Appeared in: Proceedings of: AAAI Fall Symposium "PLANNING AND LEARNING: ON TO REAL APPLICATIONS", AAAI Press, 1994.

Planning for Machining Workpieces with a Partial-Order, Nonlinear Planner*

Juergen Paulokat and **Stefan Wess**
University of Kaiserslautern, Dept. of Comp. Science
P.O. Box 3049, D-67653 Kaiserslautern, Germany
E-mail: paulokat@informatik.uni-kl.de

Abstract

We describe a hybrid architecture supporting planning for machining workpieces. The architecture is built around CAPLAN, a partial-order nonlinear planner that represents the plan already generated and allows external control decision made by special purpose programs or by the user. To make planning more efficient, the domain is hierarchically modelled. Based on this hierarchical representation, a case-based control component has been realized that allows incremental acquisition of control knowledge by storing solved problems and reusing them in similar situations.

Introduction

Planning for machining workpieces is a crucial step in the process chain of product development in mechanical engineering because it strongly influences the overall product's costs. Its task is selecting and ordering machining operations such that all features of a given workpiece can be manufactured by minimal or, at least, by low costs. Seen Abstractly, every machining operation is composed of several preparatory steps, as transporting, feeding, fixing or clamping the workpiece and changing tools, as well as the processing step itself, e.g. drilling or cutting. However, the features of a workpiece cannot be treated independently, because steps of a manufacturing plan possibly interact. Positive interactions can be utilized to decrease the cost of manufacturing a feature, e.g. sharing preparatory steps for several processing steps, or compounding manufacturing of several features in one processing step. While positive interactions should be utilized to minimize overall production costs, negative interactions must be resolved because they lead to inconsistencies. For example a processing step for a feature must not be used if it destroys or makes it impossible to manufacture another one.

Theoretically, the concepts of partial-order nonlinear planning are well suited to support the generation of process plans. Negative interactions can be solved by detecting clobberers and declobbering them (Chapman 1987). The truth criterion together with the phantomization of goals is an effective way to utilize positive interactions. But, to solve more than trivial examples, it is necessary to use domain-specific reasoning, especially geometrical reasoning, that is crucial for this and most other technical domains. However, the encoding of domain-specific knowledge in a general purpose planner is either awkward or inefficient (c.f. (Kambhampati *et al.* 1991)).

Further, research in planning is mainly concerned with the search for consistent solutions (e.g. (Chapman 1987; McAllester & Rosenblitt 1991; Barrett & Weld 1993)) and the efficiency of planning (e.g. (Minton 1988; Bergmann 1992; Borrajo & Veloso 1994)), but until now no satisfying methods are available to support optimization of plan execution costs (Pérez & Carbonell 1993). This is in contrast to the large number and importance of heuristics and expertise in real world planning (e.g. (Marcus 1988)). Besides the lack of concepts for processing control knowledge, its acquisition is further complicated, because in most cases the only source is the expert himself who only can explicate and communicate it working on a concrete example (Firlej & Hellens 1991). Therefore, to be acceptable in such domains, a planning system must support incremental acquisition or learning of heuristics.

A main characteristic of human planning for machining is the use of examples that have found to be successful in similar situations. In mechanical engineering, there are numerous attempts to build up index structures to support the classification of workpieces and the retrieval of associated manufacturing plans. How-

---

*This research was partially sponsored by the Deutsche Forschungsgemeinschaft (DFG), SFB 314: "Künstliche Intelligenz - Wissensbasierte Systeme", Projekt X9. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the DFG.

ever, these index structures are intended for manual usage and only utilize information about the geometry, material of workpieces and the applied technology. They completely lack attempts to extract structural information from plans as, e.g. as in (Veloso 1992) to make retrieval more informed.

In this paper, we describe a prototypical system for supporting planning for machining, that is implemented using CAPlan, a nonlinear, partial-order general purpose planner. Because of the impossibility to acquire complete control knowledge and the lack of concepts for deep domain-specific reasoning, CAPlan is organized as a planning assistant that has a partial model of domain-specific control knowledge and that systematically searches if no control knowledge is available. To complete the represented domain knowledge, the user can revise all planning decisions at every moment and can interactively control plan generation. In this case, the task of the planner is to represent external control decisions and to compute their effects in the overall plan. To support incremental acquisition and usage of episodic knowledge, the planning process can be additionally controlled by replaying a case that is a machining plan interactively developed with the system and kept in a case base. Replay is made by CAPlan/CbC that utilizes the user interface of CAPlan to guide its search by the decisions represented in a case.

