Rationale in Planning:
Causality, Dependencies, and Decisions

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

Austin Tate

Artificial Intelligence Applications Institute
The University of Edinburgh
80 South Bridge, Edinburgh EH1 1HN
United Kingdom
E-mail: a.tate@ed.ac.uk

Abstract

Traditional approaches to plan representation focused on the generation of a sequence of actions and orderings. Knowledge rich models, which incorporate plan rationale, provide benefits to the planning process in a number of ways. The use of rationale in planning is reviewed in terms of causality, dependencies, and decisions. Each dimension addresses practical issues in the planning process and adds value to the resultant plan. The contribution of this paper is to explore this categorisation and to motivate the need to explicitly record and represent rationale knowledge for situated, mixed-initiative planning systems.

1 Introduction

Planning is one of the oldest areas of research in Artificial Intelligence. The traditional view of the class of problems that a planner is expected to solve is: given a description of an initial world state, goal state, and domain, produce a sequence of actions that transforms the initial state to the goal state. Over time, several approaches have been developed to address these problems [Allen *et al.* 90]. While the result is a rich corpus of techniques and methods, it is proving to be a very difficult task to compare and contrast each approach. Some researchers believe the best way is to chart these results with detailed algorithmic treatment [Kambhampati *et al.* 95]. Barros, Valente, and Benjamins present a differing perspective:

> "...there is a gap in the analysis of planning systems. Algorithms and their efficiency must surely be a central concern, but if we are to engineer successful applications there must be also a more abstract analysis that highlights the capabilities of the system and the way it represents and uses knowledge..." [Barros *et al.* 96].

The intent of this paper is to contribute a review of the way planning systems have represented and used *rationale* knowledge. This can be traced back to the early beginnings of artificial intelligence planning, when the utility of recording such knowledge had been cited [Newell & Simon 63, Sussman 74, Hayes 75]. Rationale has been used in generating plans but has also been applied to other areas of planning as well (e.g. plan analysis, plan execution, etc.). This has become a large category of knowledge present in the planning process and it requires an epistemological review of the content suggested in this grouping. As these systems continue to mature, rationale representation will become an increasingly important source in addressing a variety of issues. These issues and the corresponding rationale are examined along three axis: **causality**, **dependencies**, and **decisions**. These three dimensions are strongly interrelated and contribute toward a richer model of plan representation.

"Plan rationale" has been loosely described as "why the plan is the way it is" [Wilkins 88]. Wilkins' more detailed description highlights the multidimensional basis of rationale:

> “The primary tasks of the plan rationale ... are to encode why nodes are in the plan, how nodes should be grouped together into sub-plans that accomplish a goal, how long the truth of a particular goal must be maintained and how different abstraction levels connect”

This seems to be an excellent starting point. Let’s review what was said here and address where this knowledge will be treated in the paper. The first item, “why nodes are in a plan”, will be discussed as an aspect of causal rationale. “How nodes should be grouped” is considered part of the decision rationale of the planning process. The maintenance of truth ranges spans the dependency and causal rationale while the connection of abstraction levels is reviewed as knowledge in the dependency rationale. This interpretation is summarised in table 1.

Issue	**Rationale**
Why nodes are in a plan	Causality
Choosing nodes to group into sub-plans	Decisions
Maintenance of truth ranges	Causality, Dependencies
How plan levels connect	Dependencies

Table 1. Wilkins’ definition of plan rationale.

Valente has produced a high-level CommonKADS analysis of knowledge used in planning [Valente 95]. The main division is between static (knowledge that does not change during the problem solving process) and dynamic (knowledge that does change) roles. This paper will address knowledge defined in both sets. The dynamic set contains several knowledge roles that are part of the overall rationale, such as “causal links” and “conflicts”. The static set is also explored with elements that correspond to the “world description” and “plan description” defined in their paper.

2 Causal Rationale

Causal rationale supports the planning process in a number of powerful ways. In a more general sense, McDermott pointed out: “Causality is fundamental to a lot of problem solving. A problem solver brings things about by causing other things” [McDermott 82]. The explicit recording of “what was caused” during planning or causal relationships can be traced back to the early beginnings of AI planning systems. This knowledge has been used in:

- controlling search
- connecting plan elements to their purposes
- establishing protection ranges
- ensuring correct planning results
- plan monitoring
- plan interpretation and analysis
- plan execution

Causality information is recorded as the result of decisions made by the planning system, but causality can also be explicitly represented in the domain description as well. We first take a look at how various planners maintain this knowledge throughout plan generation. We then review research that suggested possible formalisms for representing domain causality.

2.1 Generative Causality

The highly influential work of Newell, Shaw, and Simon introduced the concept of means-ends analysis [Newell & Simon 63]. This research came out of their interest in the psychology of human thinking. The GPS (General Problem Solver) program was developed as a vehicle for testing their theories in a computational setting. In this work, the means-ends analysis technique was suggested as directing the search for successful plans by looking for those operators that directly addressed outstanding goals. This connected the presence of an operator with what *caused* it to be incorporated in the plan. In this case, "achieving a goal" caused a specific operator to be searched for and selected.

Sussman recognised the value of these causes when he explored errors that occur in human problem solving [Sussman 74]. He viewed the detection and repair of a "bug" or a protection violation to be a central aspect of this process. A protection violation was traced to its underlying cause. Sussman developed the HACKER program to encompass his theories. HACKER utilised this knowledge about a bug to improve its performance. Terming this aggregate knowledge of conditions, effects, and intentions as 'teleology', he also showed how interactions between operators could be detected and resolved when planning for conjunctive goals. Teleological information introduced a new causal relationship where two actions' effects/conditions caused an ordering constraint to be added to resolve the interaction.

Using the work of Sussman as a basis, Tate developed the concept of "goal structure" (GOST) that was used to incrementally record the conditions on nodes of the network together with points where the conditions are achieved [Tate 75, Tate 77]. For example,

{supervised {scaffolding erected} = true at 6 from [4]}

would indicate that the pattern, {scaffolding erected}, had to be true at node 6 and one node that contributed this effect is node 4. This information facilitated the search process by treating the planning problem as a search through partial plans. This also provided "ranges" in which a certain pattern would be expected to retain its value. These "ranges" in turn, allowed the planning system to detect interactions and suggest intelligent node orderings that took the node effects into account. The causality recording of O-Plan [Currie & Tate 91], which draws upon Tate's earlier work on Nonlin, has widened to encompass a richer model that includes resources [Tate 93]. For example, the reason that an action may be included is to provide a resource for another part of the plan, but its post-conditions can be treated as a side effect for the particular use.

