An Engineer's Approach to the Application of Knowledge Based Planning and Scheduling Techniques to Logistics

The O-Plan Project

Prepared by
Austin Tate
Brian Drabble
Jeff Dalton

Artificial Intelligence Applications Institute
The University of Edinburgh
80 South Bridge, Edinburgh EH1 1HN, UK

14 July 1995

Contract
AFOSR F49620-92-C-0042
EOARD 92-0001

Prepared for
Rome Laboratory
Griffiss AFB, NY 13441-5700

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.

The O-Plan research has achieved a clearer understanding of the components necessary in a flexible planning system, and has shown how such components can be combined in an open systems integration architecture. The work has determined improved ways in which the knowledge available from modelling an application domain can be used effectively to restrict search in a planner. An improved characterisation of a plan as a set of constraints on activity opens up many possibilities for richer distributed, cooperative and mixed-initiative planning systems in the future. The project has created a prototype implementation which has been demonstrated on a class of realistic applications.

Acknowledgements

The O-Plan project began in 1984. Since that time the following people have participated: Colin Bell, Ken Currie, Jeff Dalton, Roberto Desimone, Brian Drabble, Mark Drummond, Anja Haman, Ken Johnson, Richard Kirby, Arthur Seaton, Judith Secker, Glen Reece, Austin Tate and Richard Tobin.

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, 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 1989 to 1995, the O-Plan project has been 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 numbers F49620-89-C0081 (EOARD/88-0044) and F49620-92-C-0042 (EOARD/92-0001) monitored by Northrup Fowler III at the USAF Rome Laboratory. The projects have been part of the ARPA/Rome Laboratory Planning Initiative (ARPI).

Additional resources for the O-Plan projects have been provided by the Artificial Intelligence Applications Institute through the EUROPA (Edinburgh University Research on Planning Architectures) Institute development project.

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

Abbreviations

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

ADS Associated Data Structure – the level of data structure in O-Plan at which a plan is represented. This is "associated" with an underlying Time Point Network (TPN).

AM O-Plan Agenda Manager – one of the main processes of the O-Plan system and the main part of the "Controller" which decides on what can be processed next in an O-Plan agent.

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

ARPI ARPA/Rome Laboratory Planning Initiative – the Knowledge-based Planning and Scheduling Initiative research and development programme.

CPE Common Prototyping Environment – a shared framework of tools and domain information used within the ARPI.

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

DM O-Plan Database Manager – one of the main processes of the O-Plan system which manages the plan state and gives access to it on behalf of other modules.

GOP Graph Operation Processor – a support routine in O-Plan used to manipulate information in graphs or networks, e.g., in the Time Point Network (TPN).

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

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

IM O-Plan Interface Manager – one of the main processes of the O-Plan system which manages inter-module, inter-agent and user communications.

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

KP O-Plan Knowledge Source Platform – one of the main processes of the O-Plan system on which Knowledge Sources can be run.

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

KSF Knowledge Source Framework – a proposed language for describing an agent's capabilities (it's Knowledge Sources).

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.

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.

PSVM Plan State Variables Manager – the Constraint Manager in O-Plan which looks after Plan State Variables (PSVs).

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 involved 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.

REA Reactive Execution Agent – an agent designed to support the execution of plans where reaction to changing circumstances is required.

RUE Resource Utilisation Entry – the form of constraint information looked after by the Resource Utilisation Manager (RUM).

RUM Resource Utilisation Manager – a constraint manager which looks after resource constraint information.

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.

Contents

List of Figures

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 become 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.

A number of publications resulting from the O-Plan project are available. These have been chosen to give more details of the principal contributions of the work. The related papers are described in the appendix, and their relationship to work described in this paper will be highlighted throughout the text by "**[See Related Paper ...]**".

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 to provide 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 embed detailed knowledge of the domain. See [2] for background reading on AI planning systems. See [9] 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. This multi-agent approach is described in greater detail in [42] **[See Related Paper A]**. The benefits of viewing the planner as situated in a command and control framework are described in [18] **[See Related Paper C]**.

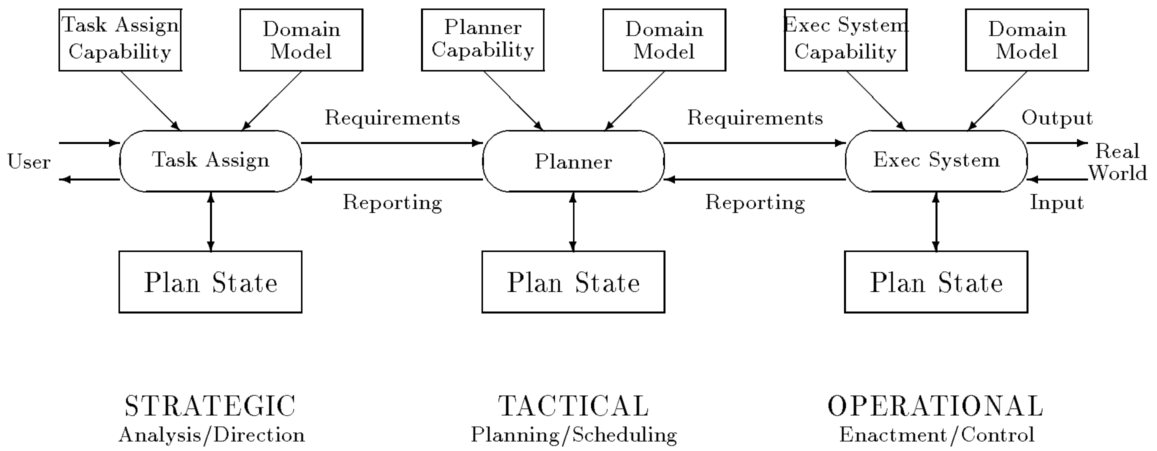

Figure 1: Communication between Strategic, Tactical and Operational Agents

Figure 1 shows the communications between the three 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* constructs a plan that would 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

[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.

outstanding issues;

- a least commitment approach;
- opportunistic selection of the focus of attention on each problem solving cycle;
- 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 [33] **[See Related Paper B]**. 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.

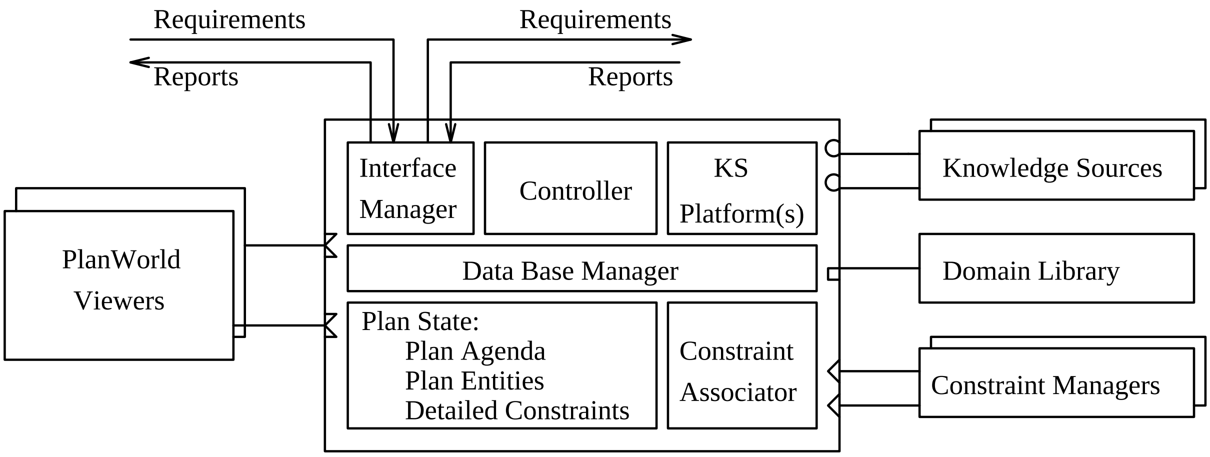

Figure 2: O-Plan Agent Architecture

The various components of the agent architecture are:

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

Knowledge Sources – Functional components which can analyse, synthesise or modify plans.

