Design Representation for Manufacturability Evaluation in CAD: Beyond Feature-based Design

K. Ishii
R.A. Miller
Engineering Research Center
College of Engineering
The Ohio State University
Columbus, OH

ABSTRACT

This paper addresses CAD representation suitable for intelligent programs that evaluate life-cycle concerns. Design for net shape manufacturing serves as a vehicle for our discussion. We argue that feature-based design alone could lead to an unmanageable number of predefined features and tagged data in order to accommodate the different perspectives of the design engineer, tooling engineer, and process engineer. The paper proposes a compromise between full scope feature-based design and pure geometry as the recommended form of design representation. This mode requires methods or formulae to infer information needed for evaluation. There are two types of methods important in manufacturability evaluation: 1) macroscopic search of problems in the entire design, and 2) microscopic evaluation of a local form feature. We illustrate our ideas with two working systems.

1. BACKGROUND OF DISCUSSION

CAD technology has made a significant impact on mechanical design. In particular, solids modeling technology (Requicha, 1980) allowed engineers to design products much faster. Engineers could evaluate the esthetics, performance, and even manufacturability of the component by looking at the realistic image of the candidate designs. The trend today is to link this powerful technology with other computer methods such as finite element methods and AI-based reasoning (Ishii, 1990), which contribute to life-cycle engineering (Suh, 1988).

For computers to deduce information associated with life-cycle values, the CAD model must contain pertinent data. A method considered promising by many is feature-based design (Luby et al, 1986; Cunningham and Dixon, 1988; Shah and Rogers, 1988; Wilson and Pratt, 1988). Cunningham and Dixon define feature as "any geometric form or entity that is used in reasoning in one or more design and manufacturing activities." Designers are encouraged to use elements from a library of features to construct their geometry model. In essence, feature-based design allows the user to "tag" relevant data into the CAD data-base which can later be used to conduct certain analysis or evaluation. Designers experience some problems constructing a product model from features alone and must use "non-feature" elements to complete the design.

On the other end of the spectrum is feature extraction (Woo, 1982; Henderson and Chang; 1988). This approach attempts to look at a conventional CAD data based such as B-rep or CSG and extract features that are of interest to certain analyses. Hence, the designers are free to utilize all the capabilities of a geometry modeler and let the extraction program worry about deducing necessary data from the model. The drawback is the complexity, incompleteness, and computational load of such an extraction program. Geometry modelers that accommodate topology information in the form of graphs, etc. (Joshi and Chang, 1988; Pinnella, et al, 1989) makes the task reasonable. However, this approach also has limitations in computational load as the model becomes complex.

This paper addresses the level of data CAD models should carry regarding life-cycle issues. Designing with features offers many advantages in making pertinent data readily available by other analysis programs. However, when the user wishes to analyze a variety of attributes about the design, the amount of data carried by the CAD model rapidly grows. This is due to the different set of parameters demanded by each analysis program which may lead to different sets of features to describe a single model. Creating different features for each life-cycle issue and tagging enormous amount of data to CAD models is obviously not the desired solution.

On the other hand, we do not agree with leaving the CAD models purely geometrical or topological and relying on a complex extraction / recognition program to infer necessary data. There has to be a happy medium in this spectrum that relates geometry information with other relevant parameters.

This paper looks at the most appropriate level of mix between designing with form features and recognition / extraction programs. Our major interest is on manufacturability evaluation of discrete parts with emphasis on net shape processes. We define net shape manufacturing as processes that use molds or dies to transform material which is shapeless, or of simple shape, into a near finished part. Manufacturability evaluation is important in the following aspects of design.

a) **Process Selection:** Starting from the functional requirements for the part, select the most appropriate manufacturing process.
b) **Design for Manufacture:** Having selected a candidate manufacturing process, the designer wants to tailor the geometry so that it is compatible with the process.

Design for net shape processes offers an excellent vehicle for investigation of our topic, i.e., the incorporation of manufacturing information into CAD models, since there are several levels of expertise involved. Design engineers usually address the functionality of a part. Tooling engineers must design dies or molds for the part economically, hence they are interested in the complexity of the tools that results from the part design. Process engineers are interested in transforming the shapeless material in the die or mold to the specified dimension in minimum time. Naturally, these experts evaluate the candidate design from different points of view.

A concurrent engineering approach to product development introduces at least one additional problem as well. In many instances several processes may be considered to fabricate a part for a given design concept. For example, die casting, injection molding, and sheet metal fabrication could all be competitive processes for many types of parts. From a CAD point of view, however, the problem is that the detailed geometry is generally different for each process. That is, even though the conceptual design is the same, the final part geometries are different for each process due to differences in reinforcement, provisions for fasteners, and process induced part cross-sectional restrictions. In particular, this paper uses die casting and injection molding to develop our discussion. We first state the fundamental issues of our discussion. Then, we describe our proposed approach based on information kept by the traditional blue prints and by inference used by engineers who read the prints. The paper also gives examples of our research efforts based on the proposed approach. We present two prototype computer programs that incorporate our philosophy and describe how they impact engineers in industry.

