CONF-910689--1

PRODUCTION READY FEATURE RECOGNITION BASED AUTOMATIC

GROUP TECHNOLOGY PART CODING

Arlo L. Ames
Advanced Technology Development Division
Sandia National Laboratories
Albuquerque, New Mexico 87185-5800

SAND--90-2641C

DE91 004789

Abstract

During the past four years, a feature recognition based expert system for automatically performing group technology part coding from solid model data has been under development. The system has become a production quality tool, capable of quickly determining the geometry based portions of a part code with no human intervention. It has been tested on over 200 solid models, half of which are models of production Sandia designs. Its performance rivals that of humans performing the same task, often surpassing them in speed and uniformity.

The feature recognition capability developed for part coding is being extended to support other applications, such as manufacturability analysis, automatic decomposition (for finite element meshing and machining), and assembly planning. Initial surveys of these applications indicate that the current capability will provide a strong basis for other applications and that extensions towards more global geometric reasoning and tighter coupling with solid modeler functionality will be necessary.

1. INTRODUCTION

This paper summarizes progress we have made in developing an expert system that automatically performs group technology part coding through feature recognition techniques. The software is named *Mitro*, after S.P. Mitrofanov, the father of modern part classification and coding. The system has become production worthy, rivaling human performance.

This paper will introduce group technology and our reasons for automating the part coding task. Then, feature recognition and its role in our application are described. Specific design details and test results follow, along with a of other applications requirements that can be supported by our feature recognition capability.

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

2. GROUP TECHNOLOGY

Group technology is a means of saving time and money in design and manufacture by grouping similar parts together to take advantage of their similarities. In design, group technology helps locate previous designs that might be suitable or adaptable for the present task; in manufacturing, we can use knowledge about previously manufactured parts to assist in planning for production of new parts.

Part classification and coding is a mechanism for implementing group technology by assigning a numeric part code to each part that is designed. The part code reflects features of the part such as shape, size, tolerance, material, and manufacturing processes. Retrieval of similar parts requires searching through a database to find parts with similar part codes.

We attempted to use group technology part classification and coding in the DOE Complex using manual techniques. A committee selected and purchased a commercial part coding system. Our design agencies were then charged with the task of assigning part codes to all new parts. Sandia trained eight people to code parts; three full-time and two part-time part coders were assigned to manually perform the task.

After coding approximately 800 parts in three years, Sandia's part coding effort ceased. Other DOE sites abandoned the effort as well. The part code database was rarely consulted, because similar parts could not be located with it.

Difficulties with part coding can be traced to a few fundamental causes:

- **Developing a Useful Database Requires Time.** In order for a part coding database to be useful, it must be populated with a large number of part codes. Coding each part requires at least 15 minutes, so thousands of person hours are required to achieve a useful system.
- **Part Coding is Difficult.** Our part coding specification consists of over 800 rules that relate over 150 part attributes. The parts being coded are commonly represented by two dimensional drawings having hundreds or thousands of geometric and annotation entities. Compounding this difficulty is the requirement that the part coder understand the wide variety of design and manufacturing disciplines (tolerancing, turning, sheet metal bending, extruding, machining) that the part coding specification reflects.
- **Differing Interpretations are Common.** The part coding specification is written in loose terms, permitting a variety of interpretations. Also, the parts being coded can be viewed in many different ways. These differences in interpretation resulted in similar parts receiving very different part codes. Under such conditions, effective retrieval of similar parts is difficult.

Because of these problems, we developed an automatic system for coding parts. Such a system would capture the experience of experts and apply that expertise uniformly on every part. Time requirements could be minimized by purchasing faster hardware. Additionally, the system could be extended to automate other design and manufacturing applications.

3. FEATURE RECOGNITION

In order to develop the automatic group technology part coding system, we require features and feature recognition.

A significant body of research in the area of features has recently developed. Many different definitions of feature have been advanced by various researchers in the field. For our purposes, we will define feature as "anything of interest". By this definition we mean that a feature is some information required for the application. Features may be function, form, or process based, and are often expressed at a higher level than the geometric description of a part.

Features play an essential role in our system, because the part coding specification is written entirely in terms of features. In fact, most design and manufacturing applications require features. It is difficult to imagine discussing an application such as machining without encountering terms such as holes, slots, and flats. [CUNN88] suggests that reasoning about design and manufacturing activities automatically develops features — that applications and features are inseparable.

