Open Planning, Scheduling and Constraint Management Architectures for Virtual Manufacturing

Howard Beck & Austin Tate

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

H.Beck@ed.ac.uk & A.Tate@ed.ac.uk

Abstract

The development of open planning and scheduling systems seeks to (i) support incremental extension and change, and (ii) facilitate communication between processing agents (both computer and human). This paper presents the open planning and scheduling approach adopted in the O-Plan and TOSCA systems at the Artificial Intelligence Applications Institute (AIAI) in Edinburgh. The purpose is to bring together a description of the concepts developed at AIAI and to relate them in a common framework. References to more detailed descriptions are provided. The paper describes:

1. the key characteristics of the open planning and scheduling systems developed at AIAI;
2. the basis of the separation of the constraint elements in planning and scheduling tasks, distinguishing a high-level model of what remains to be done, a user level view of the plan/schedule entities and the low-level detailed constraints;
3. generic constraint managers for planning and scheduling including time constraint, plan state variables and resource constraints managers.

1 Introduction

Historically, planning and scheduling tasks have been treated as static problems; now, it is generally appreciated that these tasks need to be viewed as part of a dynamic process which is subject to external impacts, be they a consequence of concurrent activities (e.g. engineering design, quality etc) or unforeseen events. In many aspects of planning and scheduling (especially in response to change), the role of the human scheduler/system operator is crucially important. To support the user's ongoing decision making, planning and scheduling systems need to be able to communicate in an understandable and useful form. The development of open planning and scheduling systems seeks to (i) support incremental extension and change, and (ii) facilitate communication between processing agents (both computer and human). The motivation for establishing an architecture which supports incremental extendability and modifiability is to enhance flexibility and provide a basis for modular system building. A generic framework and re-usable components would allow system developers the opportunity to exploit the considerable commonality in component functionality typically found in the construction of application systems. The need to support inter-process communication has become apparent from practical experience, especially in the context of increasing enterprise integration.

This paper presents the open planning and scheduling approach adopted in the O-Plan and TOSCA systems at the Artificial Intelligence Applications Institute (AIAI) in Edinburgh. The purpose is to bring together a description of the concepts developed at AIAI and to relate them in a common framework. References to more detailed descriptions are provided.

2 Open Planning and Scheduling

O-Plan (The Open Planning Archicture) [7] and TOSCA (The Open Scheduling Architecture) [3] are systems being developed at AIAI. Their approaches to planning, scheduling and control can be characterised as follows:

- open interfaces and communications protocols
- successive refinement/repair of a complete but flawed plan or schedule
- least commitment approach
- using opportunistic selection of the focus of attention on each problem
- solving cycle
- building information incrementally in "constraint managers", e.g., time point network manager
- object/variable manager
- resource utilisation manager
- using localised search to explore alternatives where advisable global alternative re-orientation where necessary.

The open planning and scheduling approach grew out of the experiences of other research in AI planning, particularly with Nonlin [15] and "blackboard" systems [12]. Some of the primary influences include: hierarchical planning [13], the notion of plan state (similar to the work of McDermott [11]), constraint posting and least commitment and temporal and resource constraint handling. The Readings in Planning volume [1] includes a taxonomy of earlier planning systems that places O-Plan in relation to the influences on its design.

The TOSCA scheduling system is heavily influenced by O-Plan and the micro-opportunistic approach to scheduling [14].

O-Plan and TOSCA have been designed as generic planning and scheduling tools applying component technologies. O-Plan has been applied to the following types of problems: (i) mission sequencing and control of space probes such as Voyager, (ii) project management, and (iii) planning and control of supply and distribution logistics. TOSCA has been applied to factory scheduling. Whereas O-Plan is concerned with the detailed construction of activity plans to achieve specific goals and handles problems of moderate scale, TOSCA is concerned with the allocation of activities to resources and start times and, comparatively speaking, handles problems of very large scale.

3 Components of the open planning and scheduling systems

O-Plan and TOSCA have as a fundamental design goal the clear separation of system components. There are two broad motivations for this design goal: (i) increased modularity can lead to re-usability, embedability and improved implementation, and (ii) the decomposition of planning and scheduling systems promotes the understandability of the working of the system. This is important for the theoretical exploration of models of the planning and scheduling tasks.