2. INTEGRATING LIFE-CYCLE INFORMATION

2.1 Combinatorial explosion of features

Feature-based design could result in an excessively large database if we attempt to embed all the data necessary for life-cycle evaluation. The explosion of data could come in the form of number of features, amount of data tagged to each feature, or need for cross referencing parameters of multiple interacting features. This issue is particularly pertinent in net shape manufacturing. If we focus on die casting and injection molding, we would have at least three experts involved in evaluating a candidate design for manufacturability: design engineer, tooling engineer, and process engineer. They view the candidate design from different angles.

a) Design engineers primarily look at functions: Ribs to stiffen a part, handles to allow manipulation of the part, fasteners to assemble the part with another, etc.
b) Tooling engineers are concerned mainly with the complexity and fabrication of the dies or molds.
c) Process engineers focus on material flow and cooling of the part in the mold.

These experts focus on different aspects of the candidate design, decompose it differently, and require different parametrization for design evaluation. Let us examine the reasons for data explosion with features.

a) Number of features

Typical feature-based design environments define feature libraries tailored for shape synthesis. As a result, most of the features are geometry-based and often parametrized to support evaluation of functionality. However, manufacturing experts may require a completely different set of parameters for evaluation. Hence, a “feature” tailored for die casting evaluation may be different from that customized for machining. For example, the manufacturability rule of a “rib” in a part will be very different for sheet metal and plastic injection molding. If we were to make a general feature-based environment, we may need to define different features for a “rib” manufactured by different processes. This approach will naturally lead to combinatorial explosion of number of features. Further, some manufacturability rules may involve relationships between multiple form features. In such cases, one approach is to define another class of features that relate all the necessary geometry information and tag data needed to check the relevant manufacturability constraints.

b) Amount of tagged data

If we assume that the number of features is optimally determined, we still have the problem of the amount of data tagged to each feature. The fact that experts view each feature differently remains. Designers look at a hole as an empty element that provides access, ventilation, or assembly of another part. Tooling engineers look them not as a hole but a need for a core pin which is a positive element that must be fabricated into or attached to the base of the mold. They must also worry about the forces subjected to the core pins by the melt. Process engineers see the hole as a flow obstruction. These experts demand different parameter sets to evaluate the compatibility of this particular feature. As mentioned above, some rules demand data from other nearby interacting features. For example, process engineers will need to know the existence of a neighboring hole or the distance to the side wall to determine the flow channel. One needs to tag a vast amount of data to conduct adequate evaluation for manufacturability.

2.2 “Base feature” problem

One key attribute of most near net shape processes, processes such as forging, casting, and injection molding, is the potential for substantial geometric complexity. They can have many sculptured surfaces which have added complexity in the form of ribs, bosses, and gussets. Similar statements are true of injection molded parts. Reinforcement of sheet metal parts and die cast parts intended for the same purpose are dramatically different due to constraints that the process imposes. Figure 1 shows a typical die cast part with substantial complexity.

Many of the manufacturability assessments require reasoning about the geometry of the part and checking compatibility with the process. For example, wall thickness must be checked. They can neither be too thick or too thin for the given material. Hole diameters and depths must be checked to insure that the core pins that will form them will not deform and will release. Clearly, all portions of the part must be checked for compliance with good design practice for the process. This means that the computerized evaluation systems must have the ability to 'look at' all regions of the part and identify significant items of data.

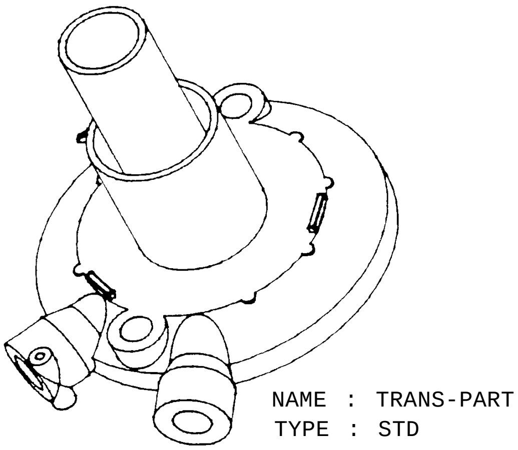


Figure 1. Die cast transmission end cover.

The majority of the work intended to provide aids for manufacturability assessment has focused on machining, particularly operations planning for machining (Cutkosky, et al, 1989; Bahns, et al, 1990). In operations planning the objective is to develop a clamping or fixturing strategy and a sequence of machining operations which will transform the stock material into the desired final shape. If this problem is viewed as a manufacturability assessment, the issue is whether or not a practical machining sequence exists for the set of tools which are available. It is no surprise that this problem is greatly facilitated by describing the part in terms of "features" which correspond in a fairly direct fashion to the machining operations which are possible. Holes are related to boring, drilling, or reaming operations depending on tolerances. Slots are related to milling operations and so on. Of particular interest for this discussion is the fact that it is assumed, at least for the automated operations planning systems developed so far, that the part is to be machined from a block or cylinder. The starting shape is always a simple geometric shape.

