TECHNICAL RESEARCH REPORT

Manufacturing-Operation Planning Versus AI Planning

by D.S. Nau, S.K. Gupta and W.C. Regli

T.R. 95-4

INSTITUTE FOR SYSTEMS RESEARCH

Sponsored by
the National Science Foundation
Engineering Research Center Program,
the University of Maryland,
Harvard University,
and Industry

Manufacturing-Operation Planning Versus AI Planning*

Dana S. Nau†
University of Maryland
College Park, MD 20742

Satyandra K. Gupta‡
Carnegie Mellon University
Pittsburgh, PA 15213

William C. Regli§
University of Maryland
College Park, MD 20742

Abstract

Although AI planning techniques can potentially be useful in several manufacturing domains, this potential remains largely unrealized. Many of the issues important to manufacturing engineers have not seemed interesting to AI researchers—but in order to adapt AI planning techniques to manufacturing, it is important to address these issues in a realistic and robust manner. Furthermore, by investigating these issues, AI researchers may be able to discover principles that are relevant for AI planning in general.

As an example, in this paper we describe the techniques for manufacturing-operation planning used in IMACS (Interactive Manufacturability Analysis and Critiquing System). We compare and contrast them with the techniques used in classical AI planning systems, and point out that some of the techniques used in IMACS may also be useful in other kinds of planning problems.

1 Introduction

AI planning techniques can potentially be useful in several manufacturing domains. However, with the exception of manufacturing scheduling, previous applications of AI planning technology to manufacturing (cf. [7]) generally have had little impact on manufacturing practices [17, 29, 34], and manufacturing engineers have tended to view AI approaches as impractical for real manufacturing problems.

One reason for this difficulty is the differences in how AI planning researchers and manufacturing planning researchers view the world. For example, the first author's work on manufacturing planning (e.g., [27, 21, 22, 28, 15, 16, 14, 13, 12, 3, 31]) has significantly influenced his research on AI planning (e.g., [9, 41, 10, 6, 5, 20, 4]), and vice versa. However, this influence is not particularly evident in the publications themselves, because they were written to address two different audiences, who have different ideas of what the important problems are and how they should be solved:

- *AI planning researchers* usually want to solve general conceptual problems, and are less interested in problem-dependent details. Thus, the AI approach to manufacturing planning has typically been to create an abstract problem representation that omits unimportant details, and then look for ways to solve the abstract problem. However, from the viewpoint of the manufacturing engineer, these "unimportant details" can often be essential parts of the problem. This leads manufacturing engineers to view AI planning techniques as impractical for solving the problems they really want to solve.

- In *manufacturing planning research*, the goal is to solve a particular manufacturing problem. Manufacturing engineers present their research results within the context of this problem—and whether or how the approach might generalize to other planning domains is usually not discussed, because it is not their primary concern. From the standpoint of AI researchers, this makes it difficult to see what the underlying conceptual problems are, or whether the approach embodies a general idea that can be applied to other problems. Thus, AI planning

*This work was supported in part by NSF Grants DDM-9201779, IRI-9306580, and NSFD EEC 94-02384. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

†Computer Science Department and Institute for Systems Research. Email: nau@cs.umd.edu. URL: http://www.cs.umd.edu/~nau.

‡Robotics Institute. Email: skgupta@isl1.ri.cmu.edu.

§Computer Science Department and Institute for Systems Research. Email: regli@cs.umd.edu. URL: http://www.cs.umd.edu/~regli.

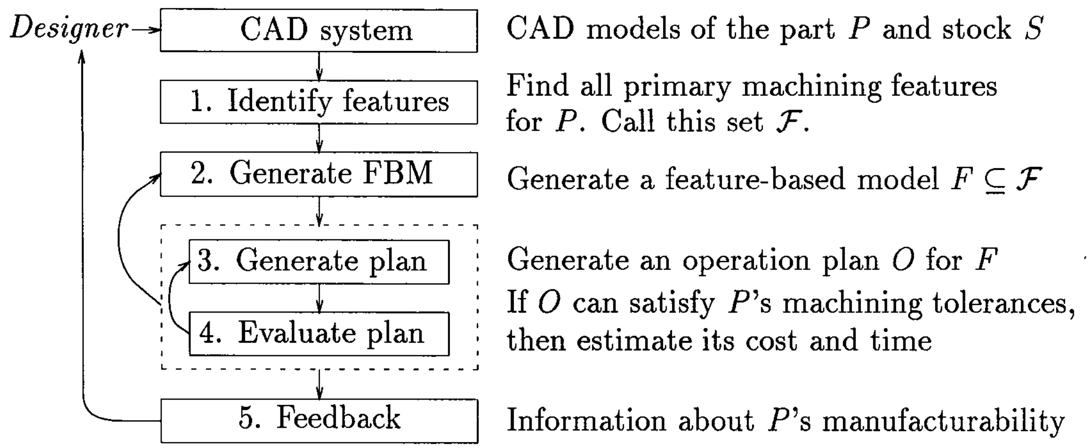


Figure 1: Basic approach used in IMACS.

