Integrating DFM with CAD through Design Critiquing

Satyandra K. Gupta
Mechanical Engineering Department
Institute for Systems Research
skgupta@src.umd.edu

William C. Regli
Computer Science Department
Institute for Systems Research
regli@cs.umd.edu

Dana S. Nau
Computer Science Department
Institute for Systems Research
Institute for Advanced Computer Studies
nau@cs.umd.edu

University of Maryland
College Park, MD 20742 USA

Accepted for publication *Concurrent Engineering: Research and Applications*,
Volume 2, Number 2.
Also available as **CS-TR-3330, UMIACS-TR-94-96, ISR TR94-11**.

Abstract

The increasing focus on *design for manufacturability* (DFM) in research in concurrent engineering and engineering design is expanding the scope of traditional design activities in order to identify and eliminate manufacturing problems during the design stage. Manufacturing a product generally involves many different kinds of manufacturing activities, each having different characteristics. A design that is good for one kind of activity may not be good for another; for example, a design that is easy to assemble may not be easy to machine. One obstacle to DFM is the difficulty involved in building a single system that can handle the various manufacturing domains relevant to a design.

In this paper we propose an architecture for integrating CAD with DFM. As the designer creates a design multiple critiquing systems analyze its manufacturability with respect to different manufacturing domains such as machining, fixturing, assembly, and inspection. Using this analysis, each critiquing system offers advice about potential ways of improving the design and an integration module mediates conflicts among the different critiquing systems in order to provide feedback to improve the overall design.

We anticipate that this approach can be used to build a multi-domain environment that will allow designers to create higher-quality products that can be more economically manufactured. This will reduce the need for redesign and reduce product cost and lead time.

keywords: design for manufacturability, design critiquing, manufacturability analysis, feature based modeling, multi-agent coordination.

1 Introduction

Survival in the modern economic climate requires that competitively-priced, well-designed and well-manufactured products be brought to market in a timely fashion. Although product design incurs only a small fraction of the total product cost, the decisions made during the design phase determine the cost of the product over its life-cycle and can be crucial to the success or failure of the product. Since the cost of making essential design changes escalates steeply with time, the ability to make crucial changes during the design phase translates into significant savings over making changes during later production.

The availability of low-cost computational power is providing designers with a variety of CAD tools to help increase productivity and reduce time-consuming *build-test-redesign* iterations. Examples include tools for finite element analysis, mechanism analysis, simulation, and rapid prototyping. The availability of such tools has become a driving force for research in concurrent engineering and engineering design. As the advantages of concurrent engineering are being realized, more downstream activities associated with the various life-cycle demands of the product are being considered during the design phase [1, 5, 20].

The *design for manufacturability* (DFM) methodology has been established as an effective way to avoid manufacturing problems. In DFM all of the design goals and manufacturing constraints are considered simultaneously and analyzed over the life cycle of a product. Such analysis can identify design elements that pose problems for manufacturing and suggest changes in the design to address these problems.

DFM expands the scope of traditional design activities, making the designer's task increasingly complex. Manufacturing consists of activities in many different domains, each having different characteristics. A design that is good for one domain may not be good for another; for example, a design that is easy to assemble may not be easy to machine. One obstacle to DFM is the difficulty involved in building a single system that can handle the various manufacturing domains relevant to a design.

In this paper we propose an architecture for integrating CAD with DFM. In our proposed framework multiple critiquing systems perform analysis as the designer creates an artifact. This analysis considers the manufacturability of the artifact with respect to different manufacturing domains (such as machining, fixturing, assembly, inspection, and so forth) that will be used in creating it. Based on this analysis, the critiquing system can provide advice about potential ways of improving the design. For each manufacturing domain we propose the use of a dedicated critiquing system that recognizes required manufacturing information from the design, evaluates the design's manufacturability, and provides feedback on how to improve the design. The analysis carried out by each critiquing system will require the following basic capabilities:

1. In order to analyze a design, each critiquing system should first be capable of understanding it and generating alternate interpretations for it with respect to the manufacturing domain at hand.

2. After understanding a design, each critiquing system will need to calculate an estimation of the design's merit for the critiquing system's particular manufacturing domain. At this stage, design attributes that are difficult or impossible to manufacture can be identified and diagnosed.

3. Finally, suggestions about potential ways of improving the design must be formulated. Design attributes that are impossible or expensive to manufacture can be systematically considered for elimination or modification.

As shown in Figure 1, the activities of these critiquing systems will need to be coordinated by an integration module. The integration module will activate different critiquing systems at appropriate times, possibly in a sequence specified by the designer. The integration module will also analyze tradeoffs and mediate conflicts in the suggestions produced by the critiquing systems, so that the suggestions transmitted to the designer will improve the overall design instead of simply optimizing one particular aspect of its manufacturability.

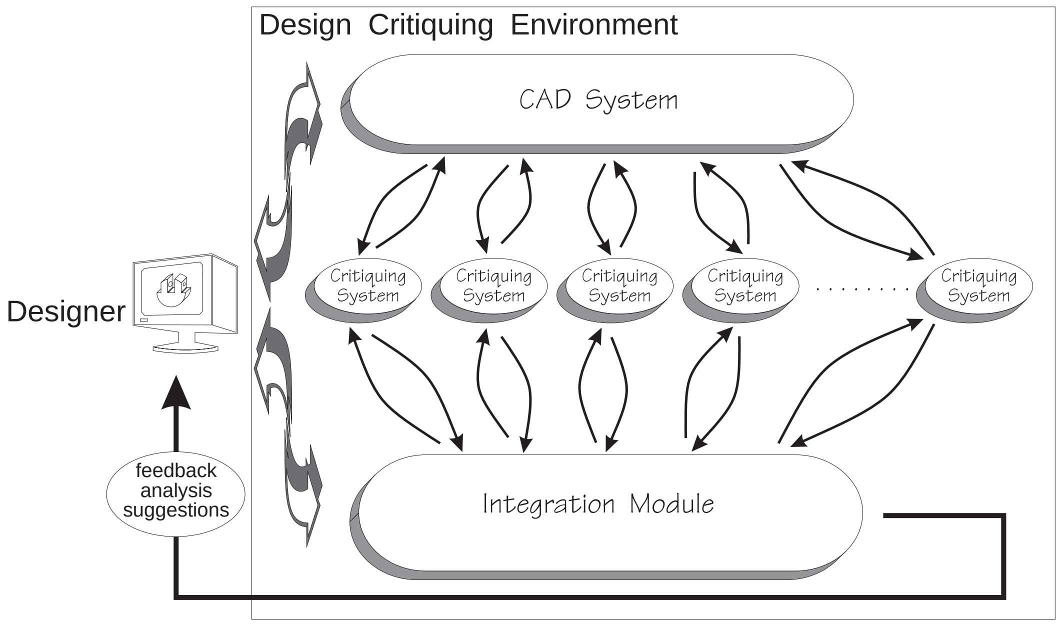


Figure 1: Integration of CAD and DFM through a multi-system critiquing architecture.

