O-Plan: the Open Planning Architecture

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

The O-Plan Project at the Artificial Intelligence Applications Institute of the University of Edinburgh is exploring a practical computer based environment to provide for specification, generation, interaction with, and execution of activity plans. O-Plan is intended to be a domain-independent general planning and control framework with the ability to embed detailed knowledge of the domain. The system combines a number of techniques:

- A hierarchical planning system which can produce plans as partial orders on actions.
- An agenda-based control architecture in which each control cycle can post pending tasks during plan generation. These pending tasks are then picked up from the agenda and processed by appropriate handlers (*Knowledge Sources*).
- The notion of a "plan state" which is the data structure containing the emerging plan, the "flaws" remaining in it, and the information used in building the plan.
- Constraint posting and least commitment on object variables.
- Temporal and resource constraint handling using incremental algorithms which are sensitively applied only when constraints alter.
- O-Plan is derived from the earlier Nonlin planner from which it takes and extends the ideas of Goal Structure, Question Answering (Modal Truth Criterion) and typed conditions.
- We have extended Nonlin's style of task description language Task Formalism (TF).

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

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

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

The Scenario

- A user specifies a task that is to be performed through some suitable interface. We call this process *task assignment*.
- A *planner* plans and (if requested) arranges to execute the plan to perform the task specified.
- The *execution system* seeks to carry out the detailed tasks specified by the planner while working with a more detailed model of the execution environment.

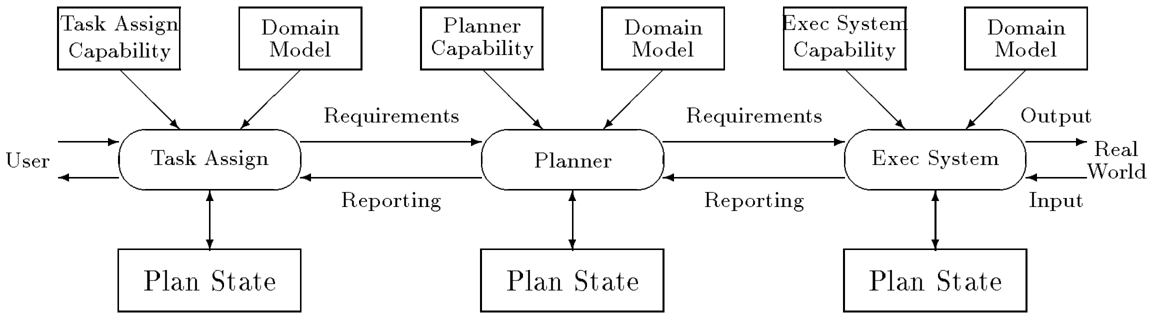

Figure 1: Communication between Task Assignment, Planning and Execution Levels

We have deliberately simplified our consideration to three levels with these different roles and with possible differences of requirements for user availability, processing capacity and real-time reaction to clarify the research objectives in our work. However, we believe that the ideas are relevant to the more general case of a *cooperative, hierarchical and distributed command, planning and control* environment.

A common representation is sought to include knowledge about the capabilities of the task assigner, the planner and the execution agent, and the information used to represent the requirements of the plan and the plan itself either with or without flaws (see Figure 1).