Domain Library – A description of the domain including a library of possible actions.

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 central control flow is 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.

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 agents, 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. Within such a collection of agents, a situated planner takes task assignments from a superior agent and creates a plan or further elaborates it before passing it to the subordinate execution support agents for further processing or enactment.

In many domains such as manufacturing, construction assembly, logistics, spacecraft control, etc. the planner needs to deal with changes occurring in two very different ways:

1. **Change of Task and Requirements**:
   The task set to the planner may change or be refined as the plan is being generated, requiring the planner to:
   - alter its focus, e.g., `plan to move the 82nd airborne now rather than later`,
   - choose alternative methods, e.g., `move the 82nd airborne by sea rather than air`,
   - abandon the task altogether, e.g., `abandon the deployment task and return the 82nd airborne to their home base`.
2. **Change in the Environment**:
   Events may occur in the domain which require the plan to be repaired by the insertion of new constraints or activities. In some cases the failure may be so severe that the entire plan needs to be abandoned and an alternative found.

The reason for taking this situated view is that planners should not be considered as functioning in isolation. In addition to being able to communicate about the overall task being performed, the planner ought to be able to interact closely with the environment in which it is placed. This allows more knowledge about the tasking and execution environment to be used during planning and replanning.

3.1 Planner to Plan-Execution System

The O-Plan architecture has been designed to support the creation of situated 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 [1] 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 [27]. This work also showed the value of using the plan intentions captured in Goal Structure to support effective reactive execution and re-planning [29] **[See Related Paper H]**.

3.2 Task Assigner to Planner

In many domains the problems of command, task setting, planning, plan analysis and plan enactment have been compartmentalised, leading to many systems having an inability to assimilate new information into existing plan options. In particular, the problem of dealing with task assignment and its link to the generative planner has been neglected by planning researchers. Future research in the O-Plan project aims to address this area and in particular the problem of allowing different situated agents to maintain their own perspective on the planning problem while at the same time allowing plans to be communicated between them. This will make it possible to communicate and use commands, plans, and tasks with improved precision, timeliness and level of detail between a number of situated agents. The O-Plan research has already addressed two key issues of the task assignment problem:

- **Plan Quality**:
  The task assigner needs to analyse the quality of the plans being generated and to provide feedback and direction concerning the options and plans which should be explored further. Joint work with USC/ISI to link O-Plan to their EXPECT system [21] has shown that plans can be generated and analysed to provide valuable feedback to human planners.

- **Role of Authority**:
  The activities of the various situated agents need to be coordinated, and authority management is viewed as one way in which this can be done [32]. 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.

These two aspects are elaborated further in the next two subsections.

3.3 Integrating Plan Quality Considerations into Planning

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. Often, the quality criteria that human expert planners consider:

- are highly dependent on the situation and the scenario at hand (some criteria may be more important if there is a certain deadline, or new criteria may need to be considered if new considerations arise), and
- include complex factors and tradeoffs that are often not represented by an automatic planner.

Research on plan analysis has concentrated on addressing two issues:

- to provide a tool – EXPECT [21] – which allows human planners to define criteria for plan quality and preferences among alternative plans and options.
- to operationalise these criteria to guide a generative planner in proposing better quality plans [14],[15],[22] **[See Related Paper G]**.

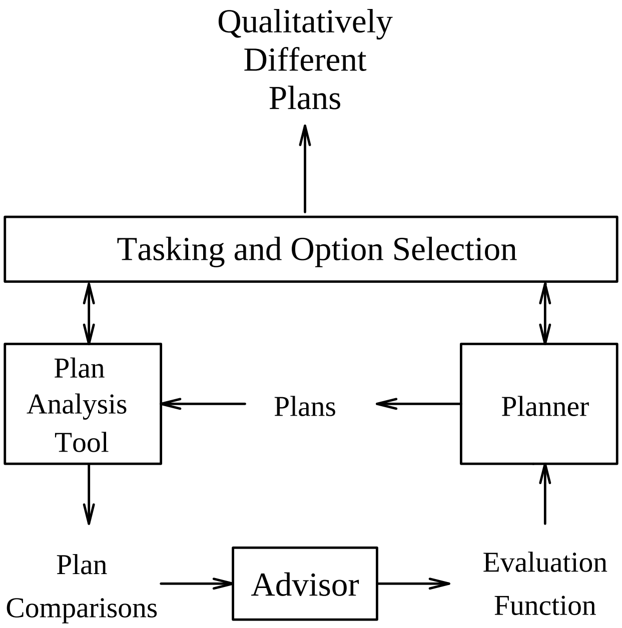

Figure 3: Combining a Planner and a Plan Analysis Tool

An approach is being investigated which combines the O-Plan planner with the EXPECT knowledge-based plan analysis system. Figure 3 describes the way in which it is proposed that O-Plan and EXPECT can be linked and the way in which plans and analysis information flows. Using these two systems, it may be possible to build an interface between the planner and the user that provides the following functionality:

- support to the user in defining criteria for evaluating plan quality through a knowledge acquisition tool,
- evaluation of the quality of plans proposed by the planner,
- justifications for judgements of plan quality,
- guidance to the planner in its search for suitable plans.

To date the O-Plan system is able to generate plans and communicate them to the EXPECT 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.

3.4 The Role of Authority for a Situated Planning Agent

At the moment, the Task Assignment agent in O-Plan informs the planner and execution agents when they can create a plan for a nominated task and when a plan can be executed. This is done through a simple menu interface. It is intended that O-Plan will support authority management in a more comprehensive and principled way in future [32] **[See Related Paper F]**. The O-Plan research has identified the need to support:

- Plan *options*: 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 with different search options or may contain a different task and environmental assumptions. It is possible to have only one plan option.
- Plan *phases*: individually provided actions or events stated explicitly in the top level task description given by the Task Assignment agent. More precise authority management is possible by specifying more explicit phases at the task level. It is possible to have only one phase in a task.
- Plan *levels*: specified degrees of detail to which plans can be produced. More precise authority management is possible by specifying more explicit levels in the O-Plan domain description language, Task Formalism (TF). It is possible to have only one level in a domain.

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.

It is anticipated that the Task Assignment agent of O-Plan will need to support such authority management capabilities. To establish an appropriate basis for future developments, and allow for some initial internal support for authority management to be incorporated, the current release of O-Plan has a simple authority scheme and reports this at the head of the Task Assignment agent menu shown here:

```
Domain: pacifica
Status: plan option 1 - planning ...
Task: Operation_Blue_Lagoon
Authority: plan(all=inf), execute(all=no)
```

This reports that the system is planning for task `Operation_Blue_Lagoon` in the domain `pacifica` and that it is currently planning within `plan option 1`. It is authorised to plan to any level of detail (infinity) for all phases (`plan all=inf`) but is not yet authorised to execute any actions (`execute all=no`).

A prototype HARDY-based[2] user interface for the Task Assignment agent has been created and connected to O-Plan.

[2] HARDY is a diagramming aid and hypermedia tool from AIAI.

4 Using Domain Knowledge in Planning

O-Plan has the ability to use domain knowledge about time constraints, resource requirements and other types of knowledge 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 [4],[16], resource constraints [17], temporal coherence of conditions [19], and Goal Structure condition type information [30],[31].

4.1 An Approach to Incorporating Constraint Management into a Planner

The O-Plan research has studied ways to enable specialised and efficient constraint handling methods to be used to manage the detailed constraints within a plan, constraints such as those on action ordering, action times, and resource use. The main, higher-level entities in plans (such as activities) are represented separately from constraints, in an *Associated Data Structure* (ADS). Separating the main plan entities from the detail of the lower-level constraints allows a more modular interface to be provided.