In the steps outlined above, understanding and analyzing the design requires relating design attributes to manufacturing activity. For this we propose a feature-based approach. A design consists of *design attributes* that need to be realized through manufacturing operations, each requiring manufacturing activities in one or more domains. Examples of design attributes include manufacturing features, feature relationships, tolerances, functional requirements, aesthetic characteristics, and properties of shape and form. In the case of a design attribute such as a hole, several of machining operations may be required to produce it and an inspection operation needed to measure dimensional accuracy. For each manufacturing domain, we use domain-specific features to represent the relationship between design attributes and manufacturing activities. For the purpose of design critiquing, it is beneficial to employ features that correspond directly to manufacturing operations.

In previous work [9, 11, 21], we worked on the problem of manufacturability evaluation for machined parts describable using a PDES/STEP-based class of features. In this paper, we are attempting to ensure that the manufacturing processes, feature classes, and evaluation and feedback criteria are not tied to any specific manufacturing domain. This paper discusses some of the issues involved in scaling up our previous work to deal with more general problems in design for manufacturability.

We expect that an architecture for a general, multi-domain, design critiquing system will help achieve DFM. An environment such as this will allow designers to create higher-quality products that are economically manufacturable by the processes at their disposal. We anticipate that this will reduce the need for redesign and result in a reduction in product cost and lead time.

The remaining paper is organized as follows. Section 2 presents a brief overview of design automation, and Section 3 gives an overview of how the individual critiquing systems would work. Section 4 describes

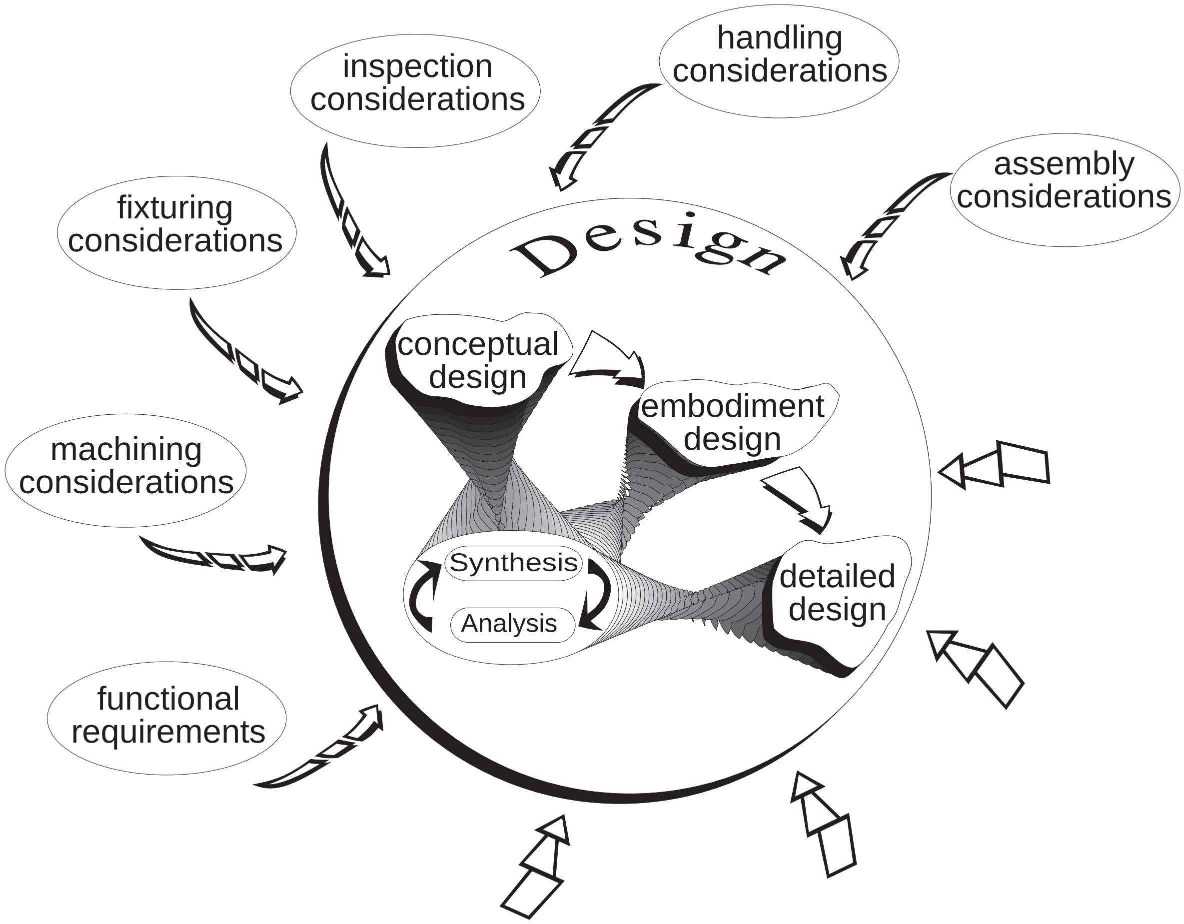


Figure 2: Design For Manufacturability.

how the design is interpreted in terms of features for various manufacturing domains. Section 5 describes how the design is critiqued for various manufacturing domains. Section 6 describes how the feedback information is generated. Section 7 contains concluding remarks and discussion of our work.

2 Design Automation

Design for manufacturability involves considering manufacturing constraints through the design process, as shown in Figure 2. DFM starts at the conceptual design stage and continues through the embodiment and detailed design stages. Different companies have tried a variety of approaches to implement the DFM methodology, ranging from building inter-departmental design teams to equipping designers with manufacturability checklists. With the advent and popularity of various CAD tools, there is increasing interest in supporting DFM through intelligent CAD systems.

