NBSIR 87-3551

PART GEOMETRY DATA IN THE AMRF

Juliana S. Tu
Theodore H. Hopp

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
National Engineering Laboratory
Center for Manufacturing Engineering
Factory Automation Systems Division
Machine Intelligence Group
Gaithersburg, MD 20899

April 1987

U.S. DEPARTMENT OF COMMERCE, Malcolm Baldrige, Secretary

NATIONAL BUREAU OF STANDARDS, Ernest Ambler, Director

PART GEOMETRY DATA IN THE AMRF

Juliana S. Tu
Theodore H. Hopp

1. INTRODUCTION

The Automated Manufacturing Research Facility (AMRF) of the National Bureau of Standards is a laboratory for studying metrology and standards issues related to the automation of small-batch, discrete-parts manufacturingNote: The change to pitch (13*) and font (1) must be converted manually.Note: The change to pitch (13*) and font (5) must be converted manually.Note: The change to pitch (13*) and font (1) must be converted manually.Note: The change to pitch (13*) and font (5) must be converted manually.Note: The change to pitch (13*) and font (1) must be converted manually.. The Geometry Modeling Project of the AMRF was established to develop the geometry modeling technology needed to support the automated, real-time, manufacturing control systems that the AMRF represents. The experience of researchers in the AMRF has shown that control processes in an automated system require real-time, on-line access to geometric data for parts, tools, fixtures, and other resources. To provide the performance required by these processes, multiple representations of part geometries must be maintained. The representation of part functionality (e.g., mating parts for assembly or fixturing) requires storing the relationships between geometric features of different parts. Also, geometric data must be related to other information in factory data bases, such as process plans, quality data, inventory, and tolerances. The management of multiple data representations, providing access to data in a uniform manner to all facility processes, and maintaining consistency among the related data distributed throughout an automated factory such as the AMRF is a major challenge.

This paper is an overview of the approach being taken toward the modeling and management of part geometry data in the AMRF. This project will develop the understanding and capabilities needed to support the AMRF through the implementation of a system, called the Geometry Data System, that will manage all model information that is needed to perform automated manufacturing operations. The Geometry Data System implements the following functions: provide uniform access to all model data required by different control processes; provide a set of tools for manipulating data in various representations; make the capabilities of particular geometry modeling systems available to application processes; and associate application-specific data to model information. The experience gained in implementing the Geometry Data System will provide input to national standardization efforts such as the Product Data Exchange Specification (PDES) [SMIT86] being developed by ANSI and the Dimensional Measurement Interface Specification (DMIS) [CAMI86] being developed by CAM-I, Inc.

The functions of the Geometry Data System were chosen because of the diverse needs of the part geometry data in automated manufacturing. Part geometry data in the AMRF consists of all of the relevant data about a part to be machined: its basic dimensions, its features, and its tolerances. Each application in the AMRF tends to have its own particular representation of these data. That is to say, each application creates, keeps, and uses the data in some particular way according to its processing requirements. For instance, the Cleaning and Deburring Workstation is interested in "deburring edges" (edges that must have burrs removed), and the order of machining of the faces adjacent to each deburring edge. The Inspection Workstation uses tolerances between functionally important geometric features for quality control. The Process Planning System uses groupings of geometry data to form machinable features such as pockets, holes, grooves, chamfers, and so on. We want to pool all these versions of part geometry data together to form an integral part of the facility, so that the various applications are relieved of the burden of geometry data management, and yet still get to use the geometry data in their special way of processing.

This paper describes our approach to the implementation of the Geometry Data System, the applications supported by the system, and how geometry data flows through the AMRF through the operation of the Geometry Data System.

2. APPROACH

Our approach to the implementation of the Geometry Data System is to apply three technologies from computer science. First, formal language syntax is used to specify geometry data and their applications as a part model. This syntax allows parsers to be built on each AMRF system to interpret the geometry data language. Second, data modeling tools are used to build the data structures of the part model, so as to provide a formal definition of the meaning of part model entities. Third, database management technology is used to manage the geometry data.

2.1. LANGUAGE

Part models are communicated within the AMRF by means of data base reports, which are ASCII text files containing the part model data. The format of these reports is formally specified by a grammar in Backus Naur Form (BNF) [HOPP87]. A typical part model report is illustrated in Figure 1. The part model report format constitutes a language by which applications communicate part geometry in the AMRF.