It began to be apparent that the complexity of planning problem domains required a mechanism that could abstract the domain to make it more manageable. Research in human problem solving suggested an approach to this issue [Pólya 45]. The approach was based on a hierarchical arrangement of knowledge with an increasing amount of detail applied at lower levels of the structure. Hierarchical planners like Sacerdoti's NOAH [Sacerdoti 75] and Tate's Nonlin [Tate 77] incorporated this idea and introduced another causal element in planning rationale. The process of searching for sub-reductions of a higher-level node resulted in the inclusion of a node or set of nodes as a detailed expansion of the plan. The set of expansion nodes could then be attributed to the higher-level node that represented an abstraction of the set. NOAH's *critics* utilised this information to ensure that the effects of the detailed plan maintained global correctness with the rest of the plan. NOAH implemented a "table of multiple effects" that was used to record expressions that were asserted or denied by more than one plan node. These entries could then be used to detect conflicts in a hierarchical plan.

In a related approach, SNLP (systematic Nonlinear planner) causal-link planners record links from a producer step to a consumer step during the planning process [McAllester & Rosenblitt 91, Barrett & Weld 94]. In this research, systematicity means that the same plan or partial plan is never explored more than once. A step (producer or consumer) is similar to a plan operator in other planning systems. A causal-link is a 3-tuple, <s,P,w>, where P is a propositional symbol, w is a step that has P as a prerequisite and s is a step that has P as an effect. This is expressed as: $s \overset{P}{\rightarrow} w$. This information is similiar to that contained in Tate's goal structure (GOST) discussed above and is used to identify threats and to resolve conflicts in its steps by providing an interval of truth for a condition. SNLP has recently been extended to CNLP (conditional Nonlinear planner) to incorporate conditional aspects [Peot & Smith 96]. The approach that was implemented was similar to an ATMS [deKleer 86]. [1] This labelling technique used by CNLP assists in tracking which actions are visible to other actions in the plan. This permits a "test" to be inserted in a plan so that the results during plan execution can lead to a more accurate set of possible steps to take through the rest of the execution.

Research continues to refine generative causal rationale and find new applications for it. Not all of these applications are restricted to generative planning. For example, this knowledge has been used in a plan execution agent to detect protection violations while carrying out a plan [Reece & Tate 94]. An execution agent that has a richer model of the plan, which includes causality, is in a better position to try to correct minor violations rather than refer to the planner for assistance. This permits a greater degree of autonomy in uncertain, dynamic environments.

2.2 Domain Causality

The causality discussed above is built extemporaneously as the planning system reasons about elements in the plan. This reasoning is largely based on causal relationships that are built implicitly or explicitly into the domain representation. Thus, domain causality represents a significant component in planning causal rationale. In this section, we point out some approaches toward domain causality representation.

One of the most well-known problems in this area is the frame problem [McCarthy & Hayes 69]. If we are to create a domain for state-based reasoning, we require a way to state facts that change as well as those that do not. The majority of planners adopt a STRIPS-style representation [Fikes & Nilsson 71] where an operator will explicitly say what will be added to or deleted from a state when an operator is applied. The "STRIPS assumption" then is that nothing else changes. Causality is implied by specifying which effects must hold before and after an operator is added to a plan. The majority of current planners today are still based on this simple STRIPS representation, but more complex domain formalisms are required to manage real-world domains.

Tate developed a declarative domain description langugefor Nonlin called Task Formalism (TF) [Tate 77]. With TF, a domain expert is able to describe causality information about actions in the domain separately from action orderings. This supports an explicit, network-style description of the operators rather than burying these ordering relationships in conditions and effects. In SIPE [Wilkins 84] and later SIPE-2 [Wilkins 88], Wilkins took an innovative approach towards extending the representation of causality by allowing a "causal theory" of a particular domain to be expressed as a set of causal rules, state rules, and init-operators. One of the key contributions of this approach is that actions whose effects are dependent upon world states can be defined without creating specialised operators that correspond to all of the possible situations in which an action takes place [Ludlow & Alguire 94]. For example, consider these two causal rules:

causal-rule: move-object-held
arguments: person1, location1, object1;
trigger: (at person1 location1);
precondition: (holding person1 object1);
effects: (at object1 location1);
end causal-rule

causal-rule: one-location-at-a-time-object
arguments: object1, location1, location2 is not location1;
trigger: (at object1 location1);
precondition: (at object1 location2);
effects: (not(at object1 location2);

[1] The relationship between plan rationale and truth-maintenance systems is examined in section 3.1.

end causal-rule

The first rule states that whatever a person is holding goes with them when (s)he moves to a new location. Without this rule, we would need to find all of the operators that involve moving of people and duplicate them all to provide a set with additional effects for when they are holding something. The second rule is used to deduce that if an object is in a new location (for any operator that moves objects) then the object cannot be at its old location.

Planning research has also focused on defining the types of causality found in domain knowledge. In Allen's formalism for reasoning about actions [Allen 84], he suggests two kinds of causation: ECAUSE, where an event occurs that causes another event to occur and ACAUSE where an agent causes an event in a way that can be considered the "action" of the agent. So for instance, if "Bill breaks the window" it would normally be interpreted to be an "ACAUSE" since Bill intentionally caused the event. It is possible though that Bill was thrown through the window, thus being the object of the situation and the event of Bill being thrown caused (ECAUSE) the event of the window being broken. This type of information assists in interpreting the causal rationale and interrelationships of plan events and actions.

A different view separates causality from eventuality [Lansky 87]. For instance, when making a plan to dine at a restaurant we may represent "making a reservation" with a causal relationship to "being seated at a table". There remains a possibility though that the mere presence of the causal event will not lead to the subsequent occurrence (e.g. the reservation is cancelled, etc.). This implies that the "causal relationship" must be monitored during execution to verify effects. In fact, this represents an entire area of "probablistic planning" which rejects the deterministic assumptions of classical planning. This has been studied by researchers examining uncertainty in AI planning [Kushmerick *et al.* 95, Dean *et al.* 93, Goldman & Boddy 94].

