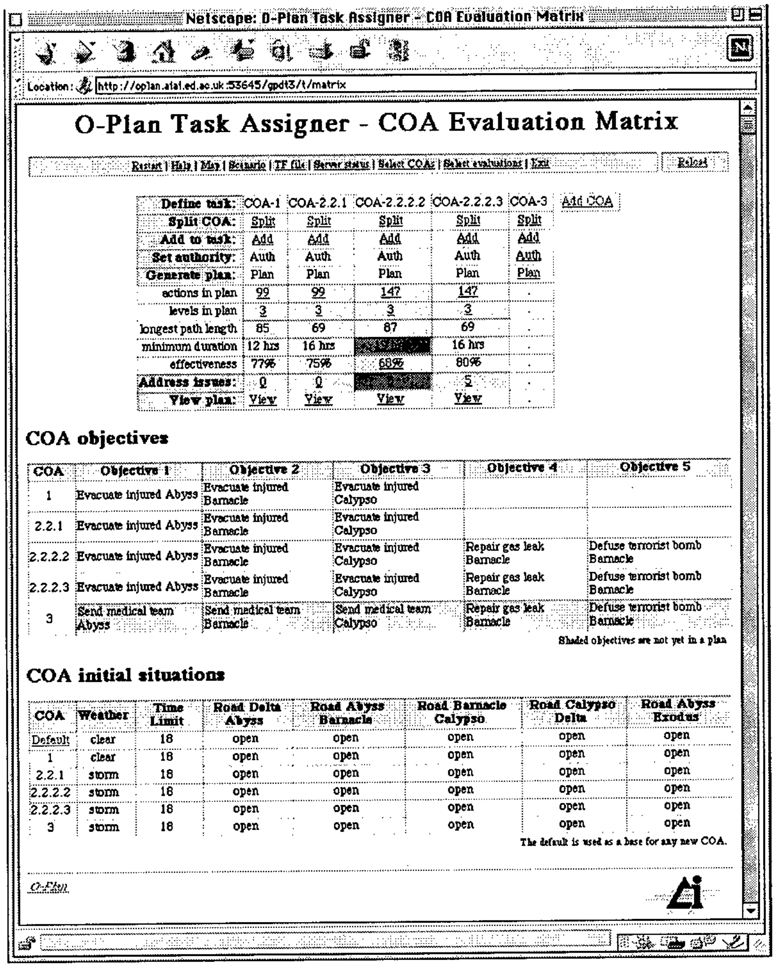
The two panels for the Task Assigner and Planner User are shown in Figures 5 and 6. Each user has control over the plan evaluation elements which are shown, to enable the critical elements of evaluation to be chosen. In the example scenario given later, the TA is only interested in the minimum duration 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 TA 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 TA. 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.

4.2 The Demonstration Scenario

The following scenario illustrates how we envisage the system being used and can be used in actual demonstrations of this work.

Initial situation: the action takes place on the island of Pacifica, with emergencies being planned for at the cities of Abyss, Barnacle and Calypso. The TA is told to deal with injured



Netscape: O-Plan Task Assigner - COA Evaluation Matrix

Location: http://oplan.aiai.ed.ac.uk:53645/gpdt3/t/matrix

O-Plan Task Assigner - COA Evaluation Matrix

Restart | Help | Map | Scenario | TF file | Server status | Select COAs | Select evaluations | Exit   Reload

Define task:	COA-1	COA-2.2.1	COA-2.2.2.2	COA-2.2.2.3	COA-3	Add COA
Split COA:	Split	Split	Split	Split	Split	
Add to task:	Add	Add	Add	Add	Add	
Set authority:	Auth	Auth	Auth	Auth	Auth	
Generate plan:	Plan	Plan	Plan	Plan	Plan	
actions in plan	99	99	147	147	.	
levels in plan	3	3	3	3	.	
longest path length	85	69	87	69	.	
minimum duration	12 hrs	16 hrs	[illegible]	16 hrs	.	
effectiveness	77%	75%	68%	80%	.	
Address issues:	0	0	[illegible]	5	.	
View plan:	View	View	View	View	.	

COA objectives

COA	Objective 1	Objective 2	Objective 3	Objective 4	Objective 5
1	Evacuate injured Abyss	Evacuate injured Barnacle	Evacuate injured Calypso		
2.2.1	Evacuate injured Abyss	Evacuate injured Barnacle	Evacuate injured Calypso		
2.2.2.2	Evacuate injured Abyss	Evacuate injured Barnacle	Evacuate injured Calypso	Repair gas leak Barnacle	Defuse terrorist bomb Barnacle
2.2.2.3	Evacuate injured Abyss	Evacuate injured Barnacle	Evacuate injured Calypso	Repair gas leak Barnacle	Defuse terrorist bomb Barnacle
3	Send medical team Abyss	Send medical team Barnacle	Send medical team Calypso	Repair gas leak Barnacle	Defuse terrorist bomb Barnacle

Shaded objectives are not yet in a plan

COA initial situations

COA	Weather	Time Limit	Road Delta Abyss	Road Abyss Barnacle	Road Barnacle Calypso	Road Calypso Delta	Road Abyss Exodus
Default	clear	18	open	open	open	open	open
1	clear	18	open	open	open	open	open
2.2.1	storm	18	open	open	open	open	open
2.2.2.2	storm	18	open	open	open	open	open
2.2.2.3	storm	18	open	open	open	open	open
3	storm	18	open	open	open	open	open

The default is used as a base for any new COA.

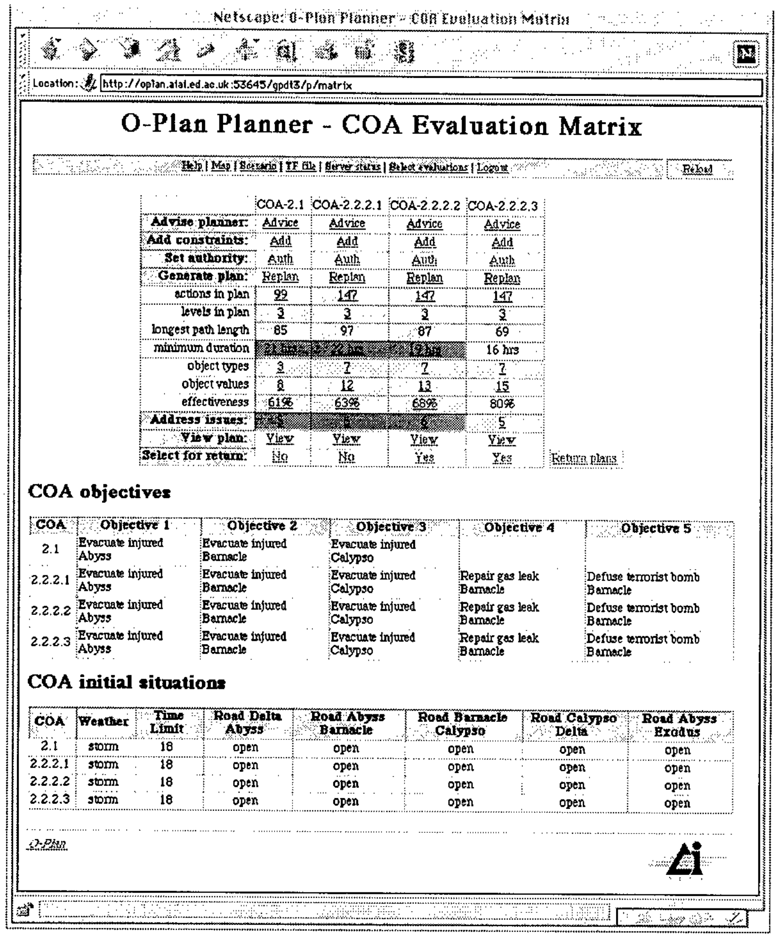
O-Plan

Figure 5: The Task Assigner's Panel

civilians at Abyss, Barnacle and Calypso within the next 18 hours. Plans are only acceptable if their effectiveness is 75% or greater. The weather forecast gives a 50% chance of a storm within the next 24 hours (Figure 7).

Initial preparations: The TA sets up the default situation, setting the time limit to 18 hrs. The weather and road situations are left with their default values pending more accurate reports.

COA-1: The TA first explores the option of evacuating the injured from all three cities in clear weather. The COA requirements are passed directly to the planner user. A plan is generated which executes in 12 hrs and has an effectiveness of 77%, which is acceptable. The plan has 3 issues outstanding. The planner user addresses these and returns the plan to the TA.



