Technical Report

O-Plan Tasking Specification

Brian Drabble, Jeff Dalton and Austin Tate

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

January 7, 1997

ARPA-RL/O-Plan/TR/29 Version 1

Contents

1 Introduction

The aim of this report is to describe the task and option management capabilities being developed for the O-Plan system to allow it to be used in a MIP framework to support Air Campaign Planning (ACP). During first two phases of the DARPA/Rome Laboratory Planning Initiative (ARPI) the O-Plan project concentrated on the development of a modular planning architecture and on the execution and repair on plans. During the current phase of the ARPI the focus of the research is into the roles of the Task Assigner (TA) and the O-Plan Planner agent and the communications link between them. The aims are to:

- identify the needs of the TA user and the support which can be provided;
- identify the type of information that needs to be communicated from the TA to the planner;
- identify ways in which the planner can make use of this information.

The final aim of the research is to build a Mixed Initiative Planning (MIP) system which allows a user (in various commanding and support roles) to work cooperatively with a planning system to solve a common task. Previous research [1] has already shown that O-Plan has a number of MIP capabilities and is capable of supporting a number of user roles [2]. Current research with the University of Rochester (through TIE#3) is investigating the problem solving model and communications needs required to link the TRAINS and O-Plan systems. This would provide a testbed for investigating the needs and requirements of an MIP system.

Central to the development of such a MIP system is the ability of the user (through the TA) to specify tasks and options for the planner to explore. Tasks and Options are central to any problem solving strategy involving both human and software agents. The options allow the user to explore different overall solutions and to return to them for further refinement later. For example, a user in the Air Campaign Planning (ACP) domain may require several options for the overall task of achieving air superiority over the theatre of operations. The options may involve alternative levels of resources e.g. planes, airfields, missiles, etc and alternative temporal constraints e.g. achieve air superiority with 72 hours from the start of conflict. In addition to exploring different overall options the user may also wish to freeze "part" of solution and concentrate on a particular aspect. For example, a user in the air campaign domain may wish to explore different sub-options for electronic combat planning while leaving the rest of the support planning unaltered e.g. refuelling, logistics, etc.

The current O-Plan system has a simple menu driven TA interface which allows the TA user to specify TF domain descriptions and tasks, request plan and world views and request the plan to be executed. The aim is to increase the functionality of this interface by allowing the TA user to specify and name options, create new sub-options, request plans and replans within an option and modify/edit a named option. Details of these aspects of the TA and planner functionality will be dealt with in later sub-sections. The structure of the report is as follows. Section 2 provides an overview of the tasking support provided by the system and Section 3 provides an overview of the option support. Section 4 provides a summary of work to date and provides details of future developments and extensions of the work.

2 Tasking Information

This section provides details of how the ideas of multi-perspective planning could be realised in an actual architecture and provides an overview of the types of system and interactions which would take place. As described in the previous section one of the main interactions is between the task assigner and the planning system. Central to this interaction is the specification of the task the task assigner wishes the planner to consider and the current option being explored. For example the task could be "gain air superiority over the amphibious landings area" and the option may have specific assumptions about the level of the forces to be committed, the different time scales, types of operation allowed, etc. In addition the task assigner may specify information which guides and/or limits the ways in which the planner can address the task. For example, the task assigner may require:

- certain named phases of the air campaign to be planned and others ignored.
- the plan to be created to a specified level e.g. CONPLAN
- certain types of planner activity to be suspended, e.g. do not assign a particular aircraft merely ensure that one is available.
- specific milestones to be achieved during the planning process, e.g. all employment activities are complete and that deployment activities should be completed only to the CONPLAN level.