Traditionally the design process has involved two main activities: synthesis and analysis. Most of the present generation of CAD tools are geared towards analysis. As higher levels of design automation are demanded, investigation of automation for synthesis-driven activities becomes necessary. Ideally one would like a system which takes customer needs as input and automatically designs a suitable product.

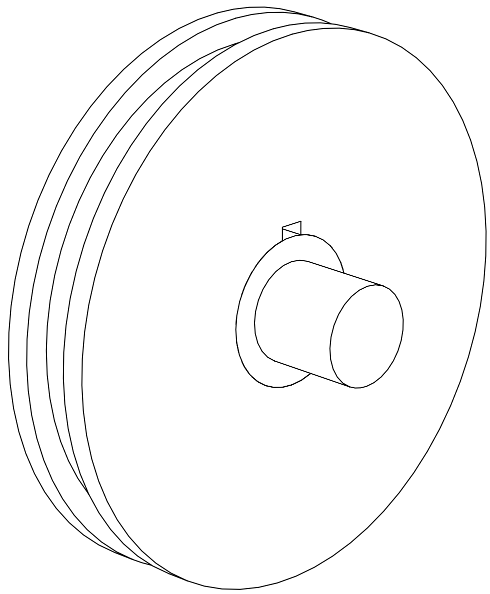


Figure 3: Shaft and Pulley Assembly.

There are no general solutions to this kind of design automation. However, several attempts have been made to achieve such automation in specific domains (e.g., certain types of mechanical designs [2, 25]).

Design problems are of varying difficulty. For example, a simple design problem is the selection of standard components; for this type of problem a considerable level of automation has been achieved. If a designer specifies the forces and dimensional constraints, a system can automatically select a suitable bearing for that application. In more difficult design problems the physical configuration for the design is known or can be derived from functional requirements; the designer's problem is then one of choosing appropriate parameters for the particular application. An example of this is the design of a transmission train for a machine tool, where some success has been reported in automating these types of problem.

The most difficult design problems are those in which designer has no prior experience with the problem and has to start from scratch. Little automation has been achieved in these cases. The more effective approaches do not try to automate the design task completely. Such approaches have attempted to develop tools for critiquing the design as it is being developed in order to guide the human designers in evaluating alternative design considerations. Research is already underway to develop design critiquing and advisory systems for different types of domain-dependent issues which need to be addressed in the design phase [8, 10, 13, 14, 24]. As the need arises for addressing additional downstream manufacturing concerns during the design phase, new analysis tools will be required to help designer in foreseeing potential manufacturing problems.

3 Multiple Critiquing Systems

In typical production environments manufacturing a product involves activities from many different manufacturing domains (machining, inspection, assembly, and so forth) each of which differs considerably from the other domains. For example, to manufacture the part shown in Figure 3, the shaft and pulley need to be machined and inspected for dimensional accuracy and then assembled. Each of these steps has different

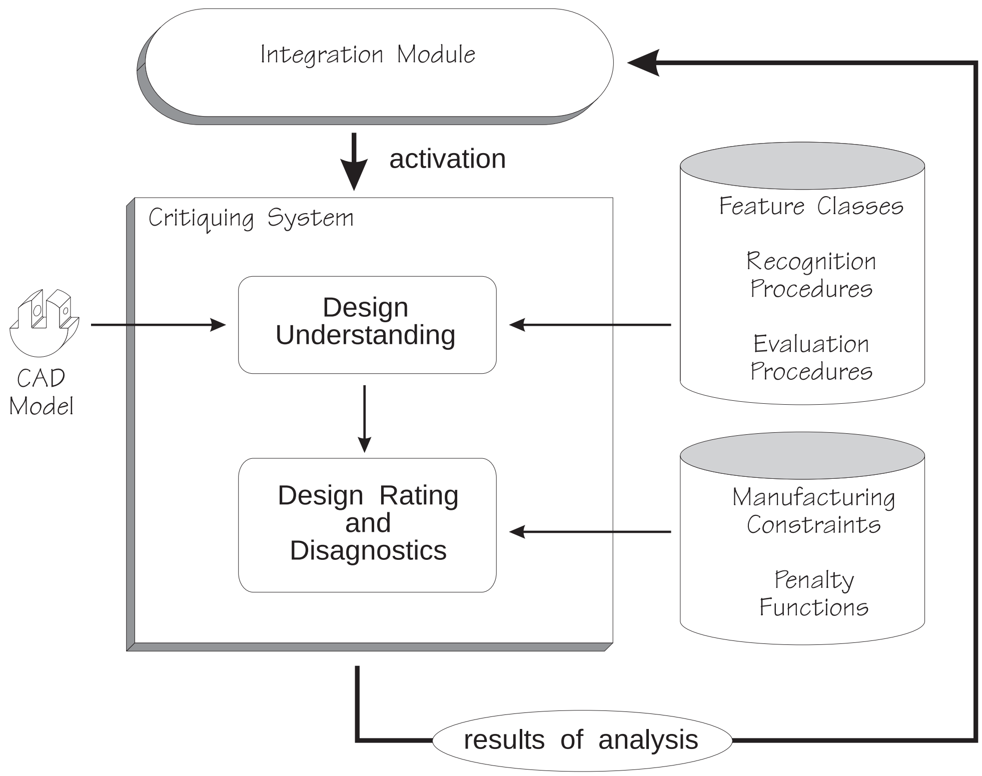


Figure 4: One of the multiple critiquing systems, and the framework within which it operates.

manufacturability considerations and it would be an imposing task to build a single critiquing system that could take all of these considerations into account. Therefore we propose the use of a separate critiquing system for each manufacturing domain. The activities of these multiple critiquing systems will be managed by an integration module, as described later in Section 6.

Figure 4 shows different components of a critiquing system along with the different knowledges base required for their operation. As shown, each critiquing system will have two modules: one for *design understanding* and another that performs *design rating and diagnostics*.