Netscape: O-Plan Planner - COA Evaluation Matrix

Location: http://oplan.aiai.ed.ac.uk:53645/gpdt3/p/matrix

O-Plan Planner - COA Evaluation Matrix

Help | Map | Scenario | TF file | Server status | Select evaluations | Logout   Reload

	COA-2.1	COA-2.2.2.1	COA-2.2.2.2	COA-2.2.2.3	
Advise planner:	Advice	Advice	Advice	Advice	
Add constraints:	Add	Add	Add	Add	
Set authority:	Auth	Auth	Auth	Auth	
Generate plan:	Replan	Replan	Replan	Replan	
actions in plan	99	147	147	147	
levels in plan	3	3	3	3	
longest path length	85	97	87	69	
minimum duration	21 hrs	22 hrs	19 hrs	16 hrs	
object types	3	7	7	7	
object values	8	12	13	15	
effectiveness	61%	63%	68%	80%	
Address issues:	[illegible]	5	6	5	
View plan:	View	View	View	View	
Select for return:	No	No	Yes	Yes	Return plans

COA objectives

COA	Objective 1	Objective 2	Objective 3	Objective 4	Objective 5
2.1	Evacuate injured Abyss	Evacuate injured Barnacle	Evacuate injured Calypso		
2.2.2.1	Evacuate injured Abyss	Evacuate injured Barnacle	Evacuate injured Calypso	Repair gas leak Barnacle	Defuse terrorist bomb Barnacle
2.2.2.2	Evacuate injured Abyss	Evacuate injured Barnacle	Evacuate injured Calypso	Repair gas leak Barnacle	Defuse terrorist bomb Barnacle
2.2.2.3	Evacuate injured Abyss	Evacuate injured Barnacle	Evacuate injured Calypso	Repair gas leak Barnacle	Defuse terrorist bomb Barnacle

COA initial situations

COA	Weather	Time Limit	Road Delta Abyss	Road Abyss Barnacle	Road Barnacle Calypso	Road Calypso Delta	Road Abyss Exodus
2.1	storm	18	open	open	open	open	open
2.2.2.1	storm	18	open	open	open	open	open
2.2.2.2	storm	18	open	open	open	open	open
2.2.2.3	storm	18	open	open	open	open	open

O-Plan

Figure 6: The Planner User's Panel

COA-2: The TA then sets up a second COA with the same evacuation tasks but this time assuming stormy weather, to check for all eventualities. This new set of COA requirements is passed to the planner user. The first plan generated takes 21hrs and has an effectiveness of 61%, both of which are unacceptable. The planner asks the O-Plan planner for an alternative plan. The new plan (COA-2.2) executes in 16 hrs and has an effectiveness of 75%, both of which are acceptable. The planner user returns COA-2.2 to the TA and deletes COA-2.1. At this point, the TA has an acceptable plan for both clear and stormy conditions.

Developing situation: the TA is now contacted by the Barnacle field station. Reports are coming in of an explosion at the power station, causing a gas leak. It is thought that this is due to a terrorist bomb, so it seems wise to fix the gas leak and send a bomb squad to defuse any remaining bombs. Meanwhile, the latest weather report indicates that a storm is brewing and has a 95% chance of hitting the island (Figure 8).

COA-2.2.2: to deal with this turn of events, the TA splits COA-2.2 (the realistic weather assumption) into two sub-options and adds two new tasks to COA-2.2.2, to repair the gas leak

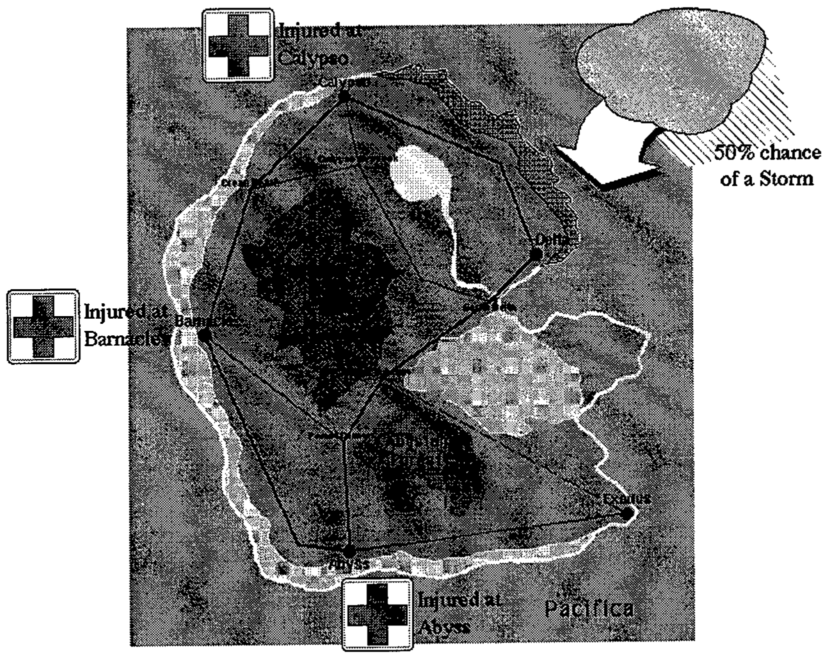


Figure 7: The Initial Situation

at Barnacle and send a bomb squad to Barnacle. COA-2.2.2 is now passed to the planner user. Since the original COA-2.2 took 16 hrs, the planner user switches schema choice on, to have fine control of the addition of the two new tasks to the existing plan. The planner user is given the option of using fast or slow vehicles for the two tasks and chooses fast vehicles. However, this plan takes 22 hrs and has an effectiveness of 63%. The planner user replans and chooses a mixture of fast and slow vehicles for the "repair gas leak" task and a fast vehicle for the "defuse terrorist bomb" task. While better, the new plan takes 19 hrs and has an effectiveness of only 68%. The TA is getting impatient and tells the planner user "this is taking too long. Just give me the best one so far." The planner user returns COA-2.2.2.2, keeping COA-2.2.2.1 for further back office work.

COA-3: The TA decides to try sending medical teams to the three cities to deal with the injured civilians rather than evacuating them. After updating the default situation to reflect the weather report, the TA starts to set up COA-3 with these tasks, and so begins to define the requirements on the screen.

COA-2.2.2.3: Meanwhile, the planner user has continued to explore the possibilities for COA-2.2.2. The plan was improved when the planner user used some slow vehicles in the plan, so it seems likely that this is because the limited number of fast vehicles are being used repeatedly, resulting in a longer (i.e. more linear) plan. The planner user presses "replan" and chooses to use a slow vehicle in the "defuse terrorist bomb" task – since sending the bomb squad is only a precaution, using the limited number of fast vehicles for evacuating the injured and fixing the known gas leak seems like a good idea. The planner user was right –

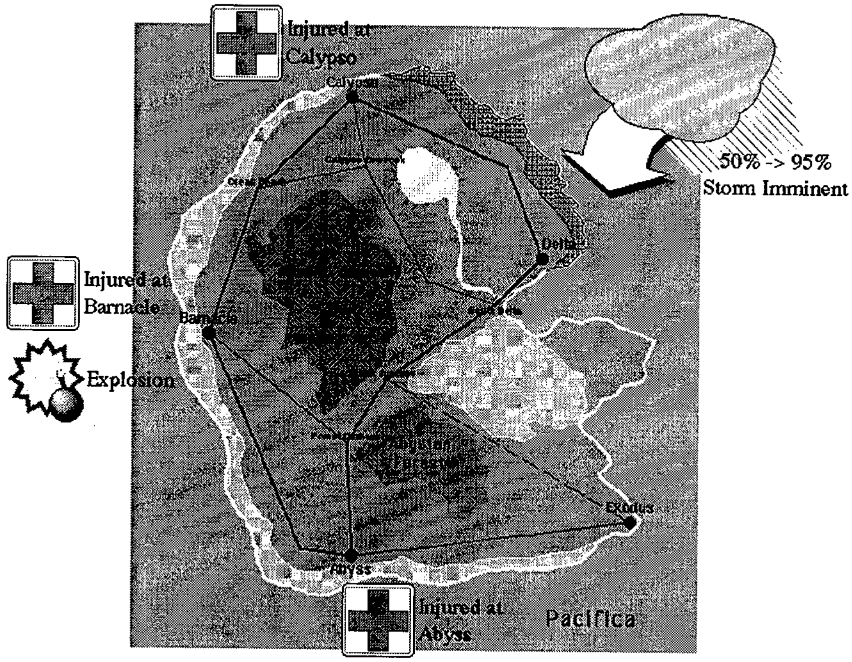


Figure 8: The Developing Situation

the resulting plan executes in 16 hrs and has an effectiveness of 80%. Viewing the plan at level 2 displays that this plan has good parallelism. The planner user now addresses the issues raised by COA-2.2.2.3 and returns this plan to the TA, saying "I think I've fixed the problem with COA-2.2.2".

Back to COA-3: The TA sees the new plan. "That looks good, now see what you can do with COA-3 as an alternative". The planner user (still in "ask user" schema selection mode) selects the fast vehicle option for 4 of the tasks, but selects a slow vehicle for the "defuse terrorist bomb" task. The resulting plan executes in 12 hrs and has an effectiveness of 79%.

Choice of COA: The TA now has a choice between COA-2.2.2.3 and COA-3. While COA-3 takes 4 hrs less, it is slightly less effective, and more importantly, it only sends medical teams to the three cities rather than evacuating the injured people. The TA could now examine other details of the two plans, using the plan views and the other elements of evaluation, in order to make an informed choice between the two or plan further.

4.3 O-Plan – Summary

The O-Plan Web demonstration illustrates mixed-initiative interaction between two human agents and one system planning agent engaged in the process of developing multiple qualitatively different courses of action. O-P^3 interfaces are provided for the two human users which are tailored to their individual user roles.