In order to benefit from advances in various technologies and to allow improved implementation of components to be used, it it necessary that the separable functions and capabilities of planners and schedulers be recognised. By separating the processing capabilities at the architecture level of a planner or scheduler from the plan or schedule representation, it becomes possible to address modularity issues of this kind. This separation underpins the generic planning and scheduling model being developed at the AIAI. The architecture and the basic processing cycle is shown in Figure 1.

The architecture of O-Plan and TOSCA has the following primary components:

- domain information
- plan/schedule states
- knowledge sources
- controller
- support modules

The processing cycle is driven by the outstanding or critical issues which need to be addressed. The Controller selects a particular issue and a Knowledge Source to address the issue. When the Knowledge Source is applied, the plan or schedule state is modified. The resulting updates to the plan or schedule state is supported by the Constraint Managers and other Support Modules.

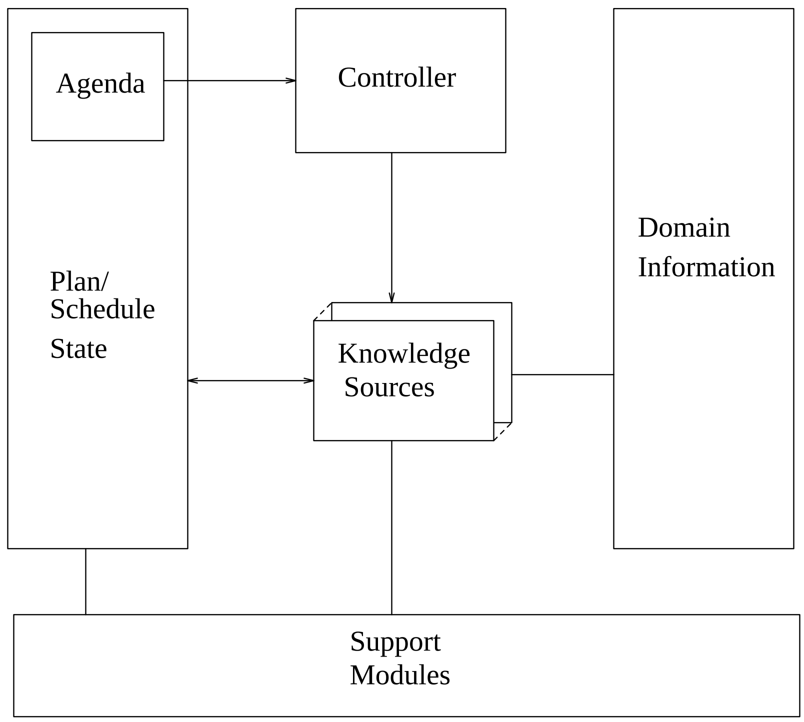

Figure 1: Open Planning and Scheduling Architecture

3.1 Domain Information

Domain descriptions are supplied to O-Plan in a language called Task Formalism (TF). This is compiled into the internal data structures to be used during planning. TF is the means through which a domain expert or domain writer can supply the domain specific information to the O-Plan system.

The domain information describes a model of an application and the tasks to be undertaken. In TOSCA, the model describes the factory, its methods of production and specific production requirements over a given scheduling period [4]. The key elements are:

Production: the manufacturing process concerned with the transformation of materials into end-products. Associated with each product is a set of process plans. Each process plan describes a method of production (*i.e.*, a set of temporally ordered operation types).

Demand for Production: imposed by the orders accepted and predicted by the manufacturing system. Demand is a description of the obligations for production that the manufacturing system has undertaken.

Capacity to Produce: the factory resources and production plans. The capacity of the factory resources are described by their capabilities, corresponding to the various operation types which they can process, and their speed of processing.

Production Constraints: conditions which must be satisfied for a schedule to be valid. Overall schedule objectives (*e.g.*, minimise Work-in-Process) are a special type of constraint in that they apply across the entire schedule.

The domain model is read in from files and converted into internal data structures. A domain description language (DDL) for factory scheduling has been formulated which serves as the basis for the generic specification of discrete factory scheduling problems [4].

