

Planform: An Open Environment for Building Planners

T. L. McCluskey*
University of Huddersfield,
UK

Maria Fox
University of Durham,
UK

Ruth Aylett
University of Salford,
UK

December 5, 2002

Project Overview

The PLANFORM project aimed to develop a high level research platform for the systematic construction of planner domain models and automatically configured planning algorithms.

Our objectives were, briefly:

- To assemble tools within an open environment for the acquisition and modelling of planning domain models;
- To create languages for modelling of planning domains and to specify characteristics of planners leading to the configuration of purpose-built planning engines;
- To create a tool which synthesises a domain model and a planning engine into a planning application;
- To evaluate the approach using realistic problem domains.

During the project, the emphasis changed slightly so that some of objectives developed in a different direction. Rather than abstractly defining planning algorithms, we decided to create a library of algorithms and use domain analysis technology to design and configure a planning application. We perceived the *knowledge acquisition bottleneck* to be a significant problem for AI Planning and concentrated more resource than planned on this area.

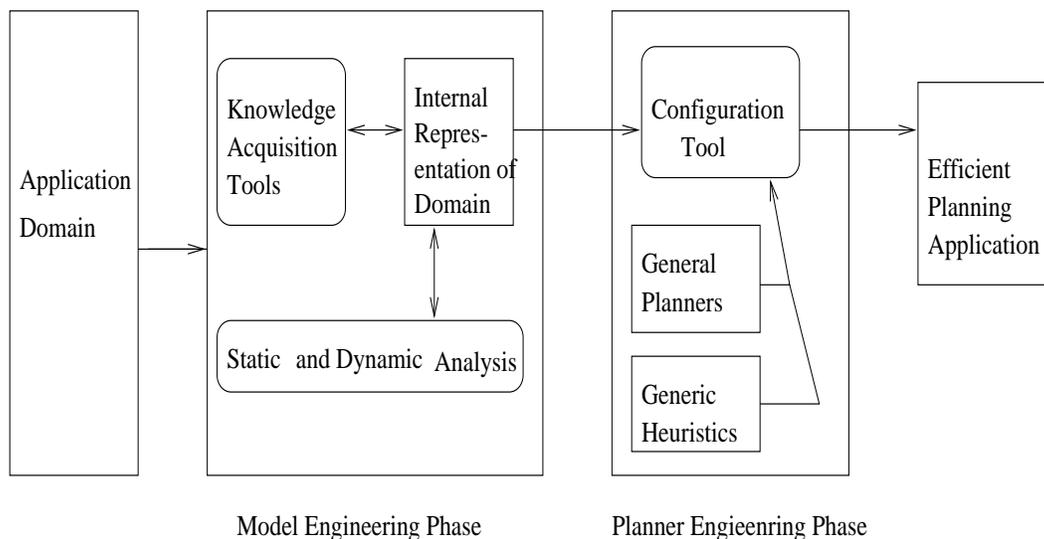


Figure 1: Architectural Breakdown of Planform

* This work was supported by EPSRC projects GRM67421/01, GRM70568/01 and GRM68237/01, held at the three participating sites.

tion and modelling of planning applications and the configuration of planning engines suited to those applications. Although time limitations prevented us from completing all aspects of the final integration and evaluation we achieved our main objectives and laid a strong foundation for development of the project.

The project consisted of a *model engineering phase* and a *planner engineering phase*. Huddersfield and Salford collaborated closely on modelling and knowledge acquisition issues, whilst Huddersfield and Durham collaborated on the development of parts of the domain modelling tools. Durham worked on the planner engineering phase of the project.