Features are application specific. Machining is concerned with the volumes of material that must be removed to create the part from the stock, and with finding a means of accessing material removal regions with a tool. Casting is concerned with features such as wall thickness. Tool accessibility is not an issue, while flow of material within the boundaries is. Each different application requires a different set of features.

The data we have chosen to use as input to our application is in the form of CAD (Computer-Aided Design) solid models. Solid models are complete, unambiguous representations of the geometry. However, CAD data is not expressed in the specific high-level terms used in the part coding specification (e.g. the part is rotational and has holes). B-rep (boundary representation) solid data is expressed as low-level geometric and topological entities (e.g. faces, edges, vertices). B-reps represent parts in a fairly unique way (aside from geometric and topological flavorings), but are lower-level than the application we are pursuing. CSG (Constructive Solid Geometry) and FBM (Feature Based Modeling) cannot be directly used because their representations are not unique. A single part can be modeled in many different ways in CSG and FBM; such variability was a primary cause of the failure of manual part coding. Additionally, directly modeling parts in terms of features requires that the features match the application. Our designers did not find the part coding viewpoint to be particularly natural, and hence did not create good feature based input for the part coding system.

Feature mapping and feature recognition are the only methods of automatically creating an application specific feature based model. Feature mapping is the transformation of one feature based model to a different one. Feature recognition recognizes patterns in a geometric model to create the feature based model. Feature recognition is more useful in the general case, because solid model data from any source (whether manually or automatically generated) can be used, while feature mapping must start with a carefully constructed, manually-defined feature based model. Additionally, feature mapping must support the interactions between elements of a very large set of entity types (e.g. slot/slot, slot/hole, slot/pocket, hole/pocket interactions), while feature recognition is only

required to support interactions between geometric entities (e.g. face/face interactions). Finally, feature mapping has only recently become a research topic, while feature recognition is a fairly mature technology. For these reasons, we have chosen to implement the part coding system using feature recognition.

4. DEVELOPMENT PHILOSOPHY

This section documents design decisions that were made in developing the part coding system. Specific details of the recognition algorithms are documented in [AMES91].

Feature recognition is a heuristic process. In developing an automated system, we are required to implement algorithms to recognize features when we have never before had to precisely define these features. Features are not only application specific; they vary from person to person. Our experience with part coding shows that two people, given identical geometries, recognize different features. For this reason, an automatic feature recognizer is only required to generate reasonable feature based models, not to match any particular human's performance.

Our features are primarily defined as sets of portions of faces. Previous feature recognition efforts (e.g. [KYPR80], [HEND84], [WOO82], [VAND90]) have concentrated on volumetric and face based features, neither of which match the semantics of our application. Volumetric features represent material to be removed in machining. Our part coding specification focuses on the final shape of the part, not on the material that was removed in creating it, so volumetric features are undesirable. Face based features are not completely satisfactory either, because there is not a one-to-one mapping between faces and features (this is documented as the face locality problem in [AMES88]). Representing features as sets of portions of faces permits us to represent the final shape of the part while addressing the face locality problem. Subface based features have been implemented here by recording which portions of edges of a face have been used in a feature; when all of the edges of a face have been used, the face is no longer available for membership in new features.¹

Exact B-rep data is required by our system. CSG models are very difficult to use because CSG models are not unique and they contain null geometry (faces and subfaces that are present in the CSG model but aren't present in the final geometry) — the form of the part cannot be determined without evaluating the model (creating a B-rep). B-rep data is also more commonly available than CSG. Algorithms for converting from CSG to B-rep exist, while the inverse mapping does not. Exact (not faceted) data is required because many features are defined explicitly in terms of specific kinds of higher-order surfaces (e.g. a taper is a conical surface). For this reason, the model must explicitly contain natural quadric surfaces and must not be faceted. Seams in the model can cause significant difficulties, especially in symmetry detection, where a symmetric part will have an assymetric model due to the seams.

¹This edge marking strategy permits portions of faces that are not next to an edge to be ignored in the features being recognized. Thus far this has not been a serious limitation.