Figure 1 provides an overview of the architecture for a mixed initiative planning systems and shows how O-Plan, ISI's EXPECT system and the University of Rochester's TRAINS system could be used to instantiate such an architecture. Figure 1 also shows two "abstract" editors which conceptually will be used to specify the task and option information and to support display to the user of the different Courses of Action (COAs) generated by O-Plan. Whether these editors are actual system components will depend upon the functionality attributed to EXPECT, TRAINS and O-Plan. The COAs generated by O-Plan will be used to populate the columns of the Course of Action/Elements of Evaluation comparison matrix and the rows will be defined by elements of evaluation appropriate to the domain being analysed. The main functions of the two editors are described in the following two subsections.

2.1 Task and Option Editor

The function of the task and option editor is to allow the user in the planner role to define the task(s) and options which they wish to explore and to create this in a form which can be given to O-Plan. The main functions of the editor are as follows:

- **Create an Option**
  This function allows the editor to:
  - Create an option name.

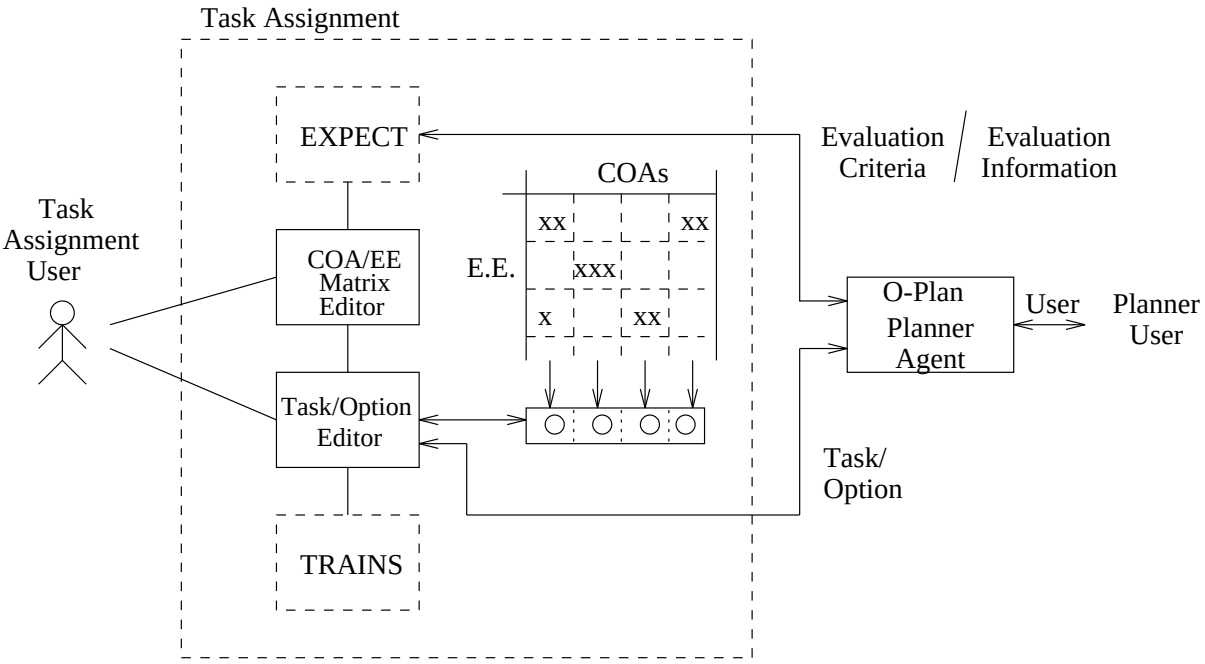

Figure 1: Communication between the Task Assigner and Planner

- Associate a specific task with an option name. This will be part of the initialisation process and will specify the task "seed" and the initial plan agenda. O-Plan will maintain separate named options, each with their own current plan and alternatives.
- specify a previously named option within O-Plan and to request that it be set up as the new current option.
- specify that a previously named option be re-initialised i.e. the task and option information remains unchanged but all alternatives and the current plan are removed.