Research at the University of Huddersfield

The Huddersfield contribution emphasised tools creation and integration aspects of the project. Tools for acquisition, validation and domain modelling were developed, in collaboration with colleagues at both Salford and Durham. As part of this effort the Huddersfield team researched and developed a GUI-based environment called GIPO¹. In addition, as the initiators of PLANFORM, Huddersfield performed a central administrative function, producing the main external website and also an internal website with additional resources such as Project Meeting minutes. *NB: in the text below 'the PLANFORM website' refers to <http://scom.hud.ac.uk/planform>.*

The first phase of the Huddersfield contribution was concerned with *applications encoding and language development*. During this phase three application domains were used to explore the adequacy of the modelling languages *OCL* and *OCL_h* [20, 19]. *OCL_h* is a hierarchical version of *OCL* enabling the modelling of planning domains as hierarchical task decompositions. Salford and Huddersfield collaborated on this effort with Salford working on the modelling of some existing domains using the two languages. Huddersfield tackled the problem of encoding the aircraft landing scheduling problem supplied by Mark Watson of the National Air Traffic Services. The constraints (e.g. separation times for each type of aircraft) were encoded in *OCL*[18] and *OCL_h*. *OCL_h* was found adequate for all three domains, although the encoding did demonstrate some required changes, and overall the pressing need for tool support.

The standard languages for communicating planning domain descriptions are the PDDL variants [1]. In order to be able to experiment with our domain models using existing planning technology we therefore created tools to map between *OCL* and PDDL. These tools continued to be developed throughout the project allowing planners, and other tools that receive input in PDDL form, to be integrated into our environment. There are some interesting technical aspects of the mapping discussed in [27].

Language manuals for *OCL* and *OCL_h* were maintained during the project [12] and an online help facility was constructed (see the PLANFORM website).

Drawing on previous development work (e.g. [19]), we assembled tools that automate the syntactic and semantic analysis of *OCL* domain models. Analyses include ensuring that invariant properties of the model are maintained and that syntactic rules are observed.

The second phase of the project was concerned with *development of the GUI and tools environment*. The focus was on building and integrating knowledge acquisition and modelling tools for AI planning into an open environment. The GUI and some of the tools described above were built in Java. Others were implemented in Prolog and it was necessary to integrate these via Sicstus Prolog's JASPER interface.

The JavaCup parser generator method was used to represent the syntax rules of *OCL* and to generate a parser for the language. This formed the input tool in GIPO (Graphical Interface for Planning with Objects). The *knowledge acquisition* part of the tool was structured using the method outlined in earlier work [20]. The methods direct the user to define objects, object sorts, relations and properties, classes and constraints on object situations, problems in the form of task specifications, and finally operators built from these components. The design of the interface was based on the need to minimise the use of syntax, and use *object* rather than *predicate* centred ideas.

Once users have entered all parts of a domain model they can utilise modelling tools to remove bugs and experiment with the encoding. We created and adapted the following tools.

- *plan stepper*: This allows the user to pick action schemas and apply them to a state, until a desired goal is

¹<http://helios.hud.ac.uk/planform/gipo>

- *internal planning engines*: this allows our own in-house planning engines to be connected up to GIPO. Sample tasks can be executed and the resulting solutions displayed.
- *interface for external planning engines*: This allows external planning engines to be ‘bolted’ into the environment. The planner needs to be able to input domain models in PDDL (from GIPO), and output solution in a prescribed format. Again, the resulting solutions are displayed through GIPO.
- *a random task generator*: This inputs the current domain model and randomly generates tasks to be used with a planner.
- *an animator*: After a domain model has been entered, and the planning engine has solved a task within that model. the animator can be used to track the transitions of each of the objects which started in the initial state.

In the third phase, *integration and evaluation*, the tools outlined above were integrated into GIPO [28]. The software was released and demonstrated at ECP’01, and again at AIPS’02. It is available on Unix, Linux and Windows platforms from the PLANFORM website. As an initial indication of GIPO’s impact, the Huddersfield website recorded 147 external downloads of the system in the period November 2001 - March 2002.

The environment has been tested using a range of common domains (details are in the resource section of the website). Further, GIPO was used as a teaching tool in a second year introductory course in Artificial Intelligence. GIPO alleviates many user interface problems by adopting an object modelling approach which seems natural to non-expert users. To ameliorate the use of GIPO by non specialists the following issues were explored:

- The use of an inductive approach to capturing operators². The *opmaker* tool was created which outputs a set of operators given a partial domain model and an example solution sequence [22, 21].
- The use of *generic types* to suggest planning *design patterns*. These ideas were developed as part of the Durham PLANFORM project, and Huddersfield is working in collaboration with colleagues at Durham to integrate design patterns into GIPO [26].

