A Feature-based Machining System using STEP

Frederick M. Proctor and Thomas R. Kramer

National Institute of Standards and Technology
Building 220, Room B124
Gaithersburg, MD 20899-0001

ABSTRACT

Discrete part manufacturing flows from a design phase in which product information is defined to a manufacturing phase in which the processes for machined parts and tooling are planned and executed. Process planning typically culminates with the generation of numerical control (NC) programs for specific equipment, such as machining centers or turning centers. These NC programs are written in the dialects of the various equipment vendors, for the specific mechanical configuration of the target machine. As a result, porting programs between machines is difficult. Worse, NC programs contain little if any of the product design information. The lack of this information at run time limits any adaptive control that could direct the process so that final parts more closely conform to the original design.

The authors have developed a prototype machining system in which product and process data replaces NC programs at run time. In this system, information models built in the EXPRESS information modeling language are used for all types of data, and data files are all in STEP Part 21 format; each Part 21 file is understandable by making reference to one of the EXPRESS models. The EXPRESS schemas of tool models proposed to ISO are used. An EXPRESS schema for ALPS (A Language for Process Specification, developed by NIST) is used for process planning. Ad hoc EXPRESS schemas are used for machining options, setup descriptions, shop and workstation operations, and tool usage rules. The system has been demonstrated on a three-axis machining center.

Keywords: STEP, manufacturing features, process planning, numerical control

1. THE USE OF FEATURES IN MANUFACTURING

A machining feature is a closed volume in space that is related to a machining operation as follows: when the operation is finished, there must be no material remaining inside the feature, and the operation may remove no material outside the feature. In use, the machining features are defined so that a boolean subtraction of the features from the original workpiece (stock or a partially finished part) will result in the desired final shape for the workpiece (finished or partially finished part). Examples of machining features include faces, pockets, slots, bosses, and holes, some of which are shown in Figure 1.

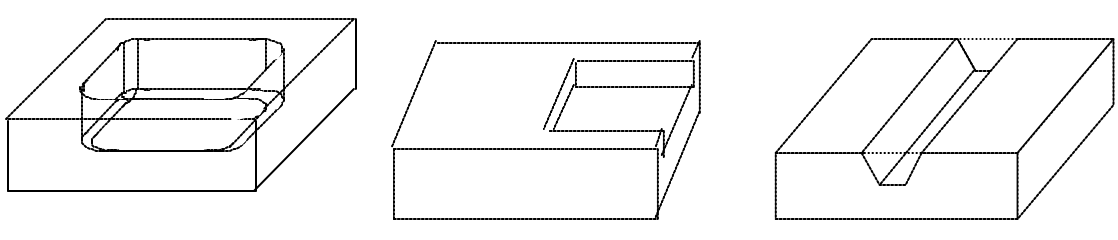

Figure 1. Examples of machining features.

The machining features used to make a specific part may be instances of a fixed library of parametrically defined features, or they may be defined without a library by making boundary or constructive solid geometry representations. There are, of course, many alternative definitions of machining features[1]. To machine a part, both the machining features and the machining operations must be defined, and the operations must be sequenced. In our Feature-Based Inspection and Control

System (FBICS), we have used operations that are instances of parametrically defined operations from a fixed library of operations.

2. THE STEP APPROACH

STEP is the common name for standard 10303 of the International Organization for Standardization (ISO). This standard is composed of individual documents known as STEP "Parts". STEP Part 11 defines the EXPRESS data modeling language. An EXPRESS model definition is contained in one or more constructs called EXPRESS "schemas". STEP Part 21 defines an exchange file format for transmitting instances of data that has been modeled in EXPRESS schemas. STEP also provides data models for various domains. The models fall in several classes. The class of model intended to be used is called an "Application Protocol" (AP).