- **Modify an Option**
  This function allows the editor to:
  - Add further constraints to the current plan in the current option. The constraints will be specified using the <I-N-OVA> constraint model of plans and will include further issues, nodes, ordering, variables, resource, authority constraints which are to be included in the option. Planning will then continue from the current plan for that option (where appropriate) or from some other alternative plan state. In some cases this may be implemented by re-initialising the option and starting the planning from scratch.

- **Refine an Option**
  This function allows the editor to:
  - refine a previously named option by adding further constraints to its current plan. The refinement process occurs when a top level plan has been defined and the user

wishes to explore the detail of the option further. The editor will allow the user to specify the ways in which the option should be refined and these include:

* refinement to a specified planning level (named by the user). The levels are consistent to the user and must be named by the user[1]. This could be achieved via an authority statement that planning was authorised down to a specific level.
* refinement of a specified phase (named by the user). This is associated with a node number in an already available option. Again this could be achieved via an authority statement that a phase is authorised for processing.
* Refinement of one or more specified constraints in the plan, e.g. nodes, variables, etc. This could be achieved via an authority statement that certains types of constraint were authorised for processing.

In each case the refinement information will be specified using the <I-N-OVA> constraint model of plans.

- **Replan an Option**
  This function allows the editor to:
  - Request a further plan from a previously named option and to leave all constraints unchanged. Replanning will continue from an alternative specified via the user or from one selected by the system.
- **Make Plan a Task Specification**
  This function allows the editor to:
  - Make the current plan with the current plan option into a task specification. In converting the plan into a task all previous alternatives associated with the plan are no longer available.
- **Obtain Planning Information**
  This function allows the editor to obtain information about the progress of the planning process and issues which remain. The information available is as follows:
  - Which of the defined milestones have been reached.
  - The status of the currently triggered and untriggered agenda entries together with their type e.g. expand, bind, or-tree, etc.
  - Number of alternatives which have been generated.
  - Number of sub-options which have been generated.
- **Obtain Plan Information**
  This function allows the editor to obtain information about the phases and levels in the plan. The information available is as follows:
  - Teleological information concerning the structure and dependencies in the plan. This will allow the user to identify possible "weaknesses" in the plan and to suggest further constraints to be imposed on the option.

[1] A simple mechanism for achieving this is to name levels via the Task Formalism definition of the domain.

  - Technical views for a previously named option.
  - Resource information and projections for resources utilisations at different points in the option

- **Define Milestones**
  This function allows the editor to define specific *testable* meta-planning milestones which must be met at different points in the plan. The types of milestones which can be specified are:

  - **Activity Template Expansion**
    Some activities should be describes using specific templates/operators and these should be specified by the user.
  - **Resource Template Expansion**
    Some activities describe incidental abd abstract resource needs. By expanding these most of the resource needs of the task can be identified.
  - **Preliminary Activity Parameterisation**
    Some of an activities parameters are needed to select appropriate resource types and further activity decompositions. This milestone is passed when the specified parameters are known.
  - **Find Temporal Bounds**
    Given the precedence structure in the plan start and end times can be found for specified activities.
  - **Estimate Resource Allocations**
    The assignment of resources could be achieved in two steps:
    * approximate times for the start and end of the activity are specified with possible overbookings allowed.
    * this identifies which resources are in short supply and where the shortages occur.
  - **Schedule Activity**
    An activity is scheduled when specific times are derived for its start and end times.
  - **Reserve Resources**
    Given a resource allocation to a specific activity over a time period is there an allocation of resources which are not over booked.
  - **Final Activity Parameterisation**
    When resources have been assigned and an activity scheduled there may be some final parameterisation required.

- **Housekeeping Information**
  This function allows the editor to specify housekeeping information to O-Plan which includes initialising itself to accept further task information and quitting when planning has been completed.