As described above, the manufacturability questions associated with near net shape processes are somewhat more complicated. There is no longer any basis for assuming a regular shaped starting point and a finite sequence of transformations to achieve the part. It is perhaps possible to define a small library of secondary form features, e.g., bosses, holes, ribs, webs, etc., and each feature can be defined by a small parameter set which provides sufficient data for evaluation (thickness, height, diameter etc.). Using this approach there is still the question of whether a round boss is the same feature as a hexagonal boss and if a boss located on a plane face is the same as a boss located on a non-planar surface. If the answers to these questions is no, the combinatorial explosion is even worse.

But, the secondary features alone are not enough. The “base feature,” the entity or entities to which these secondary features attach, must also be evaluated. It is much less obvious that a small finite library of simple shape features is practical for constructing parts with the range and variety of surface complexity that is possible in net shape manufacturing. In the absence of such a library, the manufacturability programs must posses significantly more reasoning capability in order to extract and construct the necessary data from representations which do not store it explicitly.

2.3 Feature Interaction

The feature interaction problem for manufacturability assessment arises because areas of concern on the part, and corresponding data, may be due to factors totally unrelated to the function of the part or the way the part model was constructed. This is particularly true if the shape of the part is being modified to make it more compatible with a new process. Consider figure 2 which depicts the cross section of a hypothetical two-piece component consisting of a circular bushing and a spacer. The bushing is composed of two features, a ring and a plug. The names of the features denote certain functional characteristics and were selected by the part designer prior to the point of considering the fabrication method, and the dimensional information provided is critical for the function of the part. If a die casting evaluation program is asked to evaluate this part, there are some immediate problems unless the program has sufficient reasoning capability. First, the feature names are almost certainly of no use because they do not correspond to the names used for die casting features. This means that the evaluation program, if driven only by features, would not have rules appropriate for “rings” and “plugs” and it would not know if these features correspond to other features which it does know about.

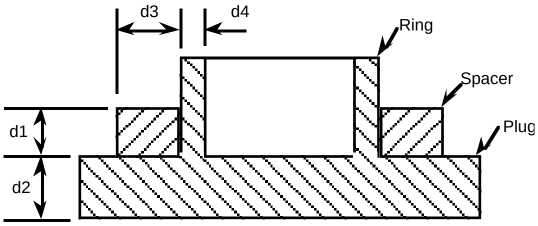


Figure 2. The base feature issue

The dimensional data absolutely critical for the proper functioning of the part, and therefore explicitly provided, namely d1, d2, d3, and d4, are only a part of the data that is needed for the casting evaluation. The thickness of the part is important in a die casting. If it is too thin, the material will not fill the cavity. If it is too thick, there will be problems created by the large thermal mass. Therefore, the evaluation program will require the total thickness, i.e., d1+d2 and d3+d4 as well as other data. These dimensions come about as an interaction of the ring, spacer and plug. None of these data items is an attribute (parameter) of a single feature alone which means that the necessary data would not be contained in the instantiation of the feature. The system must be smart enough to know that the interaction is taking place and to construct the appropriate data.

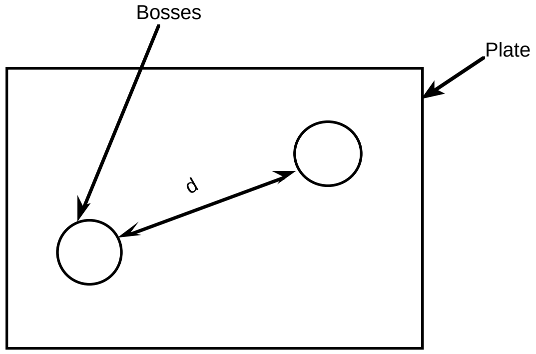


Figure 3 Feature interaction

A second, somewhat simpler example is figure 3 which shows the top view of two bosses on a flat plate. The distance between the bosses, d, is a dimension which does not exist in the absence of one or both of the features. Data instantiating this dimension would also not be found in any standard dimensioning of the part nor would it be explicitly represented in most part models. It would however be of interest when exploring flow and die life questions for die casting. Both of these examples illustrate cases in which the necessary data are not parameters of the part model but can be derived from commonly available data. These data cannot simply be retrieved from a part database but require computation.

The same examples also illustrate a potential problem with the capture of design intent. It is sometimes assumed that the utilization of features will provide a better characterization of what the designer had in mind when the part was created, and this is true to some degree. However, if multiple features are orchestrated to achieve a function, and no one feature achieves that function alone, we are still faced with the problem of capturing design intent. Considering figure 2, for instance, if the spacer and plug together fill a critical spacing or locating requirement it is the combined height of of the plug and spacer together, i.e. d1+d2, which are critical to the design. It would be this combined dimension which should be toleranced, not the individual dimensions. The combined dimension is not a form feature parameter. These and similar issues have been recognized in Dong and Wozny (1990).