The report format is defined in five sections. The header section specifies identifying information and global data about a part -- its unique name in the AMRF, its group technology code, its drawing number, the name of its designer, and the part material may all be specified. The functionality section specifies the geometric tolerances on a part with respect to its functions for quality control. The feature section specifies application-specific information about the part. A feature in the features section specifies the particular way an application uses the design data of a part -- for example, machinable features used for process planning, cleaning and deburring features for the generation of deburring paths, and inspection features for the inspection workstation. The topology section describes the connectivity relationships of the basic geometry data. These geometry data are specified in the geometry section.

As a consequence of using formal language techniques, we have parsers and data assemblers on all applications using the Geometry Data System. The parsers generate internal data structures from the database report and the data assemblers go the other direction they generate database reports from the internal data structures.

2.2. GEOMETRY DATA MODEL

In the geometry data model, the basic geometry data of a part are organized in a boundary representation [MORT85]. In this representation, a part is described by its topological entities and its geometrical entities, as shown in Figure 2. The topological entities are structured hierarchically in the order: solid, shell, face, loop, edge, and vertex. The geometrical entities are surface, curve, point, and unit vector (direction). The connectivity relationships of the geometry data of a part are provided by its topological entities, so that one can access the faces of a part, the adjacent faces of an edge, the edges on a face, and so on. The geometrical entities of a part are accessed through its topological entities, so that surfaces are accessed through faces, curves through edges, and points through vertices.

The AMRF part model was initially developed in Smalltalk-80 [GOLD83] on a Tektronix 4404 AI workstation. The purpose of the implementation was to develop and test the model, to verify that it would support the AMRF applications, and to provide a prototype implementation for use in the AMRF.

Smalltalk was chosen because it is a language in which objects and their behavior are the focus of attention for the programmer, rather than data structures and control. In Smalltalk, programs are developed and run by defining classes of objects, creating instances of these classes, and sending messages to these instances. A Smalltalk class defines both the data structure that will make up each instance of the class and the behavior of each instance (i.e., how it will respond to messages)Note: The change to pitch (13*) and font (1) must be converted manually.Note: The change to pitch (13*) and font (5) must be converted manually.Note: The change to pitch (13*) and font (1) must be converted manually.Note: The change to pitch (13*) and font (5) must be converted manually.Note: The change to pitch (13*) and font (1) must be converted manually.. A class definition is itself an instance of the Smalltalk class Behavior. Any instance of a Behavior (that is, any class definition) can respond to the message "new" and will create a new instance of that class. The class definition then determines how that instance will respond to messages.

We have implemented a class Topology to model the topological entities in the AMRF part model. A portion of the Smalltalk definition of this class is shown in Figure 3. As can be seen at the top of the lower display panel, all topological entities have a name, a set of subentities and a set of superentities. The latter two refer to the topological hierarchy shown in Figure 2. (Thus, for instance, a shell as a topological object has a solid as a superentity and a set of faces as subentities.) As shown in the third panel on the top, all topological objects share a common set of routines (not displayed) for editing and accessing object definitions, for printing, for initialization, and for error handling.

The class Topology is refined through the definition of subclasses for particular types of topological objects. The subclasses inherit all the properties of the Topology definition, while providing additional behavior (routines), data definitions, and identity to specific objects. The complete list of Topology subclasses is shown in the bottom panel of Figure 3. Reading down the figure, the EdgeLoop class associates a set of orientations to the subentities of edge loops (i.e., edges) that defines the counterclockwise traversal direction of edge loop instances. Shell objects have no additional data than that defined for all Topology objects, but are identifiable as shells by virtue of being instances of class Shell. Solid instances represent individual parts. Each Solid instance maintains a dictionary of the topological objects that constitute that solid. These dictionaries are used for efficient access to the solid. The remaining classes are refinements of a class TopologyWithGeometry, that represents the fact that faces, edges, and vertices each have an underlying geometric representation. The TopologyWithGeometry class implements a common representation and access mechanism for this model. TopologyWithGeometry has two subclasses, TopologyWithOrientedGeometry, that implements the concept of oriented geometry shared by faces and edges, and Vertex, for which the geometry (points in space) does not have an orientation. Faces and edges are implemented by parallel class hierarchies that are used to distinguish between closed topology (such as circles or spheres) which have no boundary subentities, and open topology (such as circular arcs or spherical patches) which do have a boundary.

The geometry objects associated with TopologyWithGeometry subclasses are instances of a class Geometry, shown in Figure 4. The subclasses of Geometry are the specific geometric objects that can be represented in the Smalltalk implementation and in the AMRF part model format. The geometry classes are grouped into abstract classes Curve3D, representing all curve objects, Point3D, representing all point objects, and Surface3D, representing all surface objects. A subclass of Point3D, UnitVector3D, represents directions in space. The operations on the Geometry class and its subclasses describes the properties of geometric objects and the relationships between them. While the current implementation is relatively limited in the types of geometry that can be modeled, it can easily be extended by defining additional subclasses of Geometry.