Georgeff on the other hand considers causality to be one of the following two types: an event causes the occurrence of a later event, or an event causes the simultaneous occurrence of another event [Georgeff 87]. This knowledge can be used to relate events, and to form a grouping that can be referred to singularly as a "process". Formulating the concept of process with respect to causality in planning is useful because we can focus on higher level rationale where we can refer to causality that is internal or external to a process. This area begins to show the overlap in "causality" and "dependencies" as they relate to plan rationale and is discussed in the following section.

Looking back in summary, we have seen how causality has played a role in the planning process. The techniques discussed here are still used today. Some examples reviewed were: means-ends analysis, reordering for conflict resolution, and hierarchical decomposition of operators. Also, many of the structures discussed are also present in one form or another: goal structure, table of multiple effects, and causal links. We examined various approaches in the planning literature for representing causality in domain models. By far, the most important of these has been the STRIPS representation that underpins current domain-independent planners.

3 Dependency Rationale

It can be argued that causality and dependencies in the planning process are highly interrelated. We agree with this but we present this axis of rationale knowledge separately from causal knowledge to highlight the role and contributions of dependency rationale. This knowledge has been used in:

- defining plan element interrelationships
- replanning
- backtracking search
- plan reuse and refitting

- protecting values
- revision of beliefs

As in causal rationale, dependencies can be recorded during planning but in some cases they are computed from the resultant plan network. Some planners, particularly those that support reuse of previous plans, also save dependencies along with the plan. These relationships are governed by constraints in the domain so we also examine domain dependency representations as well.

3.1 Recording Dependencies

Early work in planning motivated the need to capture the dependencies that were formed during plan generation [Hayes 75, Stallman & Sussman 77, London 78]. In order to describe this need we must review what happens during plan decision-making. Then we will see how recording plan dependencies helps to improve this process.

Dependencies arise from plan decisions. A planner typically has a number of alternatives to choose from when generating a possible solution to a planning problem. These choices may involve: selecting an operator to achieve a goal, expanding an abstract node, ordering conflicting operators, etc. An option is selected from the possible set, but in order to preserve completeness a planner may store the other alternatives as a "choice point". Choice points are typically used to "remember" a list of remaining possible expansions or linearizations for a given choice. Thus, if the plan process gets stuck, it will back up to a choice point, throw out the work done past the point, and choose again. One of the major problems with this simplified form of backtracking is that valid work done subsequent to the offending point will be simply thrown out. By "valid work" we mean that parts of the plan structure may not have "depended" on the choice that had to be remade. These parts may be still useful under the new choice. Since these parts had the unfortunate positioning to be below the choice point, work must be duplicated to re-add them to the plan.

Hayes' solution to this problem came in the form of a "decision graph". A decision graph was used to record the dependencies between planning decisions (dnodes) and nodes in a journey plan (jnodes) as the plan was being built [Hayes 75]. An example from this work is illustrated in figures 1-3. Figure 1 shows the initial world for a "box-pushing" scenario which contains a robot, represented by R1, two boxes, four doors, and four rooms. Figures 2 and 3 respectively contain a subgoal tree for a journey plan to move the two boxes next to each other and a corresponding decision graph. The dnodes in the decision graph have pointers to the jnodes that were created as a consequence of the decision. This link between dnodes and jnodes is shown by the numbers within the parentheses.

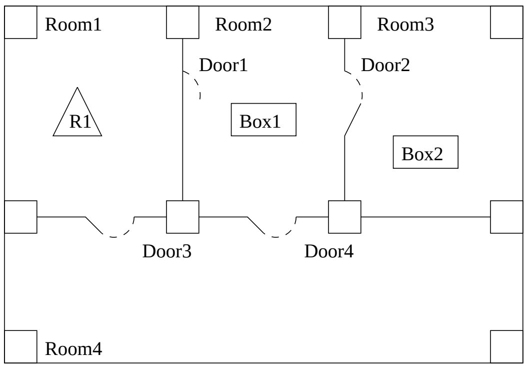

Figure 1: Initial world for decision graph example

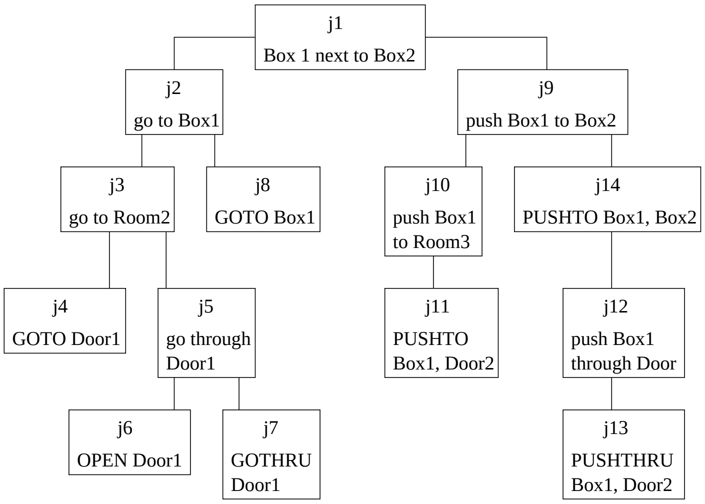

Figure 2: Hayes' subgoal tree of plan for pushing two boxes together

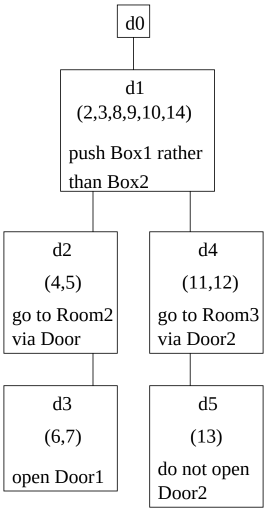

Figure 3: Hayes' decision graph

These dependencies permitted intelligent plan modifications when a new decision needed to be made. For example, a plan executor may have discovered new information that disallowed the performance of some action (jnode) in the plan. This could be the case in the box-pushing plan of figure 2 if it is discovered that Door1 was locked. The jnode that required a new choice selection was traced back to the dnode (or decision) that introduced it into the plan. In response to the locked door we discard the already executed j4 and apply the replanning procedure to j5. Next we determine that d2 was the decision that was responsible for j5. The dnode then identifies the other jnodes and dnodes that "depended" on it and they are removed. In this scenario: j5, j6, j7, d2 and d3 are all discarded. The new plan may then contain activities for the robot to get to Room2 via Room4. This allows a planner to avoid replanning potentially valid, independent aspects of the plan.