> researchers have tended to view manufacturing planning as a problem domain in which there are no general principles and approaches—just ad-hoc, domain-specific programs.

Some of the issues that arise manufacturing planning are similar to issues that have been investigated by AI planning researchers, and others are distinctly different. For the former, it may be possible to adapt existing AI planning techniques—and for the latter, it may be possible to develop new planning techniques that are useful for AI planning in general. However, one of the difficulties is that AI researchers are not aware what the interesting generalizations are, and which techniques from AI might best be applied to realistic manufacturing problems. In order to develop AI planning techniques that have a greater impact on manufacturing tasks such as process planning, AI planning researchers will need a better understanding of manufacturing concerns, and how they compare with issues of interest in AI planning.

In this paper we attempt to provide a step in this direction, by describing the planning techniques used in IMACS (Interactive Manufacturability Analysis and Critiquing System), a computer system for helping designers produce designs that are easier to manufacture. IMACS analyzes the manufacturability of proposed designs for parts to be machined in a three-axis vertical machining center, by generating and evaluating operation plans for the proposed design as shown in Figure 1. This paper compares and contrasts IMACS's planning techniques to some of the techniques used in AI planning, and describes some planning techniques used in IMACS that may also be useful in other planning domains.

2 Operation Planning in IMACS

This section describes the techniques IMACS uses to generate and evaluate operation plans. Each subsection discusses one of the steps in Figure 1.

2.1 Step 1: Finding Machining Features

A *part*, P, is the final component created by executing a set of machining operations on a piece of *stock*, S. For example, Figure 2 shows a design for a socket which we will call P_0, and Figure 3 shows the stock S_0 from which P_0 is to be produced. The annotations in Figure 2 are *tolerance specifications* that tell how much variation from the nominal geometry is allowable in any physical realization of P. As input, IMACS takes solid models of P and S, along with tolerance specifications for P.

An *operation plan* is a sequence of machining operations capable of to creating P from S. A *workpiece* is the intermediate object produced by starting with S and performing zero or more machining operations. A *machining feature* is a portion of the workpiece affected by a machining operation. The machining operations IMACS currently considers are end milling, side milling, face milling and drilling.

A *primary* feature is a machining feature whose intersection with the stock S is as large as possible, and whose intersection with the space outside the stock S is as small as possible. Figure 4 shows examples of primary and non-primary features; for a detailed definition the reader is referred to [13]. As

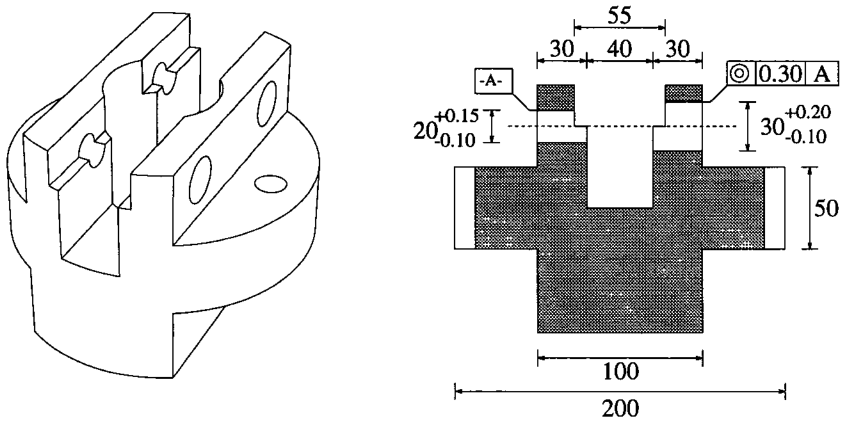


Figure 2: Design for the socket P_0.

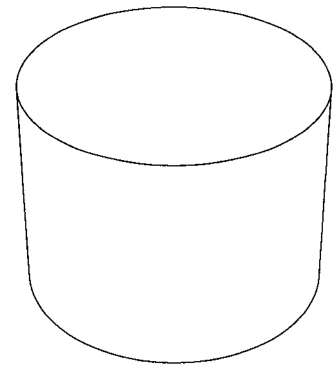


Figure 3: The stock S_0 from which P_0 is to be created.

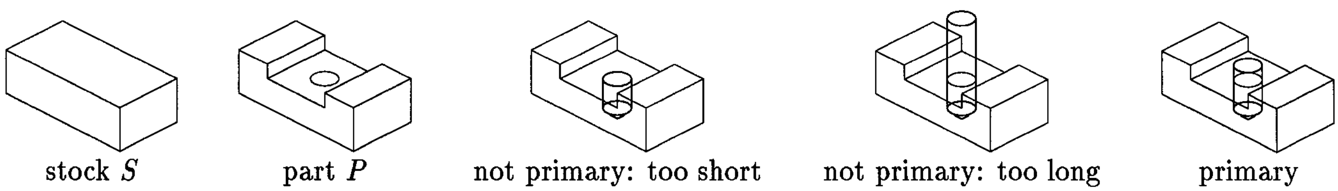


Figure 4: Example of non-primary and primary drilling features for a part P and stock S.