5 Summary of O-P^3 Technology and Future Applications

In this paper, we have introduced the generic notion of 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). We have described how O-P^3 technology has been used to implement two real applications – the Air Campaign Planning Process Panel (ACP^3) and the O-Plan two-user mixed-initiative Web demonstration of crisis response planning.

Both of these systems have an explicit notion of the planning process, which is a multi-agent interaction. The agents in both systems are assigned with roles which relate to the actions the users can carry out in the planning process. Both systems use the notion of a COA matrix which shows possible steps in the planning process for each course of action being developed. In ACP^3, this is used as a visualisation device. In the O-Plan demonstration, the population of this matrix is central to the mixed-initiative interaction between the Task Assigner, Planner User and O-Plan.

A number of other applications of O-P^3 technology are envisaged. An O-P^3 panel for the US DARPA Genoa program's intelligence gathering process is under investigation. This panel, termed G-P^3, would include the matrix sub-panel and the graph sub-panel from O-P^3. However it is thought that G-P^3 would also include new sub-panels to provide a "process product" perspective (showing the status of various information products under development) and new panels intended to give more role specific workflow status for a number of types of user. The main innovation in G-P^3 would be hooks to allow intelligent planning technology (e.g. provided by O-Plan) to be used to dynamically generate and adapt workflows and the planning process to accommodate changing requirements and situations. Such an "Intelligent Workflow Planning Aid" using O-Plan has already been demonstrated for Air Campaign Planning process (Drabble, Tate and Dalton, 1996).

Acknowledgements

The O-Plan project is sponsored by the Defense Advanced Research Projects Agency (DARPA) and the US Air Force Research Laboratory at Rome (AFRL), Air Force Materiel Command, USAF, under grant number F30602-95-1-0022. The O-Plan project is monitored by Dr. Northrup Fowler III at AFRL. The US Government is authorised to reproduce and distribute reprints for Governmental 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, AFRL or the US Government.

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.

Drabble, B., Tate, A. and Dalton, J. (1996). ACP Process Management: O-Plan IFD-5 Qualifier. O-Plan Technical Report ARPA-RL/O-Plan/TR/30, Version 1, November 1996.

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.

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.

Stillman J. and Bonissone, P. (1996). Technology Development in the ARPA/RL Planning Initiative. In Advanced Planning Technology, 10-23, (Tate, A., ed.), AAAI Press.

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

Tate, A., Dalton, J. and Levine, J. (1998). Generation of Multiple Qualitatively Different Plan Options. *Proceedings of AIPS-98*, Pittsburgh, USA.

Appendix L:

Generation of Multiple Qualitatively Different Plan Options

Austin Tate, Jeff Dalton and John Levine

Citation:
Tate, A., Dalton, J. and Levine, J., Generation of Multiple Qualitatively Different Plan Options, Proceedings of Fourth International Conference on Artificial Intelligence Planning Systems (AIPS-98), 27–34, Pittsburgh PA, USA, June 1998, AAAI Press.

Purpose:
Provides an overview of the results of the project and describes the demonstration scenario. The paper thus acts as a short version of the overall final report of the project.

Abstract:
In this paper, we present a Web-based demonstration of a Course of Action (COA) comparison matrix being used as an interface to an O-Plan plan server to explore multiple qualitatively different plan options. The scenario used for this demonstration is concerned with crisis operations on the island of Pacifica. The COA comparison matrix allows the user to explore and evaluate several different plan options based on different command-level requirements and different assumptions about the conditions on the island. This work is part of a larger effort to build a comprehensive mixed initiative planning system incorporating human users in designated user roles.

1 Introduction

Under the University of Edinburgh O-Plan Project (Currie and Tate, 1991; Tate, Drabble and Kirby, 1994), which is part of the DARPA/Air Force Research Laboratory (Rome) Planning Initiative – ARPI (Fowler *et al.*, 1996; Tate, 1996a), we are exploring mixed initiative planning methods and their application to realistic problems in logistics, air campaign planning and crisis action response (Tate, Drabble and Dalton, 1996). In preparatory work, O-Plan has been demonstrated operating in a range of mixed initiative modes on a Non-Combatant Evacuation Operation (NEO) problem (Tate, 1994; Drabble, Tate and Dalton, 1995). A number of "user roles" were identified to help clarify some of the types of interaction involved and to assist in the provision of suitable support to the various roles (Tate, 1994).

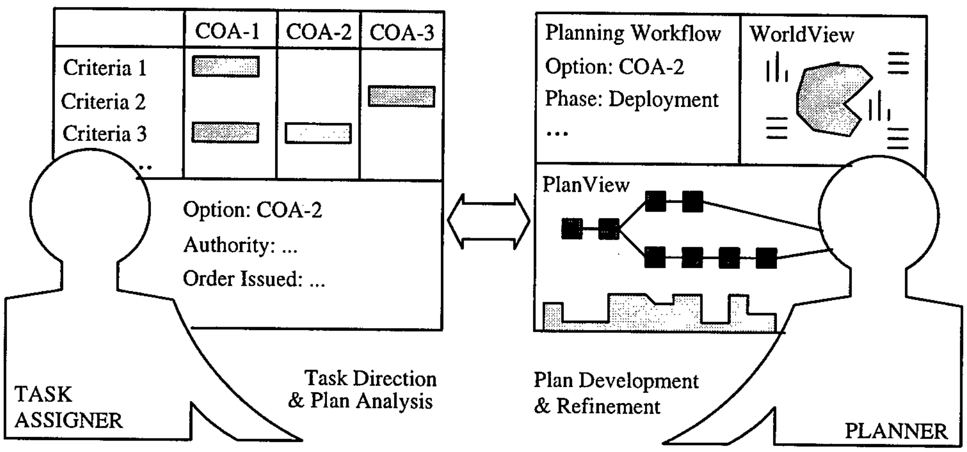


Figure 1: Communication between the Task Assigner and the Planner

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 in given roles to populate a Course of Action (COA) / Elements of Evaluation comparison matrix. The columns of this matrix are alternative options being explored as a potential solution to a (possibly underspecified) problem and the rows are evaluations of the solution being considered. The idea is that the requirements, assumptions and constraints are all refined concurrently using the elements of evaluation (EEs).

We are exploring the links between key user roles in the planning process and automated planning support aids (Tate, 1997). Research is exploring a planning workflow control model using:

- a shared model of mixed initiative planning as "mutually constraining the space of behaviour";
- the <I-N-OVA> constraint model of activity as the basis for plan communication;
- explicit management between agents of the tasks and options being considered;

- agent agendas and agenda issue handlers;
- explicit provision of authority for an agent to perform its functions.

Agents maintain their own perspective of their part in the task to hand, while cooperating with other agents who may perform parts of the task.

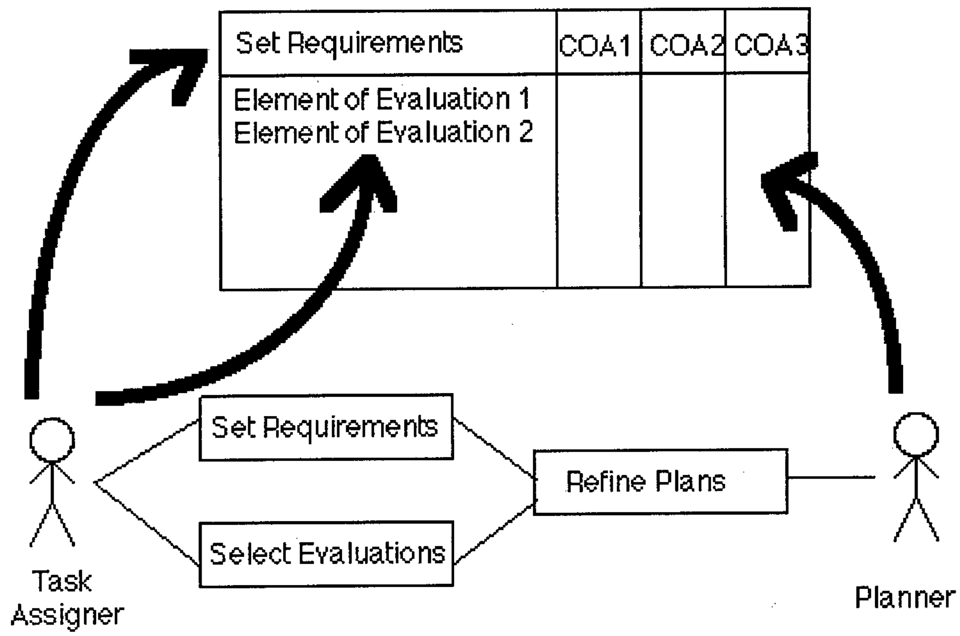


Figure 2: Roles of the Task Assigner and the Planner

As shown in Figure 1, we envisage two human agents, called the Task Assigner and the Planner, working together to explore possible solutions to a problem and making use of automated planning aids to do this. Figure 2 shows how the two human agents work together to populate the COA comparison matrix. The Task Assigner sets the requirements for a particular Course of Action (i.e. what top level tasks must be performed) and selects appropriate evaluation criteria (elements of evaluation) for the resulting plans. The Planner agent acts to refine the resulting plans by adding further constraints and splitting plans to explore two or more possible options for the same COA requirements.