2.2 COA and Elements of Evaluation Matrix Editor

The function of the COA and Elements of Evaluation Matrix (COA/EE) Editor is to provide the user in the planner role to visualise the different COAs being generated. The main function of the editor are as follows:

- **Refine Evaluation Information**
  This function allows the editor to:
  - Obtain further plan evaluation information from the user and gain an understanding of the ways and circumstances in which the evaluation information can be used in the plan evaluation process.
  - Specify a plan evaluation function which allows O-Plan to make better informed decisions which reflect the requirements of the user. For example, the user may wish to reduce resource utilisation in the first phase of the operation, prefer to use cruise missiles against a certain class of target, etc. The evaluation function will be changed (as the user focuses on different elements of evaluation) and refined as the user explores different tasks and options.

3 Option Management

This section describes the option management functionality being developed for the O-Plan system. The aim is to provide a TA user with the ability to create and name an option within the search space to incrementally create new options and sub-options as needed. For example, a user in the ACP domain may wish to explore different air to air refuelling options while keeping other commitments fixed e.g. to eliminate the enemies ability to produce weapons of mass destruction by destroying the power distribution system. The ability to change the focus of problem solving and to incrementally add new tasks were seen as two of the major changes required in the TA planner interface.

To support option management within O-Plan an Option Table has been developed with contains:

- **Option Name**:
  The name of the option provided by the TA user.
- **Root Context**:
  The node in the O-Plan context tree which serves as the root of the alternatives tree for this option. Different options may share the same root context node.
- **Current Plan Context**:
  The node in the context tree which contains the current plan for the option. If no plan has been generated then this will be root context by default.
- **Sub-options Created**:
  A list of the sub-options which have been created from this option. These will be pointers to other rows in the option table. This allows each sub-option to have its own root context and to be treated as separate entity from its parent.

3.1 Sub-Option Support

One of the main concerns in developing the option management capability was to allow the TA user to incrementally create sub-options to explore a specific aspect of the problem e.g. different air to air refuelling options. This was fairly easy to achieve with minimum changes to the current O-Plan system. However, the situation becomes more complex if the user wishes to nest sub-options i.e. while exploring a new sub-option a user may wish to create further sub-option(s). This is described in Figure 2 whichs shows the option space of `Sub-Option 2` being totally contained within that of `Sub-Option 1`. The nodes represent points in the search space at which an alternative was available to the planner, e.g. alternative schema, variable binding, plan linking, etc. The user may wish to create a sub-option to explore one of the alternatives available while leaving the rest of the plan unaltered. While exploring `Sub-Option 1`, O-Plan is only allowed to back track to alternatives within that sub-option. If the planner exhausts all planning alternatives within a sub-option then it is allowed to choose alternatives in the alternatives space of its parent. For example in Figure 2 O-Plan could choose alternatives from

`Sub-Option 1` only after it had exhausted all possibilities from `Sub-Option 2`. O-Plan handles this by explicitly maintaining a pointer from the parent to any sub-options which have been created. This link can then be traversed to reach the alternatives (if available) in the parent.