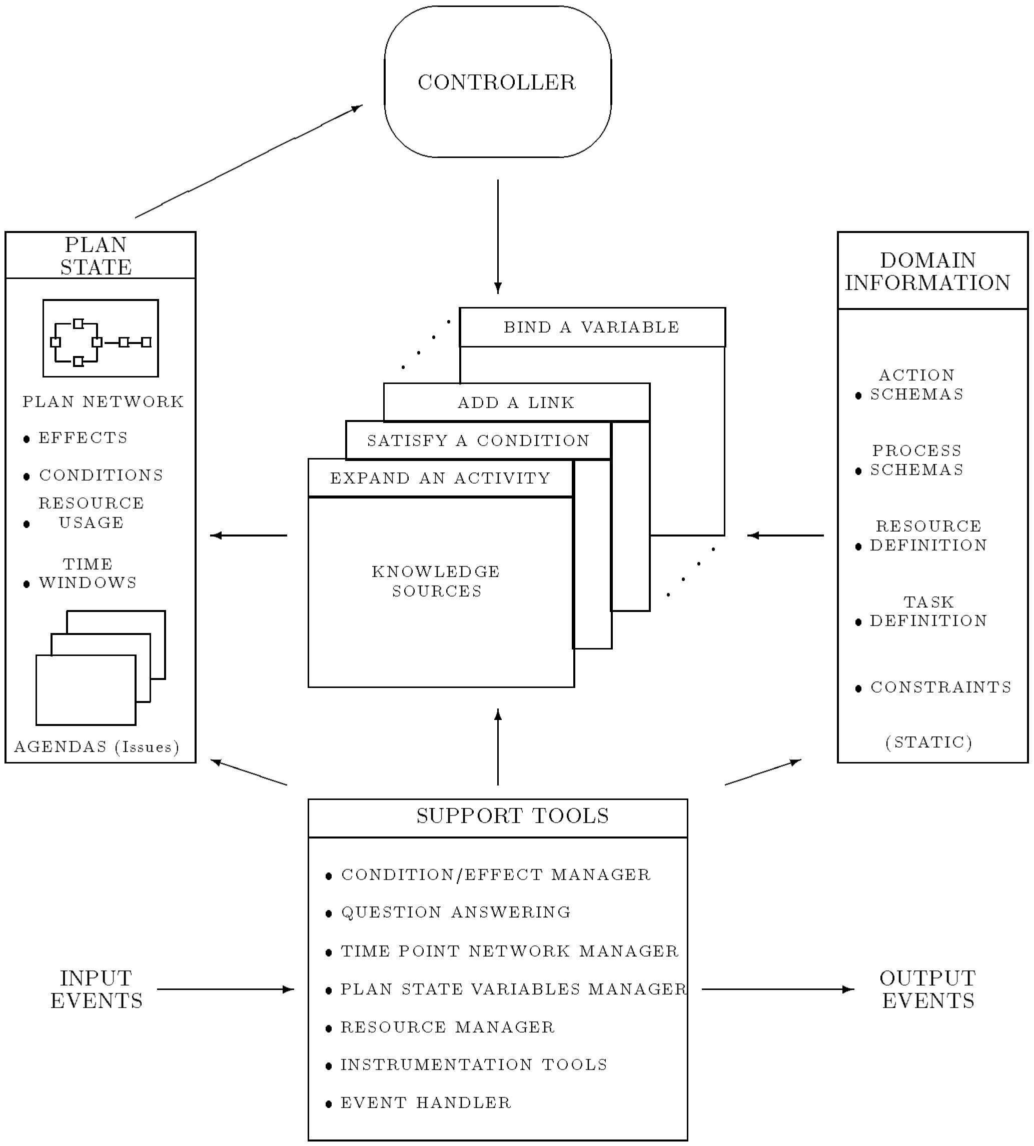

Figure 2: O-Plan Architecture

Developer Interface

O-Plan is implemented in Common Lisp on UNIX Workstations with an X-Windows interface. It is designed to be able to exploit multi-processors in future and thus has a clear separation of the various components (as shown in Figure 2). The design allows for each of these to run on a separate processor and multiple *platforms* may be provided to allow for parallelism in knowledge source processing in future. A sample screen image as seen by the O-Plan developer or an interested technical user is shown in Figure 3.

