

59-63

137245

N 93 - 1865

O-Plan2: The Open Planning Architecture

Brian Drabble, Richard Kirby and Austin Tate
Artificial Intelligence Applications Institute
University of Edinburgh
80 South Bridge
Edinburgh EH1 1HN
United Kingdom

The O-Plan2 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-Plan2 is intended to be a domain-independent general planning and control framework with the ability to embed detailed knowledge of the domain.

- 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. The algorithms for this are incremental versions of Operational Research methods.
- O-Plan2 is derived from the earlier Nonlin planner from which extended the ideas of Goal Structure, Question Answering and typed conditions.
- We have extended Nonlin's style of task description language Task Formalism (TF).

O-Plan2 could be applied to the following types of problems:

- planning and control of space probes such as VOYAGER, etc.
- project management in large scale construction projects.
- planning and control of supply logistics.

The Scenario

- A user specifies a task that is to be performed through some suitable interface. We call this process *job 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.

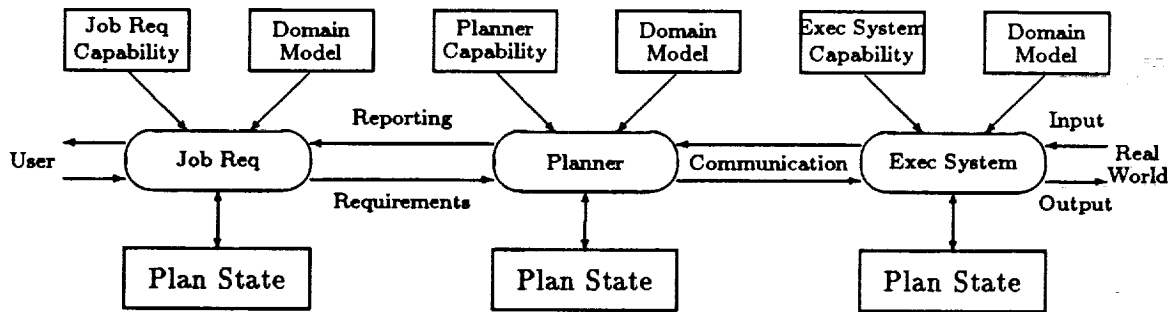


Figure 1: Communication between Central Planner and Ex. Agent

We have deliberately simplified our consideration to three agents 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.