During *design understanding*, the critiquing system will attempt to interpret the design in terms of manufacturing operations in its domain. Usually the design is available as a CAD model (in some cases this may include a description in terms of design features). This step requires coming up with a domain-specific representation of the design, using manufacturing features appropriate to the domain at hand. Much previous work has been done on design interpretation for machining [4, 15, 17, 18, 26]. A presentation of a generic framework for achieving design understanding for a multiple critiquing environment is beyond the scope of this paper.

The other key component of a generic critiquing system, and the focus of the remainder of this paper, is the facility for *design rating and diagnostics*. After generating a domain-specific feature-based representation of the design, the next step is to analyze its suitability for the specific manufacturing domain. Each critiquing system will need to maintain a knowledge base of constraints specific to the activities of the

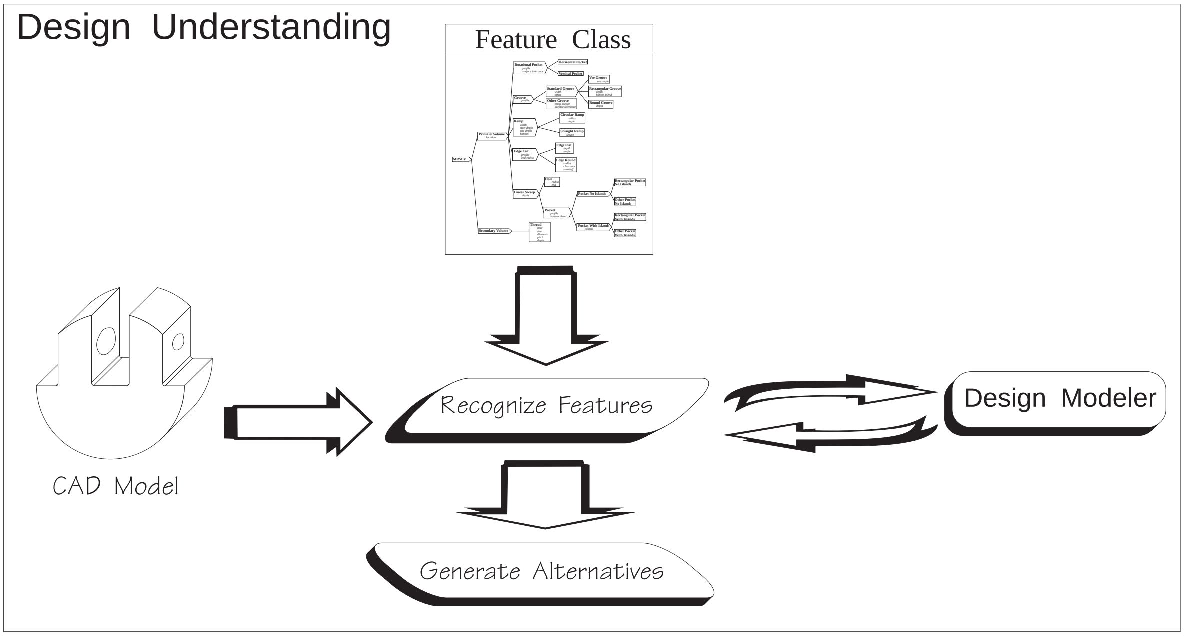


Figure 5: Understanding a design.

manufacturing domain. These constraints will represent common manufacturing practices and resource considerations. Using these constraints, the system will rate the design based on the estimated time and cost of the manufacturing operations needed in that domain. In this step it also must be verified that design can be created to the desired specifications, i.e., that the various dimensional and tolerance specifications will be satisfied by the design. If there are no manufacturing operations capable of creating certain design attributes (for example, there might be no machining process that can achieve a desired surface finish) then it will be necessary to diagnose the manufacturability problems.

Generating intelligent feedback and achieving realistic design ratings are vital components for a critiquing system. In many early rule-based DFM systems, abstract measures were used to rate designs for specific manufacturing domains. In the presence of multiple critiquing systems, the design may get a good rating for one manufacturing domain and a bad rating for another (for example, a proposed design may be easy to machine but difficult to assemble). Unless these ratings are done on a comparable scale, it will be very difficult to achieve consensus among these ratings.

Since all manufacturing operations have measurable time and cost, to maintain consistency we will perform design evaluations in terms of manufacturing time and cost. These time and cost figures can easily be combined into a overall rating. Moreover, they present a realistic view of the difficulty in manufacturing a proposed design. This data will also help the designer to make high level make-or-buy decisions.

4 Design Understanding

To critique a design with respect to a particular manufacturing domain, our first step is to obtain a domain-specific description of the design using manufacturing features; the features employed may be different for different manufacturing domains. In some cases the design will be available as a CAD model along with other technological information. In other cases designers will design parts in terms of high-level manufacturing and design features. In either situation our approach is to automatically recognize a collection of *feature instances* from the design attributes in order to analyze the design with respect to the manufacturing domain at hand.

In general there may be several alternative representations of the design as different collections of features from a particular feature class, each corresponding to a different way in which to view the part with respect to the manufacturing domain at hand. These alternatives can be generated from the set of feature instances and evaluated to determine which is optimal, as illustrated in Fig. 5. Which of these alternatives is most preferable will depend on the part's dimensions, tolerances, the availability and capabilities of manufacturing operations, and the particular kind of analysis needed by each critiquing system.

For each manufacturing domain the corresponding feature class will encapsulate the domain's particular information and functionality. Within the interface module, a generic feature recognition procedure, employing the information from the feature class for that domain, produces a description of the current design for evaluation and analysis. The information used to construct instances of features will be obtained by identifying parameterizations of the attributes that define the features, based on design information as represented in a CAD system.

Representing features and design information has been addressed by both the academic and standardization communities. Standards groups have been evolving a means for describing generic classes of features for the purposes of data exchange. STEP is the International *St*andard for the *E*xchange of *P*roduct Model Data being developed by the International Organization for Standardization (ISO). PDES (*P*roduct *D*ata *E*xchange using *S*TEP) represents the activity of corporate, government, and standards development entities in United States in support of STEP.

Within the PDES/STEP Integrated Product Information Model, one can represent different features, such as those relevant to tolerancing and process plans, and relationships between them. At present, the standard is still evolving and there is no definitive structure for representing and exchanging all the relevant information for the many manufacturing domains for which one would desire a critiquing system. A discussion of the STEP Form Features model can be found in [22, 23].