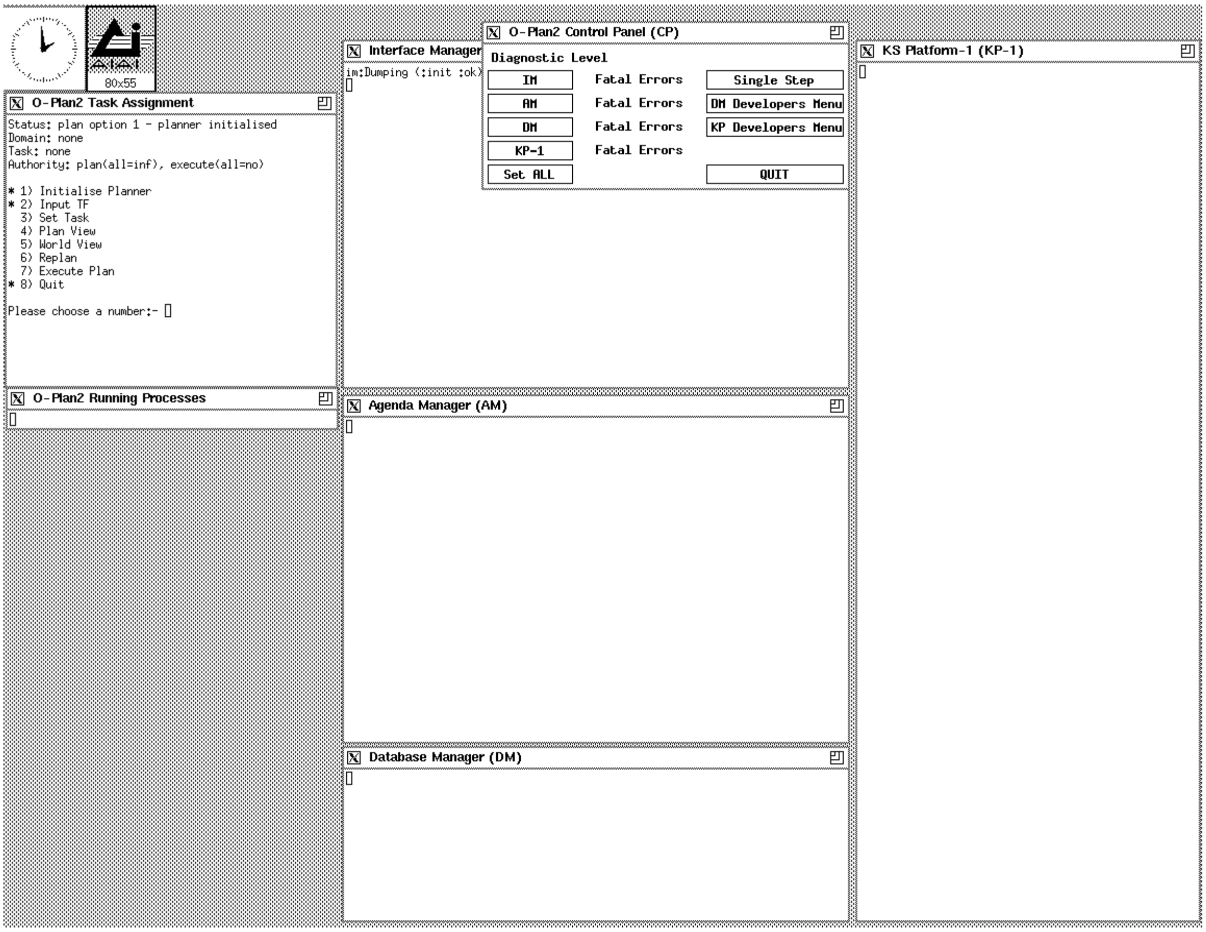

Figure 3: Example Developer Interface for the O-Plan Planning Agent

User Interface

AI planning systems are now being used in realistic applications by users who need to have a high level of graphical support to the planning operations they are being aided with. An interface to AutoCAD has been built to show the type of User Interface we envisage (see Figure 4). The lower window is the *Plan View*. This shows the plan as a graph and may be used to show other plan related displays such as PERT and Gantt charts, resource profiles, etc. The upper right window is the *World View* and can be used for simulations of the state of the world at points in the plan. The Plan Viewer and World Viewer capabilities are declared to the system and the interface between these and the planner uses a defined interface.