In this paper, we describe our current Web-based demonstration of a Task Assigner interacting with O-Plan via a COA comparison matrix, together with the background to this demonstration. We start with the generic systems architecture being used and the architecture of the O-Plan system being used as a *plan server.* We then describe mixed initiative planning where multiple agents mutually constrain the space of behaviour. The current Web-based demonstration of our ideas is then presented, followed by a summary and future directions.

2 Generic Systems Integration Architecture

The O-Plan agent architecture to be described in the next section is a specific variant of a generalised systems integration architecture shown in Figure 3. This general structure has been adopted on a number of AIAI projects (Fraser and Tate, 1995). The architecture is an example of a *Model/Viewer/Controller* arrangement.

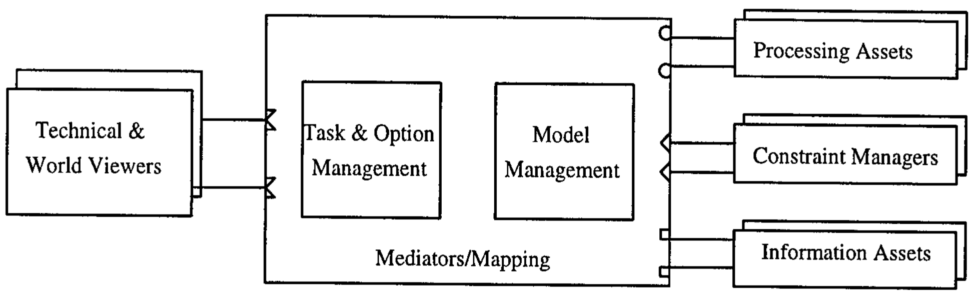


Figure 3: Generic Systems Integration Architecture

The components are as follows:

Viewers: user interface, visualisation and presentation viewers for the model.

Task and Option Management: the capability to support user tasks via appropriate use of the processing and information assets and to assist the user in managing options being used within the model. This is sometimes referred to as the *Controller*.

Model Management: coordination of the capabilities/assets to represent, store, retrieve, merge, translate, compare, correct, analyse, synthesise and modify models.

Mediators: Intermediaries or converters between the features of the model and the interfaces of active components of the architecture.

Processing Assets: functional components (model analysis, synthesis or modification).

Constraint Managers: components which assist in the maintenance of the consistency of the model.

Information Assets: information storage and retrieval components.

3 O-Plan – the Open Planning Architecture

This section describes the O-Plan architecture and the structure of individual O-Plan agents. The components of a single O-Plan agent are shown in Figure 4.

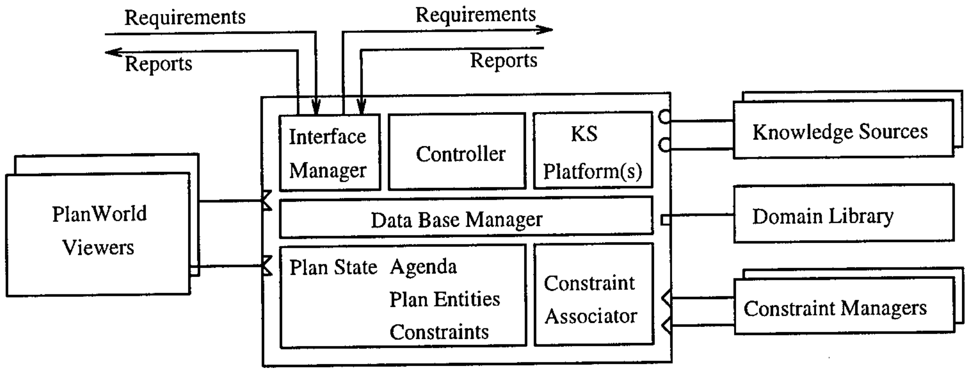


Figure 4: O-Plan Agent Architecture

3.1 Task and Option Management

Task and option management facilities are provided by the *Controller* in O-Plan. The O-Plan Controller takes its tasks from an agenda which indicates the outstanding processing required and handles these with its *Knowledge Sources.*

O-Plan has explicit facilities for managing a number of different options which it is considering. O-Plan has an agent level agenda, and agendas which relate to each option it is considering (in fact these are part of the plan representation for these options – the *issues* or I part of <I-N-OVA>). Many of these options are internal to the planning agent, and are generated during the search for a solution. Others are important for the interaction between the planner and a user acting as a task assigner.

3.2 Abstract Model of Planning Workflow – Plan Modification Operators

A general approach to designing AI-based planning and scheduling systems based on partial plan or partial schedule representations is to have an architecture in which a plan or schedule is critiqued to produce a list of issues or agenda entries which is then used to drive a workflow-style processing cycle of choosing a "plan modification operator" (PMO) to handle one or more agenda issues and then executing the PMO to modify the plan state. Figure 5 shows this graphically.

This approach is taken in O-Plan. The approach fits well with the concept of treating plans as a set of constraints which can be refined as planning progresses. Some such systems can act in a non-monotonic fashion by relaxing constraints in certain ways. Having the implied constraints or "agenda" as a formal part of the plan provides an ability to separate the plan that is being generated or manipulated from the planning system itself.

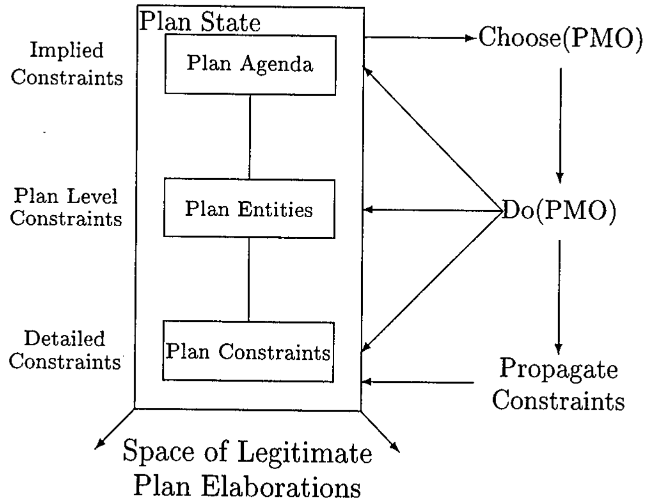


Figure 5: Planning Workflow - Using PMOs to Handle Agenda Issues

3.3 Representing Plans as a Set of Constraints on Behaviour

The <I-N-OVA> (*Issues – Nodes – Orderings / Variables / Auxiliary*) Model is a means to represent and manipulate plans as a set of constraints.