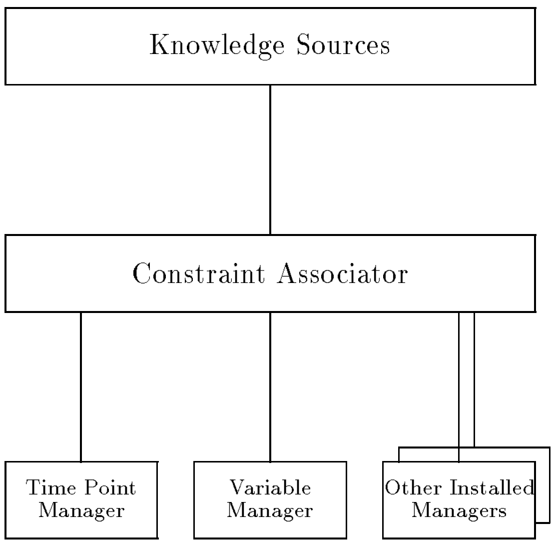

Figure 4: Associator to Mediate between Knowledge Sources and Constraint Managers

To improve the modularity of the Issue Handlers (Knowledge Sources), which must maintain detailed constraints on the main entities being manipulated, the architecture includes a *Constraint Associator* as shown in Figure 4 [35] **[See Related Paper K]**. The interface to this component allows for the handling of various types of constraint through plug-in constraint handlers[3].

Experience of writing Knowledge Sources for O-Plan and other systems has shown that it can be difficult to preserve the modularity of the code while still taking into account all aspects of detailed constraint propagation. This is particularly so for those constraints involving time and variables in a plan. Time and variables are involved in many of the manipulations performed by knowledge sources (not surprisingly in a generic temporal-activity planner). Time and variables are also often parts of other constraint descriptions for resources, Goal Structure, etc. The framework for constraint handling in O-Plan via the Constraint Associator therefore separates out the Temporal and Variable Constraint managers from any others which may be installed into O-Plan.

Being certain that there are time and variable managers installed, with a necessary minimum level of capability, allows simplifying assumptions to be made when writing Knowledge Sources in O-Plan. Knowledge Sources use the Constraint Associator to make all changes to the detailed constraints within a plan. The Constraint Associator makes appropriate calls to the detailed constraint managers installed and can cope with a range of specialised implementations of constraint handlers. All constraints can be noted, even in cases where a handler is not available[4].

The Constraint Associator maintains a uniformity of interface to the Knowledge Sources which provides some key benefits [35],[39]. The Knowledge Sources, Constraint Associator and all installed Constraint Managers may make use of a *Minimal Plan Ontology* for their communications. This provides a small set of descriptive terms about plan features and entities which may be used for communication between the components. This minimal plan ontology includes the notions of time points, the "before" ordering relation on time points, variables, and equality and inequality relations on variables. Constraints Manager responses can be:

1. **yes**, the constraint added or changes made to the constraint are valid and the changes are now under management;
2. **no**, the change could not be made to the constraint given the current constraints under management; or
3. **maybe**, if certain changes (e.g., addition of ordering links or further variable bindings) are made.

This interface is similar to the Question Answering mechanism used to establish the value of world conditions at a point in a partially-ordered plan representation [31]. That was itself a basis for the formalisation of the Modal Truth Criterion (MTC) by Chapman [7]. Such a Truth Criterion is at the heart of many current planning systems. Therefore, a common style of interface to establish the validity of constraints at points in a plan is being maintained by the O-Plan Constraint Associator in cooperation with the Constraint Managers installed into it.

[3] The current implementation does not yet make full use of this simplifying framework.

[4] O-Plan already has the concept of an *Other Constraint* type in which notes of further constraints can be kept in a plan.

The Constraint Associator can also identify potential *cross-constraint* relationships and deal with them autonomously without the Knowledge Source writer having to handle possible knock-on effects. For example, this means that a change to time constraints which may affect the current resource constraints will be identified and passed on to a resource constraint manager if one is installed. The Constraint Associator is also designed such that it can combine the results of a number of constraint manager calls and can return a single more tightly constrained set of changes to the plan state if necessary.

By using this approach to incorporating constraint management into a planner, it is possible to plug in diversified and specialised constraint handlers suited to their specific purposes. For example, a specialised spatial constraint manager using 3-D reasoning methods could be incorporated without major changes to the system design.

The following sections describe the main constraints employed by O-Plan and the managers responsible for them.

4.2 Time Constraints

O-Plan supports relative and metric time constraints for time points in actions, tasks and plans. The O-Plan constraint manager responsible for such constraints uses a Time Point Network (TPN) to support its operation and hence is called the Time Point Net Manager (TPNM) [16]. Each time point is constrained by the network to have an upper and lower bound on its temporal distance from other points in the network and from time zero.

The time points held in the TPN are indirectly linked to actions and events in a plan: the Associated Data Structure (ADS). This ensures that the TPN and the ADS can be independently changed. Moreover, the functional interface to the TPN does not reveal the underlying representation, so that a different way of handling time constraints could be substituted.

In addition to its use in the O-Plan activity-orientated planner, the current TPNM has been applied to large resource-allocation scheduling problems in the TOSCA scheduler [3], where the number of time points was in excess of 5000 and the number of temporal constraints exceeded 3000. The TOSCA scheduler was itself based on the O-Plan architecture, making use of a different Associated Data Structure based on resource reservation periods, rather than actions as in the planner.

4.3 Object/Variable Constraints

During the planning process a number of objects, and variables representing objects, can be introduced into a plan. 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.

Plan State Variables (PSVs) are created by the planner as necessary when the plan refers to an object that has not yet been identified. The Plan State Variables Manager (PSVM) is the O-Plan Constraint Manager responsible for maintaining the network of plan state variable constraints introduced into the plan.

When a PSV is created, it has stored with it a list of constraints. As more of the plan is developed, further constraints may be added. Dependencies can arise between different plan state variables and these are of two forms:

- **Same**: the variables must have the same value. (It follows from this that the constraints on the variables can be conjoined.)
- **Not-same**: the variables cannot have the same value.

In addition, each plan state variable has:

- A **type**: the set of domain objects from which the variable's value must be chosen.
- Value **constraints**: conditions the variable's value must satisfy, e.g., that its `size` be `large` and its `colour green`.

For example, plan state variable ?V1 may be constrained to be of type `movable-object`, `green` and `large`, not the same as ?V2, and the same as ?V14 and ?V8.

As with other O-Plan Constraint Managers, the responses that the PSVM can give are:

1. **yes**, the change (e.g., creating a new variable with given constraints, changing a variable's constraints) is valid, and the changes are now under management;
2. **no**, the change could not be made given the current constraints under management; or
3. **maybe**, if certain changes (e.g., addition of ordering links or further variable bindings) are made.

4.4 Resource Constraints

O-Plan uses a Resource Utilisation Manager (RUM) to manage the detailed resource constraints within a plan. The RUM can handle a number of different resource types and can reason about how resource levels change during the generation of a plan. Domain knowledge about different types of resources allows the planner:

1. to check that resource usage demands can be met from the resources available at any time;
2. to provide heuristic estimates of the quality of a plan as it is generated; and
3. to provide suggestions (if possible) on the repair of a failed plan should resource usage be the problem (reduce resource levels, produce more of the resource earlier, move actions back or forward in time, etc.)

There are two major resource types supported by the RUM: consumable resources and reusable resources. Consumable resources are ones which are consumed during the life of a plan, e.g., fuel, money, ammunition, etc. Reusable resources are ones which can be allocated to a plan for it to use and then possibly be returned (de-allocated) for re-use, e.g., trucks, manpower, runways, etc. Each of these can be further subdivided as shown in Figure 5.