Following Hayes' work, decision graphs were also added in 1977 to the Nonlin planner to assist in modifying plans [Daniel 83]. Daniel characterised two types of decisions that are made in generating a plan:

- choice of expansion for a node.
- choice of links to remove an interaction.

The type of decision being recorded determined the type of node to be stored in the decision graph. Expansion nodes linked back to their parent decision as well as to the new nodes in the plan network. Linking nodes connected expansion nodes and plan nodes with an ordering constraint (e.g. before). Truth maintenance systems (TMS) have been suggested as one way to implement this type of functionality.

Maintenance systems (e.g. TMS) are studied as a sub-area of artificial intelligence, and are a type of Reason Maintenance System (RMS). One of the useful characteristics of a RMS is that it allows the use of dependency-directed backtracking instead of chronological backtracking. The relationship of planning and TMS is actually a very interesting one. This early work on recording dependencies in planning inspired the original research in this area. Doyle pointed out that a reasoning program needed to be able to make assumptions in order to choose actions and these assumptions or beliefs may need to be revised during its course of action [Doyle 79]. These systems facilitated incremental recording of the assumptions on which an elements' justification depended and also provided a means to perform reasoned retraction of these assertions. This retraction mechanism supported a class of backtracking procedures, termed dependency-directed backtracking, which permitted the maintenance of a planning system's nonmonotonic belief set. Work continues to be done on incorporating TMS or "reason maintenance systems" into planning [Doyle 94, Doyle 96]. The current focus is on an "incremental application" that is flexible and customisable to the planning purposes.

3.2 Supporting Plan Reuse

Dependencies also play a key role in case-based planning or plan reuse. For example, the PRIAR system annotated plans with information about the dependency structure between operators [Kambhampati 89, Kambhampati & Hendler 92]. Annotations were computed after the generative planning process was complete and then this information was stored with the resultant plan. These annotations were comprised of sets of "validations". A single validation in PRIAR was a 4-tuple $\langle E, n_s, C, n_d \rangle$ where n_s and n_d are leaf nodes belonging to the hierarchical task network (HTN) and E is the supporting effect of n_s used to satisfy the applicability condition, C, of node n_d. Plan nodes were annotated with sets of validations that indicated information about the node's sub-reduction. One set would contain the validations that were supplied to other nodes, another set would contain validations that were consumed by the node, and a final set would contain validations that were required to "hold" over the node. The validations of the leaf nodes were then copied up the HTN. Validation states were then used to determine candidates for retrieval from a library of plans, "refitting" the old plan to the new situation, and replanning.

Veloso's work on Prodigy/Analogy is an excellent example of the emerging utility value of dependency rationale in the planning process [Veloso 96]. This work is also addressing areas of "decision rationale", which is discussed in the next section. Prodigy/Analogy, like PRIAR, focuses on the selection and use of past plans to fit new requirements. Dependency information is recorded in a plan's justification structure. Nodes in this structure are incrementally added at decision points. These nodes contain slots of recorded plan data. One of the three main kinds of justifications used captures links among choices in the subgoaling structure. Dependency annotations from slots like, "precond-of" and "relevant-to" are saved along with a successful solution to the problem. The Prodigy/Analogy group is currently working with MITRE to develop ways to extract dependency and ordering information from ForMAT [Mulvehill 96] plans.

3.3 Domain Dependencies

Since the early STRIPS-style domain representations, there has been support for expressing dependencies between elements in the domain. This has traditionally been achieved by listing preconditions that must hold in order to utilise a plan operator. The supplying effect would then be provided by another operator or from the initial world state. Planning domain representations have matured over time to support more complex needs.

ADL (Action Description Language) was developed as a hybrid of the STRIPS representation and the situation calculus [Pednault 87]. Pednault recognised the need for a more expressive language that could address the dynamic nature of continuous processes and simultaneous actions. The result, ADL, has an advantage over other plan representations in that explicit dependencies between circumstances and an action's effects can be efficiently expressed. The dependency relationships are encoded into the domain itself between a plan operator and dynamic aspects of the domain, rather than only being expressed statically between two operators.

Research was also contributed to provide ways of expressing various types of dependencies between operators which would indicate more about the relationship. Using Nonlin and O-Plan's Task Formalism, condition types can be used to indicate how conditions and effects can depend on each other [Currie & Tate 91]. A supervised condition type indicates that the satisfaction of a condition should be sought within the scope of the existing actions with the expansion schema (i.e. no new action may be added). An example of this would be a "carpet laying" schema that would have a supervised condition of the floor being clean. The action of sweeping the floor would be included in the carpet layer's plan. An unsupervised condition type, however, explicitly models conditions that are outside the scope of a schema's responsibility. Given the "carpet laying" schema again, there may be an unsupervised condition of a floor being present. It is not the case that the "carpet laying" schema should also be responsible for adding the action of building the floor. This indicates that somewhere earlier in the plan a floor had to be constructed and the "carpet laying" schema "depends" on being ordered after it.

A review of these examples of dependency rationale has shown the value this knowledge adds to the planning process. Hayes' early work on the journey planner motivated the need to track the dependencies that were normally compiled away during plan generation. This research had a significant impact on the concepts developed in truth maintenance systems (TMS). Work in case-based planning amplified the importance of plan dependency structures when attempting to reuse work from a similiar problem. Richer ways of expressing domain dependencies, like TF condition types, were also cited. Research continues to refine the expression and recording of this knowledge.

4 Decision Rationale

As we have seen so far, causality and dependency relationships stem from a decision-making process. Together, people and machines decide how to solve tasks, represent a domain, execute a plan and so on. Recording the rationale of these decisions adds value to the planning process also 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
- helping to steer the decision-making process

This axis represents the area of plan rationale which has received the least amount of attention in the planning research literature. This section begins with a presentation of the "multi-agent" decisions that are required from a mature planning process today. We then present parallels with research in "design rationale" to suggest ways to possibly proceed in the future. Finally, we stress the importance this knowledge in "steering" decision-making in an organisational context.

4.1 Multi-agent Decisions

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" [Burstein & McDermott 94]. With a large number of people and systems working together to produce a solution, we 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 *et al.* 93]. 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 *et al.* 96]. Clearly this shows that a richer model of plans is necessary to convey key pieces of knowledge needed to make planning decisions when human beings are involved.