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

```
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 6: <I-N-OVA> Constraint Model of Activity

The node constraints (these are often of the form "include activity") in the <I-N-OVA> model create the space within which a plan may be further constrained. The I (issues) and OVA constraints restrict the plans within that space to those which are valid. 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 ranges.

The <I-N-OVA> constraint model of activity allows planning process state as well as the current state of the plan generated to be communicated between agents involved in the planning process. This is done via the Issues part of <I-N-OVA> – which can be used to amend the task and option specific agenda which a planning agent is using for its problem solving.

3.4 Authority to Plan

As described in Tate (1993) it is intended that O-Plan will support authority management in a comprehensive and principled way. *Changes* of authority are possible via Task Assignment agent communication to the Planner agent. This may be in the context of a current plan option and task provided previously or it is possible to give defaults which apply to all future processing by the planner agent. The authorities may use domain related names that are meaningful to the user and may refer to the options, sub-options, phases and levels of tasks and plans known to O-Plan.

4 Mutually Constraining Plans for Mixed Initiative Planning and Control

Our approach to Mixed Initiative Planning in O-Plan assists in the coordination of planning with user interaction by employing a shared model of the plan as a set of constraints at various levels that can be jointly and explicitly discussed between and manipulated by any user or system component in a cooperative fashion.

The model of Mixed Initiative Planning that can be supported by the approach is *the mutual constraining of behaviour* by refining a set of alternative partial plans. Users and systems can work in harmony though employing a common view of their roles as being to constrain the space of admitted behaviour. Further detail is given in Tate (1994).

Workflow ordering and priorities can be applied to impose specific styles of authority to plan within the system. One extreme of user driven plan expansion followed by system "filling-in" of details, or the opposite extreme of fully automatic system driven planning (with perhaps occasional appeals to an user to take predefined decisions) are possible. In contrast with this, our goal is to establish a mixed initiative form of interaction in which users and system components proceed by mutually constraining the plan using their own areas of strength.

Coordination of problem solving must take place between users and the automated components of a planning system. In joint research with the University of Rochester (whose work is described in Allen, Ferguson and Schubert, 1996) we are exploring ways in which the

O-Plan controller can be given specific limitations on what plan modifications it can perform, and the specific plan options or sub-options it is working on can be coordinated with those being explored by a user supported by a suitable interface.

5 A Web-based Demonstration

This section describes our current implementation of these ideas. We have constructed a Web-based demonstration of a task assignment agent working with the O-Plan planning agent to populate and explore different options within a course-of-action matrix. We are using a general-purpose logistics and crisis operations domain which is an extension of our earlier logistics-related domains (Reece *et al.*, 1993).

This demonstration is a significant milestone on the path towards our stated vision, since it contains many of the elements which have been planned for over the last 3 to 4 years of work and which have been incorporated into O-Plan Version 3.1 since its release in January 1997. These include:

- Multiple option management: exploration of separate options and sub-options.
- Multiple initial conditions: exploration of different initial assumptions about the domain.
- Incremental tasking: adding further requirement constraints to a plan after an initial phase of planning.
- Authority to plan: authorities can be set for any COA investigated allowing for incremental plan refinement alongside user directed addition of planning constraints.
- Plan analysis: facilities for plan analysis/evaluation can be installed which have both brief and longer analysis results to present to the user.
- Evaluation selection: the evaluations presented can be selected to show the ones which are critical.
- Issue maintenance: planning or plan analysis can leave outstanding issues to be addressed, which are summarised and collected to help with planning and coordination workflow.
- Status indication: coloured "traffic lights" are used, as in other ARPI plan visualisation work (Stillman and Bonissone, 1996) to indicate that a chosen plan for a COA is complete (green), has warnings or notes to read (orange) or have issues that need attention (red).

The Web demonstration, Version 3.1 of the O-Plan code and the papers referenced here are available by following links from the O-Plan home page.

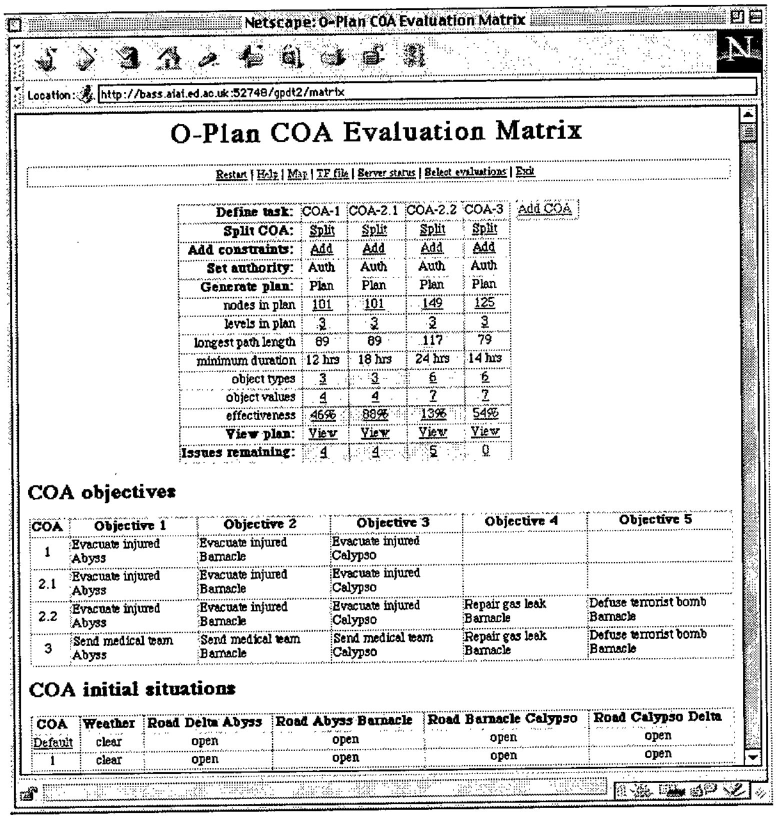


Figure 7: The Course of Action Evaluation Matrix

5.1 The COA Comparison Matrix

The user is initially given a blank COA comparison matrix which is populated by the user and O-Plan during the course of a session (Figure 7). The user acts in the role of the Task Assigner agent, setting the initial assumptions and tasking level requirements for a Course of Action (Figure 8) and selecting elements of evaluation to include in the matrix. The task assigner can split any COA into two or more sub-options and explore further within each. Additional constraints (in the form of task level requirements) can be added to any COA. The task assigner can also authorise O-Plan only to plan to a nominated level of detail. Together, these facilities allow for incremental development, exploration and evaluation of multiple qualitatively different plan options.

The COA matrix is an abstract underlying notion and may not appear in a user interface for a completed system. However, it is useful in this demonstration to show our ideas about what is being created and refined as mixed initiative problem solving takes place. In a dialogue system, such as TRAINS (Ferguson, Allen and Miller, 1996), the COA matrix would be the underlying model of the problem solving and the dialogue model would then implicitly refer to this artefact.

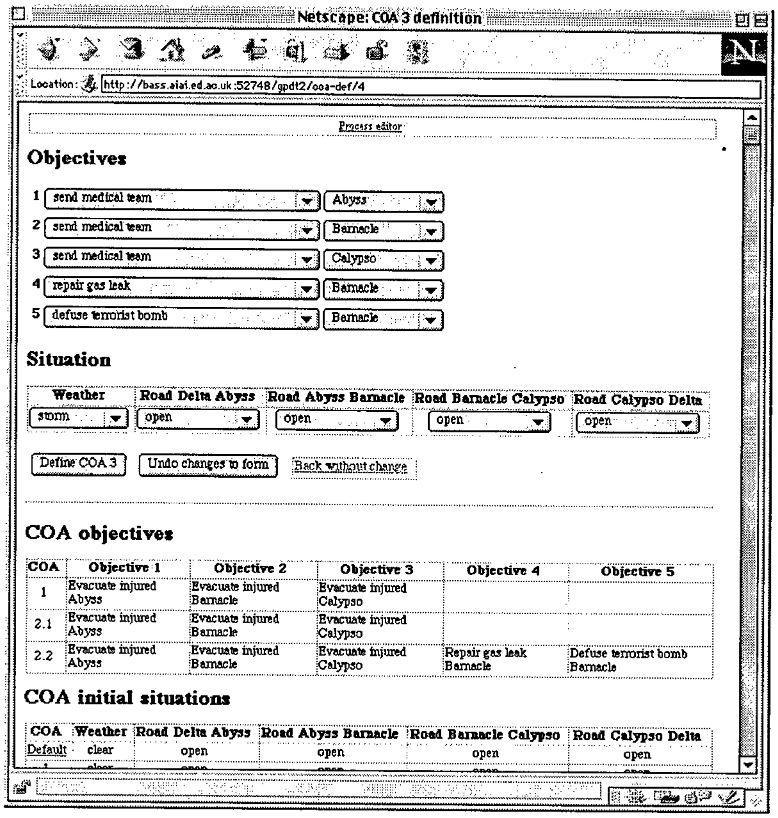


Figure 8: Defining the Requirements for a Course of Action

5.2 "Go Places and Do Things" – The Crisis Operations Domain

We have used a crisis operations domain based on the Pacifica scenarios (Reece *et al.*, 1993; Tate, Drabble and Dalton, 1996) that we call "Go Places and Do Things" (GPDT). This is a three level domain model which closely follows what we observe in large real domain models. The top level is mostly about setting objectives (i.e. COA requirements). The second level is the real planning level and where technological interactions, such as allocating limited resources, need to be resolved. The third level is needed to add detail to the skeleton plans that have been selected.