These simple examples show that a practical manufacturing evaluation system cannot be organized around the concept of non-interacting features. Even if the features carry names corresponding to the manufacturing domain in question, critical dimensions arise because of interactions. An attempt to define beforehand all of the possible interactions which might occur presents another combinatorial explosion. Similarly, since the features logically should be defined by the designer for functional reasons, the features of the basic part model will not even correspond to the manufacturing features. Both of these problems suggest that it will be more fruitful to explore approaches which enable the manufacturing evaluation functions to better reason about the part and construct, rather than retrieve, the necessary data. Some preliminary results along these lines are presented in Chen et.al. (1991a &b).

3. INFERRING NECESSARY INFORMATION

3.1 Embedding data vs. inferring information

In its extreme form, the feature-based design approach requires every question, manufacturing or otherwise, to be packaged in a feature description. That is, designers have to tag every piece of information needed for manufacturability evaluation. Another way of describing this approach is that users design with packaged or predefined features. We need a very large number of features to cover all the functional needs and manufacturing interests. Each feature will require a large piece of computer memory if it is to contain data pertinent to all the design rules one wishes to consider. Note that as discussed earlier, this may even require different pieces to describe the same basic design but detailed for different processes. This approach offers advantages when targeting a certain class of products and evaluation criteria. However, when targeting a general purpose CAD system that accommodate linkages with intelligent programs that evaluate diverse life-cycle issues, they face the “data explosion” problem outlined in the previous section.

The other end of the spectrum is CAD with only pure geometry and topology information. Pure B-rep conveys geometry and the topology of the geometric elements (faces, edges, vertices) only and hence is quite compact. However, we need to apply sophisticated, sometimes complex and computational expensive methods to obtain the data needed for manufacturability evaluation. In some cases, it is impossible to carry out evaluation from pure geometry alone. We need semantic information, information related to tolerances, surface roughness, and the like. Some of these systems are quite powerful and robust. But once again, they tend to target specific class of objects and address a narrow scope of life-cycle interests. The more generic one tries to make these extraction programs, the more complex and computational expensive they become. Also, the fact that these methods make fairly specific assumptions about the CAD data structure limits the robustness of this approach.

Somewhere in between these extremes lie the conventional “blue prints” (Figure 4). Blue prints carry messages in a slightly different way from features. Blue prints carry a relatively small yet adequate "data" for a specified manufacturing process. In many cases, they accommodate "codes" that can be interpreted by different experts according to their needs. Experts "infer" relevant parameter sets from these "codes" and necessary information from geometry. These "codes" are the keys to these inference tasks, i.e., direct experts where to look and guide them to the parameters they need for evaluation. The form of these codes and the inference methods used by the experts are what we are after. The figure below illustrates the spectrum between feature-based design which aims to embed the information in the CAD database and pure geometry to which one has to apply sophisticated feature extraction program to infer necessary data.

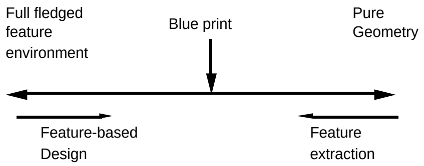


Figure 4. Embedded Data vs. Extracted Data

Blue prints often embed information that reflect, but not specifically describe, designers' intent. Dimensional specifications such as tolerances are good examples. In other words, the non-geometrical information prescribed in blue prints comprises historical pieces of the solution to embedding data in design models at the appropriate level. These solutions represent some form of standardized "codes" that do not specifically express facts, but contain data that can be interpreted by various experts who can then use some inference rules to deduce the information they need.

3.2 Data for manufacturability evaluation

Manufacturability assessment, in general terms, involves at least three distinct kinds of geometric data:

1) **Aggregate data:** This is data which in a rough sense provides an overall description of the part. The volume/surface area ratio is one example which provides a gross estimate of the average wall thickness of a part.
2) **Feature parameters:** This is the set of attributes associated with specific form features. The height, diameter and volume of a boss might be an example.
3) **Relational data:** The third type of data is the relationships among features or geometric elements.

These data can be obtained, with varying degrees of difficulty, from the part model. Some of them are explicitly stored and it is just a matter of data retrieval, others require a substantial amount of reasoning to infer the necessary data from other part attributes stored in the representation. These data provide the inputs for process specific reasoning which generally involves rules, algorithms and inference for addressing manufacturing specific questions. For example, parameters can be used to locate holes which would require long narrow core pins in the die or narrow grooves which would result in thin die steel sections. The specific meaning of 'long' and 'narrow' is material and process dependent, but for any given case, the use of such features on a die is almost a guarantee of high die maintenance cost and is a likely source of ejection and quality problems. Making judgments about flow obstructions and die erosion would involve locating any surfaces which fall directly in the flow path of the material. This entails reasoning about the relationships between the gate through which the material is injected and the surfaces of the die cavity and extracting relational data as well as tolerance information. Both parameters and relational data are required to locate and analyze potential hot spots, regions of high mass concentration which will result in shrinkage problems when the part cools.