2.3. GEOMETRY DATA MANAGEMENT

The application processes in the AMRF access the geometry data through the AMRF distributed database system, called Integrated Manufacturing Data Administration System, or IMDAS for short [BARK86]. The Geometry Data System is a database management system within the distributed architecture. It is a relational database, implemented in RIMNote: The change to pitch (13*) and font (1) must be converted manually., together with software for handling requests dealing with part model data. The operation of this architecture is shown in Figure 5. When an application issues a data request, it is parsed by IMDAS and, based on the type of request, routed to the Geometry Data System. The Geometry Data System constructs the actual relational database commands to execute the request, and packages the results in the Part Model Format for IMDAS to send out to the application.

The four data requests are INSERT, SELECT, DELETE, and UPDATE. Basically, the data requests that an application needs to issue are "insert a part model into the geometry database", "select a part model with a specific part name from the geometry database", "delete a part model from the geometry database", and "update a part model in the geometry database". The user of the database system need not be aware of the structure of the data, and much less of the relational database operations such as project, joint, select, and union.

3. APPLICATIONS

The applications currently supported by the Geometry Data System are Computer-Aided Design (CAD), Geometry Modeling, Process Planning, and Cleaning and Deburring. The applications we plan to include are Inspection, the Vertical Workstation, the Horizontal Workstation, the Turning Workstation, Robot Vision, Robot Control, and Assembly.

The CAD system is the source of geometric data for parts made in the AMRF. The data generated by the CAD system is the geometry and topology of the ideal part, along with certain administrative data about the part.

The Geometry Modeling system starts with the design data and adds the definitions of tolerances, datum reference frames, datum surfaces, and functional features. The part model produced by the Geometry Modeling system is the definition used by the rest of the AMRF systems.

The Process Planning system requires tolerance and shape information about the part so as to define machinable features and to select the proper manufacturing processes. The process planning system also generates intermediate workpiece descriptions in the form of new part models.

The Cleaning and Deburring Workstation uses information about edges and adjacent faces in planning and executing deburring programs. The information needed are the curve equations of edges to be deburred, the surface orientations of the faces that form each edge, and the order of machining of the faces. (Any burrs will be along the face that was machined first.) The latter information is part of the process plan, but is stored in the part model as a "deburring" machinable feature.

Inspection requires definitions of functional features, the controlling tolerances, and the datum reference frames used to define part coordinate systems. These data are generated by the Geometry Modeling system and are, in principle, part of the design of the part.

The Vertical, Horizontal, and Turning Workstations will require the definitions of machinable features and of fixturing requirements. These process-specific data will be generated by the Process Planning system and stored as components of the part model.

Robot Vision and Robot Control are closely coupled systems that need information about part shapes to plan and control grasping operations, identification of parts, and orienting functions performed by the robots. Much of the data used by the Robot Vision system is highly specialized part representations used for efficient vision processing algorithms. The relationship between these specialized representations and the current AMRF boundary model has not yet been fully defined.

Assembly applications require information about mating surfaces between parts, as well as the kind of material handling data needed by Robot Control. Currently, relationships between different parts are not supported, and will be a major extension to geometry modeling capabilities of the AMRF.

4. DATA FLOW

The flow of part geometry data through the AMRF is shown in Figure 6. After a part is designed on a CAD system (the AMRF currently uses CATIA running on an IBM 4381), its basic dimensions are extracted from the CAD system and stored as its design data. The design data are communicated to the AMRF database system as a database report in the format described above. These data form the topology and geometry sections of the AMRF Part Model. The part model is then retrieved by the Geometry Modeling System, where the part definition is expanded to include part feature and part functionality data, in the form of geometric tolerances. When the Geometry Modeling System stores the expanded part model in the AMRF database, it contains the geometry, topology, functional features, and tolerances of the part. This completes the design specification for the part. The part model is then available for access by all of the AMRF applications, both for manufacturing engineering and for shop floor control.

To give a specific example, Process Planning accesses the Part Model, and identifies groupings of geometric entities as machinable features. It then updates the part model with the machinable features of a part. These machinable features, stored in the part model, are referenced by the process plans produced by the Process Planning system. These features are then used by equipment-level programming systems (such as NC coding and robot control data generation) to program shop floor equipment. At this stage, a part can be manufactured automatically on the AMRF shop floor.

