Applying Design Space Analysis To Planning

Stephen T. Polyak
Department of Artificial Intelligence
The University of Edinburgh
80 South Bridge, Edinburgh EH1 1HN
United Kingdom
E-mail: Steve_Polyak@ed.ac.uk

Abstract

This paper describes a design space analysis approach towards a "complete" planning solution. A complete solution is defined as one containing the resultant plan, the context in which it applies, and the argument structure that justifies it. The focus in this paper is on defining and communicating the argument structure component. A perspective of a plan as a specialised type of design and planning as a specialised form of design activity is used. In doing so, research is drawn upon from the design rationale community for generating an explanation of a designed artifact. In particular, the method of generating a design space which represents the location of the plan within the space of possible plan elaborations is adopted. An initial implementation, Nonlin+DR, is described and its potential benefits to stand-alone and mixed-initiative planning is discussed.

Introduction

The traditional solution produced by an artificial intelligence planning system is a set of actions and ordering constraints. This result is the minimum output required to enact a plan but it represents only one component in a "complete" planning solution. The definition of a complete solution is drawn from work generated by the KADS-II project (Breuker & van de Velde 1994) which is discussed in more detail in the next section. An adaptation of this definition considers a complete planning solution to be one that contains

- a resultant plan
- a context in which the plan applies
- an argument structure that justifies the plan

The argument structure for a plan generated by an AI planning system is typically omitted from the solution. This omission limits the usefulness of the result and constrains the way a plan can be manipulated and reasoned about throughout the life-cycle of a plan. This argument structure represents the main component that is addressed in this paper. While complete solutions are not always necessary, increasing demands are being placed on solution representations for real-world planning situations. Richer knowledge about the planning process is needed to address organisational and environmental issues in these settings. The uses of a "batch solution" which is created by a sole planning agent, as well as an "incremental solution" which supports multi-perspective, mixed-initiative plan argumentation with multiple planning agents are considered.

In formulating an approach toward representing and communicating a complete solution, Tate's perspective of a plan as a specialised type of design (Tate 1996c) is utilised. Researchers in the design community have produced a number of methods and notations pertaining to the explicit representation of design rationale (DR) (Moran & Carroll 1996). Since design rationale provides the argument structure for a design artifact, it would seem fitting to apply these methods to planning as well. A previous paper pointed out the similarities between one such DR notation, QOC, and planning decision rationale (Polyak & Tate 1998). The approach behind this notation is called "design space analysis" which focuses on the output of a design as a design space rather than a single artifact (MacLean *et al.* 1991). This approach has been adapted for planning in a system, Nonlin+DR, using the University of Maryland's release of UM Nonlin (Ghosh *et al.* 1992; Tate 1977). Nonlin+DR supplements a plan solution with an externalization of the planning decision rationale. The output produced by this prototype system for a simple domain problem is reviewed.

The first section presents the definition of a complete solution as it is applied to planning. Next, the perspective of planning as a specialised type of design activity is considered. A specific approach entitled "design space analysis" is extracted from the design community and applied to planning. The prototype implementation, Nonlin+DR is then presented and discussed. The ways that Nonlin+DR could be used to assist in the overall process and possible directions which lie ahead are discussed.

What is a Complete Solution?

This definition is partially based on Newell and Simon's observation that the concept of a solution typically means different things in various situations (Newell & Simon 1972). In their work, a distinction is made between solution-objects, solution-paths, and solution-actions. A solution object is the direct result that one is typically interested in achieving. For example, in planning this would be actions and orderings and in diagnosis it would be a set of faulty components. Solution paths on the other hand consider the line of reasoning itself to be the focus. This can be seen as the result of a mathematical proof. The emphasis is not on arriving at the outcome hypothesised, but rather the way it was argued. Solution actions are plans or instructions that lead to required solutions and can be considered to be special case "solution objects". Based on this distinction and other sources, Breuker defines a complete solution as one that contains a case model, conclusion, and argument structure (Breuker 1994).

- Case Model - the understanding or conceptualisation of the problem.
- Conclusion - the answer to the question posed by the problem definition.
- Argument Structure - the reasons why the conclusion is supported.