The key to addressing the manufacturability questions, and possibly other questions not directly related to the original design synthesis, is the ability for the down stream processes to perform geometric reasoning on the part model. Features have often been proposed as a means of providing high-level, semantic information to enable this to occur. As described earlier, features alone are no guarantee that the necessary reasoning can occur at the intended level. Features provide an incomplete semantics for many cases of interest.

It is perhaps useful to ask why features, if matched to the process, provide this capability. The answer is simply that features ensure that the appropriate reasoning is brought to bear on the relevant portions of the part. Bosses are evaluated as bosses should be evaluated, ribs are evaluated as ribs. In this sense, features, by acting as tags, solve the basic perceptual problem of locating those regions on the part that are relevant to the task at hand. Since feature recognition is still far from fully developed for parts with the complexity of many net shape parts, in the absence of such cues, it is less than obvious that any evaluation system can find all of the features without human help.

For a variety of reasons then, features alone do not seem to be a particularly promising approach to providing better integration of design and manufacturing concerns. The explicit features that the designer wants to use don't match the features the manufacturing engineer wants to use and conversely. Why should a designer be required to construct a part design by 'carving' it out of a solid block just to make the machining problem easier? It is not clear that the designer knows the best way to machine it, and it may not be clear that machining is the most effective process to use. Furthermore, the above analysis and examples show that there still remain substantial geometric reasoning problems even if features of the typical sort are used as part of the part description. For these reasons, the more rationale approach would seem to be one which constructs the designs using features of particular relevance to the designer, but to enhance the part model with sufficient structural and parametric information so that down stream geometric reasoning can occur. In other words, we need to follow the 'blueprint paradigm" in which sufficient data are provided by the description of the part so that any additional data which may be necessary can be inferred by a competent actor with sufficient reasoning capability. The question is, what constitutes sufficient data.

A number of current research efforts are beginning to address some of these issues. Of particular interest are modelers such as NOODLES (Pinnella et al, 1989) which provide a much more complete representation of the topology of the part than traditional systems. Systems such as Anderson and Chang (1990) also are interesting. This system is specifically aimed at machining and process planning, but the concepts may well generalize. The TAGUS system developed at GE Corporate Research (Irani et. al. 1990) has some particularly useful characteristics for applications of the type described here. It provides a bridge between traditional modeling systems and other geometry dependent applications such as finite element modeling, design optimization etc. The system takes the geometry of the source system and reconstructs a part representation with suitable geometry and topology for the target application. Dong and Wozny (1990) and Rossignoc (1990) both address related problems and provide useful constructs. In general terms, the problem is related to the feature mapping problem (Shah, 1988, Shah and Rogers, 1988).

4. OUR RESEARCH EFFORTS

4.1 Macroscopic identification of local focus

The work on die casting focused on detailed design support which is viewed as providing detailed evaluation of the part geometry for castability and providing very high level functionality so that shape modifications can be quickly and easily made to improve the castability of the part.

The need for process specific detailed design support is based on the fact that the details of the geometry are often very process specific. The overall envelope of the part, the basic shape, will be defined by the fit and function requirements which require that parts assembled to form a component. The conceptual design will account for these requirements, but it will not, and possibly should not, commit to a specific process. This means that the design will have to be refined by adding specific geometric details which are required for the process. These details may include addition of reinforcements, modification of wall thicknesses, removing sharp corners, and adding drafts and fillets. These and other components of the geometry must be process specific. Rib patterns for cast parts and welded parts look very different, for example. The type of detailed design support provided by the die casting system helps a designer quickly transition from the partially completed conceptual design to a castable shape.

The current version of DIECAST (Liou and Miller, 1991), the system mentioned above, was constructed by integrating Pro/Engineer from Parametric Technology, a commercial CAD system, with a knowledge based system constructed in Nexpert/Object from Neuron Data. The overall architecture of the system is shown in Figure 5. The knowledge-based side contains the process specific knowledge which govern those aspects of the part geometry significant for evaluation, and it has the material specific rules which must be applied. The CAD side provides the geometric modeling capability. The integration routines provide access to geometric data and modeling functions. The knowledge based system carries out the reasoning using specific items of data obtained through queries to the modeler and through inferences made from such data. A part model with augmented topology is maintained by the knowledge based system to support the reasoning process.

The system can provide a general overall assessment of a part design and its suitability for die casting. This is done by determining the size envelop (length, width, height), the overall weight of the casting, and the nominal wall thickness computed from the volume to surface area ratio. These data, which are very easy to compute, are compared against standards for the material to determine if the design is even in the ball park. A number of more detailed assessments and design aids are also available.