3.2 Plan/schedule States

Planning and scheduling *states* can be thought of as snapshots taken during the problem solving process. Each state is associated with: the plan/schedule agenda, the plan/schedule entities and the plan/schedule constraints.

A plan/schedule state may be represented as a set of constraints which together define the range of possible plans or schedules which can be elaborated. Work on O-Plan and other practical planners has identified different entities in the plan which may be grouped into three constraint types which correspond to the high level description above.

The types of constraints are:

- Implied constraints or "Issues" — representing the pending or future constraints that will be added to the plan or schedule as a result of handling outstanding requirements, dealing with aspects of plan/schedule analysis (*i.e.*, the agenda).
- Plan/schedule entities or node constraints — the main plan entities related to external communication of a plan or schedule. They describe a set of external names associated with time points.
- Detailed constraints — associated with plan/schedule entities and representing specialised constraints on the plan or schedule. These are subdivided into three constraint subtypes: Ordering constraints, Variable Constraints and Auxiliary Constraints.

This constraint based description of a plan/schedule state space is described as the <I-N-OVA> (Issues - Nodes - Orderings/Variables/Auxiliary constraints) Model [17].

3.3 Knowledge Sources

Knowledge Sources are defined to address specific plan/schedule requirements through the application of various state modification operators. In O-Plan these include: Expand action, Choose action to satisfy required condition and Select instantiations of object variables. In TOSCA these include: Merge operations, Drop resourcing option, Restrict time window and Allocate start time. Below is a brief description of the state modification operators used in TOSCA:

1. **Merge operations**: a decision to process two or more operations consecutively on the same resource.

   Merging operations reduces the total number of setups and also alters the distribution of demand for setups over time. It is used to manage the constraint regarding the maximum number of setups at a resource and workcentre, and could also be applied to save resource processing time.

2. **Drop resourcing option**: a decision to restrict the resourcing options of an operation.

   Dropping a resourcing option redistributes the demand for capacity and demand for setups *between resources.* It is used to manage both time and setup constraints. The 'drop resourcing option' operator is iteratively applied and a resource allocation is made when the number of resourcing options is reduced to one.

3. **Restrict time window**: a decision to reduce the time window of an operation.

   Restrict an operation time window redistributes the demand for capacity and demand for setups *over time.* It is used to manage both time and setup constraints. The 'restrict time window' operator is iteratively applied and a high-level temporal allocation (*i.e.*, operation to start during a particular time period) is made when the number of high-level time period options is reduced to one.

4. **Allocate start time point**: a decision to allocate a specific time to an operation which has already been allocated a resource.

   Allocating a start time point reserves a specific start time point for an operation taking into account all constraints including those which are not monitored. It is used to check and avoid constraint violations.

3.4 Controller

Throughout the plan generation process, O-Plan identifies outstanding issues to address and these issues are posted to an agenda list. The controller computes the context-dependent priority of the agenda items and selects an item for processing. This provides the fundamental opportunism inherent in the system.

Control in TOSCA can be viewed at two levels: one based on a coarse level of problem decomposition, the other based on a much finer granularity of subproblem. At the coarse level, the current implementation of TOSCA adopts a simple linear flow from phase to phase; at the finer level (within phases), there is highly opportunistic control, corresponding closely to what Sadeh has described as a 'micro-opportunistic' approach to scheduling [14]. The phases at the top-level are:

- *Pre-scheduling* provides an opportunity to analyse the job-shop scheduling problem with the purpose of identifying infeasible constraints. When there is an excessive demand for setups, demand is reduced by merging operations.

- *High-level scheduling* deals with the monitored constraints (*i.e.*, temporal-capacity constraints including temporal preferences). During high-level scheduling, resources are allocated and the possible start times of operations restricted to a time period, the granularity of which is defined by the user.
- *Low-level scheduling* allocates operations to a specific start time.

3.5 Support Modules

In order to efficiently support the planning and scheduling functionality in O-Plan and TOSCA, a number of support modules have been separated out from the core decision making capabilities. These modules have carefully designed functional interfaces to allow for piecewise system engineering and experimentation.