In each manufacturing domain, there may be more than one way of realizing the design attributes. Correspondingly, there may be more than one way to describe the design in terms of the features available in each manufacturing domain. For example, it may be possible to machine a given design in several different ways, each corresponding to a different collection of machining features. Each such collection of features is an example of a feature model. The term "feature model" has come to mean different things to different researchers [7, 17, 21]. In our work, a *feature model* is a single, domain-specific representation of a design. Informally, a feature model is a collection of feature instances that provide a description of a design with respect to some manufacturing domain.

The output of the design understanding system is a *feature set*, —a finite set of feature instances, each of which belongs to at least one (of the possibly many) feature models for the design. After finding the feature set, the next step is to use these features to generate feature models for the design. For example, Figure 6 illustrates the concepts of feature set and feature model in the machining domain. To perform design critiquing it is crucial to consider alternative feature models—for otherwise the model that provides the optimal manufacturing cost or time may be missed. However, we do not want to have to generate every possible feature model for a given design, for there may be exponentially many of these.

In the machining domain, we have used this methodology to build a proof-of-concept implementation of the system for feature recognition and generation of alternatives for a subset of the class of machinable features definable as MRSEVs (a STEP-based feature library developed by Kramer [16]). More specific details on the implementation can be found in [9, 21]. To avoid generating every possible feature model, we use a branch-and-bound approach that estimates the design ratings for feature models while they are being generated and uses this information to prune unpromising feature models from the search space.

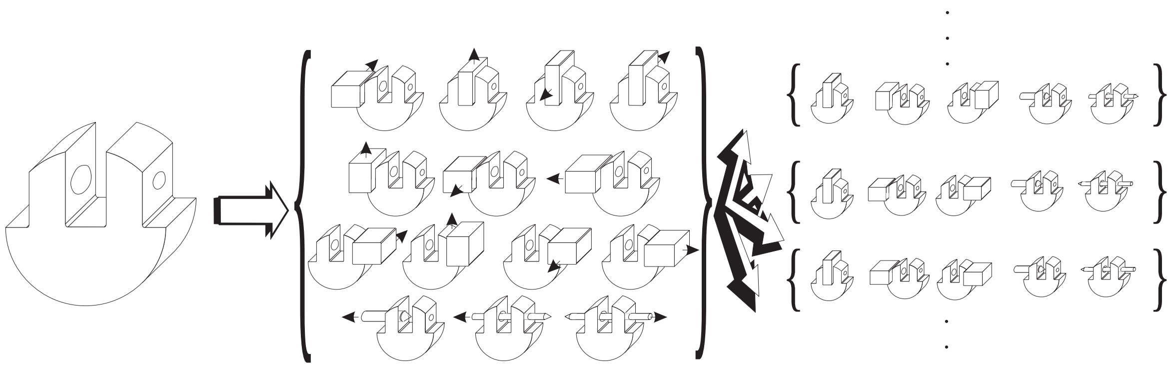


Figure 6: A design with alternate interpretations.

5 Design Rating and Diagnostics

Each critiquing system evaluates and analyzes the suitability of the proposed design for some specific manufacturing domain. A design is considered to be good for a manufacturing domain if its operation in that domain can be performed efficiently and produce the design to desired specifications.

The various features in a feature model represent manufacturing operations needed to achieve the realization of the design. In many manufacturing domains the feature model describing a design is not unique, implying that there may be many different ways to carry out the necessary manufacturing operations. For example, a given design can be machined in more than one way. Hence, if there is more than one model for the design all the models must be examined in order to rate the design based on the evaluation of the best model. Analyses performed by the critiquing system include checking common manufacturing constraints, rating the proposed design, and identifying design attributes that pose problems.

5.1 Manufacturing Constraints

Each specific domain imposes its own distinct manufacturing constraints on its available operations. Individual critiquing systems operate with their own set of manufacturing constraints, using them to evaluate the design.

One example of a manufacturing constraint is a *precedence constraint*, which specifies that one manufacturing operation must be performed before another one (for example, the “tee” portion of a T-slot cannot be machined until the stem of the slot has been machined). Conditions that introduce precedence constraints can be stored as manufacturing constraints for the particular domain; whether these conditions are applicable will often depend on relationships among the feature instances in the feature model.

Furthermore, manufacturing resources and the physical limitations of manufacturing processes can also be modeled as constraints [3]. They can be classified into two categories:

1. *Strict Constraints:* If any strict constraint is violated the proposed design cannot be manufactured. An example of these constraint includes restriction on the various tolerances: excessively tight tolerances cannot be achieved.

2. *Loose Constraints:* A loose constraint can be violated but the violation is detrimental to the manufacturability rating. Each loose constraint is associated with a *penalty function*. When a violation is

identified, the parameters are passed to this penalty function and it determines how much to lower the manufacturability rating. For example, if the wall thickness of a pocket falls below a certain critical value machining problems might occur that require more expensive tooling or more elaborate equipment. In this case, the penalty function would increase the cost and time estimates for the machining operation.

5.2 Measures of Manufacturability

The manufacturability of a given design depends on the following three factors:

1. the ability to produce the design within the specified specifications;
2. the ability to produce the design with a low production cost;
3. the ability to produce the design with a low production time.

When there are multiple critiquing systems, the manufacturability measures output by the various critiquing systems will be combined into an overall rating. Therefore, rather then using an abstract or qualitative measure, the manufacturability rating is calculated based on the manufacturing time and manufacturing cost. Manufacturing time and cost are universal measures which can be estimated for manufacturing operations across many different manufacturing domains. Extensive research has been done to estimate the time and cost associated with various kinds of manufacturing operations [27].

To compute the manufacturability rating, any combination of manufacturing cost and time can be used. The relative weights assigned to cost and time would depend upon the policy of the company using such a design critiquing system.

For each critiquing system, we first examine whether a given feature model has high probability of producing the desired design attributes. Only if a feature model can produce the design to suit the requirements does the system calculate a manufacturability rating for the feature model.