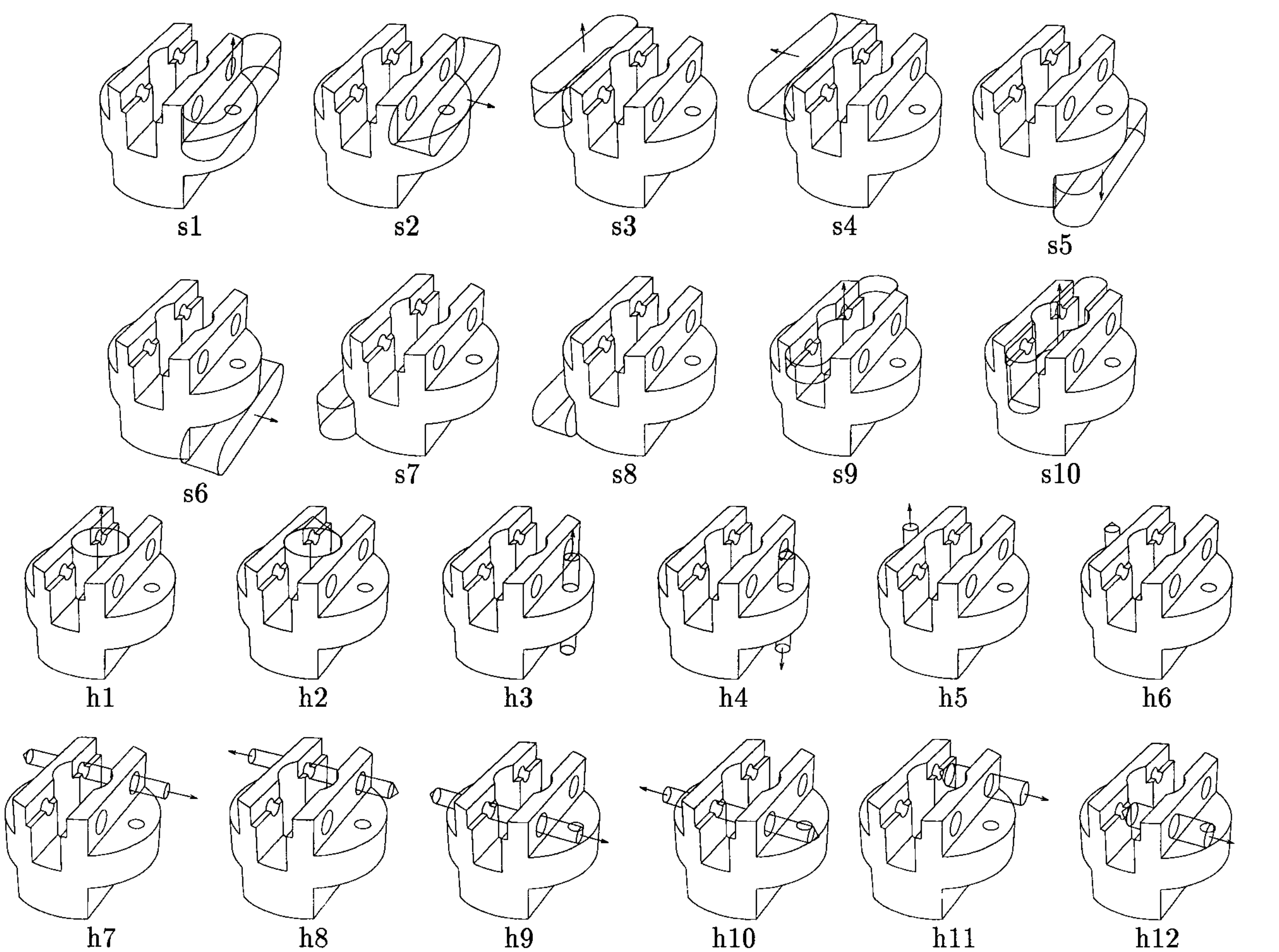


Figure 5: The set $\mathcal{F}$ of all primary features for the socket P_0. $s1, \ldots, s10$ are end-milling features; $h1, \ldots, h12$ are drilling features.

described in [11, 32], the reason why we are interested in primary features is that in every operation plan we will have any interest in considering, each machining operation will create either a primary feature or a truncation of a primary feature. Thus, primary features can be used to derive every machining operation that IMACS will ever want to consider.

$\mathcal{F}$ is the set of all primary features for P and S. IMACS generates $\mathcal{F}$ automatically from the solid models of P and S, using an algorithm described in [31, 12]. For example, there are 22 primary features for the socket P_0, as shown in Figure 5. Since the features in $\mathcal{F}$ can overlap, not all of them will always be needed in order create P from S. For example, in Figure 5, we would not need to machine both $s3$ and $s4$ in order to create P_0.