Ferguson and Allen constructed a formal model of plans based on defeasible argument systems to support "mixed-initiative" planning in their TRAINS project [Ferguson & Allen 94]. This model allows for "an explicit representation of plans as arguments that a certain course of action under certain explicit conditions will achieve certain explicit goals.". This "certainty" is achieved by developing defeasible arguments which are sets of argument steps that can play roles like: rebuts, conflicts, undercuts, etc. The overall argument then can be said to be defeated or undefeated. The incremental nature of argumentation lends itself to the construction of reasoners that have to understand the reasons of other agents and communicate with them.

Joint work between the TRAINS and O-Plan projects is described by Tate [Tate 97]. This collaboration will seek to blend the multi-modal user dialog capabilities of TRAINS with the flexible, modular O-Plan planning system. A richer interface is anticipated between these two agents to support this mixed-initiative planning environment.

As cited earlier, the Prodigy/Analogy [Veloso 96] research has also been moving toward a decision rationale perspective. The user of this system can exercise complete control over the planning decisions. This control allows manual direction of the developing plan and annotation of nodes with rationale. Involved human agents can attach "guidance" to the plan as justification for the plan structure.

Given this decision rationale of a plan, an organisation can inspect the issues that were contemplated, the alternative solutions generated for these issues, and the criteria that either supported or detracted from the alternatives. These same initiatives are present in the field of design as well. Thus, the process of design can be compared to the process of planning.

4.2 Design Rationale

Tate describes a plan as a specialised type of design [Tate 96c, Tate 96b]. He states that a design is a set of constraints on relationships between entities. A "plan" constricts the definition by specifying that the entities are agents, their purposes, and their behaviour. In fact the resultant plan is essentially a design that meets the requirements of a task within the constraints of the domain. So it is of interest to examine how researchers in design represent rationale. "Design rationale"(DR) has been developed to represent this type of knowledge.

Design rationale is essentially the explicit recording of the issues, alternatives and justifications that were relevant to elements in the design. A representation that includes design rationale can lead to a better understanding of the issues involved [Conklin & Yakemovic 91]. MacLean et al. list two major benefits from design rationale representation [MacLean *et al.* 91]:

- aid to reasoning
- aid to communication

Examples of design rationale implementations include: QOC [MacLean *et al.* 91], DRL [Lee 90], g-IBIS [Conklin & Begeman 88]. We use QOC as an example to draw comparisons between DR and planning. QOC can be characterised as a node-arc graph where the nodes are **Q**uestions, **O**ptions, and **C**riteria. The meaning of the arcs depends on which nodes are being linked. Questions are essentially issues considered in the design. Options are linked as possible alternatives to questions. Criteria are connected to an option either to support or detract from the evaluation of the option. These structures 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 95, Tate 96b] 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 a simple example, figure 4 contains a hypothetical decision rationale for one cycle of a planner working on the "sussman anomaly" problem. A cycle includes the selection of an issue to process, determining how to resolve it, and the application of the solution.

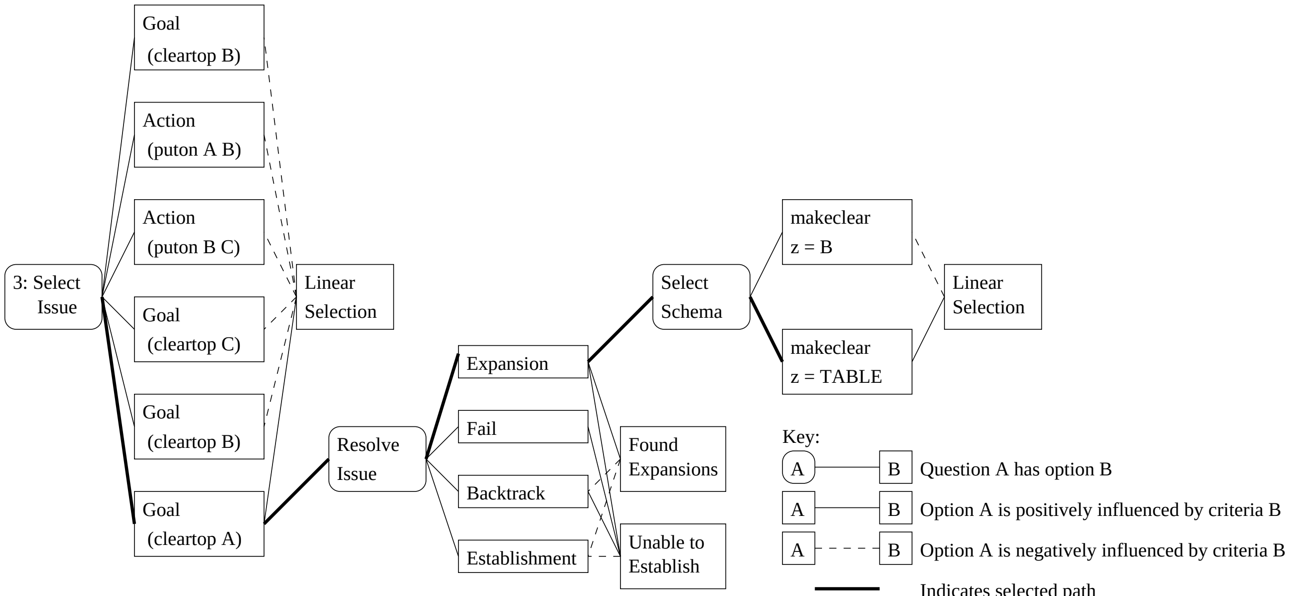

Figure 4: Decision rationale graph

During this third cycle, the planner is faced with 6 outstanding "issues" to address which can be thought of as options to the question of "Which issue should I work on next?". This simple planner's only criteria for this decision is a "linear selection" because it always processes these items in a FIFO manner. This is communicated as positive criteria for the top item and negative criteria for all of the other options. More sophisticated automated planners can treat this decision "opportunistically". This may involve the application of an evaluation function for the outstanding issues. Follow-on decisions are then linked to the selected option. In this case, the planner next considers how to resolve the selected issue (i.e. how to achieve cleartop A). The failed attempt to establish this goal in the existing plan network supports expansion, failure, or backtracking as an option. The fact that the planner found a way to expand this node results in expansion as the preferred course of action. This leads to a binding decision for the variable z (i.e. where to place the block sitting on A). This again is a simple linear selection for this example. This externalisation of the planning process is not something that is typically produced in most planners today, but is a design feature in O-Plan [Tate *et al.* 96].

