O-Plan

Introduction

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

July 12, 1995

Version 2.3

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, Glen Reece, Arthur Seaton, Judith Secker, 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 and provided technical input to the design of O-Plan.

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

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

From 1992 to 1995, the O-Plan project was supported by the ARPA/Rome Laboratory Knowledge Based Planning and Scheduling Initiative through the US Air Force Rome Laboratory through the Air Force Office of Scientific Research (AFOSR) and their European Office of Aerospace Research and Development by contract number F49620-92-C-0042 (EOARD/92-0001) monitored by Northrup Fowler III at the USAF Rome Laboratory.

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

From 1989 to 1993, research on scheduling applications of the O-Plan architecture was funded by Hitachi Europe Ltd. From 1989 to 1992, the UK Science and Engineering Research Council (grant number GR/F36545 – UK Information Engineering Directorate project number IED 4/1/1320) funded a collaborative project with ICL, Imperial College and other partners in which the O-Plan architecture was used to guide the design and development of a planner with a flexible temporal logic representation of the plan state. A number of other research and development contracts placed with AIAI have led to research progress on the O-Plan prototype.

O-Plan is a valuable asset of the Artificial Intelligence Applications Institute and must not be used without the prior permission of a rights holder. Please contact AIAI for more information.

Contact Information

The O-Plan project team can be contacted as follows:

O-Plan Team
Artificial Intelligence Applications Institute
The University of Edinburgh
80, South Bridge
Edinburgh EH1 1HN
United Kingdom

Tel: (+44) 131 650 2732
Fax: (+44) 131 650 6513
Email: oplan@ed.ac.uk

Created:	January 14, 1991 by Austin Tate
Last Modified:	July 4, 1995 (20:26) by Brian Drabble
Printed:	July 12, 1995

©1994, The University of Edinburgh

This material may be reproduced by or for the U.S. Government pursuant to the copyright license under the clause at DFARS 252.227-7032 (June 1975) – Rights in Technical Data and Computer Software (Foreign).

Contents

1 Data Sheet **3**

1.1 An Architecture for Command, Planning and Control 3

1.2 O-Plan Planner 5

1.3 Applications 5

2 Road Map to Release Materials **6**

2.1 Contents of the Release 6

2.2 Using O-Plan 7

2.2.1 Getting Started 7

2.2.2 Using O-Plan with the Demonstrations Provided 7

2.2.3 Using O-Plan for a User Defined Application 7

2.3 Understanding and Experimenting with O-Plan 8

1 Data Sheet

Planning, scheduling and control systems based on artificial intelligence techniques are now maturing and are being applied to progressively more realistic problems. The Planning and Scheduling Group at the Artificial Intelligence Applications Institute at the University of Edinburgh has been involved in the production of several complete working Artificial Intelligence (AI) planning systems with gradually improving scope and capability. The latest is the O-Plan Architecture, the O-Plan Planner based on this architecture, and a demonstration environment for inter-agent command, planning and execution using the architecture.

The O-Plan project has the following aims:

- to provide a generic, domain-independent computational architecture suitable for specialisation into command, planning and execution systems by adding new processing capabilities and domain knowledge.
- to provide a state-of-the-art AI planning system which uses an activity based plan representation.
- to provide means to allow a rich level of domain knowledge to be provided to the system and to exploit this domain information in opportune ways within the system when choices are being made and alternatives explored.
- to clarify and define the required modules and interfaces of the architecture, the planner and other parts of the system.
- to provide a portable and flexible prototype system in which new functionality can be experimented with. The design is intended to allow for experimentation with real-time distributed command, planning and control in a multi-processor systems in future.
- to demonstrate the architecture and planner on realistic problems.

Associated projects are using the O-Plan Architecture to explore resource oriented scheduling systems (TOSCA - The Open SCheduling Architecture), exploring richer descriptions of plans based on temporal logic and are looking at execution systems in greater detail.

1.1 An Architecture for Command, Planning and Control

O-Plan provides an *Architecture* in which different agents with command (task assignment), planning and execution roles can be built. The O-Plan project has sought to identify modular components within an AI command, planning and control system and to provide clearly defined interfaces to these components and modules. The various components plug into "sockets" within the architectural framework. The sockets are specialised to ease the integration of particular types of component. See Figure1.

The various components of the agent architecture are:

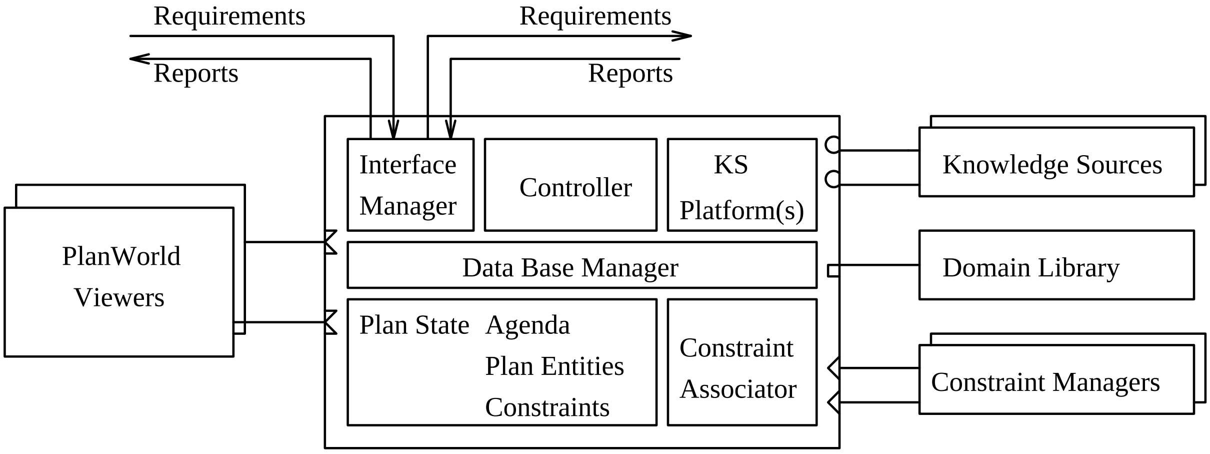

Figure 1: O-Plan Agent Architecture

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 state 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.