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

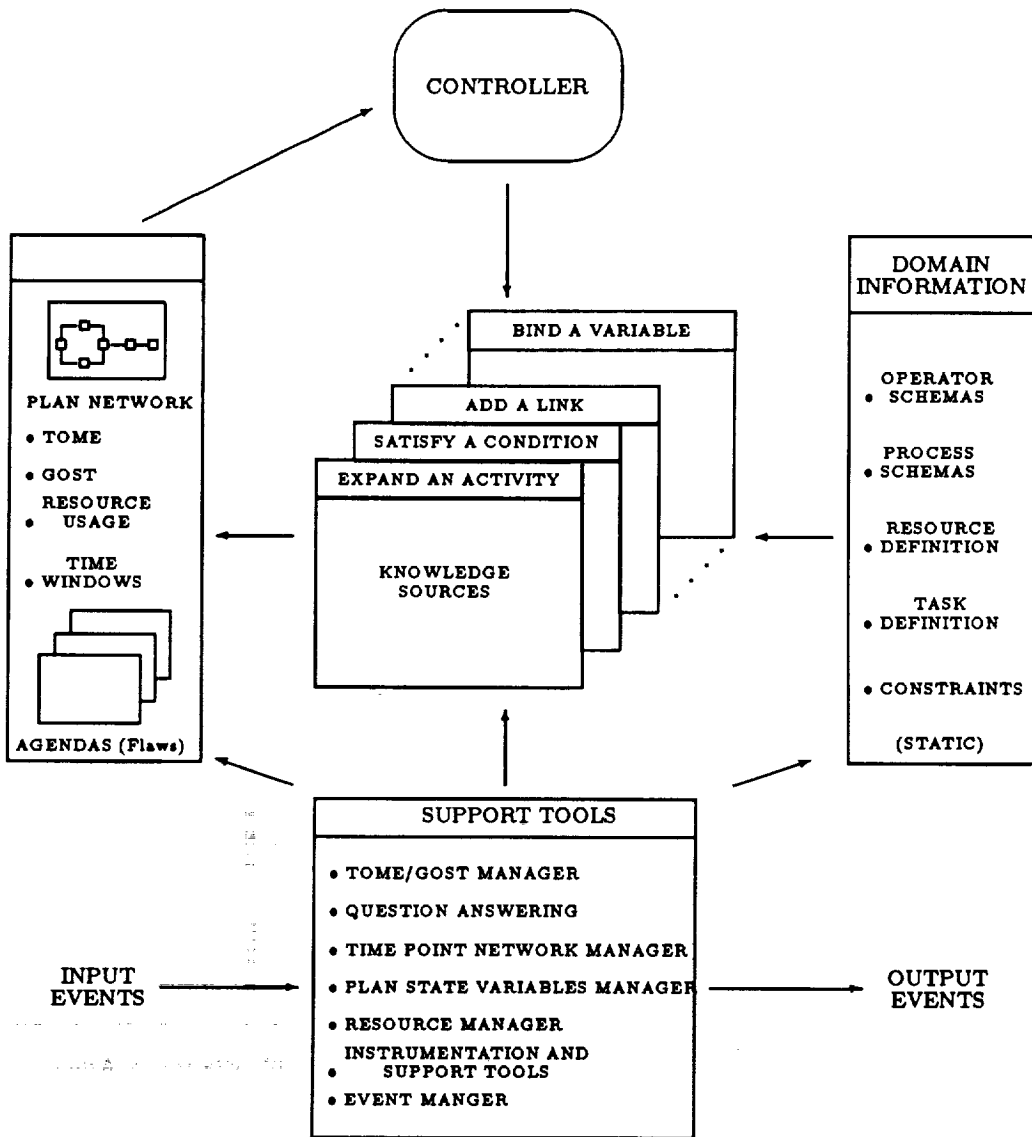


Figure 2: O-Plan2 Architecture

Developer Interface

O-Plan2 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). Each of these may be run on a separate processor and multiple *platforms* may be provided to allow for parallelism in knowledge source processing. A sample screen image as seen by the O-Plan2 developer or an interested technical user is shown in Figure 3.