In terms of planning, the case model is typically embodied by the domain knowledge and structure of the task assignment for a planning problem. The conclusion can be generally equated to the resultant plan. In most cases the argument structure is omitted or "compiled out" of the solution. While complete solutions may not be necessary in artificial settings, they are often required for real-world planning systems. We point out the need for this type of knowledge in two different planning approaches. On one hand, we consider a planning agent that plans in isolation (i.e. stand-alone), and on the other we examine the requirements that are placed on a planning agent involved in mixed-initiative planning.

Planning as Design

Recent work contributing toward international standardisation for process and plan interchange have produced new perspectives on plan representations. One of these perspectives relates plans to designs. Tate defines a plan as a specialised type of design where a "design for some artifact is a set of constraints on the relationships between the entities involved in the artifact" (Tate 1996c). A "plan" constricts this definition by specifying that the entities are agents, their purposes, and their behaviour.

Planning can then be considered to be a specialised type of design activity. Designs or plans are created by an agent or group of agents placing constraints on the developing artifact. The application of a constraint typically arises from a design decision that was made (e.g. the walls must be 4 in. thick, use expansion A rather than expansion B, etc.). We can think of these activities as repeatedly making design decisions that continually transform the artifact until it embodies the requirements necessary to enact the solution. In real-world scenarios for both planning and design we often have a need to understand the reasons behind these decisions.

Planning Decision Rationale

In a recent review of rationale in planning, Polyak and Tate describe a dimension of planning decision rationale (Polyak & Tate 1998). Decision rationale is the recording of the reasons why a specific decision was made in a particular way. Recording the rationale of these decisions adds value to the planning process in the following ways: facilitation of communication and reasoning; promoting a shared understanding of beliefs and intentions; maintaining a consistent approach; connecting agents to their responsibility in the plan process; and helping to steer the decision-making process.

Planning systems that are situated in an organisation must work in cooperation with a variety of agents. This may mean that humans and machines collaborate in the development and management of plans while sharing a common initiative. This has been termed "mixed-initiative planning". With a large number of people and systems working together to produce a solution, there is often a need to communicate intentions, beliefs, and justifications. When a decision is to be made, machine or human, the ramifications need to be considered within a "shared understanding".

Consider two human beings cooperating in the creation of a plan. What is important knowledge for them to share? Gross et. al. conducted a study in which two planners communicated via a microphone to collaborate on plan formation (Gross, Allen, & Traum 1993). In no case did the planners simply convey the plan as a set of actions. The agents identified goals and sub-goals, identified important actions, stated relevant facts that would help in the development of the plan, identified problems with what the other agent proposed, requested clarification, confirmed each others suggestions. Another study came to the same result with only a relatively small percentage of the discussion concerned with adding or refining actions (Allen, Ferguson, & Schubert 1996). This suggests that a richer model of plans is necessary to convey key pieces of knowledge needed to make planning decisions when human beings are involved. An "incremental solution" that contained this rationale could be open to argumentation, inspection, and justified modification throughout the planning process.

Rationale is also important in understanding and using a single agent planning system. This solution is considered to be "batch" in that the decision rationale is recorded in isolation and then is made available at the conclusion of plan construction along with the resultant plan. The types of decisions made by a single agent planning system are limited by the specific refinement methods that it can use. Understanding which refinement method was applied at various stages sheds light on the result of the planning process and opens new avenues of reasoning about the artifact.

Much of what has been said here about planning also applies to design. Designers cooperate by sharing rationale and often need to look behind the artifact to understand the deeper meanings behind the constructs. The research that has addressed this need in the design community is called design rationale.