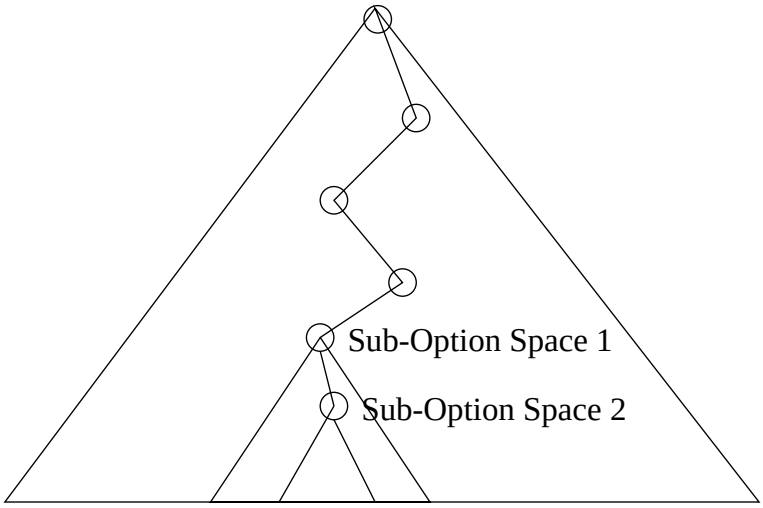

Figure 2: Option and Sub-Option Generation

This approach allows fairly simple option and sub-option management to be developed but does leave a number of other issues to be addressed. The main one is dealing with the inheritance of values through the context tree. The current context mechanism within O-Plan does not propagate downwards through the context tree the changes made to a pattern's value at a higher level. This means that the changes the user expects to be available in the sub-options will not be present and the plans generated may not meet the needs of the user. A simple scheme has been suggested to overcome this problem and will be tested in future versions of the O-Plan system.

3.2 Communication and Processing Requirements

The development of a more expressive communication language between the TA and the planner raised new issues concerning the ownership of the information being communicated and the method which should be used to route it to its destination. The various methods which were explored we as follows:

1. **Direct Controller Intepretation**
   Messages from the TA are handled directly by the agenda controller with all datastructures (e.g. the option table) being owned by the agenda controller. This does not invalidate the processing "flow" of the O-Plan architecture as the design allows for the capabilities to be "run" on different components of the system.

2. **Single Knowledge Source**:
   A single knowledge source would be provided to deal with all message from the TA. This method was discounted due to the need to place an additional entry on the agenda and to use an additional problem solving cycle to process it.

3. **Multiple Knowledge Sources**:
   A separate knowledge source would be provided for each message type set from the TA. This method was discounted due to the large number of knowledge sources which would be needed most of which would be almost identical.

4 Summary and Future Work

This section provides a summary of the work to date and pointers to future potential extensions and directions.

The report described a number of functional extensions to the current O-Plan system. The aim is to extend the current task assigner (TA) functionality and the interface between the TA and the planner. The changes to the TA functionality are motivated by:

- the problem solving model being developed by the University of Rochester through their TRAINS system. This aims at providing a natural language style interface to a problem solving system which provides a richer style of interactive problem solving, i.e. changing the focus of problem solving, adding/deleting constraints from the task specification, creating new options and sub-options to explore different aspects of the search space, etc.
- the O-Plan project's focus on the development of a Mixed Initiative Planning framework which allows for cooperative problem solving between human and computer agents. This framework allows for issues to be delegated (where appropriate) to a capable human and/or software agent and for problem solving to be focussed around the issues and problems identified by the different systems.

Work to date has focussed on identifying the areas of commonality between O-Plan Version 2.3 (the release of the O-Plan system to the ARPI in July 1995) and the Rochester model of problem solving. The aim was to identify:

1. those areas already covered in O-Plan version 2.3
2. where possible, simple ways of providing support in the short term e.g. allowing only the addition of new activities and constraints and not their removal.
3. those areas where the project has no current plans to provide support, e.g. full dependency recording to allow arbitrary decisions to be changed and their consequences altered.

This was achieved through a number of discussions between the O-Plan team members and the results have been fed back to the Rochester team.

Development work has already begun with O-Plan version 3.1 with the aim of providing a simple task and option interface which can be used to support interactions and TIE work with the University of Rochester. The aim of this work is to provide an intergrated environment within which experimentation can begin to understand the needs of a Mixed Initiative Planning (MIP) system. The development of a MIP framework is an important long term goal in other DARPA programmes aimed at developing components for the next generation command and control support tools such as JPT, JFACC and the JTF ATD. An important short term goal is to show the use and potential benefits of an MIP framework and to be in a position to offer such as framework as the focus of ARPI's IFD-6.

References

[1] Drabble, B. and Tate, A., O-Plan Mixed Initiative Planning Capabilities and Protocols, O-Plan Technical Report ARPA-RL/O-Plan/TR/24, November 1995.

[2] Tate, A. and Drabble, B. *PlanWorld Viewers*, in the proceedings of the Fourteenth UK Special Interest Group on Planning and Scheduling, (ed. Steel, S.) Wivenhoe House Conference Centre, Essex University, November 1995.