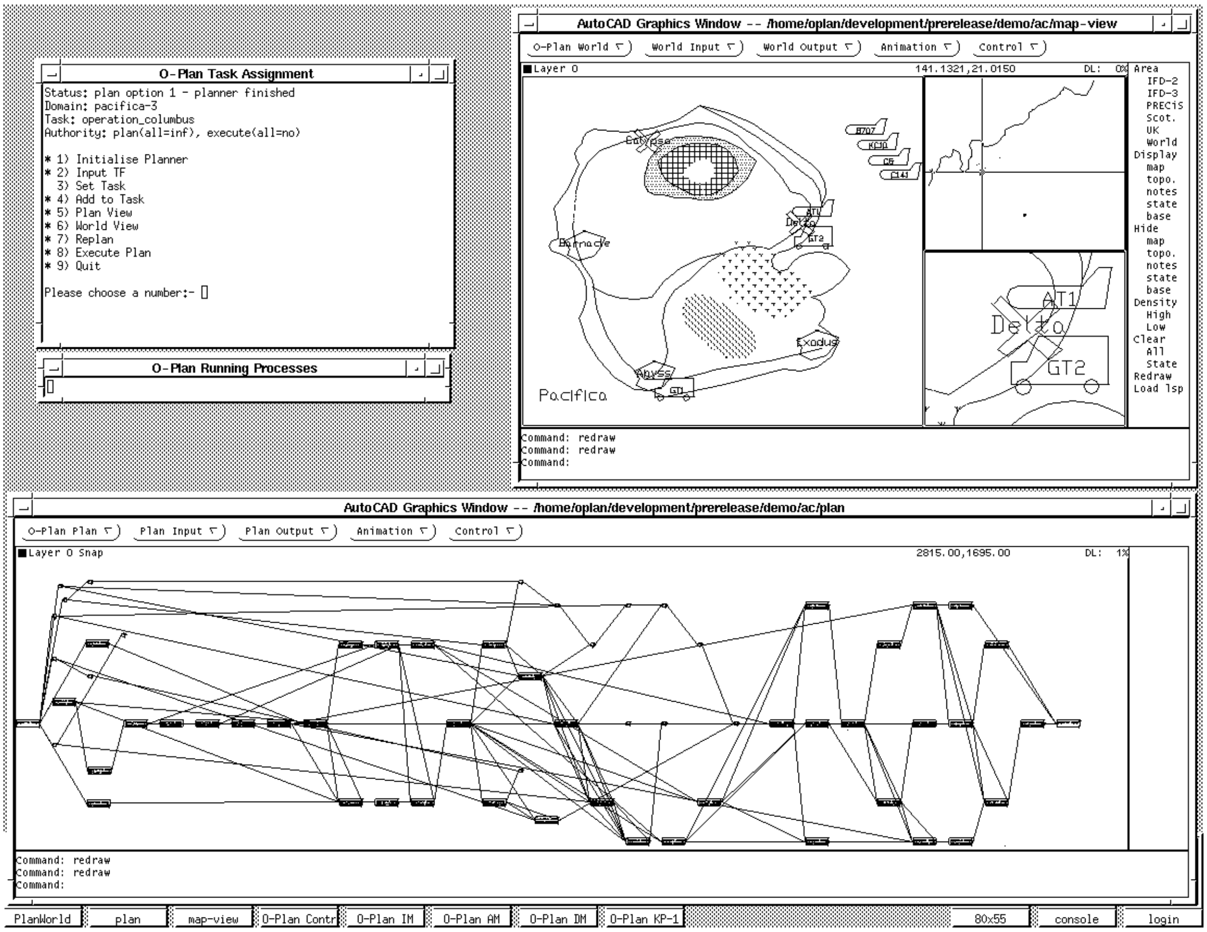

Figure 4: Example Screen for the AutoCAD-based PlanView User Interface