The Domain Characteristics

The domain we are concerned with is manufacturing planning for rotary-symmetrical workpieces to be machined on a lathe. A planning problem is given as a geometrical description of a workpiece and of the raw material that only can be cylindrical in our model (Fig. 1). The description of a workpiece is built up from geometrical primitives as cylinders, cones and toroids that describe monotone areas of the outline, possibly augmented by features as threads, grooves or special surface conditions. In most cases, the outline of a workpiece cannot be machined in one step, but repeated cutting operations are necessary to cut the difference between the raw material and the workpiece in thin horizontal or vertical layers. These layers are built up from atomic processing areas, that are automatically generated by extending the horizontal and vertical bounding lines of the geometrical primitives. Cutting an atomic area can be seen as an elementary cutting step.

For machining, the workpiece is clamped by a rotating clamping tool, while layers of material are removed by moving a cutting tool along its surface. Standard tools that are normally used for machining large areas have a fixed working direction, i.e., they only can be used to cut off material when being moved either to the left or to the right. Clamping a workpiece hides parts of its surface, therefore, after machining the part not hidden, it must be turned and clamped on the other side. Additonally, caused by the geometry of standard cutting tools only horizontal outlines or outlines that are rising along a tool's moving direction can be machined. To machine a descending outline requires tool changes which increase the overall production costs. But after turning a workpiece and clamping it from the ohter side, e.g. necessary for machining an area hidden by a clamping tool, a descending outline becomes a rising one and can be machined with the same tool. For finding maximally monotonously rising areas, geometrical primitives are grouped to rising areas at both ends of the workpiece and a horizontal area between them (Fig. 2). Each of these compound areas can contain subareas that break the monotonous course of the outline. But from an abstract point of

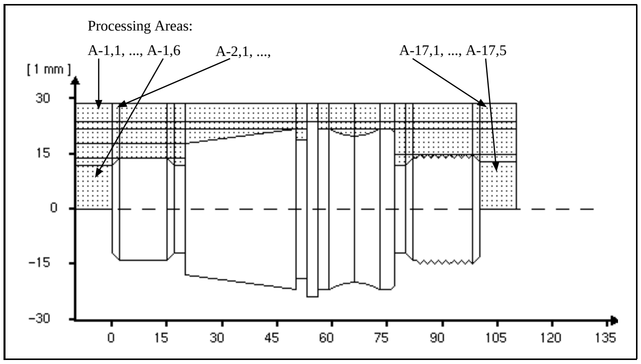

Figure 1: Geometrical representation of a workpiece and the cylindrical raw material.

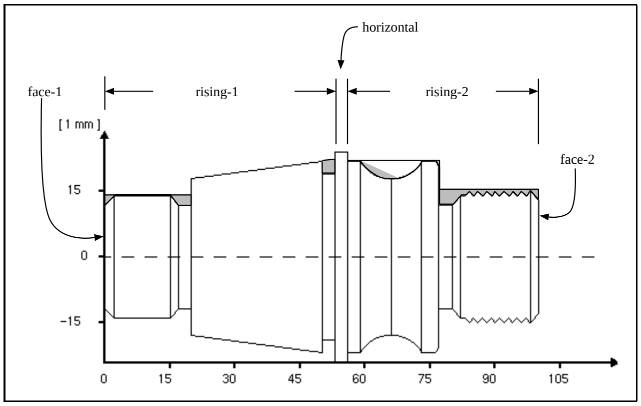

Figure 2: Geometrical representation of a manufacturing planning problem.

view they do not influence the construction because the monotonous outline necessarily has to be machined first before hidden subareas can be machined. For machining, the horizontal part can be added consistently to one of both rising areas, but choosing an alternative is one of the tasks of the planner because this choice can influence the plan's execution costs. Compound areas can be degenerated to consist of only one geometric primitive, e.g. the horizontal area in Fig. 2.