2.1. AP 203

STEP Application Protocol 203 (AP 203), entitled "Configuration Controlled Design", provides a representation for the shape of a product and related data regarding versions, approvals, authors, etc. AP 203 contains several methods of specifying shape. The AP 203 method most commonly used (and the only one used in the FBICS) is boundary representation, or b-rep. In a b-rep, shape is expressed using both geometry and topology. The b-rep describes a shape by defining its bounding faces. Each face is a portion of a geometric surface, defined by its bounding edges. Each edge is a portion of a geometric curve, defined by bounding vertices. AP 203 geometry and topology are taken largely from STEP Part 42, the STEP basic resource for shape representation.

The FBICS is able to read AP 203 STEP Part 21 files, but since the FBICS does not include software for converting AP 203 to Parasolid (the solid modeler used in the FBICS), no further use is made of AP 203 models. Such software is commercially available.

2.2. AP 213 and A Language for Process Specification (ALPS)

STEP Application Protocol 213 (AP 213) is entitled "Numerical Control (NC) Process Plans for Machined Parts." AP 213 provides a method of describing sequential process plans. At the finest level of detail the description of operations is made in terms of text strings, so that the meaning of a plan is not computer-sensible. AP 213 provides for resource description but not resource allocation. AP 213 provides for describing alternatives but not for methods of deciding among them. Because of the missing capabilities just mentioned and other shortcomings, AP 213 is not used in the FBICS.

ALPS[2] (A Language for Process Specification) is a generic language for representing plans for discrete processes. It is based on a directed graph representation. ALPS includes modeling constructs to represent concepts needed in plans for discrete processes, including: sequence, parallelism, alternatives, synchronization, resource allocation, and time. Variables and expressions are included so that tests may be made for deciding among alternatives and to add flexibility in plan specification.

In ALPS, a plan consists of a set of nodes arranged in a directed graph. The directedness is expressed by the use of the successor relation among nodes. Alternatives and parallelism are expressed by split nodes that branch out to several successors. Each split node is matched by a join node that gathers the alternatives or parallel activities back together before the plan proceeds. ALPS, developed at NIST, was originally modeled in NIAM and did not include a specification for the syntax of plans. It was later modeled in the EXPRESS language. As a consequence of being modeled in EXPRESS, a syntax specification for ALPS process plans in STEP Part 21 files is available automatically. The FBICS uses plans written in STEP Part 21 files.

For the FBICS, two significant enhancements to generic ALPS were made: expressions were modeled in EXPRESS and a new type of split node was added, providing for the use of preconditions. Earlier versions of ALPS included support for three stages of a process plan. These stages are not used in the FBICS. The additional items defined to support the stages have been removed from the version of ALPS used in the FBICS. In the FBICS, ALPS is used for process plans at both the Shop level and the Workstation level. At each of those levels, subtypes of the ALPS primitive_task appropriate to the level are defined in extensions to the generic ALPS EXPRESS schema.

2.3. AP 224

A proposed standard set of machining features is defined in STEP AP 224, "Mechanical Product Definition for Process Planning Using Machining Features"[3]. This is largely a parametric library of machining features, but also provides for defining machining features in terms of a boundary representation and provides for related data, such as design_exceptions and requisitions. AP 224 provides definitions of base shapes and 51 parametric "manufacturing_features" including, for example: boss, chamfer, circular_pattern, compound_feature, counterbore_hole, countersunk_hole, edge_round, fillet, general_pattern, general_pocket, groove, hole, pocket, rectangular_pattern, rectangular_pocket, slot, spherical cap, and thread. The AP 224 feature library provides rich methods of describing details of features. Thirteen types of (planar) feature_profile are provided, for example. Feature_profiles may be swept in various ways to produce manufacturing_features. Twenty-nine types of feature_definition_item are provided. This includes feature details such as hole_bottom_condition, path, and taper. Having features defined in terms of a boundary representation is less useful because it is very difficult, if not completely infeasible, to identify machining operations for making entities found in boundary representations without grouping the entities (i.e., recognizing features). It is correspondingly difficult to generate tool paths.