This domain is a natural extension of our earlier work in the Pacifica Non-combative Evacuation Operations (NEO) domain. In the earlier work, people are evacuated (following some crisis) from a small island using trucks and helicopters. In the new domain, the main goal is to avert a developing crisis in one of the cities on the island, using various vehicles, pieces of equipment and specialist teams. In the crisis domain, unlike previous Pacifica scenarios, the tasks to be performed are complex and may involve plans consisting of hundreds of individual actions.

This domain has been chosen for our current work to demonstrate that O-Plan is sufficiently powerful to be able to cope with these complicated logistical problems and also to provide the O-Plan team with a problem domain which is general enough to allow expansion and experimentation as our ideas and technology develop.

5.2.1 The Scenario

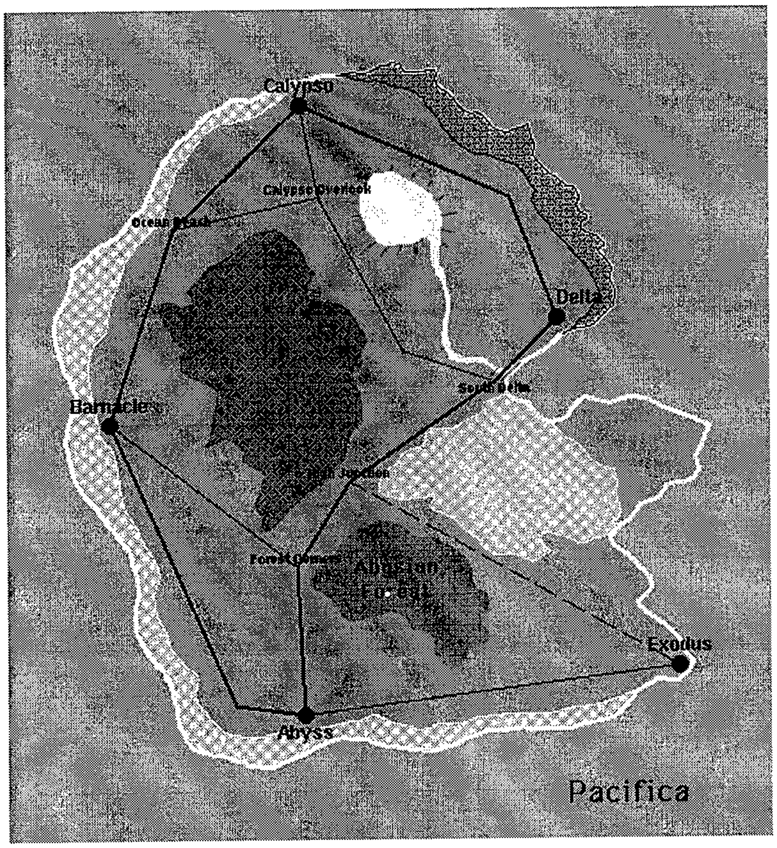


Figure 9: The Island of Pacifica

The action takes place somewhere in a network of cities, currently on the island of Pacifica (see Figure 9). A number of crisis situations can arise in the cities and on the roads joining them, such as power stations becoming inoperative or people needing medical treatment. The goal of the commander (i.e. the Task Assigner agent) is to respond effectively to the situation so that the immediate crisis situation is dealt with and appropriate repairs are made to restore the status quo.

5.2.2 World Description

The following types of objects exist in this domain:

Cities: these can contain other objects, such as teams, people and equipment.

Roads: these connect some of the cities. They may become blocked to certain classes of vehicle due to weather conditions or landslides. Some may be permanently blocked to certain classes of vehicle (e.g. mud tracks).

Vehicles: these are used to carry equipment, teams and people between cities. There are various types of vehicle which have very different capabilities, such as fast air vehicles of low carrying capacity and slow ground transports capable of carrying large pieces of equipment.

Equipment: there are various pieces of specialist equipment located in the network of cities. These are needed to perform certain tasks, such as repairs at a power station or emergency medical treatment.

Teams: there are also various specialist teams of people located in the cities. These teams perform specialist tasks, such as fast evacuation or building emergency housing.

People: people are located at cities and may need medical treatment or evacuation. As a simplification, we treat people as a single entity to be treated or moved around, rather than counting a specific number.

Weather: the weather may restrict the options available to the planner, such as not allowing helicopters to fly in thunderstorms.

The world state can be described by giving the locations and contents of the vehicles, the locations of the people, teams and pieces of equipment, and the status of the roads, people and weather.

5.2.3 Actions and Plans

In this domain, the teams, equipment and people can be moved around using a TRANSPORT action at modelling Level 2:

> TRANSPORT cargo ITEM using VEHICLE from CITY to CITY
> where ITEM is an object of type team, vehicle, equipment or people.

The result of the action is that the cargo moves from the source to the destination.

Other actions in the domain are dependent on the specific example chosen, but will typically contain around 5 actions at a lower level of detail. Typical examples are:

- Repair a turbine at a crucial power station.
- Give emergency medical treatment to people exposed to toxic fumes.
- Repair a bridge which has been broken in a storm.

- Build emergency housing for refugees.
- Perform emergency operations to make the area safe for a repair team.
- Evacuate the population of one of the cities.

An entire plan will consist of a number of TRANSPORT operations to bring the necessary teams and equipment together, followed by the main tasks. The TRANSPORT operations and main tasks may overlap, as in our demonstration example which follows.

5.2.4 Implementation Status

The current O-Plan Task Formalism (TF) file for this domain implements the crisis operations domain for the island of Pacifica, using 12 top level tasks and four cities (Abyss, Barnacle, Calypso and Delta). A Course of Action consisting of 5 tasks at the top level expands to give approximately 30 actions at the second level and 150 tasks at the third level. The exact numbers will depend on the particular Level 1 tasks selected for the Course of Action.

5.3 The Demonstration Scenario

The following scenario illustrates how we envisage the system being used and can be used in actual demonstrations of this work.

The task assigner (TA) is told that there are injured people in Abyss, Barnacle and Calypso and that these people need to be treated within the next 18 hours in order to avoid fatalities. The latest weather forecast shows a 50% chance of a storm over Pacifica during the next 24 hours.

The TA decides to try evacuating the injured from all three cities as the first possible plan, using the assumption that the weather is clear. The evaluation criteria are fine and the plan executes within the required deadline. This illustrates how the TA sets up tasks and assumptions within COAs and how the interface displays the elements of evaluation in the matrix.

The TA wants to check that the plan is still OK if the predicted storm occurs. A further COA is added with the tasks being set up as before. This time, the TA sets the weather to "storm". O-Plan is asked to generate a plan for this new set of COA requirements and finds that the time taken to execute is 18 hours – just on the deadline. This illustrates the basic use of COA columns to compare different courses of action based on different initial assumptions.

However, the TA is now interrupted by a call from the Barnacle field station. Reports are coming in of an explosion at the main Barnacle power station, causing a gas leak. It is thought that this may have been caused by a terrorist bomb. It seems wise to fix the gas leak and send a bomb squad to deal with any other bombs that may have been planted. Meanwhile, the latest weather bulletin indicates that a storm is brewing to the north-east and has a 95% chance of hitting the island within the next 5 hours.

To deal with these turns of events, the TA now splits COA-2 (the realistic weather assumption) into two sub-options and adds two new tasks to one of them – COA-2.2. The new tasks are to repair the gas leak at Barnacle and to defuse other (potential) terrorist bombs at Barnacle. This illustrates the use of plan splitting and addition of new tasks. Unfortunately for the TA, the new plan takes 24 hours, which is 6 hours over the deadline.

The TA now needs to think. The stormy weather prediction has become more definite, so the TA sets the default weather assumption to be "storm". Then a further COA column is added (COA-3). Since the original task was to simply to treat the injured people at the three cities, evacuation is perhaps an unnecessary luxury. The TA therefore sets up COA-3 to send medical teams to the three cities, repair the gas leak and defuse the terrorist bomb at Barnacle. Since the default for the weather is "storm", the TA does not need to note this explicitly. The resulting plan completes within 14 hours, so this new plan seems like the best one so far. The "traffic light" indicators in the matrix show various warnings, mostly concerned with using all available resources of a certain type within the plan. The TA marks all of these as being acceptable and the traffic lights in the column for COA-3 turn green, indicating that the plan is ready to execute.