2.2 Step 2: Generating FBMs

A *Feature Based Model* (FBM) is any irredundant subset of features $F \subseteq \mathcal{F}$ such that P can be produced from S by removing the features in F. For example, here are two FBMs for the socket P_0, composed of features from Figure 5:

$$\begin{aligned} \text{FBM1} &= \{s2, s4, s6, s8, s9, s10, \\ &\quad h1, h3, h5, h7, h9, h11, h12\}; \\ \text{FBM2} &= \{s1, s3, s5, s7, s9, s10, \\ &\quad h1, h3, h5, h7, h9, h11, h12\}. \end{aligned}$$

As described in [13, 32], each operation plan O of interest to us corresponds to an FBM, in the sense that each machining operation in O will create either a feature in F or a truncation of a feature in F.

Since each FBM is a subset of $\mathcal{F}$, FBMs can be generated using set-covering techniques. However, we usually will not want to generate *all* of these FBMs. For a given part and stock, there can be exponentially many FBMs—for example, from the 22 primary features shown in Figure 5 one can form 512 FBMs for the socket P_0. In general, only a few of the FBMs will lead to good operation plans.

As described in [14, 13], IMACS avoids enumerating all of the FBMs by doing a depth-first branch-and-bound search: as shown in Figure 1, FBMs are generated one at a time and are pruned if they appear unpromising. For example, IMACS generates only 16 of the 512 FBMs for the socket P_0.

2.3 Step 3: Generating Operation Plans

Each FBM can lead to several operation plans, of which some are better than others. Thus, to generate operation plans from a given FBM, IMACS again does a depth-first branch-and-bound search. The search procedure incorporates the following steps:

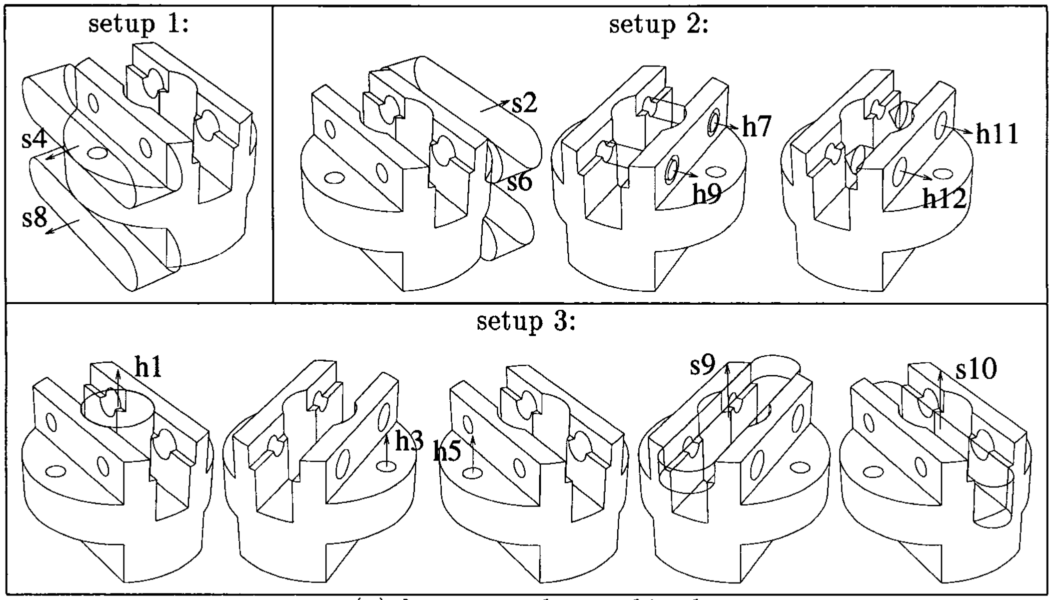
- **Find precedence constraints.** Due to various types of interactions (accessibility, setup, etc.) among the features in an FBM F, the features of F cannot be machined in any arbitrary order. Instead, these interactions introduce precedence constraints requiring that some features of F be machined before or after other features. For example, in Figure 6, the hole $h1$ must be machined before the slot $s9$ in order to achieve reasonable machining tolerances and avoid tool breakage.

- **Generate total orderings.** If the precedence constraints contradict each other (i.e., if there is no total ordering consistent with them), then we consider F to be unmachinable. Otherwise, IMACS generates the total orderings on F consistent with the precedence constraints.

- **Truncate features.** Each total ordering will require a different set of modifications to the features in F, so that the machining operations will not spend a lot of time trying to remove metal that was already removed in previous machining operations. As an example, several of the features shown in Figure 6(a) were produced by truncating the corresponding features in FBM1.