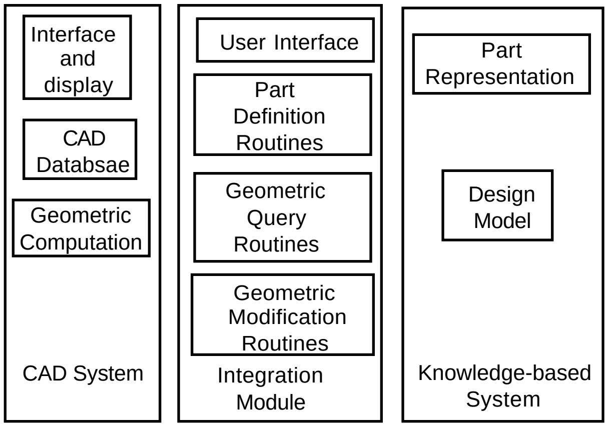
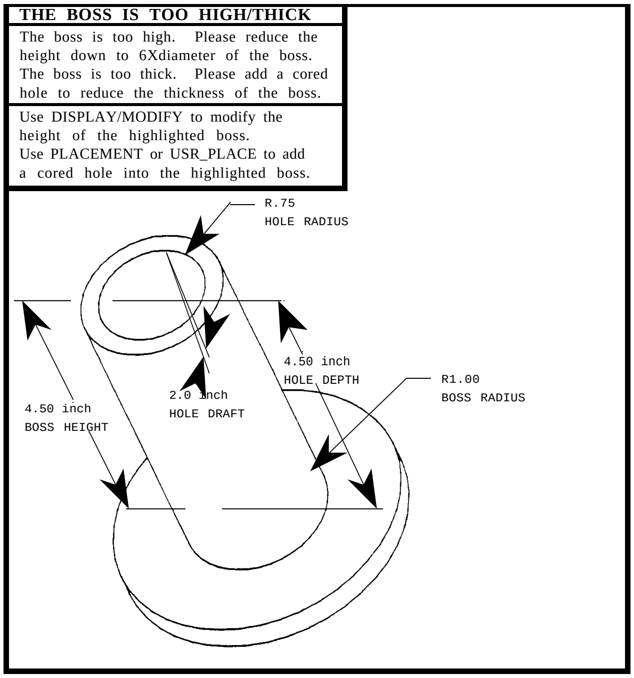


Figure 5. System architecture of DIECAST

One of the design aids provided helps to locate a parting line. Given a parting line and the die opening direction, the part is automatically evaluated for undercuts. The location of the surfaces causing undercuts is used to suggest alternate parting line locations where feasible, and an orientation with the minimum number of cores and slides can be found. With a parting line established, all surfaces on the part requiring draft are then located and the draft can be automatically applied using values from the knowledge base, or the user can specify custom values which are then automatically applied without further user intervention. Or, of course, the user can perform the function manually as in a traditional CAD system. Similarly, the edges of the part are checked for fillets and rounds as required by the material. If no round or fillet is found, or if the radius is not sufficient, modifications can be performed automatically or under user control.

The part representation used in this system is feature based in the sense that it is composed of 'user defined' features and a feature library which contains ribs, webs, bosses, slots, grooves, and cored holes. The construction in terms of features makes it possible to construct a more complete representation than is used by the CAD system alone, but since user defined features are allowed, there is no constraint on the complexity of the objects that can be designed. Since the knowledge based system has no explicit design rules for non-library features, these features cannot be evaluated in the same way as library features. However, none of the checks and modifications described above actually require features. These computations are performed using basic geometric and topological data which is found in the boundary representation of the part, independent of the feature. At present there are design rules in the knowledge base for each specific library feature. Further, there are certain local shape evaluations which can be performed on both library and user defined features as long as the user provides labels for selected feature dimensions. Most notable in this category is the evaluation of wall thickness. Note that the dimension label plays the same role for user designed features that the feature name plays for library features. This enables at least some evaluation of features whose general shape and form is specified beforehand. Figures 6 shows a typical evaluation and suggestions by DIECAST.



THE BOSS IS TOO HIGH/THICK

The boss is too high. Please reduce the height down to 6Xdiameter of the boss. The boss is too thick. Please add a cored hole to reduce the thickness of the boss.

Use DISPLAY/MODIFY to modify the height of the highlighted boss.
Use PLACEMENT or USR_PLACE to add a cored hole into the highlighted boss.

Figure 6. Evaluation and Suggestion Given by DIECAST

DIECAST in its current form does not support the degree of reasoning which is necessary. It does not support feature interaction type reasoning, and it is too dependent on the use of feature names and tags. To remove some of these deficiencies, a more abstract characterization of features is being implemented which will allow the design rules to be expressed for classes of geometries which are topologically equivalent. This together with explicit representation of the dependency relationships among features will enable much more significant reasoning to occur. Chen (1991a,b) provides some of the details of the approach and we plan to incorporate this method in our future systems.