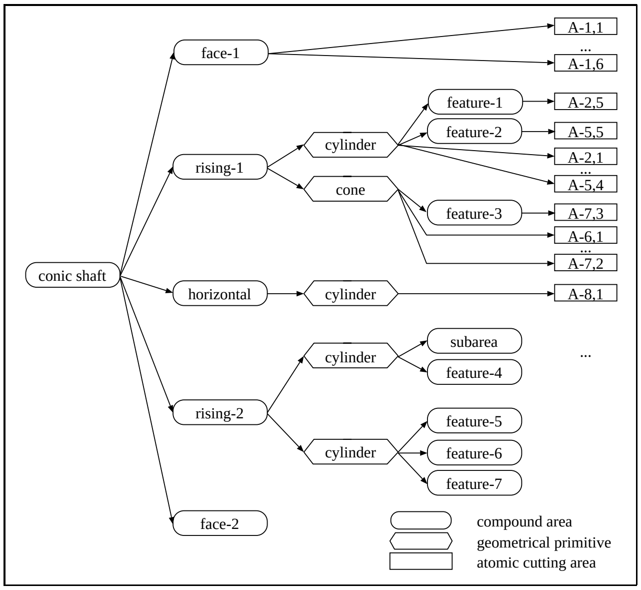

Figure 3: Hierarchical representation of a planning problem.

These areas can be seen as an abstract description of the workpiece resulting in a hierarchical representation whose root node represents the whole workpiece with the compound areas as successors (Fig. 3). Their successors are the geometrical primitives and the subareas of the workpiece. A geometrical primitive has successor nodes for its features and the atomic processing areas that are located above it. Subareas, again, can be hierarchically structured, defining their own subtrees. This hierarchical representation of the workpiece is the base for a hierarchically structured planning process.

The Planning Assistant CAPlan

Before we describe the model of the planning process for machining workpieces, we give a short summary of the CAPlan system.

CAPlan is a domain-independent nonlinear partial-order planning system implemented based on the SNLP algorithm (McAllester & Rosenblitt 1991; Barrett & Weld 1993). Although it can autonomously solve problems by depth-first search, it is mainly intended as a planning assistant that provides a control interface for a human planner or external control components, such as CAPlan/CbC that implements a replay mechanism for cases. So, every decision in a planning cycle can be made interactively. These decisions are choosing a goal to work on next, selecting an operator to reduce or satisfy a goal, adding constraints to resolve threats of causal links and retracting operators to resolve inconsistent plans by backtracking. Used as a planning assistant, CAPlan adds external decisions to its partial plan, computes their effects and signals resulting threats or inconsistencies.

In CAPlan, the concepts of redux (Petrie 1991) are used to support user interactions by minimal or conservative (Kambhampati *et al.* 1991) modifications of the current plan. The redux ontology is a general framework supporting decision making and the retraction of decisions. It supports the representation of rationals for the validity, admissibility and optimality of a decision as well as for its rejection. Every rejection describes planning situations in which the according decision is not allowed. The rationals are stored as justifications in a TMS network and are automatically supervised by TMS consistency techniques.

Conservative modification of a plan is supported by storing planning decisions in an explicit AND/OR search tree. This is in contrast to many other planning systems where they are kept in pure chronological order. So, the retraction of an early decision only forces the retraction of causally dependent decisions but does not influence any other decisions made chronologically later.

Planning for Machining Workpieces

The central idea behind our approach is that the planning system has a complete model of all steps in the domain and a partial collection of control heuristics. Control knowledge can be represented internally by control rules or externally by the user or a sophisticated program, e.g. a simulation component or a geometrical reasoner. Such an external control agent is provided with the set of alternative decisions of a decision point, the information of the workpiece model and of the current plan. It selects one alternative and optionally provides rationals for its decisions that are processed by the underlying redux system. So, we can integrate domain-specific reasoning without being forced to code the knowledge completely into the operators' description.

Planning for machining a workpiece is preceded by transforming the abstract level of the hierarchical description of the workpiece Fig. 3) in a problem repre-