Design Rationale

A design rationale is a representation of the reasoning behind the design of a system. It is essentially the explicit recording of the issues, alternatives and justifications that were relevant to elements in the design of an artifact. Examples of design rationale implementations include: QOC (MacLean *et al.* 1991), DRL (Lee 1990), gIBIS (Conklin & Begeman 1988). Each DR implementation offers some trade-off between (Lee & Lai 1991):

- expressiveness
- human usability
- computer usability

This trade-off can be expressed in the way that these notations or languages vary on a set of cognitive dimensions (e.g. premature commitment, viscosity, hidden dependencies, role expressiveness) (Shum 1991a). In reviewing these issues it is important to remember that ultimately the goal is to support design activities during the life-cycle of the design. This support addresses the design process in a number of ways. For example, a representation that includes design rationale has been shown to lead to a better understanding of the issues involved (Conklin & Yakemovic 1991). MacLean et al. list two major benefits from design rationale representation (MacLean *et al.* 1991): an aid to reasoning and an aid to communication.

All of these benefits: understanding, reasoning, and communication apply to several stages in the life-cycle of a design or plan. While the focus is usually on DR's contribution to the initial construction of the design, there is also rich support for the maintenance and reuse of the design as well. An artifact lacking rationale can often be hard to understand when revisited at a later date or by another agent who wasn't involved in the original design process. Changing requirements or environments may require incremental modifications to the design. Careful consideration for a particular implementation is necessary to achieve a balance that will facilitate, rather than hamper the planning process.

Design Space Analysis

The design space analysis (DSA) method which underlies the QOC semi-formal DR notation (MacLean *et al.* 1991) was selected for the implementation of Nonlin+DR. One of the main reasons for this choice was a similarity that can be seen between this approach and perspectives on how plans are built. QOC can be defined in the following way. Assume the existence of a finite set I of questions $\{Q_1, Q_2, ..., Q_n\}$ which reflect choices in the design/plan. Assume also a finite set J of options $\{O_1, O_2, ..., O_m\}$ and a finitie set K of criteria $\{C_1, C_2, ..., C_l\}$. Options provide alternatives $alt(O_j, Q_i)$ to questions posed during planning/design. Evaluative criteria may be be attached to options via an assesment relationship $a^+(C_k, O_j)$ or $a^-(C_k, O_j)$ which reflects whether the criteria either supports or detracts from the option. Additionally, a relationship may exist between options and questions in which the question is a sub-issue of an option $s(Q_i, O_j)$. Thus, a DSA is composed of $(I, J, K, \kappa, \lambda, \sigma)$ where κ is the set of alternative relations, λ is the set of assessments, and σ is a set of sub-issue relations. Figure 1 shows the general structure of a QOC diagram. QOC can be presented as a node-arc graph where the nodes are **Q**uestions, **O**ptions, and **C**riteria. The relations between these entities is expressed as arcs connecting the nodes.

Another reason for using QOC in Nonlin+DR is the flexibility and simplicity of the notation. The emphasis is on a representation that succinctly expresses the important relationships and does not require cumbersome inspection of the details or symbology. An empirical study of designers using QOC showed that designers required low amounts of training to pro-

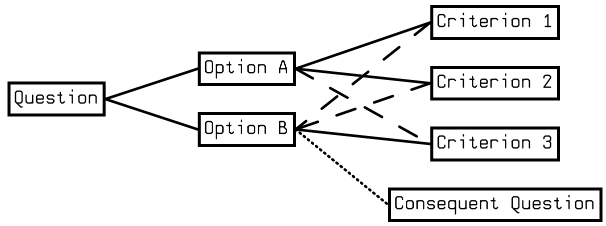

Figure 1: QOC, semi-formal notation to represent a design space. (MacLean *et al.* 1991). Dashed arcs between options and criteria denote negative influence whereas solid arcs indicate positive influence (i.e. arguing for or against an option).

ductively use QOC (Shum 1991b) for design tasks. The DSA perspective, along with its simple, straightforward presentation supports intuitive browsing to answer questions like: What are the other alternatives for this plan? How does criteria from one alternative affect another? What are the tradeoffs among them? etc.