If we were to attempt to utilise design rationale techniques in conveying planning decision rationale, it is possible that a richer DR implementation than that descibed above could be required. In DRL, Lee and Lai offer a much more semantically rich implementation. The concepts in QOC map well into DRL and DRL offers more as a representation language rather than a simple model. Each design rationale implementation offers some trade-off between [Lee & Lai 91]:

- expressiveness
- human usability
- computer usability

Arguments can be made for or against the way design rationale could be used in a particular planning context. The important factor is in determining this knowledge's applicability to a particular organisation. How to implement it could be centred around the three dimensions listed above. For instance, requirements may necessitate a highly expressive language and may be less concerned with high-end user support, but another case may stress a less detail-oriented approach that is simple, and easy to learn and use. Careful consideration is necessary to achieve a balance that will facilitate, rather than hamper the planning process.

Design rationale can justify an agent's position on facts or states. This permits an organisation to track the responsibility of factors that influence the decision-making component of the planning process. In the design process, designers need to interact with their designs. They have responsibilities for and act on their work. Plan constructors also require representations that will facilitate reasoning about and modifying plans in this fashion. Any agent may need to change the world and must take responsibility for its assertions.

4.3 Steering the Decision-Making Process

Assigning responsibility is an important first step toward steering the multiple inputs in planning decisions. There are also a number of other benefits that decision rationale could add to this process. They include:

- hierarchical management of details
- question answering
- focusing discussions and choice ordering.

Machine planners have a distinct advantage over their human counter-parts by being able to efficiently manage a great deal of low-level details. For the most part, human planners are better off working at higher levels of abstraction, allowing the computer to maintain any potential problems with detailed interactions. In fact, Tate describes,

> "...a mixed initiative form of interaction in which users and systems proceed by mutually constraining the plan using their own areas of strength" [Tate 94]

At the meeting point where human decisions and machine decisions interact, there must be an effective means to share rationale with each other. For example, a human planner may need to delve a little deeper into the details and may have to turn to the machine planner and ask it why a certain plan element is present.

"Why a certain plan element is present" is an example of question answering that needs to be supported in a decision-making process. There are also important, "what if" questions that could allow agents to test various ideas against existing plan structures [Drummond & Tate 92]. Other key questions that we see rationale addressing are the presence or absence of: variable bindings, orderings, and expansions.

These questions arise from the need to address specific issues. Representations with rationale provide an issue-based framework that focuses these questions. For example, Ballinger et. al. reports how factors may change that necessitate the consideration or reconsideration of various issues in the design of a chemical plant [Ballinger *et al.* 93]. They utilised an IBIS-type [Conklin & Begeman 88] structure to connect the new alternatives, or positions to the issue. The agents then participate in the generation of criteria that will lead to a choice.

Choice ordering is also of interest in decision rationale. The ordering indicates an implied value ranking of the potential choices. Some systems utilise a heuristic evaluation mechanism while others rely on sets of choice rules to assist in selection. For example, O-Plan [Currie & Tate 91], utilises heuristic functions that determine things like potential cost of expansion, etc. Early on, McDermott pointed out that the AI planning field must treat the process of choosing between alternatives as a basic situation of problem solving [McDermott 78]. He suggested that the choice process should have access to choice rules that assist in selecting among the alternatives. Clearly the mechanism that a system uses must be reflected in an explanation of why one element (and not another) is in a plan.

Many of these items discussed in this section have also been considered in requirements engineering (RE). Essentially, requirements and design specifications serve to constrain the possible space of a software system implementation in much the same way that a plan can be constrained. In engineering a large-scale system, a discrepancy between requirements and a system may cost in excess of 100 times the original cost if it isn't noticed early on [Roman 85]. RE researchers have looked to a rationale-based approach to drive down costs and improve quality. For example, the Remap system (representation and maintenance of process knowledge) extends an IBIS representation to track the rationale involved in functional specifications for a transaction-processing system [Ramesh & Dhar 94]. They report "an increase in quality and reduction in costly errors [that] outweigh the costs of capturing rationale". It is anticipated that research in RE systems, models [Moffett *et al.* 96] and methodologies [Scheffer *et al.* 85] will play a part in steering the "decision-making" process in our planning model.

5 Communicating Rationale

One of the keys to the value of plan rationale is the effective and efficient communication of this knowledge. This communication must support an entire mix of interaction between systems and human planners in the plan process. A mechanism must be present to record all three dimensions of rationale from the very beginning of domain creation. In fact, acquiring and maintaining domain knowledge is currently considered to be a highly significant bottleneck in planning systems [Wang 96]. Current tools that support domain creation disregard knowledge of the alternative considerations. A structured method of recording rationale, while producing a planning domain, would facilitate the acquisition and subsequent maintenance or reuse.

Other mechanisms will be necessary to support rationale as tasks are created and plans are being developed. For example, in a modular planning system this may involve various modules passing rationale along with a plan for further processing. Situated planning systems may need to communicate rationale with human or machine task assigners or plan executors.

In future work, we hope to layout a means to structure and represent this knowledge as an interlingua between the various planning agents and modules. The most reasonable approach would be to consider an extension of an existing plan ontology. Current plan ontologies focus on the interchange of plan knowledge [Tate 96c] but have not yet fully addressed the communication of some types of rationale, especially decision rationale. Ontological elements that would support planning decisions could incorporate ideas from design rationale or argumentation systems. In this way, agents in the planning process can exchange more than the plan artifact, but also the rationale behind it as well.

Overall, planning decision rationale is still in its infancy as we move toward a more integrated approach to planning. Current systems are looking for models to support the requirements that this approach places on the planning process. This review draws comparisons to solutions proposed from the design rationale community. It is possible that a modified form of DR could go a long way toward effective and efficient communication of planning rationale. Careful consideration needs to be made in how to effectively and efficiently present this knowledge on a case by case basis.