1.2 O-Plan Planner

Most effort in the current O-Plan system has been devoted to the provision of a planner which uses a hierarchical partially ordered activity representation of plans as its basis. The aim has been to replicate the functionality of earlier Edinburgh planners such as Nonlin [3] and O-Plan1 [2] but in an improved computational framework which is more flexible and can be made more widely available than those earlier systems.

1.3 Applications

The release of O-Plan includes a number of sample application domain descriptions and demonstration files to show O-Plan in use. A demonstration of the intended user interface for O-Plan has been created which uses the widely available AutoCAD package [1] to show how the system can link to such user interface packages.

References

[1] AutoDesk AutoCAD Reference Manual, 1989.

[2] Currie, K.W. & Tate, A. O-Plan: the Open Planning Architecture, *Artificial Intelligence* Vol 51, No. 1, Autumn 1991, North-Holland.

[3] Tate, A. Generating project networks. *In procs.* IJCAI-77*, 1977.*

2 Road Map to Release Materials

2.1 Contents of the Release

The following materials make up the current O-Plan release.

- **Release Image**
  a) installation notes
  b) source code of system
  c) documentation
  d) sample demonstration problems in Task Formalism (TF)
- **Introduction**
  A short introduction to the O-Plan system and a road map to the release materials. (This document)
- **Installation Guide**
  Information about systems requirements to install and run the O-Plan system and instructions for building a system suitable for use in various environments.
- **User Guide**
  Information on using O-Plan.
- **Demonstration Guide**
  Information about how to use demonstration examples which are included with the release.
- **Architecture Guide**
  The conceptual overview of the O-Plan Agent Architecture is described, followed by a description of the Task Assignment, Planner and Execution Agents and their major components.
- **TF Manual**
  The domain and task description language used to give applications to O-Plan.
- **Implementation Guide**
  A description of the current implementation, the processes in it and its data structures is given in this document.
- **Information Package**
  A collection of published papers and notes which provide relevant background material on O-Plan.
- **Handout**
  A short handout about O-Plan to accompany demonstrations of the system.

2.2 Using O-Plan

2.2.1 Getting Started

The *Installation Guide* is intended to be self-sufficient, such that it should not be necessary to read other documents before installing the system[1].

The *User Guide* is provided to explain how to use a previously installed O-Plan system. This should be read by all users.

Once a user has a basic understanding of how to run O-Plan (either from the installation guide or the user guide), the demonstrations described in the *Demonstration Guide* are intended to be self sufficient, so that an O-Plan demonstration can be run by following the instructions. understanding.

2.2.2 Using O-Plan with the Demonstrations Provided

O-Plan has a number of demonstrations provided in the release. They are of three types.

1. Examples which can be used with O-Plan acting as a planner. Most of the demonstrations are of this form. The Execution System agent is not running, and the task assignment "agent" is a simple user menu much like those provided as the user menu on earlier Edinburgh planners such as Nonlin [3] and O-Plan1 [2].
2. Examples of the system running with an Execution System active. The demonstration runs on a single workstation with rather a dense screen image.
3. Examples of the system running as a planner only once again, but this time demonstrating the type of graphics interfaces that we at Edinburgh envisage for our systems. The demonstration uses the widely available AutoCAD package [1] which must be available at the demonstration site as it is not provided with the release.

Some understanding of O-Plan will be necessary to be able to get most value from some of the demonstrations. The *Architecture Guide* provides a comprehensive overview of the purpose and architecture of the O-Plan system and the way it is structured.

The *Implementation Guide* goes into greater detail for those wishing to explore the actual data structures involved or understand how the processes that make up the system work together[2].

2.2.3 Using O-Plan for a User Defined Application

The "normal" mode of use of O-Plan in its present form is to:

---

[1] For expert users a quick guide to installation of the O-Plan system is also provided in the INSTALL file within the release image.

[2] At present the Implementation Guide provides detail on only a small part of the code of O-Plan.

- develop a domain description of the application in *Task Formalism* (TF)
- define suitable tasks in that domain to give to the O-Plan Planner
- allow the planner to seek a plan that satisfies the task and then to view the results of planning.

To do this, it will be necessary to read the *TF Manual* which describes the Task Formalism domain and task description language accepted by the current O-Plan Planner. Some of the TF demonstration files presented with the release could also be used as guidance for building a new application.

2.3 Understanding and Experimenting with O-Plan

O-Plan has been provided in a form that is intended to encourage experimentation with the replacement of components such as the knowledge sources, controller, constraint managers, support routines, etc. The architecture and system are being used this way on a number of associated projects at Edinburgh. In order to encourage this level of experimentation, the *Implementation Guide* is provided to act as an overview of the code making up the system. Also, the *Information Package* provides a number of papers and notes that describe aspects of the system.