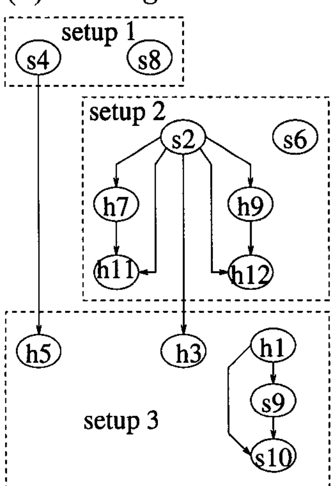
- **Identify Unpromising FBMs.** Once the features have been truncated, IMACS will discard an FBM if it contains features whose dimensions and tolerances appear unreasonable. Examples would include a hole-drilling operation having too large a length-to-diameter ratio; a recess-boring operation having too large a ratio of outer diameter to inner diameter; two concentric hole-drilling operations with tight concentricity tolerance and opposite approach directions.

- **Analyze Fixturability.** IMACS does not yet do fixturability analysis in any detailed manner—but in order to discard unpromising FBMs, it does some elementary fixturability-based pruning, based on the assumption that



(a) features to be machined

(b) ordering constraints



(c) process details

Feature name	Feature type	Tool diam (mm)	Feed (mm/min)	Number of passes	Pass length (mm)
s4	end-milling	50	166	2	225
s8	end-milling	50	166	2	225
s2	end-milling	50	166	2	225
s6	end-milling	50	166	2	225
h7	drilling	20	244	1	106
h9	drilling	20	244	1	106
h11	drilling	30	203	1	39
h12	drilling	30	203	1	39
h1	drilling	75	108	1	172.5
h3	drilling	20	244	1	56
h5	drilling	20	244	1	56
s9	end-milling	50	166	1	250
s10	end-milling	40	207	3	240

Figure 6: An operation plan derived from FBM1. This plan is the least costly one for making the socket P_0.

Table 1: Estimated production time for the operation plan of Figure 6.

Operation	Time (min)	Operation	Time (min)
drill h1	2.3	mill s2	5.0
drill h3	0.3	mill s4	5.0
drill h5	0.3	mill s6	5.0
drill h7	0.6	mill s8	5.0
drill h9	0.6	mill s9	4.0
drill h11	0.3	mill s10	4.2
drill h12	0.3	3 setups	6.0

Total Time: 39 minutes

a flat-jaw vise is the only available fixturing device. We are currently developing more sophisticated fixturability analysis techniques for IMACS; this will be described in a forthcoming paper.

- **Relax redundant constraints.** Once the truncated features have been produced, several of the resulting FBMs may have identical features but different precedence constraints. When this occurs, the precedence constraints that differ can be removed, translating the total orders into partial orders. For example, Figure 6(b) shows the partial order for the FBM of Figure 6(a).

- **Incorporate finishing operations.** For faces with tight surface finishes or tolerances, IMACS adds finishing operations, with precedence constraints so that each finishing operation comes after the corresponding roughing operation. Currently, one finishing operation per face is allowed.

- **Determine setups.** On a three-axis vertical machining center, features cannot be machined in the same setup unless they have the same approach direction. This and the partial ordering constraints can be used to determine which features can be machined in the same setup, as shown in Figure 6(b).

- **Determine process details.** To select cutting parameters for the machining operations, IMACS uses the recommendations of the Machinability Data Center's handbook [23]. The maximum recommended cutting parameters are used, rather than attempting to select optimal cutting parameters; thus IMACS's estimates involve considerable approximation.

2.4 Step 4: Operation Plan Evaluation

Designers give *design tolerance* specifications to specify how far the design can vary from its nominal geometry. To verify whether a given operation plan will satisfy the design tolerances, IMACS must estimate what tolerances the operations can achieve. Unlike typical approaches for computer-aided tolerance charting (which are computationally very intensive, and only consider limited types of tolerances [18, 26]), IMACS evaluates the manufacturability aspects of a wide variety of tolerances without getting into optimization aspects; our approach is described in [13]. For example, the operation plan shown in Figure 6 satisfies the tolerances shown in Figure 2. Thus, it is an acceptable operation plan for making P_0 from S_0.

The total time of a machining operation consists of two components: the cutting time (when the tool is actually engaged in machining), and the non-cutting time (including the tool-change time, setup time, etc.). Methods have been developed for estimating the fixed and variable costs of machining operations; our formulas for estimating these costs are based on standard handbooks related to machining economics, such as [39, 38]. As an example, Table 1 shows the estimated production time for the operation plan of Figure 6.

3 Comparison with AI Planning

Two of the most popular approaches to AI planning are STRIPS-style planning[1] [8, 2, 1, 9, 25, 24, 42, 30, 6] and hierarchical task-network (HTN) planning [33, 35, 37, 36, 40, 19, 5, 4]. In both cases, the planner typically starts with some *initial state* that is represented as a collection of logical atoms. In STRIPS-style planning, the objective is to produce a state that satisfies a *goal condition* expressed as a collection of logical atoms, and the planner produces the plan by reasoning about the preconditions and effects of STRIPS-style planning operators. In HTN planning, the objective is expressed as a set of *tasks* to be performed and constraints on how they are to be performed, and the planner produces the plan using *methods* that specify ways to decompose tasks into operators and other tasks, and *critics* that point out problems in the plan decomposition.