The feature recognizer in *Mitro* is passive; that is, it has no direct procedural links to a solid modeler. This decision was based on achieving speed and avoiding dependence on any particular vendor's modeler. Additionally, CAD vendors have been reluctant to provide *any* programmatic interface, so a passive modeler is much easier to deliver. Certain applications require stronger links to a solid modeler, but speed can be gained by limiting the amount of interaction.

The feature recognizer's architecture is based on *featurettes*, which are features that are recognized to assist in the recognition process. They are often very low level features, such as sets of parallel edges, or sets of faces having similar attributes (such as surface finish and area). Featurettes permit decomposing the recognition process into smaller, simpler steps that are easier to develop than the complex algorithms present in previous efforts. Additionally, many featurettes are usable in recognizing more than one kind of feature; maintaining featurettes saves effort. For example, Figure 1 shows a portion of the featurette hierarchy used for recognizing profiled holes, extrusion sides, and slots.

Figure 1. An example featurette hierarchy. Faces adjacent by portions of overlapping parallel edges are used to infer the presence of profiled holes, extrusions, and slots. Portions of the hierarchy have been omitted for clarity.

Balancing geometric and topological information is important in developing a competent recognizer. Previous recognizers (e.g. [KYPR80]) have been largely

topology driven, with the exception of [VAND90], which is primarily geometry driven. Purely geometry driven systems are computationally intensive, because they require making geometric tests on all pairs of entities in the model. Additionally, they ignore topological clues that permit recognizing a wider range of features. Purely topology driven systems are weak in their reasoning abilities, because topologically close entities can be very distant geometrically and vice versa. This work focuses on 2-1/2 dimensional (swept) features, using both geometry and topology to define the necessary patterns and tests. Concentrating on geometry that is 2-1/2 D allows us to relax some of the topological restrictions, while permitting us to search for geometric patterns that are relatively simple to find.

The feature recognition process is organized based on a feature precedence. This feature precedence is necessary to preserve the semantics of the part coding specification. For example, a set of faces that *could* define a machined slot can occur on the profile of an extrusion. If the part is extruded, no "slots" on its profile will have to be machined, so we must suppress recognition of such slots, while permitting slots to occur elsewhere on the part. The feature precedence guarantees that possible alternative interpretations of the geometry are not generated. Generating only one interpretation is much faster than generating many, and is appropriate for our application. The precedence we have defined places shape defining features (outer and inner diameters, sheets, bends, extrusion sides) highest, followed by shape modifying features (profiled holes, holes, slots, curved faces and flats). At each step in the recognition process, all higher precedence features must have been recognized before lower precedence features may be recognized. The portions of faces that do not belong to a feature of higher precedence are considered for membership in the current feature class.

Geometric and topological invariance is necessary for the system to be successful. The part code must be insensitive to variables such as global orientation or entity ordering in the input data to prevent the problems that occurred with human part coding. Featurettes enable each recognition step to consider sets of related entities, minimizing the impact of their orientations or topological positions.

Features may interact in a number of ways, and feature recognition algorithms have historically performed poorly on parts with interacting features. This work addresses the feature interaction problem using a number of techniques. Feature merging is avoided, because features that might be combined can easily be considered independently with no serious difficulties. Additionally, merging is expensive, requiring solid modeler verification. Faces are permitted to belong to more than one feature (using subfaces), to address the face locality problem. Finally, it is recognized that the local geometry and topology of the model may not capture global interactions, but that local interactions are more easily recognized. Recognition proceeds from featurette hints (very local information) to local tests to more global tests, in order to minimize effort while deriving correct features. For example, outer diameters of a rotational part are found by first finding a convex rotational face (a hint), then locally testing it to find any faces that are adjacent to it (especially by smooth or convex edges) in a manner that locally obscures accessibility to the face. Finally, all of the faces in the model are tested (by examining sample points) to determine whether they obscure the potential outer diameter. A global simulation by a solid modeler would be the ultimate test of a

feature. We have avoided such uses of the solid modeler because they are largely unnecessary and too expensive for this application; other applications will require such interaction.

5. TEST RESULTS

Mitro currently supports recognition of the following part coding features: overall part shape and size, symmetries, stock, thru- and blind-machined elements (profiled holes, holes, slots, flats, curved faces), tool axes, hole patterns, rotational features (outer diameters, inner diameters, diameter visibility, centering holes, chamfers, faced ends, grooves, tapers, profiles, projections, bends), sheet metal features (sheets, bends, cutouts, flattened perimeter), prismatic features (curved and slanted faces, rotational projections), extrusion features, and simple assembly features.