The final phase of our work has been to extend the internal language and the surrounding tools from *OCL* (version 1) to *OCL_h* (version 2). *OCL_h* extends *OCL* in two major ways: HTN operators can be used, and sort constraints can be put on each level of the sort hierarchy meaning that objects of a primitive sort inherit all the constraints up the hierarchy. This modelling approach is being tested using a large ‘Translog’ domain imported from a transport logistic domain constructed by members of the University of Maryland.

The priorities for future development of the Huddersfield contribution are:

- The development of a suite of *planning design patterns* and their integration with the GIPO tool;
- The evolution of the *planstepper* tool into a general mixed-initiative plan authoring tool;
- Integration of the operator induction techniques with a plan authoring interface so that operator specifications can be induced and refined interactively.
- The development of the *OCL* representation language to be on the expressive level of PDDL2.1 (temporal representations), thereby enabling GIPO to support the modelling of temporal planning domains.

Research at the University of Salford

Part of the set of objectives for the PLANFORM project was to make AI planning technology accessible to non-experts. In pursuit of this objective, work at Salford was based on the idea of the Knowledge Level [25] and Models of Expertise [6] as articulated over many years in the KBS community, in which the problem level is modelled separately from the design level. Research in KBS technology has shown that support for domain experts is feasible if it is based on the generic task concept [7], and much earlier work has been carried out round the generic task of diagnosis [24].

²This work was primarily work undertaken in conjunction with a PhD student

model at the knowledge-level which can then be used to direct a computer-based knowledge acquisition process with a domain expert.

Thus, to support domain experts, it was seen as necessary to build a knowledge-level tool as part of the PLANFORM environment, incorporating generic task components for planning, and supporting knowledge-level construction of the planning domain rather than forcing the use of the domain design language, *OCL*, used internally. However such a tool would necessarily link to the *OCL* GIPO tools being developed at Huddersfield (see Figure 1) and thus output *OCL*, so it was therefore vital to understand how planning problems would be represented in *OCL*. A strong link with Huddersfield was built through modelling activity in which the objective was to understand this mapping and derive some constraints on the knowledge-level interface, which was oriented towards users with no AI planning knowledge but with expertise in a particular planning domain. A number of domains were modelled including the multi-robot Drumstore from earlier work at Salford.

Planning *ontologies* were identified as a key foundation for such a knowledge-level tool, and the means by which the skeleton model mentioned above could be embodied. A survey of work in the field was carried out and the possibility of incorporating an existing ontology such as CYC (<http://www.cyc.com/cyc-2-1/cover.html>.) was investigated but limited time did not allow its use.

The third component researched as the basis for a knowledge-level tool was the *requirements* of the domain expert, and here an accessible domain was formulated (EVENTUS –arranging a weekend outing for a visiting researcher) and a knowledge acquisition exercise was carried out with four people. The exercise was repeated with the robot Drumstore problem. Data was analysed for concept coverage and for interface design issues, looking at the process a human expert goes through in conceptualising the domain. The details of this process are described in [2].

The KA tool was constructed in Java, and in its first version supported passive model construction by the expert with support from a small hand-coded ontology for the domains already investigated. In its second version, an active question-driven process was added based on the key planning concepts of Task, Agent and Object [5].

The methodology embodies two successive extraction-refinement processes: protocol to problem specification; and problem-specification to conceptualisation. A part of the KOD (Knowledge Oriented Design) method [30] was applied to obtain an accurate process for knowledge acquisition and to build the conceptual model through a set of examples and scenarios.

The output of the KA-Tool is *OCL*, which can then be loaded into the GIPO tool created at Huddersfield. By the formal end of the project, it was possible to generate the world model in *OCL*, and since then it has become possible to generate planning operators, seen by most people as a key problem in formulating a planning domain description. In the 26 months of the project, it was not possible to carry out any extensive evaluation programme, but it is proposed to carry on with the work for a limited period informally with this as the key task.