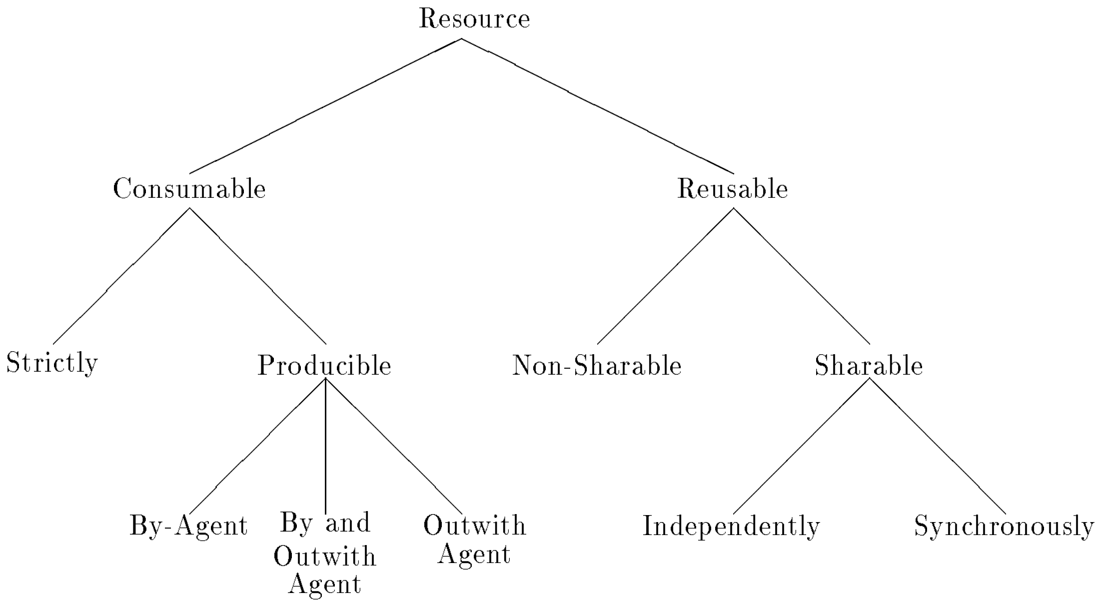

Figure 5: Example Hierarchy of Resource Types

A design for a sophisticated resource reasoning capability has been created for O-Plan [17] **[See Related Paper E]**, and a subset of this is provided by the current implementation. It is the function of the RUM to check on the levels of resources being used a certain points in the plan. The RUM is informed of resources level changes from the main planning level by means of Resource Utilisation Entries (RUE's). A RUE can effect a resource in one of five different ways:

1. **Set** a resource level to be a particular value (or within a particular range), for example to top up a fuel tank to its maximum capacity.
2. **Allocate** a certain amount of resource, i.e., reduce the amount of resource remaining as available from that point within the plan.
3. **Deallocate** a certain amount of resource back to a common pool, i.e., increase the amount of resource available from that point in the plan.
4. **Consume** a certain amount of resource.
5. **Produce** a certain amount of "new" resource.

The RUM's primary function is to manage the current set of RUEs which are part of the Plan State. It must signal to the caller when there is an inconsistent set of such RUEs. This is similar to the way that the Time Point Network Manager and other O-Plan Constraint Managers operate. The procedural interface to the RUM is the same as that of other constraint managers in that it can return any of three kinds of result:

- **yes**, the RUE can be added without any adverse impact;
- **no**, the RUE cannot be added, given the current set of resource constraints; or
- **maybe**, the RUE can be added so long as the indicated problems are handled, i.e., further temporal and/or object variable constraints are added to the plan.

This allows us to define a *Resource Criterion* [17] which is similar to the Question Answering mechanism used to establish world conditions in O-Plan.

4.5 Goal Structure and Condition Types

A lesson learned in the expert systems and knowledge-based systems field is that it is important to make maximum use of domain knowledge where it is available in order to address many real problems. 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 [30] and Nonlin [31] and subsequently used in other systems such as Deviser [43], SIPE–2 [44], and O-Plan [9],[42].

An explicit account of the *Goal Structure* or *teleology* of a plan can be kept in these systems. This records the causal relationships between actions in the plan and can show the intentions of the domain writer or planner in satisfying conditions on actions. In some circumstances, such domain knowledge can be used to prune the search of a planner. The information is provided to the planner via a planner's domain description language (e.g., Task Formalism – TF – in Nonlin and O-Plan). The domain writer takes the responsibility for a deliberate pruning of the search space or for providing preferences via condition types. This caused us to adopt the term *knowledge-based planning* to describe our work.

Nonlin and O-Plan 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). Each schema is thought of as provided by its own "manager". The schema introduces lower level actions under the direction of its manager and uses that manager's own resources. The schema may say that some specific sub-action is included in order to set up for some later sub-action as part of the overall task. In TF, such internally satisfied requirements in actions are specified as **supervised** conditions. The manager also relies on other (normally external) agents to perform tasks that are their own responsibilities, but affect the ability of this manager to do the task. These are given as **unsupervised** conditions. There are other conditions which the manager may wish to impose on the applicability of particular solutions (e.g., don't try this method for house building if the building is over five stories tall). These are termed **only_use_if** conditions in O-Plan.

A detailed description of the use of condition types to inform search in an AI planner is provided in [37] **[See Related Paper D]**. 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>[5] (*Issues – Nodes – Orderings/Variables/Auxiliary*) Model is a way to represent plans as a set of constraints [38] **[See Related Paper J]**. By having a clear description of the different components within a plan, the model allows plans to be manipulated and used separately from the environments in which they are generated.

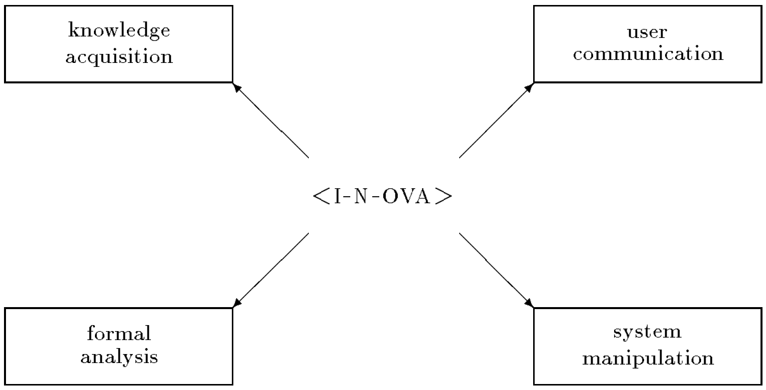

Figure 6: <I-N-OVA> Supports a Number of Requirements

As shown in figure 6, the <I-N-OVA> constraint model underlying plans is intended to support a number of different uses of plan representations:

- automatic manipulation of plans and to act as an ontology to underpin such use.
- human communication about plans.
- 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 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.

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

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

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 7.

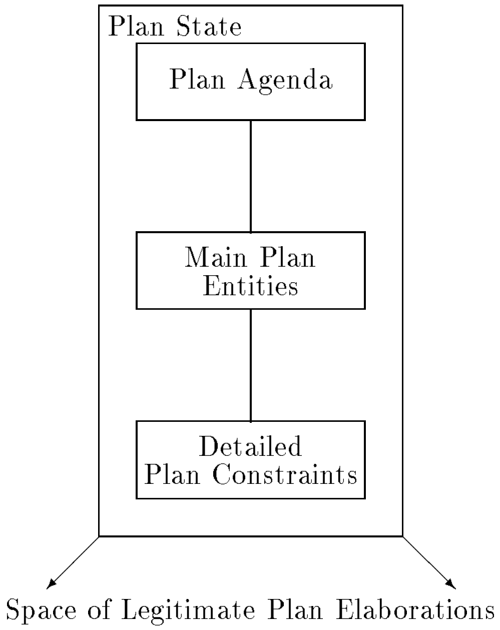

Figure 7: Various Plan Constraints Define a Space of Plan Elaborations

The three types of constraint in a plan are:

1. Plan Agenda – a set of "Issues" that must be addressed and thus define a set of "implied constraints" on legitimate future states. The agenda implies 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 agenda is a "to-do" list which can be used to decide what plan modifications should be made to a plan by a planner (user or system).

