On the Relation of Classical and Temporal Planning

Alexander Horz (horz@gmd.de)

German National Research Center for Computer Science (GMD)
PO Box 1316, W-5205 Sankt Augustin 1 (Germany)

Abstract

This paper explores a research strategy for a uniform and rational reconstruction of AI planning techniques. The strategy relies on two assumptions: (1) classical planners like STRIPS or SNLP are restricted variants of temporal planners like DEVISER, and (2) temporal planners may be best constructed atop a time map manager (TMM). The strategy aims at a reconstruction of timeless, classical as well as temporal systems in a TMM-based architectural framework. However, this paper shows that assumed restricted variants of DEVISER cannot be adequately recast in the TMM framework: this result is shown to hold for classical nonlinear planners like SNLP, and one reasonable extension by possibly simultaneous actions. Hence, in accordance with recent complexity results, this paper calls the intutively appealing research strategy into question.

1 Motivation

Classical planning in the tradition of STRIPS [7] is concerned with the construction of task sequences that instantaneously transform some initial state of the world into a state satisfying a certain goal condition. Even in nonlinear planning [18, 5, 9], a partially ordered set of tasks is considered only a compact representation of the set of its linearizations. DEVISER [19], on the other hand, combines partially ordered plans and PERT networks in order to extend the classical framework by quantitative temporal annotations and concurrent actions. Temporal planners seem to naturally extend the classical framework [2, 17]. Hence, we may assume a spectrum of planning systems with growing expressiveness from STRIPS-style to DEVISER-style.

The rational reconstruction of planners described in the literature is an important objective of current research. Chapman [5], for instance, reports about severe problems in reimplementing the "well known", classical planner NOAH. In this paper we investigate a research strategy for a uniform reconstruction of different points in the assumed spectrum of growing expressivity. We therefore additionally assume that general, temporal planners are appropriately constructed atop a TMM, according to Dean's original motivation for his research on time map management [6]: temporal projection is considered to be fundamental to the planning task[1], and a divide-and-conquer approach where temporal reasoning and the plan construction process are separated is much easier to understand and to modify.

We are interested in planners which may be located in the middle-ground between classical nonlinear planners like NOAH [18] or SNLP [13], and more general, temporal systems like DEVISER. These intermediate systems may be characterized by the following features: (1) any action has *unit duration*, and (2) the *extended interpretation* of nonlinear plans considers unordered tasks to be executable in any arbitrary order as well as simultaneously, subject to some simultaneity criterion. Nonlinear plans have been originally considered a compact representation of a set of sequences. Under the extended interpretation, however, we may also reduce plan execution time, and obtain even more solutions. For the purpose of this paper, extended planners simply represent additional points in the assumed spectrum.

When pursuing the research strategy assuming that any point in the spectrum is a restricted variant of DEVISER-style systems, and that general temporal planners may be adequately constructed on top of a TMM, two unexpected problems arise: equivalent recasting of neither (1) classical nonlinear planners nor (2) one reasonable extension by simultaneous actions is possible within the TMM framework, when keeping the idea of clearly separating temporal reasoning and plan construction. These problems are caused by the facts that (1) a conflict-free time map does not neccessarily represent a correct nonlinear plan, and (2) there is (at least) one reasonable but not recastable alternative to TMM's built-in criterion for simultaneous executability of actions.

The paper has the following structure: After introducing basic definitions in section 2, two reasonable approaches to extending the planning framework by simultaneous actions are described in section 3. The TMM-based construction of the temporal, nonlinear planner

[1] Recent complexity considerations [3] show that general temporal projection is hardly a neccessary ingredient of the planning task.

TRIPTIC [17] is sketched in section 4. Section 5 shows that the TMM-framework may adequately recast one extension described in section 3 but neither classical planners nor the alternative extension. Section 6 concludes.

2 Basic Definitions

Let A be the set of *atomic* actions, each of which is a partial function from W to W, where W is the set of possible world states. For any state $w \in W$ and any formula ϕ let $w \models \phi$ denote that w is a model of ϕ, i.e. ϕ is true in w. World states are represented by subsets of $\mathcal{E}$, the set of *essential* [11] sentences. A set $S \subseteq \mathcal{E}$ is *maximally legal* iff the set $W_S = \{w \in W | \forall \phi \in S : w \models \phi\}$ of world states represented by S is a singleton.