Support modules are distinguished from the other processing capability of the system in that they do not take decisions themselves, but serve to provide efficient support to the higher level Knowledge Sources where decisions are taken. The support modules include database management and retrieval facilities, context layered access to the plan/schedule state, instrumentation and diagnostics as well as the constraint managers, some of which are described in Section 4. Other support modules not included here are described in [18].

4 Constraint Management in Planning and Scheduling

Both O-Plan and TOSCA use a number of *constraint managers* to maintain information about a plan while it is being generated. The information can then be utilised to prune search (where plans are found to be invalid as a result of propagating the constraints managed by these managers) or to order search alternatives according to some heuristic priority.

To improve the modularity of our planning and scheduling systems, we have separated the management of detailed constraints from the explicit manipulation of planning and scheduling entities. This is done via the Associated Data Structure (ADS) abstraction. Data maintained by constraint managers is indirectly linked to the activities, resources, events *etc.* through the ADS level.

Below is a description of *temporal*, *variable* and *resource* constraint managers.

4.1 Management of Temporal Constraints

O-Plan and TOSCA use a point based temporal representation with range constraints between time points and with the possibility of specifying range constraints relative to a fixed time point [5]. This provides the capability of specifying relative and metric time constraints on time points. The functional interface to the Time Point Network (TPN), as seen by the

Associated Data Structure (ADS) has no dependence on a particular representation of the plan or schedule [8].

The Time Point Network is the lowest level of temporal data structure and consists of a set of points (and associated time constraints between two points) each of which has an upper and lower bound on its temporal distance from:

1. other points in the network
2. a (user defined absolute) start time reference point

This is strong enough for both representing metric and relative time constraints between time points. The points are numbered to give an index with a constant retrieval time for any number of points. This structure allows points to be retrieved and compared through a suitable module interface and with a minimum of overhead. The interface is important and reflects the *functionality* required of the TPN, and hides the detail. This ensures that we have no absolute reliance on points as a necessary underlying representation. The TPN is maintained by the Time Point Network Manager (TPNM). Through application in TOSCA, the current TPNM has been proven on large resource allocation scheduling problems where the number of time points was in excess of 5000 and the number of temporal constraints exceeded 3000.

4.2 Management of Plan State Variables

In the O-Plan system, the Plan State Variable Manager is responsible for maintaining the consistency of restrictions on plan objects during plan generation. O-Plan adopts a least commitment approach to object handling in that variables are only bound as and when necessary. The constraints are specified as:

- **Sames:** This specifies that this plan state variable should be the same as another plan state variable
- **Not-Sames:** This specifies that this plan state variable should not be the same as another plan state variable
- **Constraint-list:** This specifies a list of attributes which the value to which the plan state variable is bound must have.

4.3 Management of Resource Constraints

O-Plan and TOSCA employ different mechanisms for tracking resource demand and availability: O-Plan uses a simple Resource Utilisation Manager (RUM) [9]; TOSCA uses a more comprehensive model based on habographs [2].

The Resource Utilisation Manager monitors resource levels and utilisation. Resources are divided into different types such as:

1. Consumable: these are resources which are "consumed" by actions within the plan. For example: bricks, petrol, money, etc.

2. Re-usable: these are resources which are used and then returned to a common "pool". For example, robots, workmen, lorries, etc.

Consumable resources can be subcategorised as *strictly consumed* or may be *producible* in some way. Substitutability of resources one for the other is also possible. Some may have a single way mapping such as money for petrol and some can be two way mappings such as money for travellers' cheques. Producible and substitutable resources are difficult to deal with because they *increase* the amount of choice available within a plan and thus *open up* the search space.

The current implementation uses the same mechanism for maintaining resource constraints as did the original O-Plan system [7]. A new scheme is however under study which is based on the maintenance of optimistic and pessimistic resource profiles with resource usage events and activities tied to changes in the profiles [9].