5.3 Critiquing Procedure

In our proposed approach various feature models will be produced in a generate-and-test loop. Each time a new model is generated, it will be analyzed using heuristic techniques to discard unpromising feature models that are not expected to result in a better manufacturability rating than the best one seen so far. To analyze a feature model, each critiquing system could use the steps shown in Figure 7. If there is more than one feature model for the design these steps could be carried out on several alternative feature models to determine which one is best. Below these steps are described in more detail:

1. *Examine precedence constraints.* The first step is to determine precedence constraints on the order in which the features are to be created during manufacturing. In the case of machining, accessibility [10], tolerance-datum dependencies, setup [12] and other types of interactions among features in the feature model will introduce precedence constraints requiring that some of them be machined before or after other features in the model. Our previous work on identifying precedence constraints for machining operations can be found in [10, 11].

   If these constraints are not cyclic, then there are one or more possible orders in which the features can be manufactured. However, if some of the precedence constraints form a cycle, then there is no way that the features can be created. For example in the case of machining, if two holes intersect then it might be necessary to create each of them before the other, which of course is impossible.

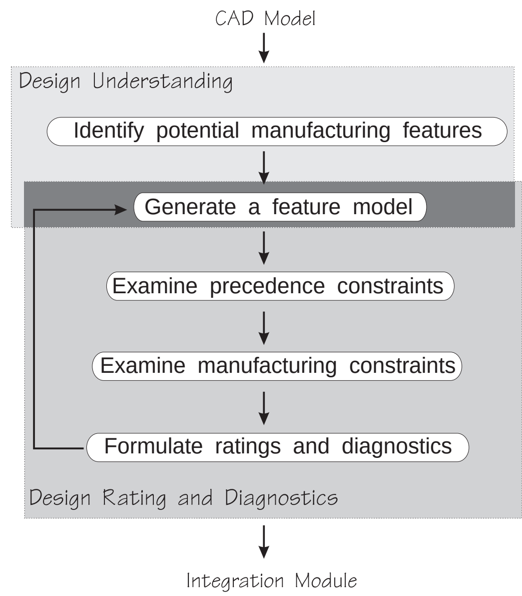


Figure 7: Critiquing Procedure

2. *Examine other manufacturing constraints.* The next step is to determine various parameters of the manufacturing operations associated with various features in the models. Use these parameters to determine what design attributes are achievable from the given feature model and precedence constraints and whether they satisfy the design requirements.

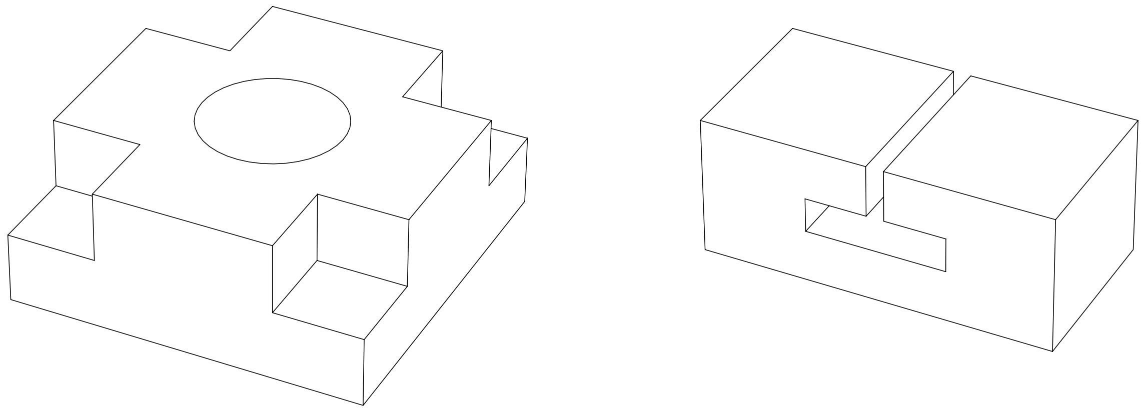
   For example, in the machining domain, each operation creates surfaces on a part which have geometric variations compared to their nominal geometry. Designers normally assign tolerance specifications to the nominal geometry in order to specify how large these variations are allowed to be. By calculating cutting speeds, feed rates, depths of cut, and other machining parameters, one can estimate whether it is possible to machine each feature in such a manner as to achieve the tolerance specifications. We have addressed these issues in our previous work on tolerance estimation in the machining domain [10, 19].

3. *Formulate Ratings and Diagnostics.* For this step, there are two possibilities:

   (a) If problems were found during the above steps, then perform diagnostics to determine which design attributes are responsible for the problems. If possible, suggest modifications to the design that will eliminate these problems.

   For example, in Figure 8(a), none of the notches can be machined because they have no corner radii; and in Figure 8(b) the T-slot cannot be machined because the access slot is too narrow. In these examples, the most obvious modifications would be to change the shape or dimensions of the features in question. However for some cases it is possible that adding additional features might improve the design's manufacturability rating; an example would be a request to add positioning surfaces for use in assembly operations. In general there may be several modifications that could be made to improve the manufacturability of the design; which modification is preferable would depend on the purpose of the design.

   In suggesting modifications to the design, the functional requirements for the design can be used to discard suggestions that are inconsistent with the design's intended functionality. The



(a) No corner radius (b) Narrow access slot

Figure 8: Design posing problem for machining.

remaining suggestions can be ranked by how effective they are in improving the design's manufacturability rating. For example, we use both of these techniques in the approach we are developing for automatically generating design changes to reduce the setup cost for machined parts [6]. In most cases, there will be a limited number of possible design changes that improve the manufacturability while maintaining consistency with the design's functional specifications—and these possibilities can be returned to the integration module.

(b) If no problems were found during the above steps, then estimate the manufacturing cost and time for each feature in the feature model and calculate a manufacturability rating for the feature model as a whole using the estimates for the individual features. This analysis is used to identify any *problem features* that might be bottlenecks or expensive to manufacture.