DSA explains design rationale as defining how a given artifact is located in the space of possible design alternatives. Sets of these structures collectively define a "design space" of possible design realizations. This process of "design space" elaboration is similar to the work performed in planning. Tate stresses the importance of issues in his <I-N-OVA> framework (Tate 1996c; 1996b) which could be mapped to the use of questions in QOC. At a high level, a planning session could be defined by the issues (questions) considered (achieving a goal, assigning a resource, ordering nodes, etc.), the alternatives (options) posed (use operator A or B or C) and the justification (criteria) for those choices (using operator B requires less resource commitment). As it was pointed out before, this externalization of the planning process is not something that is typically produced in most planners today[1].

As these uses illustrate, representations are now required which weave together expertise on a variety of topics, techniques, and standards involved in complex domains. In each of these applications of AI-based plan representations we can see a set of rich plan/process elements at the core. This core may not only entail knowledge about the possible elaborations of behaviour that are valid for the plan specification (i.e. the artifact) but also knowledge about the planning, modelling, or (re)design process itself. For example, we may wish to capture and relate knowledge from both the space of decisions as well as the space of behaviour as shown in Figure 2.

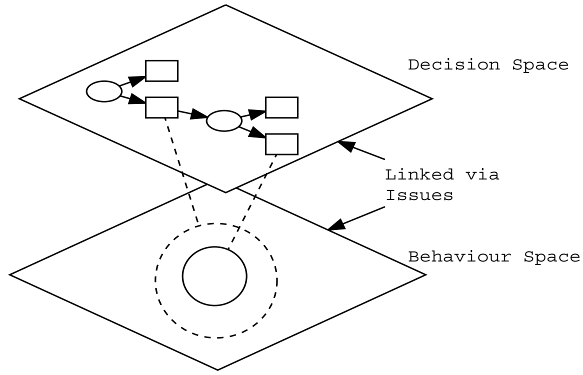

Figure 2: Capturing and relating decisions and behaviour.

In this diagram, decisions are represented by ellipses and boxes represent alternatives. Alternatives considered and selected in the decision space define new boundaries of possible actions in the behaviour space. These spaces are connected in part by the issues that drive this process. Different uses will require specialisations of this decision/behaviour knowledge to suit particular needs.

Recording Planning Decisions

In this section, a prototype system is described which was designed to record planning DSA rationale. A plan is contextualised as a specific elaboration in the possible space of planning decisions. This DSA method can be used to support activities in both mixed-initiative and classical AI planning (stand alone) settings. Currently the system only addresses a stand alone approach, but its mixed-initiative potential is examined in the following discussion section.

Nonlin+DR

A design space analysis approach has been implemented using the publicly available University of Maryland release of UM Nonlin (Ghosh *et al.* 1992). UM Nonlin is a Common Lisp implementation of some aspects of Nonlin, a hierarchical, Nonlinear, domain-independent planning system that was originally developed by Tate (Tate 1977).

This version, entitled Nonlin+DR, is capable of producing semi-formal rationale output in graph description language (GDL). GDL output can be visualised using the publicly available tool, XVCG (X-windows Visualization of Compiler Graphs). XVCG provides automatic formatting of the design space graphs expressed in GDL and effective management of high-level browsing with built-in interactive scaling. A visual interface for this core planning system was created using

[1] Exceptions to this include O-Plan (Currie & Tate 1991) which incorporates this as a design feature and research on explicit meta-plan driven systems.

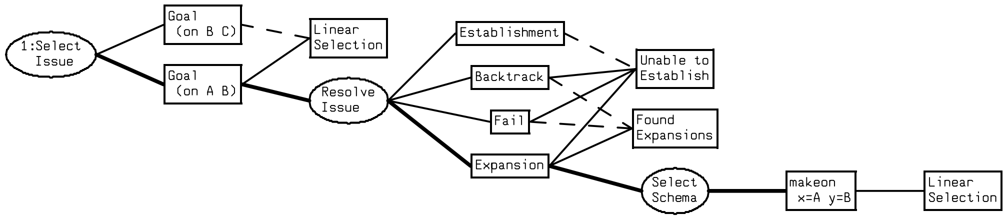

Figure 4: Nonlin+DR local design space for processing a single agenda item.

Tcl/Tk. This interface integrates simple task selection, option configuration, and viewing of the plan and associated rationale.