The software has been tested using a suite of approximately 200 parts. Of these, 90 are simple- to medium-complexity Sandia designed parts. The part codes the system generates have been compared with human generated part codes. The results are very encouraging.

The system is reasonably fast. The entire test suite can be coded in less than twelve hours on a Symbolics XL400 workstation, averaging 7 minutes per part on medium complexity parts. Human part coders required 15 minutes per part to generate codes. *Mitro* is faster than humans, and can be improved with faster hardware.

The software performs reliably, coding all of the parts in the test suite in a consistent manner. All of the similarities of the parts are captured in the part codes, without regard to orientation, relative size of features, or any of the other problems that plagued manual efforts. The system is very thorough in analyzing parts; often discrepancies between human and machine generated part codes can be traced to details the humans missed. The software's results are not identical to the humans' results. In cases where they differ, one of the following situations occurs:

- The software was wrong and the human right.
- The human was wrong and the software ~~wrong~~ right.
- Both the software and the human were right.
- Both were wrong.

The software errors are generally related to infrequently encountered global interactions; further development can correct these cases. Human errors are errors in perception; these errors are much more difficult to correct. In cases where the both are correct, the software's answer is preferable because a single set of algorithms is used in generating the software's answers; humans tend to custom tailor their algorithms for each new case. Thus, humans tend to generalize better to new kinds of data, but don't provide uniform answers.

The software will generalize well in domains where it has been programmed. Experience in testing parts that the system (and its developer) have not previously seen indicates that the software will perform well on new parts. Human intervention will be required to determine when a new part category or new feature recognition capabilities should be added. The software cannot check itself, and will require periodic review of its performance. The system is truly an expert system — it performs expertly in narrow domains.

Figure 2 shows some example parts from the test suite that demonstrate system capabilities. The software's interpretations of these parts are as follows. The first example is a rotational segment (more than one segment can be cut from a single turning) that does not have a turnable inner diameter. The part has holes, a round-bottomed slot, a large concave curved face, and machined flats. The second example is a prismatic part (that would be completely machined); it is not turned because protrusions are too large. It has machined slots, holes, flats, and curved faces, and a rotational protrusion. The third example is a bent sheet metal part. It has small corner cutouts, small and large side cutouts, and holes. The fourth example is an extrusion with one slot. It has holes and machined flats.

Figure 2. Example parts that are properly coded by *Mitro*.

6. NEW DIRECTIONS

The feature recognition capability has been examined in light of a number of automatic applications it might support. Some of the requirements for these applications are presented here, along with commentary regarding the difficulty of extending the current work to fill these requirements.

An automatic manufacturability analysis tool can provide designers with early information regarding how easily and cheaply their designs can be manufactured. The group technology information (what kind of part is it, does it have holes, etc) can form a foundation for such an effort. Additionally, manufacturability analysis considers whether the parameters of features (diameters, etc) are standard, and could be manufactured from standard stock with standard tools. Further development will center around adding additional classification of features (e.g. countersinks, counterbores and undercuts), but appears to be rather direct.

Automatic finite element mesh generation can remove much of the drudgery in performing finite element analysis. Hexahedral elements are often preferred for analysis (especially nonlinear analysis), but no fully automatic system for generating them has yet been developed. Inputs to mapped mesh generators must be hexahedral volumes, and manual decomposition of a part into these hexahedra is difficult. [RAZD89] and [PHIL88] performed early work in automatic decomposition. An automatic decomposition could use low and high-level feature information as hints for locations to perform dissections. Visibility information would be helpful in determining whether certain decompositions would interfere with the part geometry in undesired ways. Finally, boolean operations, a simple geometry evaluator (recognizer of hexahedral parts) and a planning algorithm would complete the decomposer. Work here entails using or extending the current work to provide good dissection hints, interfacing the system to a solid modeler directly, performing boolean operations and interpreting their results.

Machining planning is being addressed by many feature recognition efforts. Tool accessibility is a problem that must be addressed; this is similar to the visibility information in mesh generation. The feature recognition capability developed here would be useful in generating strong hints for developing a volumetric decomposition, much as is performed in [VAND90], except that this work recognizes those hints more quickly.