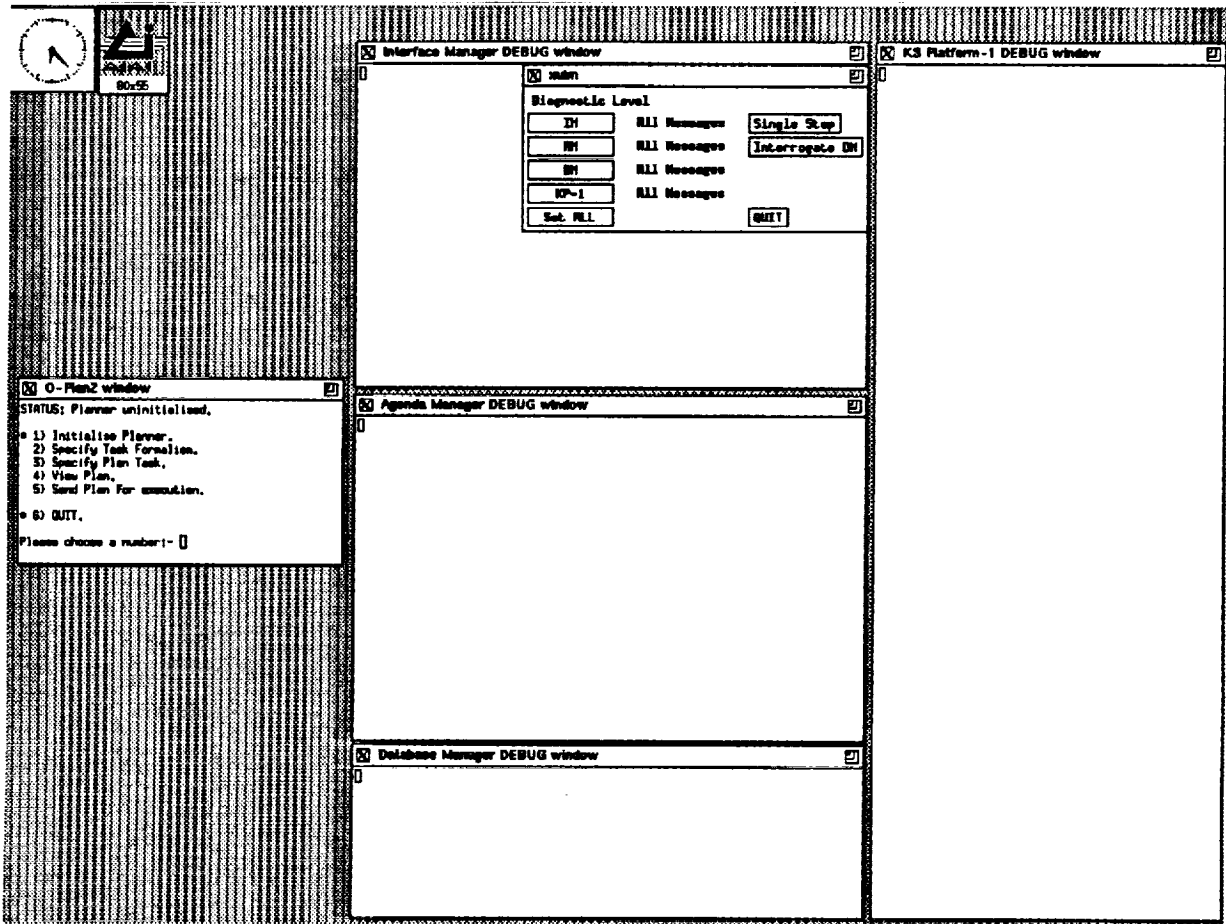


Figure 3: Example Developer Interface for the O-Plan2 Planning Agent

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 draws the plan as a graph, and the upper right window can be used for simulations of the state of the world at points in the plan.

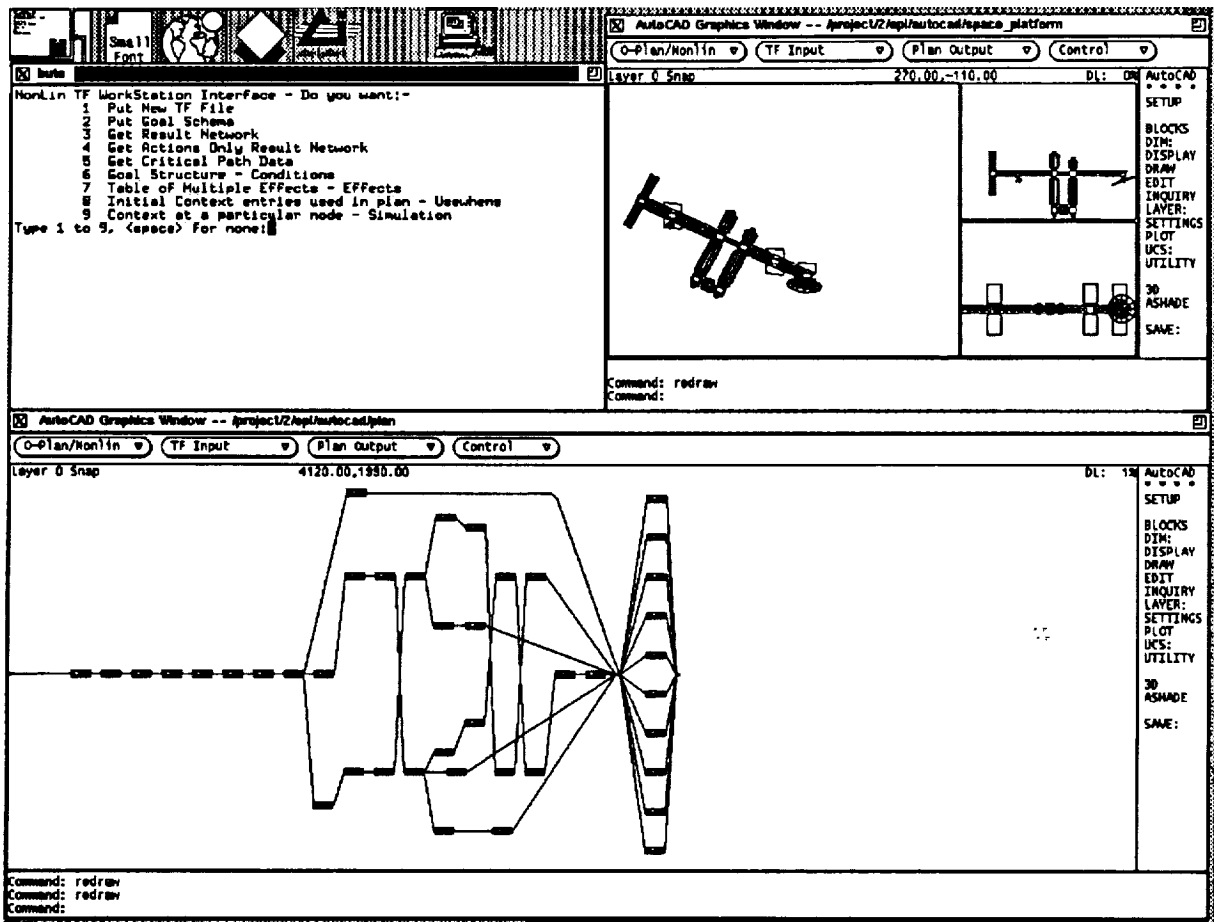


Figure 4: Example Output of the AutoCAD-based User Interface