Classical planning in the tradition of STRIPS [7] describes actions by triples (P_o, D_o, A_o), the precondition, delete set and add set of the *operator* o which is interpreted as the (atomic) action f_o. Since it is sufficient to show our main result for a specific type of classical planning tasks, we further consider a specific type of operators, only: we restrict P_o to a set (conjunction) of essentials, enforcing that $S \cup K \vdash P_o \Leftrightarrow S \supseteq P_o$, where K is the (unused) background knowledge about (maximally) legal sets of essentials. Operator o is *legal* iff $\mathtt{result}(o, S)$ is maximally legal, for any maximally legal $S \supseteq P_o$, and $D_o \cap A_o = \emptyset$.

Soundnes of operators [11] is not sufficient for our purpose: if (P_o, D_o, A_o) is sound with respect to f_o, then so are, e.g., $(P_o, D_o, \emptyset)$ or $(P_o, \mathcal{E}, A_o)$. Sound operators may happen to predict the state of affairs resulting from action execution very weakly while legal operators are maximally predictive with respect to essential sentences.

Conflict management of classical nonlinear plans is a central issue for this paper. Due to the lack of space, the reader is referred to the formal model in [9]; the simple example in figure 1 may serve as an illustration. The plan contains four *dependencies*: One between **start** and **t1**, and one between **start** and **t2**, both concerning condition **p**. The third dependency is between **t1** and **goal** concerning **q**, the fourth between **t2** and **goal** concerning **r**. Furthermore, the plan contains two *conflicts*: task **t1** denies the precondition **p** of task **t2**, may be executed before **t2**, and violates the dependency between **start** and **t2** concerning **p** (and vice versa). Hence, the plan is no solution under the classical interpretation (see [9] for further details).

3 Simultaneous Actions in the Classical Framework

Classical planning considers actions as sequentially and instantaneously transforming the world. Even when ignoring durations of actions, planners which take possible

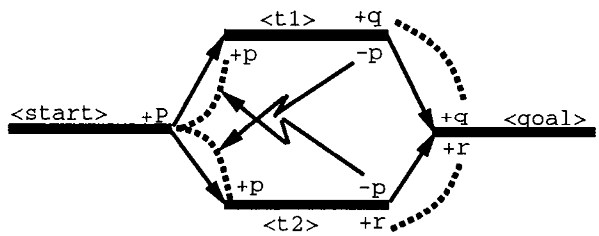

Figure 1: A simple, nonlinear plan; dashed curves show dependencies

simultaneity into account may produce novel solutions, and help reduce plan execution time.

A first design of such planners would require cumbersome domain models including explicit descriptions of any possible *complex* action formed by simultaneous execution of two or more atomic actions. As the major limitation, the domain modelling effort would, in the worst case, grow unacceptably from linear to exponential in the original number of atomic actions. Pelavin's [15, pp. 176ff] approach, which requires explicit modelling of noninterference conditions between atomic actions, as well as Große and Waldinger's [8] "possible" predicate also suffer from combinatorial explosion.

In essence, we are looking for a rule predicting the applicablity and effects of complex actions, given only the descriptions of the respective atomic constituents. The rule for description composition is required to preserve soundnes and legality, i.e. we require the rule's result to be a sound and legal description of the complex action.

The complex action resulting from simultaneous execution of a and b, again a partial function from W to W, is denoted $a||b$; sequential execution is denoted $a \circ b$. Viewing atomic actions as functions gives us no hint about the result of simultaneous execution. However, we may reasonably identify simultaneous executability with (1) independence or (2) coherence of actions. These criteria are not equivalent, as described below.

3.1 Independence

One reasonable criterion for simultaneous executability may be easily understood by applying the extended interpretation to classical nonlinear solutions:

Definition 1 (Independence Assumption) *The* independence assumption *holds if and only if*

- *any two actions a and b are simultaneously* **executable** *in world state w if and only if sequences $a \circ b$ and $b \circ a$ are executable in w, and*
- *simultaneous execution yields a* **result** *identical to execution in any arbitrary order.*

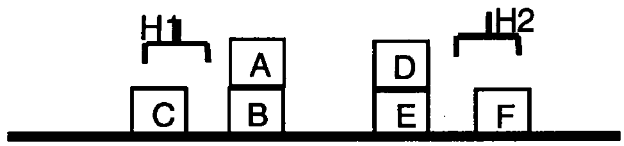

Figure 2: A situation in the two-handed blocks world.