2. Main 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 Plan Constraints – specialised constraints on the plan associated with the Main Plan Entities. Work on the O-Plan planner has identified the desirability of distinguishing two special types of detailed constraint and categorising all others as "auxiliary":

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

Auxiliary Constraints are other detailed constraints related to input (pre-), output (post-) and protection conditions, and to resources, authority requirements, spatial constraints, etc. The auxiliary constraints are grouped into 4 categories:

- Authority Constraints
- World Condition/Effect Constraints
- Resource Constraints
- Other Constraints

Ordering and Variable constraints are highlighted since they may form part of other detailed constraints in a temporal reasoning domain such as occurs in planning and scheduling problems. That is, an auxiliary constraint may themselves involve ordering or variable constraints. Knowing that these constraints have such "cross-associations" has been found to simplify the design of constraint handling mechanisms and to ease implementation issues [33],[35]. It has also proved to be helpful in formalising planners and their plan representations (e.g., [23],[24]).

Auxiliary Constraints may be expressed as occurring at a time point ("point constraints") or across a range of points ("range constraints"). Point constraints can be used to express input and output constraints on nodes and 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.

There is a deliberate and direct mapping of the model of plans and activity used within O-Plan and the <I-N-OVA> Constraint Model of Plans to existing structured analysis and diagraming methods such as IDEF, R-Charts, etc. [34] **[See Related Paper I]**. Other researchers have also recognised the value of merging AI representation concepts with structured analysis and diagramming techniques for systems requirements modelling (e.g., [5],[25]).

6 Abstract View of the O-Plan Control Flow

The O-Plan research described in previous sections has allowed us to simplify previous descriptions of the O-Plan architecture, and in particular to present a simplified abstraction of the core working of an O-Plan agent.

O-Plan operates on a workflow principle, being driven by an agenda of "issues". A simple abstraction of this is shown in (figure 8).

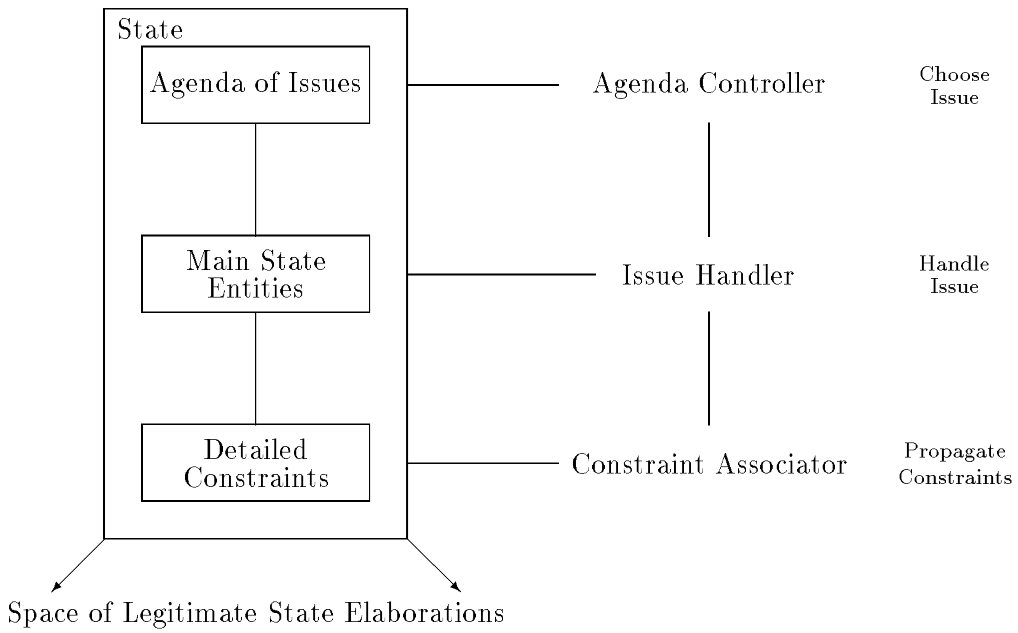

Figure 8: Framework of Components in the O-Plan Agenda-based System

O-Plan refines a "current state". It maintains one or more *options* within the state in which the previous alternative decisions that can be taken restrict the space of state elaborations which can be reached from that point[6]. The system needs to know what outstanding processing requirements exist in the state (shown in figure 8 as 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 *Agenda or Option Controller*). This calls up processing capabilities (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).

We have found it to be useful to separate the entities representing the decisions already made during processing into a high level representing the *Main State Entities* shared across all system

[6]State constraint relaxation may also be possible to increase the space of state elaborations in some cases.

components and known to various parts of the system, and more *Detailed Constraints* which form specialised areas of the representation of the state. These lower level more compartmentalised parts can represent specialised constraints within the state such as time, resource, spatial and other constraints. This separation can assist in the identification of modularity within the system.

7 Working with the User

An interface to AutoCAD has been built to show the type of User Interface we envisage. This is called the PlanWorld Viewer Interface [40] **[See Related Paper L]**. Figure 9 shows an example screen using this interface. The window in the top left corner shows the Task Assignment menu and supports the management of authority [32] to plan and execute plans for a given task. The lower window shows a *Plan View* (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 PostScript pre-viewers for plan networks, 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.

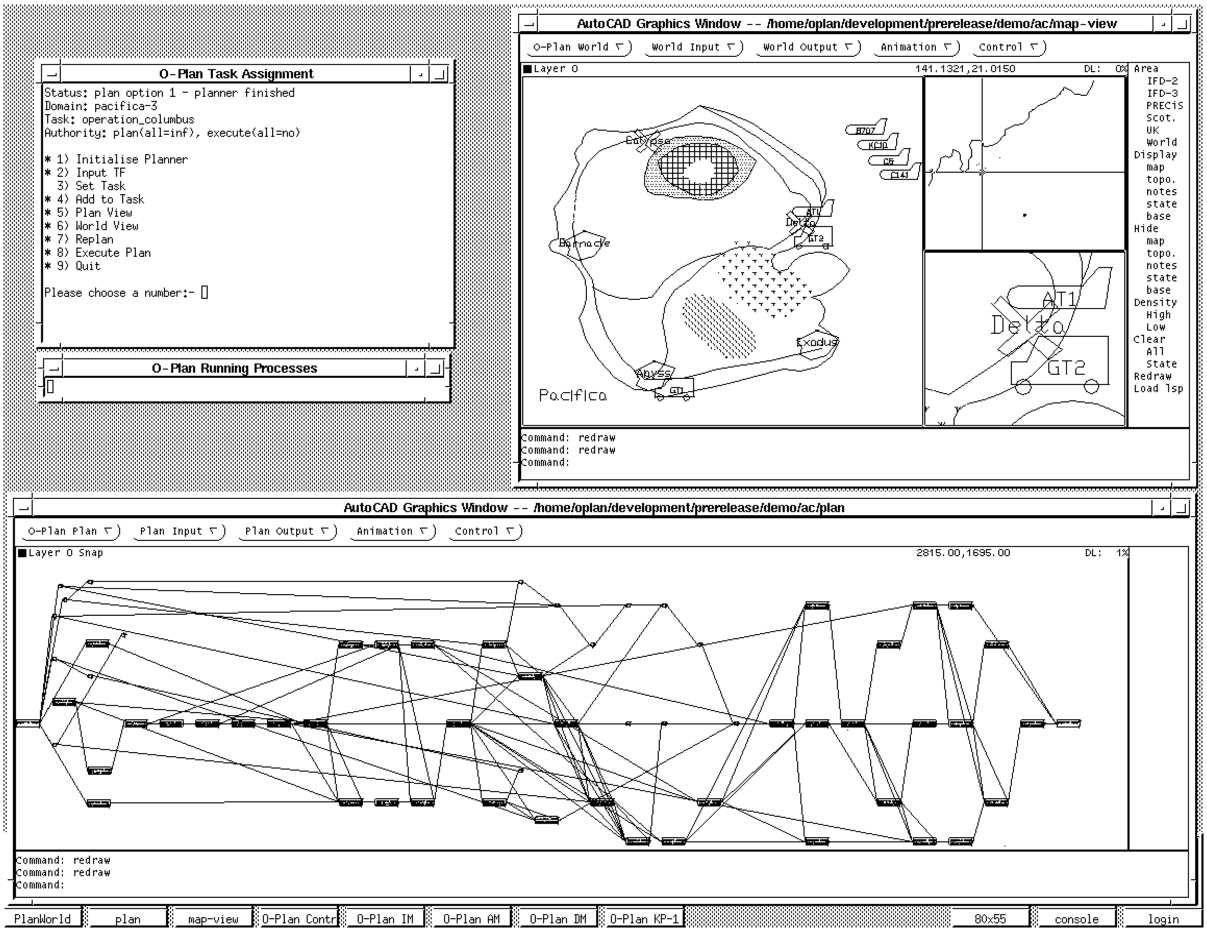

Figure 9: Example Output of the PlanWorld Viewer User Interface

Recent work on the O-Plan user interface has focussed on the representation and management of constraints in planning, particularly in order to simplify some aspects of the user's role in the architecture and to act as a mechanism for user/system mixed initiative planning [36] **[See Related Paper M]**.