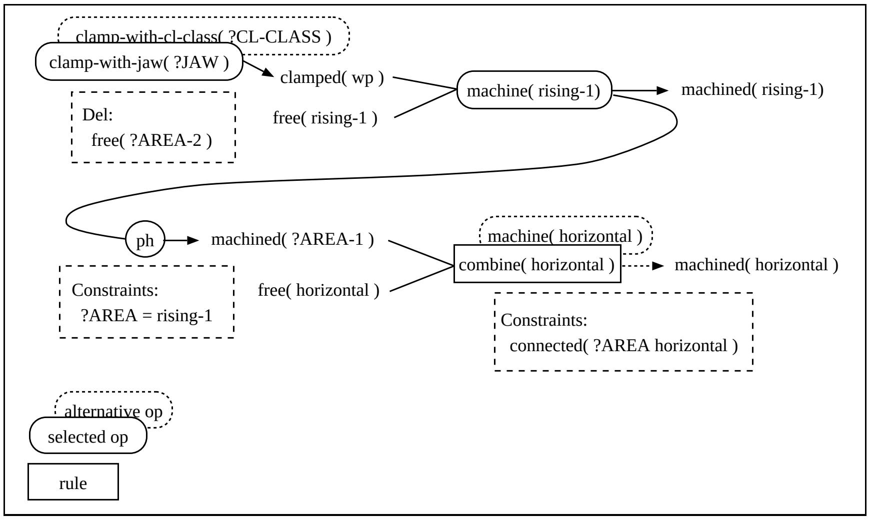

Figure 5: Partial plan on the abstract planning level.

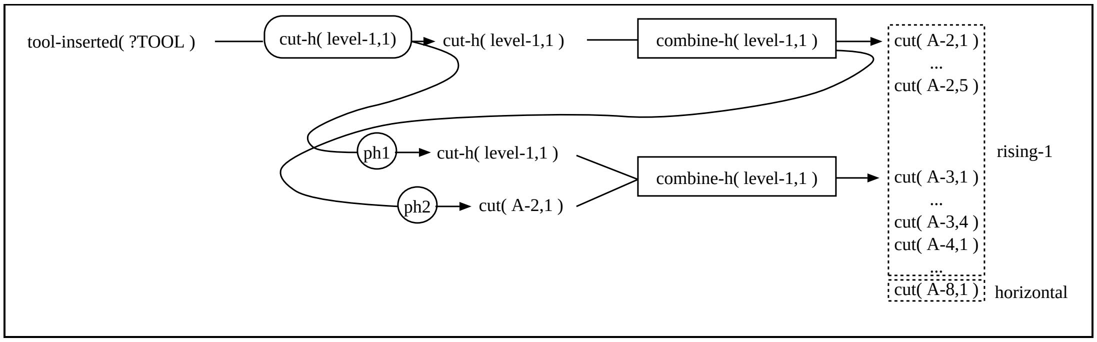

Figure 6: Partial plan on the concrete planning level for the abstract operator machine(rising-1) of Fig. 5.

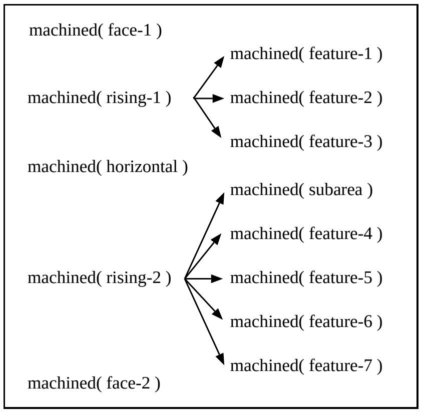

Figure 4: Set of initial goals and their ordering relations.

sentation based on propositional logic that is used by the planner. This analysis step extracts a set of planning goals as well as a set of geometrical and technological constraints, e.g. the technological requirement that a groove can only be machined if the outline which it is contained in, has been already machined. These constraints are represented by ordering relations between goals representing the planning problem (Fig. 4). In classical planning, orderings are only defined between steps. In this sense the orderings between goals are used to imply orderings between the subplans to reach these goals. The effect of these orderings is that the size of the search space can be significantly reduced without any further reasoning or representation effort, because processing of these constraints is completely done by the analysis step and by internal consistency mechanisms of the planner. Names, as rising-1 used in the problem description, provide a link to the geometrical representation. They can be used to access further information not explicitly represented in the problem description but which are necessary for control decisions.

Figure 5 shows a part of the plan on the abstract planning level of the hierarchical planning process. First, the planner decided to work on goal machined(rising-1). Therefore, the abstract operator machine(rising-1) is selected, which introduces subgoals to clamp the workpiece and to force the area rising-1 to be free. Then, planning continues with the goal machined(horizontal). Here, the planner can choose between introducing a new step to machine the horizontal component isolated from any other compound component or to utilize the side effects of another plan step. In this example the side effect of step machine(rising-1) is used. This is supported by the rule combine(horizontal) that introduces a goal that a connected area is machined. (Note: these rules are different from control rules; they are comparable to operators but do not represent actions and are not part of the resulting process plan.) The new goal machined(?AREA) can be matched to one of the goals of the initial problem description and can be statisfied by a phantomization. As different phantomizations are possible, this is a decision point that can strongly influence the execution costs of the plan.