Obviuosly, descripiton composition according to the independence assumption preserves soundnes and legality, as required. For an example of independent actions, consider a slightly extended version of the blocks world where two robot hands are available. The actions respectively corresponding to `move(H1,A,B,C)` and `move(H2,D,E,F)` are independent according to our definition, and they are simultaneously executable in the situation depicted by figure 2. However, the actions corresponding to `move(H1,A,B,F)` and `move(H2,D,E,F)` are neither independent nor simultaneously executable.

3.2 Coherence

Although the independence assumption is reasonable, there are temporal planners (e.g. DEVISER [19]) which implicitely employ a different criterion.

For any action descriptions $o1 = (P_{o1}, D_{o1}, A_{o1})$ and $o2 = (P_{o2}, D_{o2}, A_{o2})$ let $o1|o2$ denote $(P_{o1} \cup P_{o2}, D_{o1} \cup D_{o2}, A_{o1} \cup A_{o2})$. Actions f_{o1} and f_{o2} are *coherent in world state* w if and only if $\forall \phi \in P_{o1} \cup P_{o2} : w \models \phi$, and $o1|o2$ is legal.

Definition 2 (Coherence Assumption) *The* coherence assumption *holds if and only if*

- *Any two actions f_{o1} and f_{o2} are simultaneously executable in world state w if and only if f_{o1} and f_{o2} are coherent in w, and*
- *$o1|o2$ is a sound and legal description of $f_{o1}||f_{o2}$, for any sound and legal descriptions $o1$ and $o2$ of f_{o1} and f_{o2}, respectively.*

Recall the example: as required, `move(H1,A,B,C)` and `move(H2,D,E,F)` are coherent, while `move(H1,A,B,F)` and `move(H2,D,E,F)` aren't. Obviously, coherence is necessary but insufficient for independence. Tasks `t1` and `t2` (cf. figure 1) form a simple example.

Note that neither independence nor coherence are generally equivalent to simultaneous executability. However, they are assumed to be in [16, 14] and [19, 6, 4], respectively. Thus, the respective assumption built into a specific planner must be explicitely stated, since the domain modeller has to take it into account. A more detailed discussion about independence respectively coherence may be found in [10].

3.3 Classical Planning with Simultaneity

Assuming independence, a nonlinear plan of sound operators solves a problem under the classical interpretation if and only if it solves the problem under the extended interpretation where unordered, independent tasks may be simultaneously executed. Hence, when assuming independence, we may simply employ an unmodified, classical nonlinear planner.

However, when executing the plan, we still have to check independence: In a correct nonlinear plan, two tasks may remain unordered if there is no conflict between either of the tasks and a dependency concerning a precondition literal or a produced effect of the other task. Hence, neither of two respectively unordered tasks may deny the other task's preconditions or the other task's effects that are used later in the plan. Since some effects may remain unused, unorderedness is not a sufficient condition for independence.

Let us now consider the impacts of the coherence assumption on the design of an extended nonlinear planner. Provided the coherence assumption, we may only state that any classical nonlinear solution is also a solution under the extended interpretation. The plan in figure 1 is a counterexample for the reverse direction: Under the classical interpretation, it is impossible to resolve the double-cross conflict formed by two right-forks without introduction of a new task. When assuming coherence of `t1` and `t2`, however, we may resolve both conflicts by merging `t1` and `t2` into `t1|t2`.

In general, a new conflict resolution rule may be introduced: Not only may we prevent the conflictor `D` from being executed between the producer `P` and the user `U`, or introduce a different *c*-producing task `P'` between `D` and `U` (cf. [9]). Provided coherence of the actions corresponding to `U` and `D`, we may also resolve the conflict by merging the user `U` and the conflictor `D` into a legal task `U|D`.

Consequently, planners assuming coherence may be considered a little more general than those assuming independence: they may produce more solutions.

4 TRIPTIC: A Temporal, Nonlinear Planner

This section explores the second assumption underlying the research stratgey under consideration. It describes TRIPTIC, a temporal nonlinear planner [17] which is implemented in Common Lisp on top of a corrected and extended reconstruction [12] of Dean's TMM. The lack of space requires concentration on aspects relevant for the purpose of this paper.