8 Logistics Applications

8.1 Target Applications for O-Plan

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 platforms in the future.

O-Plan is intended 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 and ERS-1.

These applications fit midway between the large scale, but regular, 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 [3].

8.2 Crisis Action Planning

The application emphasis of the O-Plan project has been to aid in the definition, generation and support of the military crisis action planning process. There are six phases identified in reponding to a crisis as shown in figure 10.

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

Figure 10: Crisis Action Planning Phases

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

The requirements for task statement and plan generation by O-Plan in crisis action planning have been tested in the PRECiS domain [28] **[See Related Paper N]** and in a simplified version of an Integrated Feasibility Demonstration scenario (IFD-2) from the ARPA/Rome Laboratory Planning Initiative [20]. 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 that are flexible, robust and responsive to changing task requirements and to changes in the operational situation. Current planning aids are too inflexible. The aim of the O-Plan project is to show how a planner using extensive domain knowledge and situated in a task assignment (command) and execution (control) environment can allow for better flexible, distributed, collaborative and mixed initiative planning.

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, which is not currently viewed as an iterative process in which several sources of knowledge and techniques (e.g., planning, scheduling, tasking, resourcing, 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 and to consider more options and constraints where appropriate within the planning process.

8.3 The 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 provided by the PRECiS (Planning, Reactive Execution and Constraint Satisfaction) environment defines the data and hypothetical background for logistics planning and reacting scenarios and has been used for demonstration and evaluation purposes within the project.

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 (IFD-3) [20]; Reece and Tate to define an openly accessible fictional environment based on the island of Pacifica [26], 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 [28] and in 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 generation, 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.

8.4 Demonstration of O-Plan Generating Courses of Action for NEOs

The aim of this section is to describe the evaluation experiments conducted as part of the O-Plan project and to show how each one can be related to the categorisation of experiment types defined in the ARPI Evaluation Handbook [8]. The main demonstration domain has been Non-combatant Evacuation Operation (NEO) planning in the PRECiS environment.

A number of planned demonstrations were conducted at the end of each year of the project. In addition to these planned demonstrations a further demonstration was conducted to link the O-Plan system with a plan analysis tool (USC/ISI's EXPECT system) to show the value of being able to analyse plans and to provide feedback to the planner on how to improve the quality of the plans being generated.

The following items describe some of the experiments carried out in each of main categories of the ARPI handbook: *Programmatic*, *Demonstration* and *Scientific*. Each section describes the aims of the experiment, an overview of the approach and the results obtained. A full list of the experiments and their results can be found in [13] **[See Related Paper O]**.

Year 1 – 1993: Generation of Plans from the IFD-2 Scenario

The was a demonstration experiment using the IFD-2 SOCAP Tunisian scenario run on SIPE-2 [6]. From the start of the experiment it was recognised that SIPE-2 was a more developed system than O-Plan and as such this could only be an approximation to IFD-2. However, using the Task Formalism (TF) (O-Plan's domain input language) then supported within O-Plan Version 2.1 it was possible to encode the SOCAP domain and to identify a number of shortcomings in O-Plan TF [10, 11]. The schema library for this domain contained 63 schemas which defined alternative missions, deployment and employment plans, sea and airlift resources, etc. The Courses of Action (COAs) generated contained an average of 150 actions and were developed in approximately 40 seconds. O-Plan was able to generate plans in the SOCAP domain for two tasks:

- **Task 1**: "Deter three threats"
  The task requires a plan to deter one army, one air force and one navy threat by specified dates. The threats are forces which have crossed the protected border.

- **Task 2**: "Deter three threats and counter a further nine"
  The task requires a plan to deter the same three threats as well as countering a further nine threats: three army, navy and air force respectively. These nine forces are threatening to cross the border but have not yet done so.

The outcome of the experiment was that we identified a problem of incorrect constraint posting and as a result very large search spaces were being generated. The reason for the large search spaces was the combinatorics of the domain mainly arising from the codesignation of cross constraints involving time and resources. The research on the O-Plan architecture had identified the need for improved handling of these types of constraint but it was not reflected in the implementation. Similar problems were found in the earlier Nonlin system in a domain investigating the problem of Replenishment At Sea (RAS) [41]. In the RAS problem ships need to be moved from one battle group to another while others are sent for replenishment. Again the problem was one of selecting a particular ship too early rather than waiting until further constraints could be identified and posted.

Off line analysis of the problem showed that the problem could be solved with little or no search being involved. For example, many of the forces which could be chosen for a particular mission were similar and consequently the planner should have left the decision over which force to use until it was forced upon it, i.e., developing the force's employment plan.

Year 2 – 1994: Use of a Rich Resource Model in an Activity Planner Framework

This was a demonstration experiment and was designed to show the ways in which a rich model of domain resources, e.g., trucks, aeroplanes, fuel, runways, etc, could be encoded and used within an activity planner. As part of the preparation for the demonstration a study was carried out into the different types of resources present in planning domains and into previous planning approaches to resource reasoning [12]. The results of this study were twofold.

1. It became possible to identify the type of resource reasoning support which should be possible with an activity planning framework.
2. It resulted in the design of a flexible Resource Utilisation Manager (RUM) for use in an activity planner such as O-Plan and SIPE-2.

The support provided by the new RUM design would allow a range of resources types to be represented and manipulated and went beyond those types supported to date in other systems. The demonstration successfully showed that plans could be generated for a number of different resource constrained tasks specified in the PRECiS domain. The schema library for this domain contained 20 schemas which defined alternative evacuation methods, e.g., trucks or helicopters, fuel supplies, transport aircraft, etc. The COAs generated contained an average of 30 actions and were developed in approximately 40 seconds.

A number of techniques were explored and validated which showed how resources could be defined and manipulated using a range of methods. These methods made explicit use of O-Plan's Resource Utilisation Manager to track consumable resources and O-Plan's TOME and

GOST Manager to track reusable/sharable resources. Whilst these method allow the same breadth of coverage as was expected with the new RUM design, they do not have the same level of flexibility and support. In tasks where the resources were limited, e.g., small amounts of diesel fuel, the system was able to use knowledge of resources to rule out certain options as being impossible. In tasks where the choices were more extensive, e.g., use several transport types with no temporal restrictions, the system was still able to find a solution in an acceptable period of time.

Year 3 – 1995: Coordinated Command, Planning and Control

This was a demonstration experiment which showed O-Plan solving a number of tasks from a command, planning and control scenario. 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.

In both these cases the changes were to be made to an ongoing and executing plan.

The types of changes explored in this demonstration include failures of trucks due to blown engines and tyres and the inclusion of new objectives, e.g., pick up an extra group of evacuees. The PRECiS domain used for the demonstration has been deliberately simplified to allow a number of different aspects to be explored while keeping the plan to a manageable size. This is for viewing purposes only so that the user could follow what was happening in the demonstration. However, while being a simplification, the types of problem encountered and the solutions proposed by the planner are of relevance to military crisis action planning. Larger and more complex plans are available in other Pacifica domains. The schema library for this domain contained 12 schemas which defined alternative evacuation methods, e.g., trucks or helicopters, fuel supplies, transport aircraft, etc. The COAs generated contained an average of 20 actions and were developed in approximately 40-60 seconds. 4 different repair plans were used in the demonstration and they were as follows:

- To repair a blown engine on a ground transport.
  - The engine can only be fixed by a repair crew which is dispatched from Delta with a tow truck. The transport is then towed to Delta for repairs. The evacuees remain with truck 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.
  - This is similar to the previous repair plan except that the evacuees are moved by another ground transport instead of by helicopter.