Below, we compare and contrast the techniques used in IMACS to the techniques used in STRIPS-style planning and HTN planning.

- **The overall goal.** In manufacturing planning, the goal to be achieved is represented by a design specification such as the one in Figure 2. In AI planning systems, the goal is typically

[1] By this, we mean planning systems that use STRIPS-style operators (with no decompositions), ignoring algorithmic differences among them that are not relevant to the current work. This includes partial-order planners such as ABTWEAK [42] and UCPOP [30].

something that must be achieved exactly—but in planning a sequence of machining operations, it is physically impossible to produce the *exact* nominal geometry of the design. Thus, the objective is to find any plan that can produce an approximation of the design geometry that satisfies various design tolerances, such as those shown in Figure 2.

- **Goal modification.** AI planning systems typically treat the goal as a fixed entity. However, IMACS is intended to operate as part of a "design loop" such as the one shown in Figure 7, in which the designer proposes a design, uses IMACS to evaluate its manufacturability, and modifies the design accordingly. In the current implementation of IMACS, the design modifications are proposed only by the designer, but in [3] we discuss ways to extend IMACS to automatically make suggestions to the designer about ways to modify the design that will improve its manufacturability while satisfying the designer's objectives. From an AI perspective, this would correspond to changing the goal to make it easier to achieve.

- **Finding subgoals.** In principle, design specifications such as the one in Figure 2 could be expressed as collections of logical atoms: for example, each face, edge, and vertex in the CAD model could be represented by a different atom. However, this would not be very useful. In order to make use of the design specifications, IMACS does feature extraction in order to transform them into something quite different: a set of machining features such as those shown in Figure 5.

  Since the machining features correspond one-for-one to machining operations that will create them, feature extraction can be thought of as finding subgoals to achieve. In AI planners, subgoals normally arise during plan construction, because they are specified in task decompositions or occur as preconditions of planning operators. However, the set of primary features $\mathcal{F}$ found by IMACS corresponds to all of the subgoals it will care to consider—and IMACS finds these subgoals during the feature extraction step, before it ever tries to construct a plan.

- **Alternative sets of subgoals.** Since $\mathcal{F}$ corresponds to the set of all possible subgoals that might occur during planning, each FBM $F \subseteq \mathcal{F}$ corresponds to a collection of subgoals that is sufficient to produce the final goal. In principle, it would be possible to combine the features in $\mathcal{F}$ into a goal formula like those used in classical AI planners, but this formula would be a *disjunct of conjuncts* of the form $F_1 \vee F_2 \vee \ldots \vee F_n$, where each FBM $F_i \subseteq \mathcal{F}$ is taken to represent the conjunctive goal of creating all features in F_i. Since there can be an exponential number of FBMs, representing this goal formula explicitly would require exponential time and space in the worst case. Rather than constructing this formula explicitly, IMACS uses a branch-and-bound approach to generate the FBMs one at a time, pruning the unpromising ones before they are fully generated.

- **Resolving goal interactions.** Each feature in an FBM corresponds to a machining operation, so the entire FBM corresponds to a partially ordered plan. If the interactions among these features cannot be resolved by creating precedence constraints, then IMACS discards the plan. There would be no point in adding additional operators to the plan, because these operators would just create redundant features.

- **Subgoal modification.** During planning, IMACS sometimes truncates some of the features, so that the resulting operation plans won't end up spending too much time machining air. Truncating the features corresponds to *modifying* the subgoals in such a manner that the ultimate goal will still be achieved—something that usually does not occur in traditional AI planners.

- **Finding optimal plans.** Most AI planning systems stop as soon as they have found a plan that achieves the goal—but IMACS looks for the least costly plan capable of producing the design. Thus, IMACS uses a branch-and-bound search to continue generating and evaluating plans until it is evident that none of the remaining plans will be any better than the best one seen so far. In order to do this efficiently, IMACS prunes a plan whenever various cost computations make it evident that the plan is unpromising.

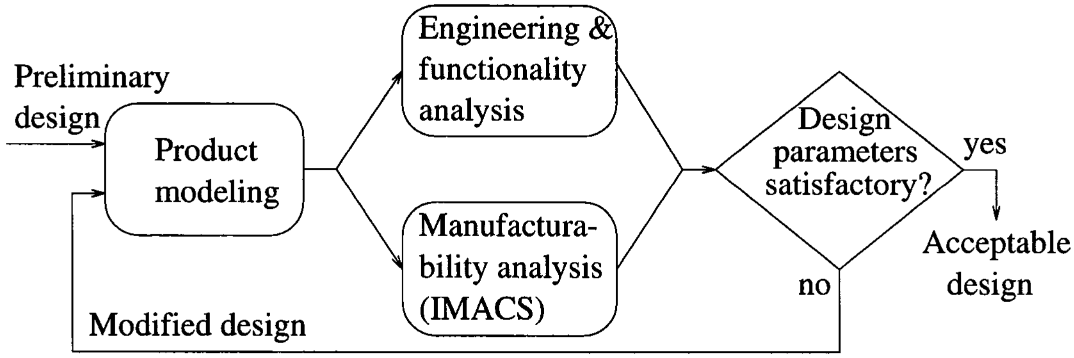


Figure 7: Using IMACS as part of a design loop.