4.2 Representation for "local" reasoning

Another research effort focuses on the representation of manufacturability guidelines or design rules that apply to a more general class of geometrical form rather than specific functional form features. Many conventional design rules require a predefined set of parameters that describe the geometry of the feature in question. Our aim is to redefine these rules in terms of geometry characteristics that can be inferred using relatively simple inference from various CAD data structures. A good example of such design rules is the identification of "hot spots" in injection molding that dictate the cycle time and possibility of undesirable sink marks. Most injection molding handbooks provide distinct sinkmark prevention rules for each form feature that are commonly used: ribs, rib intersections, bosses, gussets, snaps, and other thickness variations.

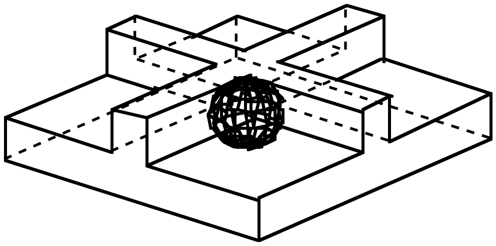
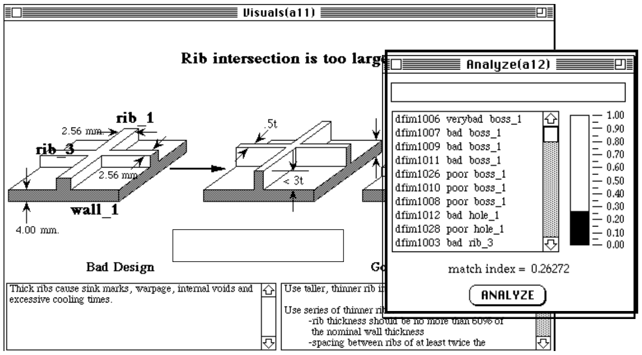


Figure 7. Geometry-based sinkmark index

Our efforts focused on strategic experiments with sample parts together with parametric use of FEM based simulation programs to develop a generic, geometry-based index for sinkmark prediction and cycle-time estimation. Basically, the sinkmark formation is a function of the cooling surface surrounding a unit volume of material. Hence, the proposed geometry index is the radius of a sphere or a cylinder that fits in a local thickness region such as a ribs, rib intersection, a boss, etc. Figure 7 describes the geometry index. Naturally, the threshold for the acceptable sinkmark index depends on the process condition that are economical for production. Hence, the threshold to be used in the design guideline will depend on the feasible process condition, which in turn, reflect the processing cost. We are currently developing a similar geometry index for warpage in injection molding. Our proposed geometry index uses symmetry of volume vs. cooling surface ratio.

We are currently incorporating these generic manufacturability rules into our computer-aid for layout design, HyperDesign/Plastics (Ishii and Hornberger, 1989). HyperDesign is a tool that targets 1) engineer training and 2) compatibility check of preliminary design. Current implementation allows the user to sketch a two dimensional layout of his/her concept and attach "form" icons to represent geometry form features. The system uses design compatibility analysis (DCA; Ishii, et al., 1988) to evaluate the manufacturability of the design and give suggestions for improvement. Figures 8 shows example screen dumps from our latest system.

Obviously, compiling manufacturability rules for every possible geometry feature results in the explosion of the number of design rules. The development of generic geometry-based rules, which may involve some level of reasoning, allows the application of a small number of design rules to a wide variety of geometry features. The current HyperDesign runs on Macintosh computers using HyperCard front-end and Apples Logic Manager (LM) for inference.



Rib intersection is too larg

dfim1006 verybad boss_1
dfim1007 bad boss_1
dfim1009 bad boss_1
dfim1011 bad boss_1
dfim1026 poor boss_1
dfim1010 poor boss_1
dfim1008 poor boss_1
dfim1012 bad hole_1
dfim1028 poor hole_1
dfim1003 bad rib_3

match index = 0.26272

Thick ribs cause sink marks, warpage, internal voids and excessive cooling times.

Use taller, thinner rib i

Use series of thinner ri
-rib thickness should be no more than 60% of
the nominal wall thickness
-spacing between ribs of at least twice the

Figure 8. HyperDesign Plastics

5. CONCLUSION

This paper addressed representation of candidate designs in CAD suitable for intelligent programs that evaluate life-cycle concerns. We argued that feature-based design alone could lead to an unmanageable number of predefined features and tagged data in order to accommodate the different perspectives of the design engineer, tooling engineer, and process engineer. Unlike machined parts, the lack of a starting or "base" feature and the complexity of possible geometry form adds to the difficulty of constructing a generic "feature-set" for net shape manufacturing. However, the use of pure geometry as a form of representing designs leads to the need for a very sophisticated, possibly computationaly expensive computer programs, i.e., ultimate feature extraction programs, to deduce the information needed for evaluation.