Currently, Nonlin+DR can be used in a classical AI "batch solution" mode. Once the planning process is complete it exports the recorded decision rationale to be presented by the XVCG tool. The rationale is composed of a set of local decision space graphs. The global decision space can be conceptualised as an aggregation of local decision spaces. Each local decision space maps to the processing of a single issue or agenda item. A review of a simple "sussman anomaly" problem will help to explain this approach.

A standard blocks-world domain is used for this example. In this domain there are two operators corresponding to higher level "operator" schemas: makeon and makeclear. One primitive action schema, puton, is used to define low level activity[2]. The task that is sent to the planner is shown in Figure 3.

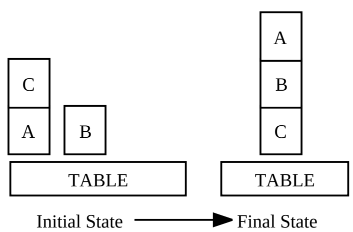

Figure 3: Task Formulation for the Sussman Anomaly Problem.

It is the classic sussman anomaly which is a conjunction of two interacting goals. This problem is typically used in AI planning to show that the simple "linear" approach to solving the two goals in any order will fail. The first local design space[3] generated by Nonlin+DR is represented in Figure 4.

[2]See the UM Nonlin manual (Ghosh *et al.* 1992) for more detail on these operators

[3]The global design space for this problem is available from the author.

Select Issue

The first decision that Nonlin+DR was faced with was which goal to work on. The alternatives considered are connected to the right of the decision. At this point, the planner was able to either select (on a b) or (on b c). Nonlin+DR does not have a very sophisticated mechanism for agenda selection as it only relies on one very basic criteria, linear selection. The algorithm is hardwired to always process these items in a FIFO manner and is unable to treat this decision opportunistically. This is modelled as a single decision criteria that has an influence on each item in the agenda. Solid criteria links represent positive influence and dashed links represent negative influence (i.e. arguing for or against an alternative). In this case, linear selection criteria will always argue for the first in line and against all others. A bold link from a decision to an alternative indicates the selected course of action. Obviously here (on a b) is selected. A bold link that carries on from a selected alternative indicates the deliberation of a subsequent decision.

Resolve Issue

In this local design space, Nonlin+DR next considered how to resolve the issue. At a high level, the alternatives for resolving a goal are establishment or expansion (Tate 1977). It is also possible that the planner may decide to backtrack or fail at this point as well. The planner considered the argument for establishment and realized that there is no support for this. Nonlin+DR records this criteria as arguing against establishment and favouring expansion, backtracking, or failing. When considering the expansion option the planner noted that there was at least one expansion that corresponded to the goal. This favoured expansion over backtracking or failure. The selection to expand then lead the planner to another, rather simple, decision of how to expand.

Select Schema

Since there was only one possibility the planner chose it as the way to update the plan in progress. Even if there was more than one way to perform this expansion

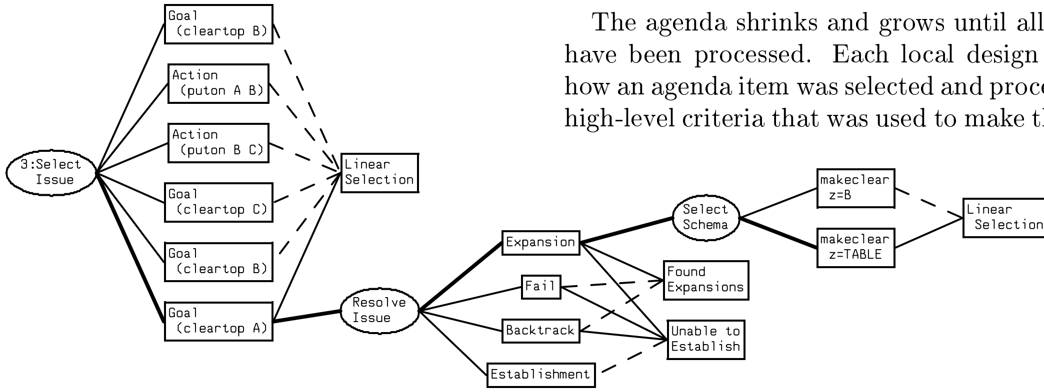

Figure 5: Design space resulting from different variable binding choices.

the decision would still have been very straight-forward because the schema selection only considers linear selection criteria again. An update may add items to the agenda as it does in this case. The planner then moved on to select the next agenda entry which is then described in the next local design space.