Assuming that an adequate design of temporal planners reflects the division of labor mentioned in section 1, TRIPTIC is designed according to the following princi-

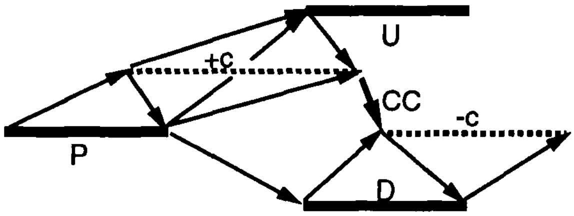

Figure 3: Solving a right-fork; bold and dashed lines show task resp. condition tokens.

ples: The TMM handles all temporal relations, in particular it is responsible for the detection of conflicts and possible helpful interactions. The planner has to explicitely establish dependencies between tasks, especially their organization towards the goal.

Conflict management in TRIPTIC is of further interest, here. Technically, the TMM detects conflicts by detecting possibly intersecting, contradictory token pairs. Conflicts may be resolved by introduction of clipping constraints. When there is only one allowable clipping, this is automatically done by the TMM. Otherwise, the planner has to eventually select one alternative clipping.

Tasks are represented by task tokens. Figure 3 shows the TMM-representation of the (solved) right-fork conflict [9]. Bold lines show task tokens, dashed lines show condition tokens, and (clipping) constraints are represented by (bold) arrows. The precondition of tasks are represented by condition tokens which are constrained to be true at the start of the respective task; the effects are constrained to be true at the task's end and to start after the task starts. Note that here and in the sequel, "after" may alternatively be read "after or exactly when", since (clipping) constraints may be closed intervals of length 0; the same holds for "before." Dependencies are established by "stretching persistences" [6]. This implies a truth criterion ignoring "white knights" [5].

Consider the example in figure 3, where the dependency between `P` and `N` concerning `c` is conflicted by `D`. This "right-fork" would be detected and resolved by the TMM, because token `+c` may intersect token `-c`. Since `+c` definitely starts before `-c`, the unique allowable clipping `CC` constraining `+c` to end before `-c` starts will be automaically introduced. Note that, as a "side-effect", `CC` constrains `N` to start before `D` ends. We may alternatively mimic DEVISER's behavior by additionally constraining a task's precondition to persist at least until the task ends, and constraining the effects to start exactly when the task ends. In this case, `CC` would additionally constrain `N` to end before `D` ends.

Summarizig in short, the TMM-based framework supports a conceptually clear design of temporal planners. An adequate task representation causing no problems in the calculation of interval intersections is still subject to future research: neither a mixture of open and closed interval boundaries nor introduction of `*pos-tiny*` and `*neg-tiny*` seem to be generally sufficient. However, we won't go into further details, here.

5 Dubious Intution

In section 1, we sketched a research strategy for a unifying view of planning systems. It assumes the spectrum of growing expressivenes, which has been explored in section 3, as well as the appropriateness of the TMM-framework, which has been confirmed in section 4. Surprisingly, the intuitively appealing assumptions must be called into question due to the following reason: TMM-based temporal planners do not generalize classical nonlinear planners and extended classical planners assuming independence.

The impossibility of recasting classical nonlinear planners may be explained at the plan presented in figure 1: this plan is not conflict-free, hence no solution. The TMM, however, would constrain intervals `t1` and `t2` to be equal, since the "double-cross" conflict is a combination of two "right-fork" conflicts. TRIPTIC's built-in criterion for simultaneous executability of tasks is coherence: the resulting time map would not contain any pair of intersecting, contradictory tokens, hence would be considered a representation of a solution plan.

The only resort would be a means for enforcing mutual exclusion of different task intervals. However, this would not only contradict the essential idea of least commitment in nonlinear planning; it is not possible by partially ordering interval endpoints [1, p. 16]. Here, we could replace the TMM by a interval-based reasoner as it is done in ILP [1, p. 16]. However, this would be no solution for the problem of recasting extended planners assuming independence, where simultaneity should be possible.

Extended planners assuming independence cannot be recast in the TMM-framework, since independence is equivalent to the conditions that (1) preconditions are required to persist until the end of the task, (2) postconditions are already produced when the task starts, and (3) contradictory tokens "belonging" to the same task have to be ignored. Obviously, condition (3) contradicts the division of labor between the planner and the TMM.

Hence, at least one of the underlying assumptions must be invalid: temporal planners do not generalize (extended) classical planners, i.e. there is no spectrum of growing expressivenes, or temporal nonlinear planners may not be adequately designed in the TMM framework.