4 Conclusions

In developing IMACS, we did not care whether or not we were using AI planning techniques; the goal was to find a useful solution to a real manufacturing problem. Thus, although there are some similarities between the techniques used in IMACS and those used in classical AI planning systems, there are also some significant differences. Some of these differences can be generalized in ways that may be useful in other domains as well.

One example is IMACS's use of primary features and feature-based models. Each primary feature corresponds to a subgoal to be achieved—and (except for finishing operations, which are handled separately), the set $\mathcal{F}$ of all primary features includes all subgoals that might ever be relevant for achieving the overall goal. This simplifies the task of resolving goal interactions, in the following manner. Each FBM $F \subseteq \mathcal{F}$ is a set of subgoals whose achievement is sufficient to achieve the overall goal, and if it contains a goal interaction that cannot be resolved by introducing precedence constraints, then there is no point in introducing new operators into the plan. If a promising plan exists for achieving the overall goal, then it can instead be found among the other FBMs. Thus if IMACS cannot resolve goal interactions in an FBM by introducing precedence constraints, it discards the FBM and tries another one.

In [11] we point out that this approach is useful not only in producing operation plans for machined parts, but also in other manufacturing domains. The same kind of approach should be useful in other planning problems regardless of whether or not they are manufacturing problems, provided that they are problems for which one can enumerate in advance all of the goals or tasks that one might need to achieve.

In order to develop realistic and robust approaches to manufacturing planning, it is important to address some of the details of manufacturing that AI researchers have typically ignored. The development of IMACS illustrates that it is possible to do this in a principled manner. Furthermore, some of the principles that are developed in this way may be relevant for planning in general.

References

[1] Tom Bylander. Complexity results for planning. In *IJCAI-91*, 1991.

[2] David Chapman. Planning for conjunctive goals. *Artificial Intelligence*, 32:333–379, 1987.

[3] D. Das, S. K. Gupta, and D. Nau. Reducing setup cost by automated generation of redesign suggestions. In *ASME Conf. Computers in Engineering*, 1994. Best-paper award winner. Also available as ISR TR 94-12.

[4] K. Erol, J. Hendler, and D. Nau. Complexity results for hierarchical task-network planning. *Annals of Mathematics and Artificial Intelligence*, (CS-TR-3240, UMIACS-TR-94-32), 1994. To appear.

[5] K. Erol, J. Hendler, and D. S. Nau. HTN planning: Complexity and expressivity. In *AAAI-94*, 1994.

[6] K. Erol, D. Nau, and V. S. Subrahmanian. Complexity, decidability and undecidability results for domain-independent planning. *Artificial Intelligence*, 1994. To appear.

[7] F. Famili, D. S. Nau, and S. Kim, editors. *Artificial Intelligence Applications in Manufacturing.* AAAI Press/MIT Press, 1992.

[8] R. E. Fikes and N. J. Nilsson. Strips: a new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2(3/4):189–208, 1971.

[9] Naresh Gupta and Dana S. Nau. Complexity results for blocks-world planning. In *Proc. AAAI-91*, 1991. Honorable mention for the best paper award.

[10] Naresh Gupta and Dana S. Nau. On the complexity of blocks-world planning. *Artificial Intelligence*, 56(2-3):223–254, August 1992.

[11] S. Gupta, W. Regli, and D. Nau. Manufacturing feature instances: Which ones to recognize? In *ACM Solid Modeling Conference*, 1995. To appear. Available as CS-TR-3376, UMIACS-TR-94-127, ISR-TR-94-81.

[12] S. K. Gupta, T. R. Kramer, D. S. Nau, W. C. Regli, and G. Zhang. Building MRSEV models for CAM applications. *Advances in Engineering Software*, 1994. To appear. Also available as ISR TR 93-84, CS-TR-3331, UMIACS-94-97.

[13] S. K. Gupta and D. S. Nau. A systematic approach for analyzing the manufacturability of machined parts. *Computer Aided Design*, 1994. To appear. Also available as ISR TR 93-76, CS-TR-3158, UMIACS-TR-93-105.

[14] S. K. Gupta, D. S. Nau, W. C. Regli, and G. Zhang. A methodology for systematic generation and evaluation of alternative operation plans. In Jami Shah, Martti Mantyla, and Dana Nau, editors, *Advances in Feature Based Manufacturing*, pages 161–184. Elsevier/North Holland, 1994.

[15] S. K. Gupta, D. S. Nau, and G. Zhang. Concurrent evaluation of machinability during product design. *IEEE Computer*, 26(1):62–63, January 1993.