Note that the alternatives for an expansion also contain the variable bindings selected for the schema.

Expansion alternatives may be due to different schemas that have the same ":todo" pattern but they may also be different instances of the same schema with different bindings. For example, consider the way that the planner addressed the goal "(cleartop A)" in Figure 5. For the "select schema" decision, the planner had the choice of either placing C on B or placing C on the table. The table was chosen because this variable binding set was ordered before the other alternative. This was rather fortunate because if the variable binding for B was selected instead it would have led to an inefficient plan where C was unstacked onto B and then subsequently unstacked onto the table.

Resolve Conflict

In Figure 6 the schema "puton" was selected to address the "(puton A B)" issue. The planner detected a conflict between an effect from this proposed action and a condition in another part of the plan. Specifically, this action would negate "(cleartop B)" needed to place B on C. In order to utilise this schema, the planner had to make a subsequent decision on how to resolve this conflict. Thus we see that the design space is further defined by alternatives for conflict resolution. These alternatives are either: link "(puton B C)" before "(puton A B)" or link "(puton A B)" before "(cleartop B)". In this case, the planner chooses to link the stacking of B C before the stacking of A B. Again this was a straight-forward linear selection from a list of possible ways to address this problem.

The agenda shrinks and grows until all of the items have been processed. Each local design space shows how an agenda item was selected and processed and the high-level criteria that was used to make the selections.

Thus, the global design space is an aggregation of the local design spaces explored for each agenda item and represents the overall decision rationale of the plan.

Discussion

This example used here is rather simplistic in two respects. Firstly, this blocks world domain is particularly sparse and does not offer much in the way of "interesting" alternatives. Secondly, the underlying UM Nonlin planner considers only very basic criteria for option selections (e.g. agenda selection, schema instance selection, etc.). The focus of this example though was to clearly explain how DSA could be applied in a basic Nonlinear planning session before moving on to more challenging domains and planners. Work on this example has produced a list of items to work on and has shown potential issues to consider when scaling up this approach for more difficult domains and sophisticated planning situations.

Items for future work include an enumeration of a wider set of decisions that are made by planning agents (humans or machines). Some of these decisions will naturally come out of a move toward richer situations (e.g. selecting a resource, associating a task executor, etc.). DSA may also be used to show how various planning systems utilise different approaches[4] and criteria (e.g. linear selection, random selection, smart selection) for the same problems.

A determining factor for this progression will be its application to mixed-initiative planning. The design space approach is seen as a unique way of placing the plan in its broader context. This context could help to focus mixed-initiative discussion on the relevant alternatives and criteria for a specific part of the plan. It may also indicate criteria/alternative interaction that was unforeseen or alternatives that may have

[4] For example, the alternatives for Nonlin+DR's decision rationale is reflected by it's backward state space, HTN, and plan space refinement methods.