The FBICS uses AP 224 for representing the shapes of designs and workpieces, as well as for machining features. The FBICS software drives the Parasolid modeler from AP 224 models. AP 203 is now widely used for exchanging shape descriptions. In the long run, it would be desirable in the FBICS to have an AP 203 to Parasolid converter, and to add the capability to generate a machinable AP 224 shape description from an AP 203 file describing the same shape. At least one automatic AP 203 to AP 224 converter is said to be commercially available.

3. THE FEATURE-BASED INSPECTION AND CONTROL SYSTEM

3.1. Earlier Work

In the Vertical Workstation of the NIST Automated Manufacturing Research Facility[4], a primitive CAD system was used with a design protocol[5]. This system constrained the user to design in terms of machining features (although the features did not conform fully to the definition just given, because several operations might be required to make one feature). The operations for cutting the features were automatically defined and partially sequenced by a generative process planner[6]. The individual components of this system were unsophisticated but they were very well integrated. Within a limited range of design, a part could be designed and cut within an hour using this system. An NC-code generator, controllers, and a feature recognition module which could extract features of the required sort from a boundary representation were included[7]. The Vertical Workstation System stopped just short of feature-based control, using short turnaround off-line NC code generation instead.

From experience with the Vertical Workstation, it became apparent that using machining features for design, while a good technique for a few special situations, is not a general solution for piece part manufacture. It is common to be able to machine a part more effectively (faster, cheaper, tighter tolerances, etc.) if machining features may be used that are not explicit in the design. Many other researchers have come to the same conclusion.

We then started work on the Off-Line Programming System (OLPS), an NC-code generation system that is intended to be used in a larger system in which machining features are defined separately from the design[8]. A library of parametric machining features was defined for use with OLPS[9].

3.2. The Current System

The current Feature-Based Inspection and Control System is a hierarchy of planners and executors which transform a high-level, feature-based description of a piece part into a sequence of plans that are executed to achieve material removal. The primary purposes of the FBICS are:

1. to demonstrate feature-based control in an open-architecture control system.
2. to serve as a testbed for solving problems in feature-based manufacturing, particularly the partitioning of manufacturing activities into separate activities and the definition of interfaces between activities.
3. to test the usability of STEP methods and models for machining and inspection.

The FBICS is based on the NIST Real-Time Control Systems (RCS) reference model architecture[10, 11, 12]. RCS specifies a task-oriented problem analysis that results in a hierarchy of control modules, each which decomposes higher-level tasks into lower-level actions. Control modules at the bottom of the hierarchy interface directly to sensors and actuators. The FBICS architecture, shown in Figure 2, follows the RCS reference model.