- To repair 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.

Closely allied to the third year O-Plan demonstration, an associated Ph.D student project by Glen Reece showed the link between a proactive planner and a more comprehensive reactive execution agent [27] based on the O-Plan architecture. This agent 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.

Linking of O-Plan and the EXPECT Plan Analysis Tool

This was a demonstration experiment conducted with USC/ISI in which the O-Plan system was linked with their EXPECT plan analysis tool [14],[15]. The Tunisian scenario used for IFD-2 was chosen for the evaluation domain. The schema library for this domain contained 63 schemas which defined alternative missions, deployment and employment plans, sea and airlift resources, etc. The Courses of Action (COAs) generated contained an average of 150 actions and were developed in approximately 40 seconds. The different COAs were generated using alternative mission profiles and force packages.

	COA 1	COA 2	COA 3	COA4
AIRPORTS				
- number of airports	1	1	1	2
- sorties per hour	315	315	315	480
- sq. ft. aircraft parking	2M	2M	2M	3M
SEAPORTS				
- number of seaports	1	1	1	2
- number of piers	6	6	6	15
- number of berths	6	6	6	16
- max. vessel size in ft.	600	600	600	765
- number of oil facilities	1	1	1	3
CLOSURE DATE	C + 29	C + 22	C + 23	C + 23
LOGISTICS PERSONNEL	1154	5360	5396	7362
LINES OF COMMUNICATION				
- number of locations	1	5	7	6
- max. distance in miles	20	99	140	120
- air and sea?	yes	yes	yes	yes

Figure 11: EXPECT's Evaluation of Several Alternative Plans Generated by O-Plan

EXPECT allows military planners to analyse these alternative COAs generated by O-Plan against a number of user defined domain evaluation criteria and to create an evaluation matrix for a number of chosen COAs. From the analysis, military planners are able to identify aspects of the COAs which are acceptable, e.g., low number of support personnel and those which are not, e.g., a closure date greater than 29 days. An EXPECT evaluation matrix from a series of different COAs generated by O-Plan for a logistics scenario is shown in Figure 11. This information could then be used to impose addition requirements on the planning system to provide a better quality solution.

8.5 Bringing O-Plan Technology into Productive Use

The O-Plan system has been developed to be as modular as possible, with open interfaces to allow easy integration with the work of other ARPA/Rome Laboratory Planning Initiative participants. This has led to discussions with several groups within the Initiative and an exchange of ideas and research results. We see one of our contributions within the Initiative as providing a common framework in which the specialised contributions of different groups can be explored. We have also passed results to other ARPA programs under the ARPA Knowledge Sharing effort, e.g., to define the context handling facilities in the LOOM system. The widespread publication of the results of the project is the main way in which the project seeks to disseminate its results to the technical community.

The transition path to eventual productive use for O-Plan, and the concepts in O-Plan, is through a series of releases to the Common Prototyping Environment (CPE) of the ARPA/Rome Laboratory Planning Initiative [20] and through involvement in Technology Integration Experiments (TIEs) with other participants [6]. The transition should then involve the integration of aspects of the demonstrated technology into Integrated Feasibility Demonstrations (IFDs) within the Initiative.

O-Plan has been released in three annual versions to the ARPI CPE and through that has been made available to a number of sites. The latest version at the end of the project is O-Plan release 2.3.

A Technology Integration Experiment (TIE) has been conducted with USC/ISI to look at linking plan generation and plan evaluation (described in section 3.3). This is central to the O-Plan project's aim of situating planning in a task assignment and execution setting. It is a topic that is vital for successful military crisis planning and response. Extension of this joint work is now proposed.

The O-Plan project has also begun discussions to establish ways in which O-Plan could be incorporated into an Integrated Feasibility Demonstration (IFD) for air campaign planning within the ARPI.

9 Conclusions

The O-Plan research has achieved a clearer understanding of the components necessary in a flexible planning system and has shown how such components can be combined in a systems integration architecture. The work has determined improved ways to restrict search in a planner by using the knowledge available from modelling an application domain, and it has developed a better characterisation of plans as sets of activity constraints, opening up many possibilities for richer distributed, cooperative and mixed-initiative planning systems in the future. The project has created a prototype implementation and demonstrated it on a class of realistic applications.

References

[1] 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.), Published by Morgan Kaufmann Inc, San Francisco, USA, 1994.

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

[3] 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.

[4] 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.

[5] 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.

[6] 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: Intelligent Systems and their Applications*, Vol. 10, No. 1, pp. 17-26, February 1995, IEEE Computer Society.

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

[8] Cohen, P., Dean, T., Gil, Y., Ginsberg, M. and Hoebel, L. Handbook of Evaluation for the ARPA/Rome Laboratory Planning Initiative, February, 1994.

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

[10] Drabble, B., Conversion of SIPE-2 Domain Descriptions to O-Plan Task Formalism, O-Plan Technical Report ARPA-RL/O-Plan/TR/1, March 1993.

[11] Drabble, B., Applying O-Plan2 to Military Logistics Planning, O-Plan Technical Report ARPA-RL/O-Plan/TR/9, September 1993.

[12] Drabble, B. *Comparison of the CAMPS system with O-Plan2: Architecture and Reasoning Capabilities*, O-Plan Technical Report ARPA-RL/O-Plan2/TR/12 Version 1, January 1995.

[13] Drabble, B. and Tate A. Applying O-Plan to the NEO Scenarios, O-Plan Technical Report ARPA-RL/O-Plan/TR/23,July 1995.

[14] Drabble, B. and Gil, Y. Acquiring Criteria for Plan Quality Control, Proceedings of the AAAI Spring Symposium workshop on Integrated Planning Applications, June 1995. Stanford Univeristy, CA, USA. Published by the American Association for Artificial Intelligence, Menlo Park, California.

[15] Drabble, B. and Gil, Y. Yes, but why is that plan better?, Proceedings of the International Conference on Artificial Intelligence in the Petroleum Industry, September 1995, Lillehammer, Norway.

[16] Drabble, B. and Kirby, R., Associating AI 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.

[17] 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.

[18] 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.

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

[20] 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.

[21] Gil, Y. Knowledge Refinement in a Reflective Architecture, 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.

[22] 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.

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

[24] 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.

[25] Ramesh, B. and Dhar, V., Representing and Maintaining Process Knowledge for Large-Scale Systems Development, IEEE Expert, pp. 54-59, April 1994.

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

[27] 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.

[28] 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.

[29] Reece, G.A. and Tate, A., Synthesizing Protection Monitors from Causal Structure, Proceedings of the Second International Conference on AI Planning Systems (AIPS-94), pp. 146-152, AAAI Press, Chicago, USA, 1994.

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

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

[32] 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.

[33] 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.

[34] Tate, A., Putting Knowledge Rich Process Representations to Use, *IOPener – The Journal of the IOPT Club for the Introduction of Process Technology*, Vol. 2 No. 3 pp 12-14, March 1994, UK Introduction of Process Technology (IOPT) Club, c/o Praxis Ltd, UK.

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

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

[37] 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.

[38] 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.

[39] Tate, A., Integrating Constraint Management into an AI Planner, *Artificial Intelligence in Engineering*, Vol. 9, No. 3, pp 221-228, 1995.

[40] 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.

[41] Tate, A. and Whiter, A., Multiple Resource Constraints and an Application to a Naval Planning Problem, Proceedings of the First Conference on Artificial Intelligence Applications, pp. 410-416, AAAI, Denver, Colorado, USA, December 1984.

[42] 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.

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

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

Appendix

Figure 12 shows how a number of related papers relate to the main themes of the O-Plan research.