As a final optimisation, the TA adds another column (COA-4) and sets this up as for COA-3, but with the injured being evacuated from Barnacle rather than a medical team sent (because of the additional danger in Barnacle due to the gas leak and/or terrorist bombs). This plan executes within 17 hours, which is 1 hour less than the deadline.

5.4 Future Work

The current demonstration still has some limitations, and we plan to address these in our final project demonstration (due in June 1998). The most important item to be addressed is to add the human planner agent into the demonstration, with the task assigner, planner and O-Plan agent all acting together to explore the plan space in a true mixed initiative interaction. This will require that new facilities be added to support the human planner agent and that communication between agents be provided via Web interaction and teleconferencing. We envisage that the planner agent and the task assigner will have different interface views onto the COA matrix, as illustrated in Figure 1. We also intend to improve the treatment of the crisis operations domain and allow plans to be specified, visualised and refined via a graphical Java-based process editor and plan viewer.

6 Summary

Five concepts are being used as the basis for exploring multi-agent and mixed-initiative planning involving users and systems: Together these provide for a *shared* model of what each agent can and is authorised to do and what those agents can act upon.

1. *Shared Plan Model* – a rich plan representation using a common constraint model of activity (<I-N-OVA>).

2. *Shared Task Model* – Mixed initiative model of "mutually constraining the space of behaviour".
3. *Shared Space of Options* – explicit option management.
4. *Shared Model of Agent Processing* – handlers for issues, functional capabilities and constraint managers.
5. *Shared Understanding of Authority* – management of the authority to plan (to handle issues) and which may take into account options, phases and levels.

Using these shared views of the roles and function of various users and systems involved in a command, planning and control environment, we have demonstrated a planning agent being used to support mixed initiative task specification and plan refinement over the world wide web. It has been applied to the generation of multiple qualitatively different courses of action based on emerging requirements and assumptions. The demonstration takes place in a realistic crisis management domain.

Acknowledgements

The O-Plan project is sponsored by the Defense Advanced Research Projects Agency (DARPA) and the US Air Force Research Laboratory at Rome (AFRL), Air Force Materiel Command, USAF, under grant number F30602-95-1-0022. The O-Plan project is monitored by Dr. Northrup Fowler III at AFRL. The US Government is authorised to reproduce and distribute reprints for Governmental 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, AFRL or the US Government.

References

Allen, J.F., Ferguson, G.M. and Schubert, L.K. (1996). Planning in Complex Worlds via Mixed-Initiative Interaction. In Advanced Planning Technology, 53–60, (Tate, A., ed.), AAAI Press.

Currie, K.W. and Tate, A. (1991). O-Plan: the Open Planning Architecture. Artificial Intelligence, 51(1), Autumn 1991, North-Holland.

Drabble, B., Tate, A. and Dalton, J. (1995). Applying O-Plan to the NEO Scenarios, in An Engineer's Approach to the Application of Knowledge-based Planning and Scheduling Techniques to Logistics. Appendix O, USAF Rome Laboratory Technical Report RL-TR-95-235, December 1995.

Ferguson, G.M., Allen, J.F. and Miller, B.W. (1996). TRAINS-95: Towards a Mixed-Initiative Planning Assistant. Proceedings of the Third International Conference on AI Planning Systems (AIPS-96), 70–77, (Drabble, B., ed.), AAAI Press.

Fowler, N., Garvey, T.D., Cross, S.E., and Hoffman, M. (1996). Overview: ARPA-Rome Laboratory Knowledge-Based Planning and Scheduling Initiative (ARPI). In Advanced Planning Technology, 3–9, (Tate, A., ed.), AAAI Press.

Fraser, J. and Tate, A. (1995). The Enterprise Tool Set – An Open Enterprise Architecture. Proceedings of the Workshop on Intelligent Manufacturing Systems, International Joint Conference on Artificial Intelligence (IJCAI-95), Montreal, Canada, August 1995.

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.

Stillman J. and Bonissone, P. (1996). Technology Development in the ARPA/RL Planning Initiative. In Advanced Planning Technology, 10-23, (Tate, A., ed.), AAAI Press.

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, 512–516, (Burstein, M., ed.), Tucson, Arizona, USA, Morgan Kaufmann.

Tate, A. (1996a) (ed.). Advanced Planning Technology. AAAI Press.

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, (Drabble, B., ed.) Edinburgh, Scotland, AAAI Press.

Tate, A. (1997). Mixed Initiative Interaction in O-Plan. Proceedings of AAAI Spring 1997 Symposium on Computational Models for Mixed Initiative Interaction, Stanford University, March 1997.

Tate, A., Drabble, B. and Kirby, R. (1994). O-Plan2: an Open Architecture for Command, Planning and Control. In Intelligent Scheduling, (eds, M.Zweben and M.S.Fox), Morgan Kaufmann.

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

☆U.S. GOVERNMENT PRINTING OFFICE: 1999-610-130-81113

DISTRIBUTION LIST

addresses	number of copies
DR. NORTHRUP FOWLER AFRL/IFT 525 BROOKS ROAD ROME, NY 13441-4505	10
ARTIFICIAL INTELLIGENCE APPLICATIONS INSTITUTE 80 SOUTH BRIDGE EDINBURGH EH1 1HN UNITED KINGDOM	5
AFRL/IFOIL TECHNICAL LIBRARY 26 ELECTRONIC PKY ROME NY 13441-4514	1
ATTENTION: DTIC-OCC DEFENSE TECHNICAL INFO CENTER 8725 JOHN J. KINGMAN ROAD, STE 0944 FT. BELVOIR, VA 22060-6218	2
DEFENSE ADVANCED RESEARCH PROJECTS AGENCY 3701 NORTH FAIRFAX DRIVE ARLINGTON VA 22203-1714	1
ATTN: NAN PFRIMMER IIT RESEARCH INSTITUTE 201 MILL ST. ROME, NY 13440	1
AFIT ACADEMIC LIBRARY AFIT/LDR, 2950 P.STREET AREA B, BLDG 642 WRIGHT-PATTERSON AFB OH 45433-7765	1
AFRL/MLME 2977 P STREET, STE 6 WRIGHT-PATTERSON AFB OH 45433-7739	1

AFRL/HESC-TDC 1
2698 G STREET, BLDG 190
WRIGHT-PATTERSON AFB OH 45433-7604

ATTN: SMDC IM PL 1
US ARMY SPACE & MISSILE DEF CMD
P.O. BOX 1500
HUNTSVILLE AL 35807-3801

COMMANDER, CODE 4TL000D 1
TECHNICAL LIBRARY, NAWC-WD
1 ADMINISTRATION CIRCLE
CHINA LAKE CA 93555-6100

CDR, US ARMY AVIATION & MISSILE CMD 2
REDSTONE SCIENTIFIC INFORMATION CTR
ATTN: AMSAM-RD-OB-R, (DOCUMENTS)
REDSTONE ARSENAL AL 35898-5000

REPORT LIBRARY 1
MS P364
LOS ALAMOS NATIONAL LABORATORY
LOS ALAMOS NM 87545

ATTN: D'BORAH HART 1
AVIATION BRANCH SVC 122.10
FOB10A, RM 931
800 INDEPENDENCE AVE, SW
WASHINGTON DC 20591

AFIWC/MSY 1
102 HALL BLVD, STE 315
SAN ANTONIO TX 78243-7016

ATTN: KAROLA M. YOURISON 1
SOFTWARE ENGINEERING INSTITUTE
4500 FIFTH AVENUE
PITTSBURGH PA 15213

USAF/AIR FORCE RESEARCH LABORATORY 1
AFRL/VSOSA(LIBRARY-BLDG 1103)
5 WRIGHT DRIVE
HANSCOM AFB MA 01731-3004

ATTN: EILEEN LADUKE/D460 1
MITRE CORPORATION
202 BURLINGTON RD
BEDFORD MA 01730

OUSD(P)/DTSA/DUTD 1
ATTN: PATRICK G. SULLIVAN, JR.
400 ARMY NAVY DRIVE
SUITE 300
ARLINGTON VA 22202

DR JAMES ALLEN 1
COMPUTER SCIENCE DEPT/BLDG RM 732
UNIV OF ROCHESTER
WILSON BLVD
ROCHESTER NY 14627

DR YIGAL ARENS 1
USC-ISI
4676 ADMIRALTY WAY
MARINA DEL RAY CA 90292