[16] S. K. Gupta, W. C. Regli, and D. S. Nau. Integrating DFM with CAD through design critiquing. *Concurrent Engineering: Research and Applications*, 1994. To appear. Also available as ISR TR 94-11, CS-TR-3330, UMIACS-TR-94-96.

[17] Inyong Ham and Stephen C.-Y. Lu. Compute-aided process planning: The present and the future. *Annals of the CIRP*, 37(2):591, 1988.

[18] Ping Ji. A tree approach for tolerance charting. *International Journal of Production Research*, 31(5):1023–1033, 1993.

[19] S. Kambhampati and J. Hendler. A validation structure based theory of plan modification and reuse. *Artificial Intelligence*, May 1992.

[20] S. Kambhampati and D. S. Nau. On the nature and role of modal truth criteria in planning. *Artificial Intelligence*, 1994. To appear. Also available as ISR-TR-93-30, CS-TR-3049, UMIACS-TR-93-27.

[21] R. Karinthi and D. Nau. An algebraic approach to feature interactions. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(4):469–484, April 1992.

[22] R. Karinthi and D. Nau. Geometric reasoning using a feature algebra. In F. Famili, S. Kim, and D. S. Nau, editors, *Artificial Intelligence Applications in Manufacturing*, pages 41–59. AAAI Press/MIT Press, 1992.

[23] Machinability Data Center. *Machining Data Handbook.* Metcut Research Associates, Cincinnati, Ohio, third edition, 1980.

[24] David McAllester and David Rosenblitt. Systematic nonlinear planning. In *AAAI-91*, pages 634–639, July 1991.

[25] S. Minton, J. Bresna, and M. Drummond. Commitment strategies in planning. In *Proc. IJCAI-91*, July 1991.

[26] R. O. Mittal, S. A. Irani, and E. A. Lehtihet. Tolerance control in the machining of discrete components. *Journal of Manufacturing Systems*, 9(3):233–246, 1990.

[27] D. S. Nau. Automated process planning using hierarchical abstraction. *TI Technical Journal*, pages 39–46, Winter 1987. Award winner, Texas Instruments 1987 Call for Papers on AI for Industrial Automation.

[28] D. S. Nau, G. Zhang, and S. K. Gupta. Generation and evaluation of alternative operation sequences. In A. R. Thangaraj, A. Bagchi, M. Ajanappa, and D. K. Anand, editors, *Quality Assurance through Integration of Manufacturing Processes and Systems, ASME Winter Annual Meeting*, volume PED-Vol. 56, pages 93–108, November 1992.

[29] James L. Nevins and Daniel E. Whitney, editors. *Concurrent Design of Products & Processes.* McGraw-Hill Publishing, 1989.

[30] J. Penberthy and D. S. Weld. UCPOP: A sound, complete, partial order planner for ADL. In *Proc. Third International Conference on Knowledge Representation and Reasoning.* October 1992.

[31] W. C. Regli, S. K. Gupta, and D. Nau. Feature recognition for manufacturability evaluation. In *ASME Conf. Computers in Engineering*, 1994. To appear. Also available as ISR TR 94-10.

[32] W. C. Regli and D. S. Nau. Recognition of volumetric features from CAD models: Problem formalization and algorithms. 1993. Submitted for journal publication. Available as ISR-TR-93-41.

[33] E. D. Sacerdoti. *A Structure for Plans and Behavior.* American Elsevier Publishing Company, 1977.

[34] Jami Shah, Martti Mantyla, and Dana Nau, editors. *Advances in Feature Based Manufacturing.* Elsevier/North Holland, 1994.

[35] Austin Tate. Generating project networks. In *Proc. 5th International Joint Conf. Artificial Intelligence*, 1977.

[36] David E. Wilkins. *Practical Planning: Extending the Classical AI Planning Paradigm.* Morgan-Kaufmann Publishers, Inc., San Mateo, CA, 1988.

[37] David E. Wilkins. Domain-independent planning: Representation and plan generation. In James Allen, James Hendler, and Austin Tate, editors, *Readings in Planning*, pages 319–335. Morgan Kaufman, 1990. Originally appeared in *Artificial Intelligence* 22(3), April 1984.

[38] F.W. Wilson and P.D. Harvey. *Manufacturing Planning and Estimating Handbook.* McGraw Hill, 1963.

[39] W. Winchell. *Realistic Cost Estimating for Manufacturing.* Society of Manufacturing Engineers, 1989.

[40] Q. Yang. Formalizing planning knowledge for hierarchical planning. *Computational Intelligence*, 6:12–24, 1990.

[41] Q. Yang, D. S. Nau, and J. Hendler. Merging separately generated plans with restricted interactions. *Computational Intelligence*, 8(2):648–676, February 1992.

[42] Q. Yang and J. D. Tenenberg. Abtweak: Abstracting a nonlinear, least commitment planner. In *AAAI-90*, pages 204–209, 1990.