6 Conclusions

Rationale has been a key component in the planning process and will continue to to increase in importance. It has been used to improve the way a planner reasons about a plan and manages the details of plan element relationships. The three dimensions reviewed are strongly interrelated and highlight a multidimensional contribution. Some aspects, especially causal and dependency rationale, can be traced back to early work in planning. Other aspects have yet to be implemented. Planning decision rationale is beginning to gain more attention as deeper levels of organisational integration are required. Plan agents are expected to mature and evolve and they will surely need to turn toward rationale to support a mixed-initiative exchange of information.

Given this categorisation of rationale, a methodology and representation are required to record and communicate this knowledge. Tools need to be enhanced or developed to capture and present this information in an effective manner. These approaches should include rationale that has been presented both in the domain and during generative planning stages.

Work in design rationale suggests one method for tackling the new forays into decision rationale. Research into applying DR techniques to planning could help define new ontological elements required to represent plan knowledge. These elements could then be added as extensions to core plan representations. We hope to address some of these open issues as we continue to move toward rich plan representations.

Acknowledgements

The O-Plan project is sponsored by the Defense Advanced Research Projects Agency (DARPA) and Rome Laboratory, Air Force Materiel Command, USAF, under grant number F30602-95-1-0022. The O-Plan project is monitored by Dr. Northrup Fowler III at the USAF Rome Laboratory. One 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. 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, Rome Laboratory, AFOSR or the U.S. Government.

Thanks to Dave Robertson for reviewing the paper and for comments and suggestions.

References

[Allen 84] J.F. Allen. Towards a general theory of action and time. In Allen et al. [Allen *et al.* 90], pages 464–479.

[Allen *et al.* 90] J.F. Allen, J. Hendler, and A. Tate, editors. *Readings in Planning.* Morgan Kaufmann, Palo Alto, CA, 1990.

[Allen *et al.* 96] J.F. Allen, G.M. Ferguson, and L.K. Schubert. Planning in complex worlds via mixed-initiative interaction. In Tate [Tate 96a], pages 53–60.

[Ballinger *et al.* 93] G. Ballinger, R. Bañares-Alcántara, and J. King. Using an IBIS to record design rationale. Department of Chemical Engineering ECOSSE Technical Report 1993-17, University of Edinburgh, Edinburgh, Scotland, 1993.

[Barrett & Weld 94] A. Barrett and D. Weld. Partial order planning : evaluating possible efficiency gains. *Artificial Intelligence*, 67(1):71–112, 1994.

[Barros *et al.* 96] L. Barros, A. Valente, and R. Benjamins. Modeling planning tasks. In Drabble [Drabble 96], pages 11–18.

[Burstein & McDermott 94] M.H. Burstein and D. McDermott. Mixed-initiative military planning: Directions for future research and development. In Burstein [Burstein 94], pages 467–483.

[Burstein 94] M.H. Burstein, editor. *Proceedings of the ARPA/Rome Laboratory Knowledge-based planning and Scheduling Initiative Workshop*, Tuscon, AZ, 1994. Morgan Kaufmann.

[Conklin & Begeman 88] E.J. Conklin and M. L. Begeman. gIBIS: A hypertext tool for explanatory policy discussion. *ACM Transactions on Office Information Systems*, 6:303–331, 1988.

[Conklin & Yakemovic 91] E.J. Conklin and KC.B. Yakemovic. A process-oriented approach to design rationale. *Human Computer Interaction*, 6:357–391, 1991.

[Currie & Tate 91] K. Currie and A. Tate. O-Plan: the open planning architecture. *Artificial Intelligence*, 52:49–86, 1991.

[Daniel 83] L. Daniel. Planning and operations research. In *Artificial Intelligence: Tools, Techniques and Applications.* Harper and Row, New York, 1983.

[Dean *et al.* 93] T. Dean, L.P. Kaelbling, J. Kirman, and A. Nicholson. Planning with deadlines in stochastic domains. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 574–579, 1993.

[deKleer 86] J. de Kleer. An assumption-based TMS. *Artificial Intelligence*, 28:127–162, 1986.

[Doyle 79] J. Doyle. A truth maintenance system. *Artificial Intelligence*, 12(2):231–272, 1979.

[Doyle 94] J. Doyle. A reasoning economy for planning and replanning. In Burstein [Burstein 94], pages 35–43.

[Doyle 96] J. Doyle. Toward rational planning and replanning rational reason maintenance, reasoning economies, and qualitative preferences. In Tate [Tate 96a], pages 130–135.

[Drabble 96] B. Drabble, editor. *Proceedings of the Third International Conference on Artificial Intelligence Planning Systems (AIPS-96)*, Edinburgh, Scotland, May 1996. Morgan Kaufmann.

[Drummond & Tate 92] M.E. Drummond and A. Tate. PLANIT interactive planner's assistant – rationale and future directions. AIAI AIAI-TR-108, University of Edinburgh, 1992.

[Ferguson & Allen 94] G. Ferguson and J. Allen. Arguing about plans: Plan representation and reasoning for mixed-initiative planning. In Burstein [Burstein 94], pages 123–131.

[Fikes & Nilsson 71] R.E. Fikes and N.J. Nilsson. Strips: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2:189–208, 1971.

[Georgeff & Lansky 87] M. Georgeff and A. Lansky, editors. *Reasoning About Plans.* Morgan Kaufmann, Palo Alto, CA, 1987.

[Georgeff 87] M. Georgeff. Actions, processes, and causality. In Georgeff and Lansky [Georgeff & Lansky 87], pages 99–122.

[Goldman & Boddy 94] R. Goldman and M. Boddy. Epsilon–safe planning. In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, pages 253–261, 1994.

[Gross *et al.* 93] D. Gross, J. Allen, and R.D. Traum. The TRAINS-91 dialogues. Department of Computer Science TRAINS Technical Note 92-1, University of Rochester, Rochester, NY, 1993.

[Hayes 75] P.J. Hayes. A representation for robot plans. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-75)*, pages 181–188, 1975.

[Kambhampati & Hendler 92] S. Kambhampati and J. Hendler. A validation-structure-based theory of plan modification and reuse. *Artificial Intelligence*, 55:193–258, 1992.

[Kambhampati 89] S. Kambhampati. *Flexible Reuse and Modification in Hierarchical Planning: A Validation Structure-based approach.* Department of computer science, University of Maryland, College Park, 1989.

[Kambhampati *et al.* 95] S. Kambhampati, C. Knoblock, and Yang Q. Planning as refinement search: a unified framework for evaluating design tradeoffs in partial–order planning. *Artificial Intelligence*, 76, 1995.