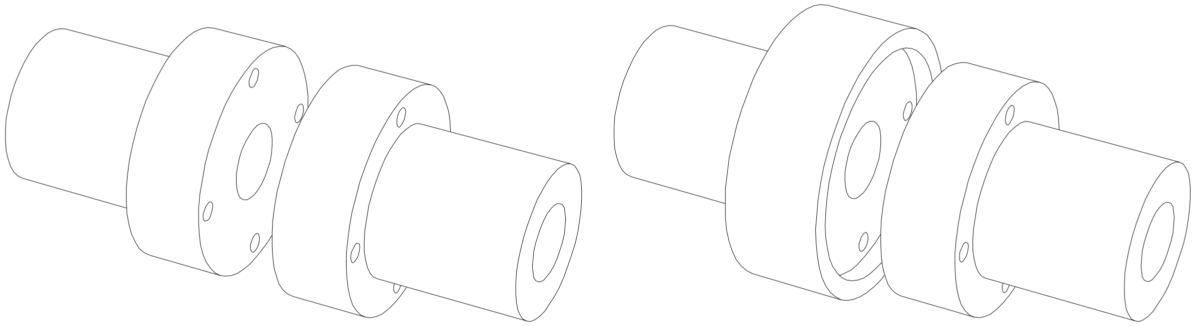
Next, for each of the problem features, calculate the sensitivity of the time and cost ratings to changes in the design attributes to be created by that feature. This information is used to help the designer decide whether it is worthwhile to change the design to improve its manufacturability rating. Design changes involve manipulating the parameters for the design attributes, modifying the problem features to satisfy these new attributes, and estimating the resulting changes in the time and cost of manufacturing the product.

6 Integration Module

6.1 Requirements

In the approach we are proposing in this paper, each critiquing system will evaluate the design from the point of view of its respective manufacturing domain. However a design that good for one manufacturing domain may be bad for some other. For example, Figure 9(a) shows a design that is inexpensive to machine but difficult to assemble, and Figure 9(b) shows a design that is expensive to machine but easy to assemble. Thus, the evaluations of different critiquing systems may be in conflict as well as whatever recommendations these systems make for changes to the design.

The above situation is similar to the situation in which a designer takes advice from several people and receives a conflicting set of opinions. To handle such situations, we will need a module for integrating the



(a): Easy to machine (b): Easy to assemble

Figure 9: An example of conflicting requirements.

operation of the various critiquing systems. This module will need to mediate the recommendations of the different critiquing systems and find a compromise which improves the overall manufacturability of the design. In addition, the following communication activities are a vital aspect of this integration module:

1. *Communication with Designer:* The system should be able to communicate with the designer effectively, letting the designer know about potential problems with the proposed design and presenting ways of improving it. When interacting with the designer, the system should be able to explain the reasoning behind its redesign suggestions. This improves the credibility of the system in the eyes of a designer and can train designers to design for manufacturability.

2. *Communication with CAD Systems:* If the designer wants to try out any of the proposed changes, the system should be capable of automatically generating the instructions to the CAD system that update the design.

3. *Communication with Manufacturing Engineers:* Any design critiquing system for manufacturability analysis will rely on plan-specific manufacturing information. For most companies manufacturing resources are dynamic in nature—availability of tools, personnel, and materials are constantly changing. Therefore, effective communication interfaces must be established to let the manufacturing engineers update the knowledge base of these systems.

6.2 Proposed Approach

Each critiquing system may operate at a different level of design detail. During the design process, the designer may wish to use only those critiquing systems appropriate for the current level of design detail. An integration module should allow the designer to the appropriate critiquing systems to use.

Figure 10 shows our proposed integration module. This module will coordinate the activities of the critiquing systems through the following series of steps:

1. *Activate the critiquing systems.* First, activate the appropriate critiquing systems in a pre-defined sequence. Each critiquing system will analyze the design and return the following information:

   (a) A list of non-manufacturable design attributes.

   (b) A list of the most costly attributes of the design along with a sensitivity rating each (i.e., how much the manufacturability rating will be changed by changes in each of these attributes).

   (c) A list of proposed modifications to design attributes to improve the manufacturability rating.

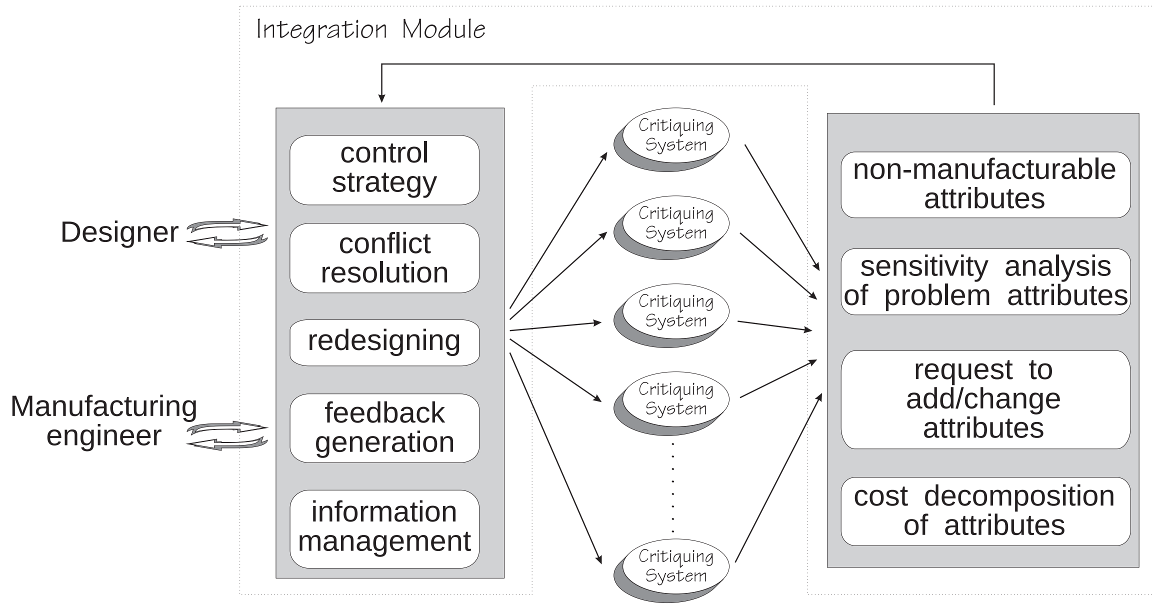


Figure 10: Integrating the operation of multiple critiquing systems.

(d) A detailed manufacturing time and cost decomposition for each design attribute.

2. *Handle non-manufacturable design attributes.* If there are conditions that make the design unmanufacturable, present them directly to the designer along with the manufacturing constraints they violate. The designer can then modify them to enable them to be manufactured.

3. *Compute the overall manufacturability rating:* Based on the time and cost estimations performed by the various critiquing systems, compute an overall manufacturability rating. This rating reflects the time and cost of the complete set of manufacturing efforts needed to produce the design.