6 Conclusions

We have explored a research strategy for a unifying view of planning systems. The strategy assumes a spectrum of growing expressivenes from STRIPS-style to DEVISER-style systems, and that temporal planners may be adequately designed in a TMM-framework. However, we obtained the contradiction that less expressive systems cannot be recast by a TMM-based temporal planner, i.e. TMM-based temporal planners do not generalize (extended) nonlinear planners. In accordance with recent complexity results [3], this contradiction calls the intuitively appealing research strategy into question.

Future work, thus, has to investigate the following issues: The idea of a spectrum is still appealing, although it may not have the assumed shape. Although seeming questionable, the TMM-based construction has advantages over previous approaches. What should be kept from this idea?

7 Acknowledgements

This work has been partially supported by the German Ministry for Research and Technology (BMFT) under grant ITW 8900 A7. Thanks to Tom Gordon, Gerd Große, Joachim Hertzberg and Eric Rutten for valuable comments on earlier drafts.

References

[1] James Allen. Temporal Reasoning and Planning. In James Allen, Henry Kautz, Richard Pelavin, and Josh Teneneberg, editors, *Reasoning About Plans*, chapter 1. Morgan Kaufmann, San Mateo (CA), 1991.

[2] James Allen and Johannes Koomen. Planning Using a Temporal World Model. In *Proc. IJCAI-83*, pages 741–747, 1983.

[3] Christer Bäckstrom and Bernhard Nebel. On the Computational Complexity of Planning and Story Understanding. In *Proc. ECAI-92*, 1992.

[4] Priklis Belegrinos and Michael Georgeff. A Model of Events and Processes. In *Proc. IJCAI*, pages 506–511, 1991.

[5] David Chapman. Planning for Conjunctive Goals. *Artificial Intelligence*, 32:333–377, 1987.

[6] Thomas Dean and Drew McDermott. Temporal Database Management. *Artificial Intelligence*, 32:1–55, 1987.

[7] Richard E. Fikes and Nils J. Nilsson. STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. *Artificial Intelligence*, 2:189–208, 1971.

[8] Gerd Große and Richard Waldinger. Towards a Theory of Simultaneous Actions. In Joachim Hertzberg, editor, *Proc. 1st European Workshop on Planning (EWSP-91)*, pages 78–87, 1991.

[9] Joachim Hertzberg and Alexander Horz. Towards a Theory of Conflict Detection and Resolution in Nonlinear Plans. In *Proceedings of the 11th IJCAI*, pages 937–942, 1989.

[10] Alexander Horz. Relating Classical and Temporal Planning (Preliminary Report). In *Proc. 7th German Workshop on Planning and Configuration*, 1993.

[11] Vladimir Lifschitz. On the Semantics of STRIPS. In Michael P. Georgeff and Amy L. Lansky, editors, *Reasoning about Actions and Plans*, pages 1–9. Morgan Kaufmann, 1987.

[12] Stefan Materne and Joachim Hertzberg. MTMM - Correcting and Extending Time Map Management. In Joachim Hertzberg, editor, *Proc. 1st European Workshop on Planning (EWSP-91)*, pages 88–99, 1991.

[13] David McAllester and David Rosenblitt. Systematic Nonlinear Planning. In *Proc. AAAI-91*, pages 634–639, 1991.

[14] Edwin P. D. Pednault. Formulating Multi Agent Dynamic World Problems in the Classical Planning Framework. In Michael P. Georgeff and Amy L. Lansky, editors, *Reasoning about Actions and Plans.* Morgan Kaufmann, 1987.

[15] Richard Pelavin. Temporal Reasoning and Planning. In James Allen, Henry Kautz, Richard Pelavin, and Josh Teneneberg, editors, *Reasoning About Plans*, chapter 3. Morgan Kaufmann, San Mateo (CA), 1991.

[16] Pierre Regnier and Bernard Fade. Complete Determination of Parallel Actions and Temporal Optimization in Linear Plans of Actions. In Joachim Hertzberg, editor, *Proc. 1st European Workshop on Planning*, pages 100–111, 1991.

[17] Eric Rutten and Joachim Hertzberg. Temporal Planner = Nonlinear Planner + Time Map Manager. TASSO Report 40, Gesellschaft für Mathematik und Datenverarbeitung mbH (GMD), Sankt Augustin, 1992. (revised version will appear in *AI Communications, vol. 6, 1993*).

[18] Earl D. Sacerdoti. The Non-linear Nature of Plans. In *Proc. of the 4th IJCAI*, pages 206–214, 1975.

[19] Steven A. Vere. Planning in Time: Windows and Durations for Activities and Goals. *IEEE Trans. PAMI*, 5:246, 1983.