By this phantomization, a decision is made that the compound areas rising-1 and horizontal are processed by one step. This influences the expansion of the abstract plan step machine(rising-1) into a new planning problem on the concrete planning level (Fig. 6). Depending on this decision, the new planning problem is to select and to order cutting operations for the atomic processing areas of the compound areas rising-1 and horizontal.

Case-based Planning

Planning in CAPLAN is done by depth-first search or guided by user interactions. A user interaction plays the role of control knowledge to find a plan with low execution costs. To improve the behavior of the planner incrementally, a partial or complete solution can be stored as a case, that consists of the hierarchical problem description as well as of the planning decisions and their rationals. This offers an opportunity to add control knowledge to the system and to increase its planning expertise because a case possibly contains user interactions that complement the insufficiencies of the model.

In the current implementation, case retrieval is made by comparing the hierarchical description of a new problem with the problem descriptions of stored cases. Using the tree-based problem representation the case match is realized by tree matching. The similarity measure is computed as weighted sum of all adding and deleting operations necessary for transforming the hierarchical problem description of the case into the description of the current problem. Every transformation step is associated with specific costs. A case qualifies for solving a new problem if the computed measure is greater than a given limit.

If there is a case matching the current problem, the decisions of its plan are replayed by CAPLAN/CBC, that uses the user interface of CAPLAN to control the planning process. But a planning decision of the case is only reused if its rationals are compatible with the problem description of the new problem. Therefore, most problems can only be solved partially by case replay such that the partial solution must be completed by first-principle planning. Because both first-principle planning and case-based planning use the same planner, all replayed planning decisions can be retracted during the completion of a partial solution.

Discussion

We have described a hybrid planning architecture for supporting planning for machining workpieces. A main characteristic of this domain is that planning is driven by minimizing plan execution costs. Human planners use a large number of heuristics and domain specific reasoning but their usage is not sufficiently supported by known planning techniques. Therefore, we support classical planning by extensive preprocessing to unburden the planner as far as possible from domain specific reasoning. To overcome the remaining gap we give the user control over the planning process that makes necessary the storage of decision rationals for supporting conservative plan changes.

A further main problem is the incremental acquisition of control knowledge. Currently, it is supported by storing complete solutions as cases. Case retrieval is guided by the problem description only, without considering the relevance of features for the associated plan. Nevertheless, it seems to be a basis for future work on incremental elicitation and learning of domain specific control knowledge for optimizing plans.

References

Barrett, A., and Weld, D. S. 1993. Partial-order planning. *Artificial Intelligence.*

Bergmann, R. 1992. Learning abstract plans to speed up hierarchical planning. In Tadepalli, P., ed., *Proceedings of Workshop "Knowledge Compilation and Speedup Learning" at the Machine Learning Conference.* Scotland: University of Aberdeen.

Borrajo, D., and Veloso, M. 1994. Incremental learning of control knowledge for nonlinear problem solving. In *Proceedings of the European Conference on Machine Learning.*

Chapman, D. 1987. Planning for Conjunctive Goals. *Artificial Intelligence* 32:333–377.

Firlej, M., and Hellens, D. 1991. *Knowledge Elicitation — A Practical Handbook.* Prentice Hall.

Kambhampati, S.; Cutkosky, M.; Tenenbaum, M.; and Lee, S. H. 1991. Combining specialized reasoners and general purpose planners: A case study. In *Proceedings of AAAI-91.* Menlo Park, California: MIT Press.

Marcus, S. 1988. *Automating Knowledge Acquisition for Expert Systems.* Kluwer Academic Publishers.

McAllester, D., and Rosenblitt, D. 1991. Systematic nonlinear planning. In *Proceedings of AAAI-91.* Menlo Park, California: MIT Press.

Minton, S. 1988. *Learning Search Control Knowledge — An Explanation-Based Approach.* Kluwer Academic Publishers.

Pérez, M. A., and Carbonell, J. 1993. Automated acquisition of control knowledge to improve the quality of plans. Technical Report CMU-CS-93-142, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213.

Petrie, C. 1991. Context Maintenance. In *Proceedings of AAAI-91.* Menlo Park, California: MIT Press.

Veloso, M. M. 1992. *Learning by Analogical Reasoning in General Problem Solving.* CMU-CS-92-174, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213.