What we need is a “happy medium” between these extremes. In particular, we need to develop coded forms of information and methods to infer, from the code, various data needed for evaluation. Specifically, our research focuses on: 1) Method of where to look based on the physical characteristics of the problem and 2) Method of inferring manufacturing information from a set of focused parameters.

Our future research will address verification of our proposed concept via further development of computer-aids for die casting, injection molding, and sheet forming.

ACKNOWLEDGEMENT

This research was supported by the US Army Material Command, NSF/DDM, NSF/ERC Apple Computer, Inc., and Hewlett Packard Laboratories.

REFERENCES

Anderson, D.C. and Chang, T.C. (1990) Geometric reasoning in feature-based design and process planning. Computer and Graphics, Vol. 14, No. 2, pp 225-235.

Bahns, C.B., Barash, D.D., Cescate, D.J., Schneider, M.L., and Hagen, G.B. (1990) Intelligent machining workstation initiative. Cincinnati Milacron Marketing Co.

Chen, Y. M., R. A. Miller, and K. R. Vemuri, (1991a) "A Framework for Feature Based Part Modeling," *Computers in Engineering - Volume One*, ASME, pp. 357-365, Aug. 1991.

Chen, Y. M., R. A. Miller, and K. R. Vemuri, (1991b) "On Implementing an Integrated Design-Manufacturability Assessment Environment," *Computers in Engineering - Volume One*, ASME, pp. 407-413, Aug. 1991.

Cunningham, J.J. and Dixon, J. R. (1988) Designing with features: the origin of features. ASME Computers in Engineering. August, 1988, San Francisco, California.

Cutkosky, M.R., D.R. Brown, and Tanenbaum, J.M. (1989) Extending concurrent product and process design toward earlier design stages. ASME Winter Annual Meeting, December, 1989, San Francisco, CA. pp. 65-73.

Dong, X. and M. Wozny (1990). Managing feature type dependency in a feature-based modeling system. ASME Computers in Engineering. pp. 45-51.

Henderson, M.R. and Chang, G.J. (1988) FRAPP: Automated feature recognition and process planning from solid model data. ASME Computers in Engineering, San Francisco.

Irani, R.K., Safena, M. and Finnagan, P.M. (1990) Boundary-based feature modeling utility. ASME Computers in Engineering, August, 1990, Boston, MA. pp. 45-51.

Ishii, K., Adler, R, and Barkan, P. (1988) Application of design compatibility analysis to simultaneous engineering. *Artificial Intelligence in Engineering Design and Manufacturing (AI EDAM).* Vol. 2, No.1

Ishii, K., Hornberger, L, and Liou, M. (1989c) Compatibility-based Design for Injection Molding. ASME Winter Annual Meeting, San Francisco, CA. pp. 153-160.

Ishii, K. (1990) Role of computers in concurrent engineering. *ASME Computers in Engineering.* August 1990, Boston, MA. Vol.1. ISBN 0-7918-0515-8. pp. 217-224.

Joshi, S. and Chang T.C. (1988) Graph-based heuristics for recognition of machined features from a 3D solid model. Computer Aided Design, Vol. 20, No.2, march 1988.

Liou, S. Y., and Miller, R.A. (1990) Design for die casting. *International Journal of Computer Integrated Manufacturing*, to appear.

Luby, S. C., Dixon, J.R., and Simmons, M.K. (1986) Designing with features: creating and using a features database for evaluation of manufacturability of casting. ASME Computers in Engineering, 1984.

Pinella, J.M., Finger, S., and Prinz, F.B. (1989) Shape feature description and recognition using an augmented topology graph grammar. 1989 NSF Engineering Design Conference.

Requicha, A. A. G. (1980) Representations of Rigid Solids: Theory, Methods, and Systems. *ACM Computing Surveys*, Vol. 12, No. 4.

Rossignoc, J. R. (1990) Issues on feature-based editing and interrogation of solid models. Computer and Graphics, Vol. 14, No. 2, pp. 149-172.

Shah, J.(1988) Feature transformations between application-specific feature spaces, *Computer-Aided Engineering Journal*, pp.247 - 255, December 1988.

Shah, J. and Rogers, J.T. (1988) Feature based modeling shell: Design and Implementation. ASME Computers in Engineering Conf., 1988. San Francisco, California.

Shah, J., Hsiao, D., Robinson, R. (1990) A framework for manufacturability evaluation in a feature based CAD system. NSF Design and Manufacturing Conference, Tempe.

Suh, N., (1988), Basic Concepts in Design for Producibility, Annuals of the CIRP, 37/2, 1-9.

Wilson, P.R., and Pratt, M.J. (1988) A taxonomy of features for solid modeling. Geometric modeling for CAD applications, M.J. Wozny et al (ed.), IFIP 1988.

Woo, T.C. (1982) Feature extraction by volume decomposition. Proc. of Conf on CAD/CAM technology in mechanical engineering, Cambridge, MA. pp. 76-94.