The TOSCA system is particularly concerned with managing high resource contention, and provides mechanisms referred to as habographs to deal more precisely with demand from multiple sources which need to be considered as an aggregate. The aims are: (i) identify constraint violations as early as possible, and (ii) monitor threats of possible constraint violations. The basic insight underlying the habographs representation is described in [2]. The fundamental distinction between habographs and other temporal-capacity constraint representations [10, 14] is in the way that the demand imposed by an operation over time is estimated — specifically, in terms of the assumptions underlying the estimations. Most systems *assume* a demand profile for each operation. Any resource demand profile based upon an aggregation of operation demands is also subject to those assumptions, and as a result, are unable to distinguish between constraint threats and constraint violations. Habographs, by not making this assumption, are able to distinguish constraint violations from threats and more accurately identify constraint threats.

The identification of constraint violations and the monitoring of constraint threats plays a central role in schedule generation both in terms of (i) directing the scheduling process and (ii) informing scheduling decisions.

Habographs extend on similar resource profiling methods [6, 10, 14] in a number of ways. As well as monitoring *temporal* demand on resources, they also provide:

1. a lookahead for and *setup capacity* constraints to allow setup constraints to be dynamically maintained throughout schedule generation.

2. a *hierarchical model* of strategic knowledge constraints to support the analysis of demand for resources over time. This allows compound constraints applying to *machine groups* with overlapping capabilities to be analysed.

3. a separate representation reflecting *temporal preferences*. By distinguishing the temporal preference from the temporal range (limits) of valid allocations of each operation a mechanism is provided for exploring optimality without introducing infeasibility.

In important respects, these extensions support the scheduling of more complex factory domains: *viz.*, the management of setups, the allocation of resources of overlapping capabilities, and the management of the trade-off between hard and preference temporal constraints.

4.4 The Constraint Associator

To improve the separation of functionality with respect to constraint management in O-Plan, we wish to localise the interactions between changes in one type of constraint that can lead to changes in other types of constraint. This has been problematic in O-Plan to date. In particular, changes in constraints on time points and changes to constraints on plan state variables can have implications for most other constraints being managed (such as effects/conditions, resources, etc.). Previously, Knowledge Sources had to be written such that any change in one constraint type that could influence another was programmed in.

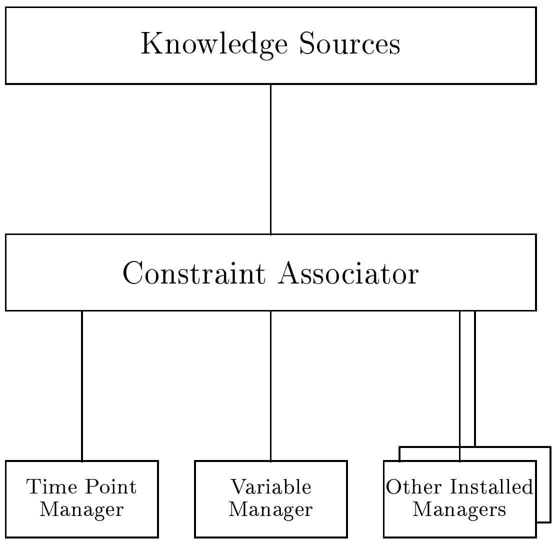

Figure 2: Constraint Associator to mediate between Knowledge Sources and Constraint Managers

The clarification of the constraint manager interface for O-Plan as described in [9] has made us realise the special requirements for the handling of time point constraints and variable constraints in the architecture. These form the core elements in the shared ontology in which communication occurs between the plan entity (ADS) layer and the constraint managers in O-Plan. By recognising that there is a normal constraint management function for time points and variable, but also an *additional* function of association and mutual constraints with other constraint types, we can design better and more modular support for constraints handling in O-Plan and simplify the writing of Knowledge Sources [17].

Accordingly, the O-Plan agent architecture design in future will allow for an "Associator" component as part of the data base manager which looks after plan states. The Associator mediates between the decisions made by Knowledge Sources and the underlying constraint managers (see figure 2). A number of constraint managers can be "installed" into an O-Plan agent. As a minimum, each agent will have a time point manager and a variables manager installed into the Associator. Any number of other constraint managers may then be added depending on the requirements. To give the functionality of the current O-Plan planner this will include the effect/condition manager, the resource utilisation manager, and an "other constraints" manager to keep annotations of other requirements on a plan state. In other applications it may be necessary to include spatial constraint managers, etc.