The modelling exercise enabled the development of strong links with Huddersfield, where frequent (almost weekly) visits were made at some points. A two-week visit to Durham was also organised to strengthen understanding of the requirements of the generative planning back-end.

Further development of the KA-Tool is still being carried out in house, with the generation of Planning Operators now possible. If sufficient resources can be found, the next steps would include:

- Incorporation of a large ontology, such as CYC, into the KA tool;
- Integration of Durham's *generic types* into this ontology;
- Full integration of the KA tool with GIPO.

Two publications were generated jointly with the Huddersfield team (Simpson et al 2001a, 2001b) and two more by the Salford team on their own [4] and [2]. One further paper is under review [3] and a journal article is in preparation.

Research at the University of Durham

The Durham contribution to PLANFORM focussed on the development of planner configuration technology. The objective was to develop techniques by which, given a domain model elicited from a user, planner components suited

been to maintain a library of planner components, including heuristics, specialised solution strategies and problem-specific control rules, and to access these by means of pattern-matching techniques once patterns have been identified in the domain model.

The techniques used to identify domain patterns are based on static domain analysis algorithms developed at Durham prior to the start of the PLANFORM project [8]. The objective in the project was to extend these algorithms to enable the recognition of *generic types* and associated patterns of behaviour in planning domains, and to associate these generic patterns with special purpose solution strategies [14, 13, 16]. Briefly, a generic type is a collection of types sharing some fundamental behaviour. For example, the generic type of *mobility* contains all types of objects that are capable of movement while the generic type of *construction* contains all types that are capable of being combined into compounds (and subsequently recovered by *destruction* of the compound).

When a generic type is present its associated patterns of behaviour are present and these can be used both to assist a domain designer in refining the model and to suggest appropriate solution approaches.

The analysis techniques developed at Durham can identify certain key generic types and associated patterns. For example, the pattern associated with mobility comprises the mobile types, their maps and the predicate that defines locatedness of a mobile object on its map. A specific problem associated with mobility is route-planning, and a special purpose solution strategy suited to this problem can be to exploit travelling salesman heuristics. We were able to automate the configuration of planners with specialised route-planning capabilities enabling route-planning sub-problems to be handled using specialised approaches instead of by search [9, 13, 10].

The following software was developed at Durham as part of the PLANFORM project:

- Versions 4 and 5 of the STAN planning system. STAN [9, 10] performs generic type analysis and configures a special purpose planner suited to the associated generic patterns;
- Extensions to TIM [8, 15] to recognise a range of generic types in a domain model;
- The TIM API providing access to the generic type analyses performed by the TIM system, enabling their exploitation by other planning systems. The API is being exploited by other researchers in the international community;
- The OODL domain modelling language, supporting the construction of domains around generic types, and associated domain modelling tool DRAUGHTSMAN. These were developed by an MSc student and contributed to our collaboration with Huddersfield on the GIPO tool.

In the scope of the project a number of other generic types have been identified (for example, construction, resource-allocation, portability and others) and associated with specialised solution strategies. The configuration problem becomes complex when several generic patterns co-occur and their solution strategies must be integrated, and we have not completed the work required to support arbitrarily complex patterns of integration. We have, however, categorised the forms of integration that need to be handled and made progress with configurations based on several of these categories [17, 11, 9].

The library contains stored parametrized solution strategies (such as travelling salesman solvers, multi-processor scheduling heuristics etc) appropriate for sub-problems that commonly arise in planning domains. We do not try to guarantee complete coverage of all such sub-problems – the configuration system defaults to search if no generic types or suitable library components can be found. At present the library contains only one solution strategy per identified generic type, so extraction of a suitable strategy is simple. In general the extraction problem is more difficult because there may be different forms in which generic patterns arise and these might need to be matched in some intelligent way against the library. We have not explored this issue in the scope of PLANFORM. A recent extension we have made to the library is the addition of generalised control rules which can be selected and instantiated to fit a specific problem domain [23].