DR MARIE A. BIENKOWSKI 1
SRI INTERNATIONAL
333 RAVENSWOOD AVE/EK 337
MENLO PRK CA 94025

DR MARK S. BODDY 1
HONEYWELL SYSTEMS & RSCH CENTER
3660 TECHNOLOGY DRIVE
MINNEAPOLIS MN 55418

DR PIERO P. BONISSONE 1
GE CORPORATE RESEARCH & DEVELOPMENT
BLDG K1-RM 5C-32A
P. O. BOX 8
SCHENECTADY NY 12301

DR MARK BURSTEIN 1
BBN SYSTEMS & TECHNOLOGIES
10 MOULTON STREET
CAMBRIDGE MA 02138

DR THOMAS L. DEAN 1
BROWN UNIVERSITY
DEPT OF COMPUTER SCIENCE
P.O. BOX 1910
PROVIDENCE RI 02912

DR WESLEY CHU 1
COMPUTER SCIENCE DEPT
UNIV OF CALIFORNIA
LOS ANGELES CA 90024

DR PAUL R. COHEN 1
UNIV OF MASSACHUSETTS
COINS DEPT
LEDERLE GRC
AMHERST MA 01003

DR JON DOYLE 1
LABORATORY FOR COMPUTER SCIENCE
MASS INSTITUTE OF TECHNOLOGY
545 TECHNOLOGY SQUARE
CAMBRIDGE MA 02139

DR. BRIAN DRABBLE 1
CIRL, 1269
UNIVERSITY OF OREGON
EUGENE, OR 97403

MR. SCOTT FOUSE 1
ISX CORPORATION
4353 PARK TERRACE DRIVE
WESTLAKE VILLAGE CA 91361

DR MICHAEL FEHLING 1
STANFORD UNIVERSITY
ENGINEERING ECO SYSTEMS
STANFORD CA 94305

RICK HAYES-ROTH 1
CIMFLEX-TEKNOWLEDGE
1810 EMBARCADERO RD
PALO ALTO CA 94303

MS. YOLANDA GIL 1
USC/ISI
4676 ADMIRALTY WAY
MARINA DEL RAY CA 90292

MR. MARK A. HOFFMAN 1
ISX CORPORATION
1165 NORTHCHASE PARKWAY
MARIETTA GA 30067

DR RON LARSEN 1
NAVAL CMD, CONTROL & OCEAN SUR CTR
RESEARCH, DEVELOP, TEST & EVAL DIV
CODE 444
SAN DIEGO CA 92152-5000

DR CRAIG KNOBLOCK 1
USC-ISI
4676 ADMIRALTY WAY
MARINA DEL RAY CA 90292

DR JOHN LOWRENCE 1
SRI INTERNATIONAL
ARTIFICIAL INTELLIGENCE CENTER
333 RAVENSWOOD AVE
MENLO PARK CA 94025

DR. ALAN MEYROWITZ 1
NAVAL RESEARCH LABORATORY/CODE 5510
4555 OVERLOOK AVE
WASH DC 20375

ALICE MULVEHILL 1
BBN
10 MOULTON STREET
CAMBRIDGE MA 02238

DR ROBERT MACGREGOR 1
USC/ISI
4676 ADMIRALTY WAY
MARINA DEL REY CA 90292

DR DREW MCDERMOTT 1
YALE COMPUTER SCIENCE DEPT
P.O. BOX 2158, YALE STATION
51 PROPSPECT STREET
MEW HAVEN CT 06520

DR DOUGLAS SMITH 1
KESTREL INSTITUTE
3260 HILLVIEW AVE
PALO ALTO CA 94304

DR. AUSTIN TATE 1
AI APPLICATIONS INSTITUTE
UNIV OF EDINBURGH
80 SOUTH BRIDGE
EDINBURGH EH1 1HN - SCOTLAND

DIRECTOR 1
DARPA/ITO
3701 N. FAIRFAX DR., 7TH FL
ARLINGTON VA 22209-1714

DR STEPHEN F. SMITH 1
ROBOTICS INSTITUTE/CMU
SCHENLEY PRK
PITTSBURGH PA 15213

DR JONATHAN P. STILLMAN 1
GENERAL ELECTRIC CRD
1 RIVER RD, RM K1-5C31A
P. O. BOX 8
SCHENECTADY NY 12345

DR EDWARD C.T. WALKER 1
BBN SYSTEMS & TECHNOLOGIES
10 MOULTON STREET
CAMBRIDGE MA 02138

DR BILL SWARTOUT 1
USC/ISI
4676 ADMIRALTY WAY
MARINA DEL RAY CA 90292

GIO WIEDERHOLD 1
STANFORD UNIVERSITY
DEPT OF COMPUTER SCIENCE
438 MARGARET JACKS HALL
STANFORD CA 94305-2140

DR KATIA SYCARA/THE ROBOTICS INST 1
SCHOOL OF COMPUTER SCIENCE
CARNEGIE MELLON UNIV
DOHERTY HALL RM 3325
PITTSBURGH PA 15213

DR DAVID E. WILKINS 1
SRI INTERNATIONAL
ARTIFICIAL INTELLIGENCE CENTER
333 RAVENSWOOD AVE
MENLO PARK CA 94025

DR. PATRICK WINSTON 1
MASS INSTITUTE OF TECHNOLOGY
RM NE43-817
545 TECHNOLOGY SQUARE
CAMBRIDGE MA 02139

DR STEVE ROTH 1
CENTER FOR INTEGRATED MANUFACTURING
THE ROBOTICS INSTITUTE
CARNEGIE MELLON UNIV
PITTSBURGH PA 15213-3890

DR YOAV SHOHAM 1
STANFORD UNIVERSITY
COMPUTER SCIENCE DEPT
STANFORD CA 94305

MR. LEE ERMAN 1
CIMFLEX TECKNOWLEDGE
1810 EMBARCARDERO RD
PALO ALTO CA 94303

DR MATTHEW L. GINSBERG 1
CIRL, 1269
UNIVERSITY OF OREGON
EUGENE OR 97403

MR JEFF GROSSMAN, CO 1
NCCOSC RDTE DIV 44
5370 SILVERGATE AVE, ROOM 1405
SAN DIEGO CA 92152-5146

DR ADELE E. HOWE 1
COMPUTER SCIENCE DEPT
COLORADO STATE UNIVERSITY
FORT COLLINS CO 80523

DR LESLIE PACK KAELBLING 1
COMPUTER SCIENCE DEPT
BROWN UNIVERSITY
PROVIDENCE RI 02912

DR SUBBARAO KAMBHAMPATI 1
DEPT OF COMPUTER SCIENCE
ARIZONA STATE UNIVERSITY
TEMPE AZ 85287-5406

DR CARLA GOMES 1
AFRL/IFTB
525 BROOKS RD
ROME NY 13441-4505

DR KAREN MYERS 1
AI CENTER
SRI INTERNTIONAL
333 RAVENSWOOD
MENLO PARK CA 94025

DR MARTHA E POLLACK 1
DEPT OF COMPUTER SCIENCE
UNIVERSITY OF PITTSBURGH
PITTSBURGH PA 15260

DR RAJ REDDY 1
SCHOOL OF COMPUTER SCIENCE
CARNEGIE MELLON UNIVERSITY
PITTSBURGH PA 15213

DR EDWINA RISSLAND 1
DEPT OF COMPUTER & INFO SCIENCE
UNIVERSITY OF MASSACHUSETTS
AMHERST MA 01003

DR MANUELA VELOSO 1
CARNEGIE MELLON UNIVERSITY
SCHOOL OF COMPUTER SCIENCE
PITTSBURGH PA 15213-3891

DR DAN WELD 1
DEPT OF COMPUTER SCIENCE & ENG
MAIL STOP FR-35
UNIVERSITY OF WASHINGTON
SEATTLE WA 98195

MR JOE ROBERTS 1
ISX CORPORATION
4301 N FAIRFAX DRIVE, SUITE 301
ARLINGTON VA 22203

DR TOM GARVEY 1
SRI INTERNATIONAL
ARTIFICIAL INTELLIGENCE CENTER
333 RAVENSWOOD AVE
MENLO PARK CA 94025

DIRECTOR 1
DARPA/ISO
3701 NORTH FAIRFAX DRIVE
ARLINGTON VA 22203-1714

OFFICE OF THE CHIEF OF NAVAL RSCH 1
CODE 311
800 N. QUINCY STREET
ARLINGTON VA 22217

DR GEORGE FERGUSON 1
UNIVERSITY OF ROCHESTER
COMPUTER STUDIES BLDG, RM 732
WILSON BLVD
ROCHESTER NY 14627

DR STEVE HANKS 1
DEPT OF COMPUTER SCIENCE & ENG'G
UNIVERSITY OF WASHINGTON
SEATTLE WA 98195

DR CHRISTOPHER OWENS 1
GTE
10 MOULTON ST
CAMBRIDGE MA 02138

DR JAIME CARBONNEL 1
THE ROBOTICS INSTITUTE
CARNEGIE MELLON UNIVERSITY
DOHERTY HALL, ROOM 3325
PITTSBURGH PA 15213

DR NORMAN SADEH 1
THE ROBOTICS INSTITUTE
CARNEGIE MELLON UNIVERSITY
DOHERTY HALL, ROOM 3315
PITTSBURGH PA 15213

DR TAIEB ZNATI 1
UNIVERSITY OF PITTSBURGH
DEPT OF COMPUTER SCIENCE
PITTSBURGH PA 15260

DR MARIE DEJARDINS 1
SRI INTERNATIONAL
333 RAVENSWOOD AVENUE
MENLO PARK CA 94025

MR.ROBERT J. KRUCHTEN 1
HQ AMC/SCA
203 W LOSEY ST, SUITE 1016
SCOTT AFB IL 62225-5223

DR. DAVE GUNNING 1
DARPA/ISO
3701 NORTH FAIRFAX DRIVE
ARLINGTON VA 22203-1714

GINNY ALBERT 1
LOGICON ITG
2100 WASHINGTON BLVD
ARLINGTON VA 22204

ADAM PEASE 1
TECKNOWLEDGE
1810 EMBARCADERO RD
PALO ALTO CA 94303

DR STEPHEN WESTFOLD 1
KESTREL INSTITUTE
3260 HILLVIEW AVE
PALO ALTO CA 94304

DR. STEPHEN E. CROSS, DIRECTOR 1
SOFTWARE ENGINEERING INSTITUTE
CARNEGIE MELLON UNIVERSITY
4500 FIFTH AVE 15213
PITTSBURGH PA 15213

DIRNSA 1
R509
9800 SAVAGE RD
FT MEADE MD 20755-6000

NSA/CSS 1
K1
FT MEADE MD 20755-6000

PHILLIPS LABORATORY 1
PL/TL (LIBRARY)
5 WRIGHT STREET
HANSCOM AFB MA 01731-3004

THE MITRE CORPORATION 1
D460
202 BURLINGTON ROAD
BEDFORD MA 01732

DR. DAVID ETHERINGTON 1
CIRL, 1269
UNIVERSITY OF OREGON
EUGENE, OR 97403

DR. MAREK RUSINKIEWICZ 1
MICROELECTRONCS & COMPUTER TECH
3500 WEST BALCONES CENTER DRIVE
AUSTIN, TX 78759-6509

MAJOR DOUGLAS DYER/ISO 1
DEFENSE ADVANCED PROJECT AGENCY
3701 NORTH FAIRFAX DRIVE
ARLINGTON, VA 22203-1714

DR. STEVE LITTLE 1
MAYA DESIGN GROUP
2100 WHARTON STREET S&E 702
PITTSBURGH, PA 15203-1944

NEAL GLASSMAN 1
AFOSR
110 DUNCAN AVENUE
BOLLING AFB, WASHINGTON, D.C.
29332

AFRL/IFT 1
525 BROOKS ROAD
ROME, NY 13441-4505

AFRL/IFTM 1
525 BROOKS ROAD
ROME, NY 13441-4505

DR. CHARLES L. MOREFIELD 1
ALPHATECH, INC.
2101 WILSON BLVD, SUITE 402
ARLINGTON VA 22201

MR. GARRY W. BARRINGER 1
TECHNICAL DIRECTOR
AEROSPACE CZ ISR CENTER
LANGLEY AFB VA 23665

DR. JAMES HENDLER 1
DEFENSE ADVANCED PROJECT AGENCY
3701 NORTH FAIRFAX DRIVE
ARLINGTON, VA 22203-1714

MISSION
OF
AFRL/INFORMATION DIRECTORATE (IF)

The advancement and application of information systems science and technology for aerospace command and control and its transition to air, space, and ground systems to meet customer needs in the areas of Global Awareness, Dynamic Planning and Execution, and Global Information Exchange is the focus of this AFRL organization. The directorate's areas of investigation include a broad spectrum of information and fusion, communication, collaborative environment and modeling and simulation, defensive information warfare, and intelligent information systems technologies.