As mentioned earlier, part functional feature data and geometric tolerances of a part are entered on the Geometry Modeling System. This system was built on top of a geometry modeling package called UNICADNote: The change to pitch (13*) and font (1) must be converted manually., running on a Silicon Graphics IRIS 2400 workstation. The user can define features, datum reference frames, and tolerances through a combination of interactive graphics and filling forms on the screen. A typical screen display is shown in Figure 7. The Geometry Modeling System is described in detail in [CLAR87].

5. CONCLUSION

As has been described, part model data in the AMRF is a unifying thread from design, through modeling, process planning, equipment programming, to shop floor operations. A key to the successful implementation of this approach has been to provide a structured methodology for the preparation and handling of part data at all stages of the manufacturing process. This methodology was developed through the application of formal language techniques, data modeling, and database management technologies.

The approach being taken to handling part models in the AMRF is evolutionary. The current system supports some, but not all, of the applications envisioned. As geometry data requirements are developed for other AMRF systems, and change for systems currently supported, the Geometry Modeling System described in this paper will change to meet those needs.

There are several types of geometry data that are not supported by the current system. These include such things as part drawings and various types of relationships between part models. Part drawings are generally not needed within the AMRF. The intent of this work is to support automated or semi-automated systems. The emphasis has been on communicating mathematically precise and complete part models, rather than on blueprints that require human interpretation. As such, the objectives of this work are complementary to such standards as IGES [ANSI81].

There is one set of requirements that has not been addressed that is expected to be very important. This is the modeling and representation of relationships between different part models. Such relationships are needed for a variety of applications, including revision control, the relationship of a part design to intermediate workpiece geometries, and assembly descriptions (both for assembled products and for fixturing operations). This will require storing cross-references between part models, as well as between components of part models (such as between mating faces on different parts). These issues are extremely complex, from the point of view both of the engineering needs and of the data management requirements. While we anticipate addressing some of these issues in the near future, we expect that it will be many years before the technology will have been developed to a point comparable to that available for representing individual parts. Developing this technology within the framework of the AMRF will be the goal of our future work on geometric modeling.

6. REFERENCES

[ANSI81] American National Standards Institute, Digital Representation for Communication of Product Definition Data, American National Standard ANSI Y14.26M-1981, American Society of Mechanical Engineers, New York, 1981.

[BARK86] Barkmeyer, E., Mitchell, M., Mikkilineni, K., Su, S., and Lam, H., An Architecture for Distributed Data Management in Computer Integrated Manufacturing, NBSIR 86-3312, National Bureau of Standards, Gaithersburg, MD, 1986.

[CAMI86] CAM-I, Inc., CAM-I Dimensional Measuring Interface Specification Project 1986, PR-86-ASPP-02.4, Computer Aided Manufacturing International, Inc., Arlington TX, 1986.

[CARD85] Cardelli, L. and Wegner, P., "On Understanding Types, Data Abstraction, and Polymorphism," ACM Computing Surveys, 17(4), Association for Computing Machinery, New York, 1985, pp. 471-522.

[CLAR87] Clark, S. and Ressler, S., The Geometry Modeling System User’s Guide, NBSIR 87-3508, National Bureau of Standards, Gaithersburg, MD, 1987.

[GOLD83] Goldberg, A. and Robson, D., Smalltalk-80: the Language and its Implementation, Addison-Wesley, Reading, MA, 1983.

[GUTT77] Guttag, J., "Abstract Data Types and the Development of Data Structures,", Communications of the ACM, 20(6), Association for Computing Machinery, New York, 1977, pp. 396-404.

[HOPP87] Hopp, Theodore H., AMRF Database Report Format: Part Model, National Bureau of Standards internal report, 1987 (to appear).

[MORT85] Mortenson, Michael E., Geometric Modeling, John Wiley & Sons, New York, 1985.

[NANZ84] Nanzetta, P. and Wellington, J., Publications of Center for Manufacturing Engineering: 1978-1983, NBSIR 84-2840, National Bureau of Standards, Gaithersburg, MD, 1984. (Supplements issued occasionally.)

[SIMP82] Simpson, J., Hocken, R., and Albus, J., "The Automated Manufacturing Research Facility," Journal of Manufacturing Systems, 1(1), Society of Manufacturing Engineers, 1982, pp. 17-32.

[SMIT86] Smith, B., Product Data Exchange Specification; The PDES Project: Objectives, Plans, and Schedules, Internal Memorandum, National Bureau of Standards, June, 1986.