PAPER A: 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.

Provides an overview of the O-Plan architecture with its task assignment, planning and execution agents, giving information about the aims of the work and its applications.

PAPER B: 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.

Provides an overview of the module specifications, interfaces and protocols used within the O-Plan architecture.

PAPER C: 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.

This PAPER explains the importance of exploiting the task assignment and execution framework within which a planner is situated. The benefits of being able to use a rich model of this environment are explained.

PAPER D: 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.

O-Plan can make use of domain knowledge of various kinds to restrict its search for plans. It can thus be applied to larger problems than would otherwise be the case. This paper describes one strong contribution of the O-Plan research to finding ways to encode domain knowledge in forms which can be used by a planner.

PAPER E: 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.

Accounting for resource availability is an important requirement when planning. This paper describes the novel mechanisms used within O-Plan for managing resources using incremental algorithms. The generic interface to such constraint managers within O-Plan is also described.

PAPER F: 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.

In a cooperative planning environment, a planner should not be considered to work in isolation, simply producing plans. This paper describes early work on modelling the role

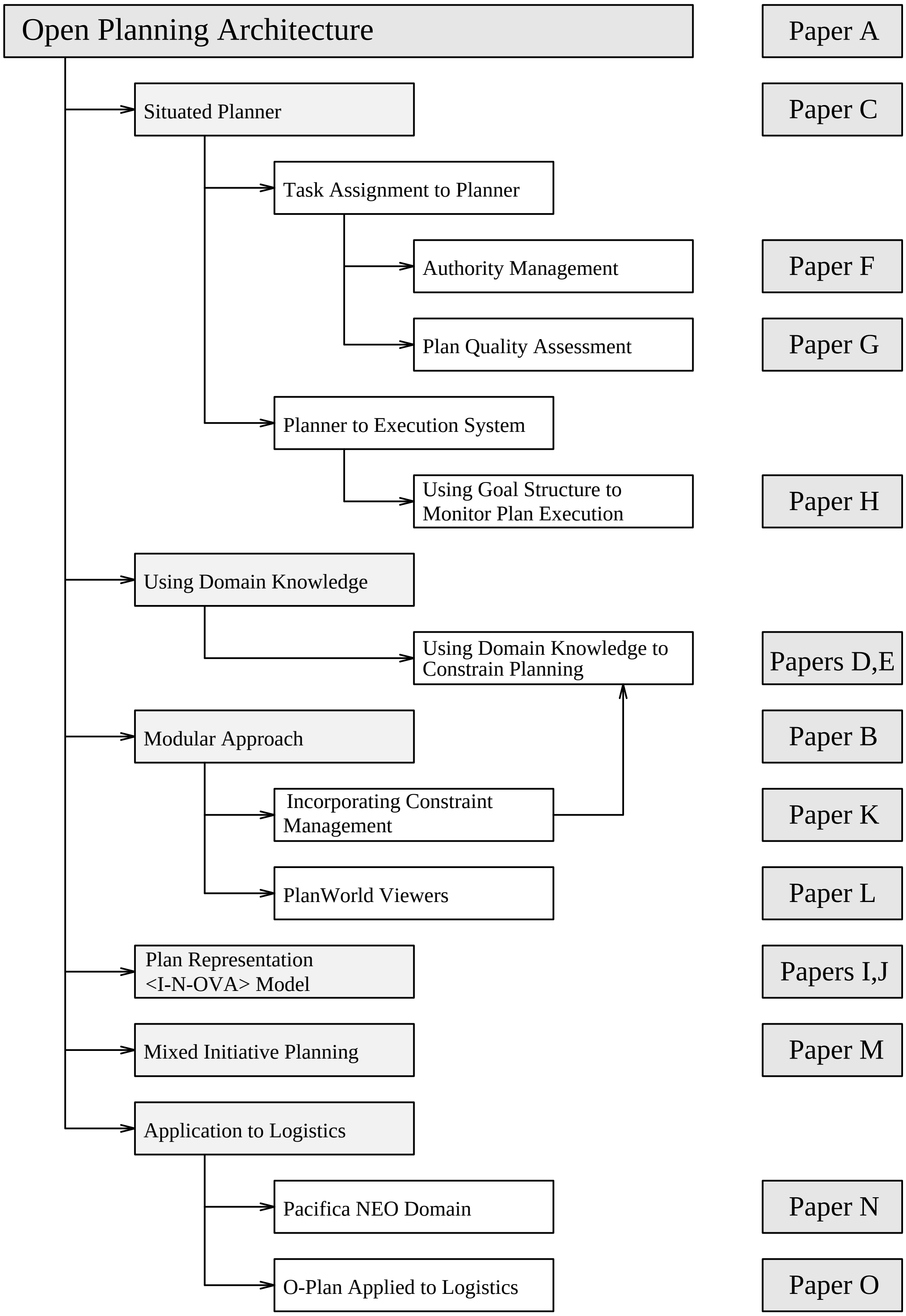

Figure 12: Road Map to Related Papers

of authority and delegation within a command, planning and control environment in such a way that it can be used to effectively coordinate planning activities.

PAPER G: 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.

Producing useful, effective plans requires improved information about the quality of plans and about the features of plans from which quality can be determined. This paper explores the general factors that may be used to analyse differences between plans and examines a specific domain of military Non-combatant Evacuation Operations (NEOs) to provide examples of domain criteria used to assess plan quality.

PAPER H: 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.

O-Plan produces plans that can be executed. Work in the O-Plan project on a Reactive Execution Agent was performed on a linked Ph.D. This paper describes the use for plan execution support of knowledge embedded in O-Plan plans that captures the rationale, or Goal Structure, of the plan steps. It is shown that reactive plan change support can be provided using such knowledge.

PAPER I: Tate, A., Putting Knowledge Rich Process Representations to Use, *IOPener – The Journal of the IOPT Club for the Introduction of Process Technology*, Vol. 2 No. 3 pp 12-14, March 1994, UK Introduction of Process Technology (IOPT) Club, c/o Praxis Ltd, UK.

The knowledge rich plan structures used within O-Plan are themselves of value in other contexts. This paper describes how the O-Plan plan model can support improved process modelling, analysis and workflow in organisations.

PAPER J: 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.

In order to promote convergence of work in software engineering, process management, AI planning and formal mathematical work on planning, a model of plans as a set of constraints on behaviour or activity is being explored by the O-Plan project. This paper describes the <I-N-OVA> constraint model employed and relates it to other work.

PAPER K: Tate, A., Reasoning with Constraints in O-Plan2, Extended version, containing a number of additional sections, of a paper in the Proceedings of the ARPA/Rome Laboratory Planning Initiative Workshop, (M.Burstein, ed.), Tucson, Arizona, USA, Morgan Kaufmann, 1994.

The O-Plan research has simplified the way in which detailed constraints within plans can be managed, and has introduced a way in which constraint managers could be plugged into a planner using a well defined protocol. This paper describes the approach adopted.

PAPER L: 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.

The user interface to O-Plan makes use of plug-in viewers which can support technical plan views (e.g., workflows and charts) or domain related world views (e.g., maps and animations). This paper describes the ways in which a user can interact with O-Plan and describes the PlanWorld Viewer interface.

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

Planning is not done as an isolated activity. It relies on cooperative work between many people and systems. This paper describes O-Plan's approach to mixed initiative and co-operative planning. It is based on the mutual process of placing constraints on behaviour.

PAPER N: 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.

O-Plan research and the prototype implementation have been applied to specific logistics problems related to military Non-combatant Evacuation Operations (NEOs). This paper describes a non-confidential demonstration and test environment and example scenarios in which the requirements for O-Plan were established and demonstrations provided. It involves NEOs from the fictional island of Pacifica.

PAPER O: Drabble, B., Tate, A. and Dalton, J. Applying O-Plan to the NEO Scenarios. O-Plan Technical Report, Artificial Intelligence Applications Institute, University of Edinburgh, July 1995.

This paper describes and evaluates the application of O-Plan to the PRECiS/Pacifica NEO Scenarios.