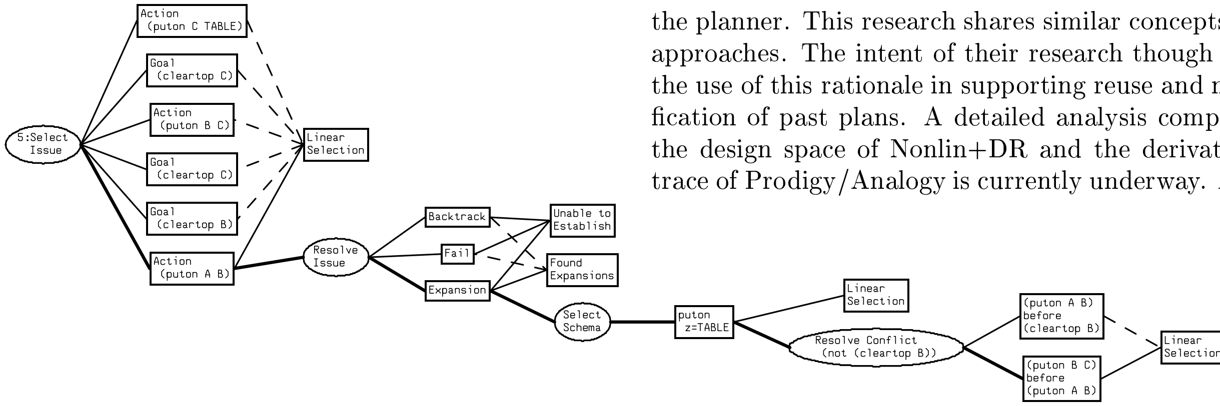

Figure 6: Conflict resolution in the design space.

been left undiscovered. In order to achieve this level of interaction though it will be necessary to open up the planning interface to allow a user or group of users to control and inspect the planning choices during plan generation. This is similar to what has been done for Prodigy/Analogy (Veloso 1996) and earlier in the work on PLANIT (Drummond & Tate 1992), an interactive planner's assistant.

The DSA approach also has several potential benefits in a stand alone setting. One aspect is in debugging a problem found in a planning result. Chien identified two common problems resulting from knowledge encoding errors: Incorrect plan generation; and Failure to generate a plan (Chien 1996). In both instances, the DSA rationale can be used to quickly localise the error and fix the precondition, effect, or variable specification that may have caused the error. Domain additions and modifications can be reviewed as contributing to the plan space even if they weren't part of the "selected" plan solution.

Related work

PLANIT was a prototype system used for project management, process planning, and job shop scheduling for the UK Alvey Programme PLANIT club (Drummond & Tate 1992). This system used rich plan representations which, among other things, could help a user determine how the original plan was constructed and what the aims of the plan were. While this paper focussed on the automatic generation of rationale from a planning system, the PLANIT work served as an open representation to support human agent plan generation, analysis, and modification.

Another system concerned with the capture and representation of plan rationale is Prodigy/Analogy (Veloso 1996). Veloso and Carbonell developed a language for capturing the reasons that support choices made by the planner. This research shares similar concepts and approaches. The intent of their research though is on the use of this rationale in supporting reuse and modification of past plans. A detailed analysis comparing the design space of Nonlin+DR and the derivational trace of Prodigy/Analogy is currently underway. Allen and Ferguson suggest another perspective on representing plan decision rationale (Ferguson & Allen 1994). Rather than seeing the plan and its argument structure as two separate structures they suggest a method for combining the two. They describe a formal model of plans based on defeasible argument systems. One of the problems with this approach though is that it requires existing planning algorithms to be recast in terms of this formalism. The DSA approach, on the other hand, requires a lower level of commitment to incorporate these ideas.

Summary

In demanding, real-world planning situations we need "complete solutions" to address the associated requirements. Since planning can be viewed as a special type of design activity it makes sense to try to apply design rationale methods to planning as well. The design space approach views the solution as located in a space of possible elaborations. Capturing and externalizing these elaborations creates a more robust solution that supports an intuitive inspection of the decisions made, the alternatives considered, and the influence of certain criteria on these alternatives.

The potential benefits of this approach were described for both a mixed-initiative and classical AI planning settings. Outstanding items and issues have been raised to address more challenging settings. It is anticipated that the application of this approach to richer domains and more sophisticated planning situations will elicit a greater set of elements for a model of planning rationale.

Acknowledgements

The author is sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant number F49620-96-1-0348 – an AASERT award monitored by Dr. Abe Waksman and associated with the O-Plan project F30602-97-1-0022. The U.S. Government is authorised to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation hereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing official policies or endorsements, either express or implied, of DARPA,AFOSR or the U.S. Government.

References

Allen, J.; Ferguson, G.; and Schubert, L. 1996. Planning in complex worlds via mixed-initiative interaction. In Tate (1996a), 53–60.