We believe that this style of interface between the higher level decision making level of the planner and the various Constraint Managers could improve modularity in planning systems.

5 Conclusion

This paper has described the open planning and scheduling approach adopted in the O-Plan and TOSCA systems at the Artificial Intelligence Applications Institute in Edinburgh. One particular area highlighted has been the interface between planning systems, scheduling systems and constraint managers responsible for certain specialised aspects of planning and scheduling states. An interface to such constraints managers has been developed to show how improved packaging can be beneficial for the re-use of components. We view this work as a necessary development of recent attempts to re-use components of planning and scheduling systems, particularly specialist constraint managers [16].

References

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

[2] H. Beck. Constraint Monitoring in TOSCA. In A.Tate M.E.Drummond, M.Fox and M.Zweben, editors, *Working Notes from the AAAI Spring Symposium on Practical Approaches to Planning and Scheduling*, 1992. Also available as Technical Report AIAI-TR-121.

[3] H. Beck. TOSCA: A novel approach to the management of job-shop scheduling constraints. In C. Kooij, P.A. MacConaill, and J. Bastos, editors, *Realising CIM's Industrial Potential: Proceedings of the Ninth CIM-Europe Annual Conference*, pages 138–149, Amsterdam, 12-14 May 1993.

[4] H. Beck, K. Currie, and A. Tate. A Domain Description for Job-Shop Scheduling. Technical Report AIAI-TR-137, A.I.A.I., 1993.

[5] C.E. Bell and A. Tate. Using Temporal Constraints to Restrict Search in a Planner. Technical Report AIAI-TR-5, A.I.A.I., 1986.

[6] P. Berry. *A Predictive Model for Satisfying Conflicting Objectives in Scheduling*. PhD thesis, Dept. of Computer Science, Strathclyde University, Glasgow, 1991.

[7] K.W. Currie and A Tate. O-Plan: the Open Planning Architecture. *Artificial Intelligence*, 52(1), 1991.

[8] B. Drabble and R. Kirby. Associating A.I. Planner Entities with an Underlying Time Point Network. In *European Workshop on Planning (EWSP '91)*. Springer-Verlag Lecture Notes in Artificial Intelligence, 1991.

[9] B. Drabble and A. Tate. The use of optimistic and pessimistic resource profiles to inform search in an activity based planner. In Chris Hammond, editor, *Proceedings of the Second International Conference on AI Planning Systems*, University of Chicago, Chicago, Illinois, June 1994.

[10] B. Liu. Scheduling via reinforcement. *Journal of AI in Engineering*, 3(2), 1988.

[11] D.V. McDermott. A Temporal Logic for Reasoning about Processes and Plans. *Cognitive Science*, 6:101–155, 1991.

[12] H.P. Nii. Blackboard systems: the blackboard model of problem solving and the evolution of blackboard architectures. *AI magazine.*, 7(2):38–53., 1986.

[13] E.D. Sacerdoti. *A Structure for Plans and Behaviour*. Artificial Intelligence Series. North Holland, 1977.

[14] N. Sadeh. *Look–ahead Techniques for Micro–oportunistic job shop scheduling*. PhD thesis, School of Computer Science, Carnegie Mellon University, 1991. Technical Report CMU-CS-91-102.

[15] A. Tate. Generating project networks. In *Proceedings 5th IJCAI*, pages 888–893, 1977.

[16] A. Tate. The Emergence of 'Standard' Planning and Scheduling System Components — Open Planning and Scheduling Architectures. In *European Workshop on Planning (EWSP '93)*, 1993.

[17] A. Tate. Characterising Plans as a Set of Constraints — the <I-N-OVA> Model — A Framework for Comparative Analysis. *To appear in ACM SIGART Bulletin*, 6(1), January 1995.

[18] A. Tate, B. Drabble, and R. Kirby. O-Plan2: An Open Architecture for Command, Planning and Control. In M. Fox and M. Zweben, editors, *Knowledge Based Scheduling*, pages 213–239. Morgan Kaufmann., Palo Alto, California, 94303, USA, 1994.