Most recently our work in these areas has considered the use of generic patterns as a basis for the development of *planning design patterns*. Using these, a domain construction tool can prompt the user for the components of generic patterns in a way that makes it simple for the user to enter that information. Initial work on a tool capable of supporting this idea was done by an MSc student [29] who was temporarily employed on the PLANFORM project at Durham. The work was continued in collaboration with the Huddersfield site which has focussed on the development of tool support [26].

ternational planning competition in 2000. STAN 4 can automatically detect mobility and resource-dependence patterns in planning domains and can extract route-planning and resource-allocation strategies from its library. Selected strategies are integrated, by means of a simple constraint-based interface, to a forward-search-based planner [10]. STAN 4 was one of the prize-winning systems, selected for the promise it showed in utilizing novel approaches to solving complex planning problems. Its plan quality was generally superior to that produced by the other competing systems.

Work remains to be done on increasing the sophistication of the integration techniques that support coordination of different specialised solution strategies within the overall framework. An important aspect of making the configuration tools available to the general planning community is to provide a clean means of access to the library, pattern recognition techniques and planner interface. A priority for future development is to supply an API to the suite of tools we have, which other planning systems could exploit. Although we made progress with the design and implementation of such an API we did not complete its implementation and it remains a topic for future work.

Overall Conclusions

The PLANFORM project set out to construct an open environment for planner development, bringing knowledge acquisition tools, domain construction tools, modelling languages and planner configuration components into an integrated organisation making planning accessible to the non-specialist.

The domain construction tool developed at Huddersfield produces PDDL domain descriptions providing a simple connection to the planner configuration architecture developed at Durham. The knowledge acquisition tools developed at Salford assist a user in confronting the task of domain construction through the GIPO tool. At this stage it is possible for a naive user to follow the entire process of modelling, and planning with, a specific problem domain without requiring detailed knowledge of any internal representation language (*OCL* or PDDL). The level of abstraction at which such a user can work within the environment will be further raised when the implementation of design patterns, as a guiding principle in the modelling process, is completed.

More time and work is necessary to evaluate the environment in terms of how successful it is at making planning accessible to the non-expert. There are several key aspects of the environment that require evaluation:

- The extent to which the object-oriented approach to modelling ameliorates the modelling task for the planning non specialist;
- The extent to which design patterns can further ameliorate this effort;
- The extent to which the proposed knowledge acquisition techniques can capture the modelling intentions of the user, and how transparent the environment can make the process of iterating over the modelling task until effective capture is achieved.

The last of these concerns the issue of how to provide useful feedback to the user when the modelling process fails to result in a consistent model (due to missing, or conflicting, information). Without a consistent domain model the plan configuration tools can do nothing useful and it is therefore desirable that the user be able to develop a correct domain model incrementally. We believe this is one of the most interesting technical challenges facing the PLANFORM project at this stage.

References

- [1] AIPS-98 Planning Competition Committee. PDDL - The Planning Domain Definition Language. Technical Report CVC TR-98-003/DCS TR-1165, Yale Center for Computational Vision and Control, 1998.
- [2] R. S. Aylett and C. Doniat. The PLANFORM-KA Tool. In *Proceedings of the 20th UK Planning and Scheduling Workshop (PLANSIG-2001)*, Edinburgh, Scotland, 2001.
- [3] R. S. Aylett and C. Doniat. KA Tool and domain construction for AI planning applications (submitted to ES2002). Technical report, University of Salford, 2002.