4. *Handle expensive design attributes:* If possible, modify design attributes that are expensive to manufacture in order to improve the manufacturability rating. If two different critiquing systems require conflicting changes to a design attribute, then the attribute's sensitivity ratings in both of these domains should be used to determine how to make a change that best improves the overall manufacturability rating. It may be necessary, in this case, to presented several possible modifications to the designer as alternatives.

5. *Handle requests to add or change design attributes:* Use the results of the previous step to create a modified version of the design incorporating the changes requested by the critiquing systems. Use the critiquing systems to evaluate the modified design and compute its overall manufacturability rating. If the modified design has a better rating than the unmodified design, advise the designer to incorporate the changes into the design.

7 Discussion and Conclusions

As concurrent engineering requires more and more downstream manufacturing issues to be addressed during the design phase, we anticipate a growth in the number of design critiquing and advisory systems. As designers begin to use multiple critiquing tools, we anticipate problems in coordinating these tools. Since different critiquing tools are written to address different manufacturing objectives, the recommendations given by these tools will sometimes conflict with each other. Thus it will be necessary to develop ways to reconcile these conflicts objectives so as to avoid giving the designer confusing and contradictory advice.

In this paper, we have outlined an approach for integrating the operation of multiple critiquing tools, balancing their individual recommendations to to provide feedback to the designer in an integrated and consistent manner. We anticipate that such a system could considerably reduce the need for redesign iterations, resulting in reduced lead time and product cost.

Acknowledgements

This work was supported in part by NSF Grants NSFD CDR-88003012, IRI9306580, and DDM-9201779.

References

[1] R. Bakerjian, editor. *Design for Manufacturability*, volume 6 of *Tool and Manufacturing Enginners Handbook*. Society of Manufacturing Engineers, 1992.

[2] T. Bardasz and I. Zeid. Cognitive models of memory for mechanical design problems. *Computer Aided Design*, 24(6):327–342, June 1992.

[3] Tien-Chien Chang. *Expert Process Planning for Manufacturing*. Addison-Wesley Publishing Co., 1990.

[4] S. H. Chuang and M. R. Henderson. Three-dimensional shape pattern recognition using vertex classification and the vertex-edge graph. *Computer Aided Design*, 22(6):377–387, June 1990.

[5] M.R. Cutkosky, R.S. Engelmore, R.E. Fikes, M.R. Genesereth, T.R. Gruber, W.S. Mark, J.M. Tenenbaum, and J.C. Weber. PACT: An experiment in integrating concurrent engineering systems. *IEEE Computer*, 26(1):28–37, January 1993.

[6] D. Das, S.K. Gupta, and D.S. Nau. Reducing setup cost by automated generation of redesign suggestions. In *ASME Computers in Engineering Conference*, 1994. To appear.

[7] Bianca Falcidieno and Franca Giannini. Automatic recognition and representation of shape-based features in a geometric modeling system. *Computer Vision, Graphics, and Image Processing*, 48:93–123, 1989.

[8] R. Gadh, E.L. Gursoz, M.A. Hall, F.B. Prinz, and A.M. Sudhalkar. Feature abstraction in a knowledge-based critique of design. *Manufacturing Review*, 4(2):115–125, 1991.

[9] Satyandra K. Gupta, Thomas R. Kramer, Dana S. Nau, William C. Regli, and Guangming Zhang. Building MRSEV models for CAM applications. Technical Report ISR TR93-84, The University of Maryland, December 1993. Accepted for publication in *Advances in Engineering Software*.

[10] S.K. Gupta and D.S. Nau. A systematic approach for analyzing the manufacturability of machined parts. *Computer Aided Design*, 1994. To appear.

[11] 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 J. Shah, Martti Mäntylä, and Dana S. Nau, editors, *Advances in Feature Based Manufacturing*, pages 161–184. Elsevier Science Publishers, 1994.

[12] C. C. Hayes and P. Wright. Automatic process planning: using feature interaction to guide search. *Journal of Manufacturing Systems*, 8(1):1–15, 1989.

[13] W. Hsu, C.S.G. Lee, and S. F. Su. Feedback approach to design for assembly by evaluation of assembly plan. *Computer Aided Design*, 25(7):395–409, July 1993.

[14] M. J. Jakiela and P. Y. Papalambros. Concurrent engineering with suggestion-making cad systems: Result of initial users tests. In *ASME Design Automation Conference*, volume DE-Vol. 66, Montreal, Canada, September 1989.

[15] S. Joshi and T. C. Chang. Graph-based heurisitcs for recognition of machined features from a 3D solid model. *Computer-Aided Design*, 20(2):58–66, March 1988.

[16] Thomas R. Kramer. A library of material removal shape element volumes (MRSEVs). Technical Report NISTIR 4809, The National Institute of Standards and Technology, Gaithersburg, MD 20899, March 1992.

[17] Timo Laakko and Martti Mäntylä. Feature modelling by incremental feature recognition. *Computer Aided Design*, 25(8):479–492, August 1993.

[18] M. Marefat and R. L. Kashyap. Geometric reasoning for recognition of three-dimensional object features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):949–965, October 1990.

[19] 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.

[20] N.P.Suh. Design axiom and quality control. *Robotics and Computer Integrated Manufacturing*, 9(4/5):367–376, 1992.

[21] William C. Regli and Dana S. Nau. Recognition of volumetric features from CAD models: Problem formalization and algorithms. Technical Report TR 93-41, The University of Maryland, Institute for Systems Research, College Park, MD 20742, USA, April 1993.

[22] Jami J. Shah. Assessment of features technology. *Computer Aided Design*, 23(5):331–343, June 1991.

[23] Jami J. Shah. Experimental investigation of the STEP form-feature information model. *Computer Aided Design*, 23(4):282–296, May 1991.

[24] B.G. Silverman and T.M Mezher. Expert critics in engineering design: Lessons learned and research needs. *AI magazine*, 13(1):45–62, 1992.

[25] R. H. Sturges, K. O'Shaughnessy, and R. G. Reed. A systematic approach to conceptual design. *Concurrent Engineering: Research And Applications*, 1(2):93–105, 1993.

[26] J. H. Vandenbrande and A. A. G. Requicha. Spatial reasoning for the automatic recognition of machinable features in solid models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(12):1269, December 1993.

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