Three dimensional tolerance analysis work being researched [ROBI89] requires discovering where objects in an assembly contact each other, classifying the type of contact being made, and forming a kinematic model that can be analyzed for small parameter deviations. A reasonable approach would intersect offset solids to locate the contacts and augmenting *Mitro's* recognition algorithms to classify the intersections.

Robotic assembly requires discovering contacts between parts in an effort to locate valid disassembly routes. These contacts could be discovered identically to the tolerance analysis work, leading to degrees of freedom. More work in global interactions would be required to infer good disassembly routes. Accessibility, as required in machining planning and mesh generation, would be required here.

In each case, the existing capability provides a good starting point for the application, but must be extended to provide adequate support. Many of the

requirements for one application can be leveraged against other applications, speeding their development. Note, though, that the recognition capability is shared, while the view of the geometry is typically rather different. The same recognizer is often applied to the data in a very different way.

7. CONCLUSIONS

Feature recognition is a powerful, useful technology. It has enabled us to develop a competent, fully automatic group technology application. Contrary to opinions voiced in [SHAH88] and [PRATT84], feature recognition is not redundant effort. By recognizing features *uniformly*, the present system is much better than humans at grouping parts, even though the features are "information which the designer was fully aware of when he created the model". Feature recognition enables us to generate many different application-specific views, addressing applications that the designer could not originally foresee.

A balanced use of geometry and topology, featurettes, and feature precedence can combine to define a feature recognizer that is robust even when dealing with interacting features.

The part coding application has provided a rich environment to develop our feature recognition capability. We have developed an excellent base for attacking new applications. Future developments are likely to require more global reasoning and stronger ties to solid modeler functionality.

8. REFERENCES

[AMES88] Ames, A.L., "Automated Generation of Uniform Group Technology Part Codes From Solid Model Data", *Proceedings of the 1988 ASME International Computers in Engineering Conference and Exhibition*, ASME, 1988.

[AMES91] Ames, A.L., "Mitro: Automatic Feature Recognition Based Uniform Group Technology Part Coding from Solid Model Data", in preparation.

[CUNN88] Cunningham, J.J. and J.R. Dixon, "Designing with Features: The Origin of Features", *Proceedings of the 1988 ASME International Computers in Engineering Conference and Exhibition*, ASME, 1988.

[HEND84] Henderson, M.R., "Extraction of Feature Information from Three Dimensional CAD Data, PhD Thesis, Purdue University, 1984.

[KYPR80] Kyprianou, L.K., "Shape Classification in Computer-Aided Design," PhD. Thesis, Christ's College, University of Cambridge, 1980.

[PHIL88] Phillips, L.R., J.L. Mitchiner, T.D. Blacker, Y.T. Lin, "A Knowledge System for Automatic Finite Element Mesh Generation: AMEKS", *Proceedings of the First Annual International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, Vol. 2, June 1988.

[PRATT84] Pratt, M.J., "Solid modeling and the interface between design and manufacture," *IEEE CG&A*, 1984.

[RAZD89] Razdan, A., M.R. Henderson, P.F. Chavez, P.A. Erickson, "Feature Based Object Decomposition for Finite Element Meshing" *The Visual Computer*, Vol. 5, No. 5, 1989.

[ROBI89] Robison, Richard H., "A Practical Method for Three-Dimensional Tolerance Analysis Using a Solid Modeler, ADCATS Report No. 89-3, Master's Thesis, Brigham Young University, 1989.

[SHAH88] Shah, J.J., P. Sreevalsan, M. Rogers, R. Billo, A. Mathew, *Current Status of Features Technology, Revised Report*, R-88-GM-04.1, Computer Aided Manufacturing-International, Inc., 1988.

[VAND90] Vandenbrande, Jan H., "Automatic Recognition of Machinable Features in Solid Models," PhD Thesis, University of Rochester, 1990; published as Report IRIS#260, Computer Science Department and Institute for Robotics and Intelligent Systems, University of Southern California, 1990.

[WOO82] Woo, T.C., "Feature Extraction by Volume Decomposition," CAD/CAM Technology in Mechanical Engineering, 1982.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

END

DATE FILMED

05 / 08 / 91