[Kushmerick *et al.* 95] N. Kushmerick, S. Hanks, and D. Weld. An algorithm for probabilistic planning. *Artificial Intelligence*, 76:239–286, 1995.

[Lansky 87] A.L. Lansky. A representation of parallel activity based on events, structure, and causality. In Georgeff and Lansky [Georgeff & Lansky 87], pages 123–159.

[Lee & Lai 91] J. Lee and K. Lai. What's in design rationale? *Human Computer Interaction*, 6:251–280, 1991.

[Lee 90] J. Lee. Sibyl: A qualitative decision management system. In P.H. Winston and S. Shellard, editors, *Artificial Intelligence at MIT: Expanding Frontiers*, pages 104–133. MIT Press, Cambridge, MA, 1990.

[London 78] P.E. London. Dependency networks as a representation for modelling in general problem solvers. Department of Computer Science TR-698, University of Maryland, 1978.

[Ludlow & Alguire 94] C.O. Ludlow and K.M. Alguire. Looking at O-Plan2 and SIPE-2 through missionaries and cannibals. In Burstein [Burstein 94], pages 453–464.

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

[McAllester & Rosenblitt 91] D. McAllester and D. Rosenblitt. Systematic nonlinear planning. In *Proceedings of the AAAI-91*, pages 634–639, Anaheim, CA, 1991.

[McCarthy & Hayes 69] D. McCarthy and P. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer and D. Michie, editors, *Machine Intelligence 4*, Edinburgh, UK, 1969. Edinburgh University Press.

[McDermott 78] D. McDermott. Planning and acting. In Allen et al. [Allen *et al.* 90], pages 225–244.

[McDermott 82] D. McDermott. A temporal logic for reasoning about processes and plans. *Cognitive Science*, 6:101–155, 1982.

[Moffett *et al.* 96] J. Moffett, J. Hall, A. Coombes, and J. McDermid. A model for a causal logic for requirements engineering. *Requirements Engineering*, 1(1):27–46, 1996.

[Mulvehill 96] A.M. Mulvehill. Building, remembering, and revising force deployment plans. In Tate [Tate 96a], pages 201–205.

[Newell & Simon 63] A. Newell and H.A. Simon. GPS, a program that simulates human thought. In Allen et al. [Allen *et al.* 90], pages 59–66.

[Pednault 87] E. Pednault. Formulating mulitagent, dynamic-world problems in the classical planning framework. In Allen et al. [Allen *et al.* 90], pages 675–710.

[Peot & Smith 96] M.A. Peot and D.E. Smith. Conditional nonlinear planning. In Tate [Tate 96a], pages 210–217.

[Pólya 45] G. Pólya. *How to Solve It.* Princeton University Press, Princeton, N.J., 1945.

[Ramesh & Dhar 94] B. Ramesh and V. Dhar. Representing and maintaining process knowledge for large-scale systems development. *IEEE Expert*, pages 54–59, April 1994.

[Reece & Tate 94] G.A. Reece and A. Tate. Synthesizing protection monitors from causal structure. In *Proceedings of the Second International Conference on AI Planning Systems*, pages 146–151, Chicago, IL, June 1994.

[Roman 85] G. Roman. A taxonomy of current issues in requirements engineering. *IEEE Expert*, pages 14–21, April 1985.

[Sacerdoti 75] E.D. Sacerdoti. The nonlinear nature of plans. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-75)*, pages 206–214, 1975.

[Scheffer *et al.* 85] P.A. Scheffer, A.H. Stone, and W.E. Rzepka. A case study of SREM. *IEEE Computer*, pages 47–54, April 1985.

[Stallman & Sussman 77] R.M. Stallman and G.J. Sussman. Forward reasoning and dependency-directed backtracking in a system for computer-aided circuit analysis. *Artificial Intelligence*, 9(2):135–196, October 1977.

[Sussman 74] G.J. Sussman. The virtuous nature of bugs. In *Proceedings of the AISB Summer Conference*, July 1974.

[Tate 75] A. Tate. Interacting goals and their use. In *Proceedings of the International Joint Conference on Artificial Intelligence(IJCAI-75)*, Tbilisi, USSR, 1975.

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

[Tate 93] A. Tate. Putting knowledge-rich plan representations to use. In *Papers of the 14th Machine Intelligence Workshop*, Tokyo, Japan, November 1993.

[Tate 94] A. Tate. Mixed initiative planning in O-Plan2. In Burstein [Burstein 94], pages 512–516.

[Tate 95] A. Tate. Characterising plans as a set of constraints – the < I-N-OVA > model – a framework for comparitive analysis. *ACM Sigart Bulletin*, 6(1), January 1995.

[Tate 96a] A. Tate, editor. *Advanced Planning Technology: Technological Advancements of the ARPA/Rome Laboratory Planning Initiative*, Menlo Park, CA, 1996. AAAI Press.

[Tate 96b] A. Tate. Representing plans as a set of constraints – the < I-N-OVA > model. In Drabble [Drabble 96], pages 221–228.

[Tate 96c] A. Tate. Towards a plan ontology. *AI*IA Notiziqe (Quarterly Publication of the Associazione Italiana per l'Intelligenza Artificiale), Special Issue on Aspects of Planning Research*, 9(1):19–26, March 1996.

[Tate 97] A. Tate. Mixed initiative interaction in o-plan. In *Proceedings of the AAAI Spring Symposium on Computational Models for Mixed Initiative Interaction*, Stanford, California, March 1997.

[Tate *et al.* 96] A. Tate, B. Drabble, and J. Dalton. O-plan: a knowledge-based planner and its application to logistics. In Tate [Tate 96a], pages 259–266.

[Valente 95] A. Valente. Knowledge-level analysis of planning systems. SIGART *Bulletin*, 6(1), 1995.

[Veloso 96] M.M. Veloso. Toward mixed-initiative rationale-supported planning. In Tate [Tate 96a], pages 277–282.

[Wang 96] X. Wang. Planning while learning operators. In Drabble [Drabble 96], pages 229–236.

[Wilkins 84] D. Wilkins. Domain-independent planning: Representation and plan generation. *Artificial Intelligence*, 22:269–301, 1984.

[Wilkins 88] D. Wilkins. *Practical Planning: Extending the Classical AI Planning Paradigm*. Morgan Kaufmann, 1988.