- [5] R. S. Aylett and S. Jones. Planner and Domain: Domain Configuration for a Task Planner. *Int. Journal of Expert Systems*, 9(2):279–318, 1996.
- [6] J. Breuker and B. Wielinga. Models of Expertise in Knowledge Acquisition. In *Topics in Expert Systems Design: Methodologies and Tools* (eds: G. Guida and C. Tasso), 1989.
- [7] B. Chandrasekaran and T. R. Johnson. Generic Tasks and Task Structures: History, Critique and New Directions. In *Second Generation Expert Systems* (eds: J. M. David, J. P. Krivine and R. Simmons), 1993.
- [8] M. Fox and D. Long. The Automatic Inference of State Invariants in TIM. *Journal of AI Research*, 9:367–421, 1998.
- [9] M. Fox and D. Long. Hybrid STAN: Identifying and Managing Combinatorial Optimization Sub-problems in Planning. In *Proceedings of the International Joint Conference on AI*, 2001.
- [10] M. Fox and D. Long. STAN 4: A Hybrid Planning Strategy based on Sub-Problem Abstraction. *AI Magazine*, AAAI Press, 22(4), 2001.
- [11] M. Fox and D. Long. Integrating Multiple Sub-solving Strategies into a Hybrid Planning System (submitted to the Journal of AI Research). Technical report, Department of Computer Science, University of Durham, 2002.
- [12] D. Liu and T. L. McCluskey. The OCL Language Manual, Version 1.2. Technical report, Department of Computing and Mathematical Sciences, University of Huddersfield, 2000.
- [13] D. Long and M. Fox. Extracting Route-Planning: First Steps in Automatic Problem Decomposition. In *AIPS Workshop on Analyzing and Exploiting Domain Knowledge*, 2000.
- [14] D. Long and M. Fox. Recognising and exploiting generic types in planning domains. In *Proceedings of the International Conference on AI Planning and Scheduling*, 2000.
- [15] D. Long and M. Fox. Recognizing and Exploiting Generic Types in Planning Domains). In *Proceedings of the Fourth International Conference on AI Planning and Scheduling (AIPS-2000)*, 2000.
- [16] D. Long and M. Fox. Multi-Processor Scheduling Problems in Planning. In *Proceedings of the International Conference on AI*, 2001.
- [17] D. Long, M. Fox, and M. Hamdi. Reformulation in Planning. In *Proceedings of the Symposium on Abstraction, Reformulation and Approximation*, 2002.
- [18] T. L. McCluskey. Planform: Draft Progress Report. *Technical Report, School of Computing and Maths, Univ of Huddersfield*, 2001.
- [19] T. L. McCluskey and D. E. Kitchin. A Tool-Supported Approach to Engineering HTN Planning Models. In *Proceedings of 10th IEEE International Conference on Tools with Artificial Intelligence*, 1998.
- [20] T. L. McCluskey and J. M. Porteous. Engineering and Compiling Planning Domain Models to Promote Validity and Efficiency. *Artificial Intelligence*, 95:1–65, 1997.
- [21] T. L. McCluskey and N. E. Richardson. The induction of operator descriptions from examples and structural domain knowledge. In *Proceedings of the 20th Workshop of the UK Planning and Scheduling Special Interest Group*, pages 181–192, 2001.
- [22] T. L. McCluskey, N. E. Richardson, and R. M. Simpson. An Interactive Method for Inducing Operator Descriptions. In *The Sixth International Conference on Artificial Intelligence Planning Systems*, 2002.
- [23] L. C. J. Murray. Reuse of Control Knowledge in Planning Domains. In *AIPS Workshop on Knowledge Engineering Tools and Techniques for AIP Planning*, 2002.
- [24] M. Musen. Modern Architectures for Intelligent Systems: Reusable Ontologies and Problem-Solving Methods. In *AMIA Annual Symposium*, 1998.
- [25] A. Newell. The Knowledge Level. *Artificial Intelligence*, 1982.
- [26] R. M. Simpson, T. L. McCluskey, Maria Fox, and Derek Long. Generic Types as Design Patterns for Planning Domain Specifications. In *Proceedings of the AIPS'02 Workshop on Knowledge Engineering Tools and Techniques for AI Planning*, <http://scom.hud.ac.uk/planet/aips02kett/>, 2002.

to OCL_h Translation. In *Proceedings of the 12th International Symposium on Methodologies for Intelligent Systems*, 2000.

- [28] R. M. Simpson, T. L. McCluskey, W. Zhao, R. S. Aylett, and C. Doniat. GIPO: An Integrated Graphical Tool to support Knowledge Engineering in AI Planning. In *Proceedings of the 6th European Conference on Planning*, 2001.
- [29] M. C. Tully. Object-Oriented Domain Modelling for Planning. Technical report, Department of Computer Science, University of Durham, 2001.
- [30] C. Vogel. *Le Genie Cognitif*. 1988.