Breuker, J., and van de Velde, W. 1994. *The CommonKADS Library for Expertise Modelling: reusable components for artificial problem solving.* Amsterdam, Tokyo: IOS Press.

Breuker, J. 1994. A suite of problem types. In breuker94a (1994), 57–87.

Chien, S. 1996. Intelligent tools for planning knowledge base development and verification. In Shadbolt, N.; O'Hara, K.; and Schreiber, G., eds., *Advances in Knowledge Acquisition, 9th European Knowledge Acquisition Workshop*, 321–337.

Conklin, E., and Begeman, M. L. 1988. gIBIS: A hypertext tool for explanatory policy discussion. *ACM Transactions on Office Information Systems* 6:303–331.

Conklin, E., and Yakemovic, K. 1991. A process-oriented approach to DR. *Human Comp. Inter.* 6:357–391.

Currie, K., and Tate, A. 1991. O-Plan: the open planning architecture. *Artificial Intelligence* 52:49–86.

Drummond, M., and Tate, A. 1992. PLANIT interactive planner's assistant. Artificial Intelligence Applications Institute AIAI-TR-108, University of Edinburgh.

Ferguson, G., and Allen, J. 1994. Arguing about plans: Plan representation and reasoning for mixed-initiative planning. In Burstein, M., ed., *Proceedings of the ARPA/Rome Laboratory Knowledge-based planning and Scheduling Initiative Workshop*, 123–131. Tuscon, AZ: Morgan Kaufmann.

Ghosh, S.; Hendler, J.; Kambhampati, S.; and Kettler, B. 1992. UM Nonlin Version 1.2.2 User Manual. Department of Computer Science, University of Maryland, College Park, MD.

Gross, D.; Allen, J.; and Traum, R. 1993. The TRAINS-91 dialogues. Department of Computer Science TRAINS Technical Note 92-1, University of Rochester, Rochester, NY.

Lee, J., and Lai, K. 1991. What's in design rationale? *Human Comp. Inter.* 6:251–280.

Lee, J. 1990. SIBYL: A qualitative decision management system. In Winston, P., and Shellard, S., eds., *Artificial Intelligence at MIT: Expanding Frontiers.* Cambridge, MA: MIT Press. 104–133.

MacLean, A.; Young, R.; Bellotti, V.; and Moran, T. 1991. Design space analysis: Bridging from theory to practice via design rationale. In *Proceedings of Esprit '91*, 720–730.

Moran, T., and Carroll, J., eds. 1996. *Design Rationale: Concepts, Techniques, and Use.* Lawrence Erlbaum Associates.

Newell, A., and Simon, H. 1972. *Human Problem Solving.* Prentice Hall.

Polyak, S., and Tate, A. 1998. Rationale in planning: Causality, dependencies, and decisions. *Knowledge Engineering Review* 13(2):1–16.

Shum, S. 1991a. Cognitive dimensions of design rationale. In Diaper, D., and Hammond, N., eds., *People and Computers VI*, 1–13. Cambridge: Cambridge University Press.

Shum, S. 1991b. Cognitive issues in representing design reasoning as hypertext. *SIGCHI Bulletin* 23(1):38–40.

Tate, A. 1977. Generating project networks. In *Proceedings of the International Joint Conference on Artificial Intelligence(IJCAI-77)*, 888–893.

Tate, A., ed. 1996a. *Advanced Planning Technology: Technological Advancements of the ARPA/Rome Laboratory Planning Initiative.* Menlo Park, CA: AAAI Press.

Tate, A. 1996b. Representing plans as a set of constraints – the $<$ I-N-OVA $>$ model. In Drabble, B., ed., *Proceedings of the Third International Conference on Artificial Intelligence Planning Systems (AIPS-96)*, 221–228. Edinburgh, Scotland: Morgan Kaufmann.

Tate, A. 1996c. Towards a plan ontology. *AI*IA Notiziqe (Publication of the Associazione Italiana per l'Intelligenza Artificiale), Special Issue on Aspects of Planning Research* 9(1):19–26.

Veloso, M. 1996. Toward mixed-initiative rationale-supported planning. In Tate (1996a), 277–282.