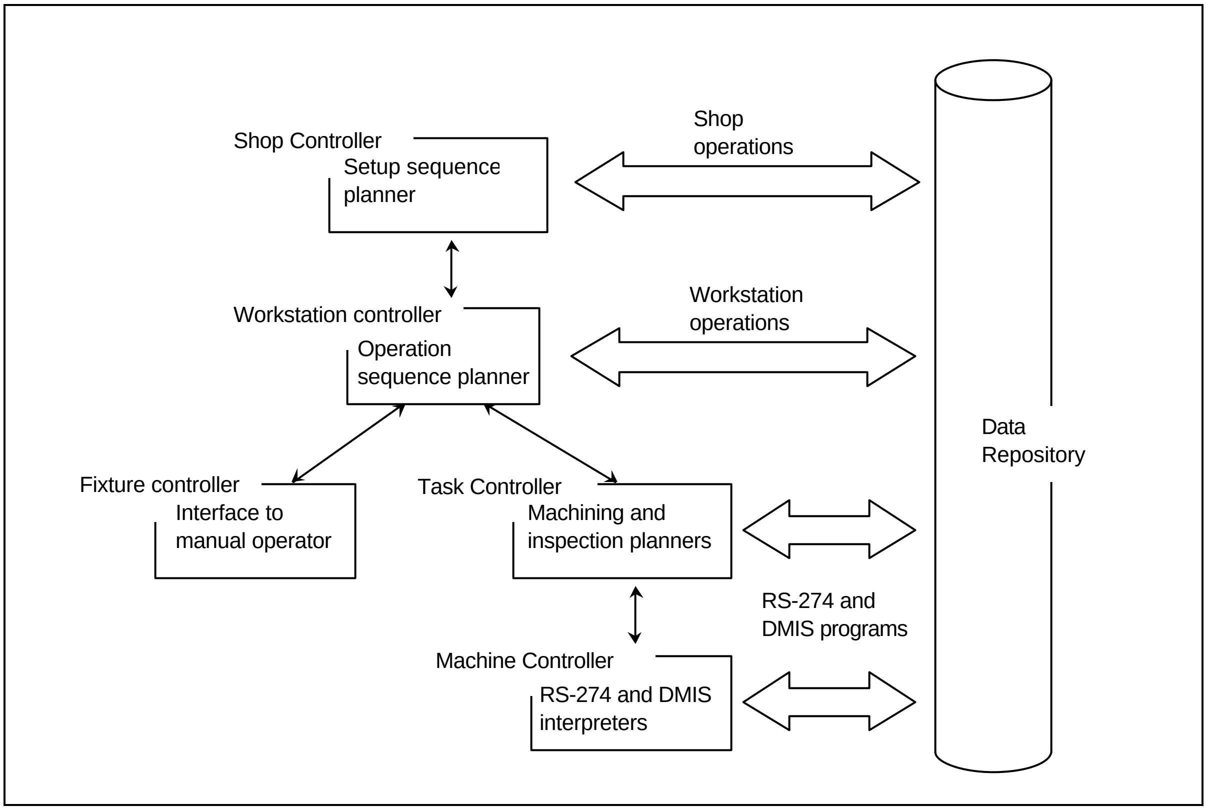

Figure 2. The FBICS Architecture.

The lowest FBICS control module, the Machine Controller, corresponds to a conventional numerical control machine with direct numerical control (DNC) and networked file system capabilities. DNC allows the Task Controller to send requests to the Machine Controller to run RS-274 numerical control (NC) programs for machining[13], or DMIS[14] programs for inspection. The Data Repository is simply a set of files available via a networked file system.

3.3. Data Types

Twelve different data sets drive the FBICS. The first five sets of data are independent of the part to be machined. The other seven types of data depend upon the work being done in a specific setup of a part. These files are listed in Table 1, and their description is detailed in the subsequent text.

Data Type	**Description**
Shop Option	preferences for inspection strategies
Task Options	preferences for inspection (number of points, etc.) and machining
Tool Catalog	a catalog of types of tools that are usually available
Tool Inventory	an inventory of specific tools that are currently available
Tool Use Rules	rules for setting speed, feed, stepover, pass depth, and coolant use
Machining Features	a description of the machining features to be cut away
Shop Process Plan	a plan for one of several setups
Workstation Process Plan	a plan that includes at least one operation to make each machining feature in the setup
Fixture	a description of the fixture that holds the workpiece being machined
Workpiece	a description of the shape of the workpiece as it is before machining is begun
Design	a description of the intended shape of the workpiece after machining
Setup	a list of files associated with a setup; locations of fixtures, workpiece, design, and features

Table 1. Data types used in the FBICS.

In addition to the data types liste in Table 1, all of which are modeled in EXPRESS, data files are also used for executable operations (EXPRESS), NC code (RS-274), inspection code (DMIS), and graphics (non-standard).

Shop Options include four inspection options: what to do if a feature is out of tolerance, what to inspect, when to inspect, and how intensively to inspect.

Task Options include machining options and inspection options. Machining options describe user preferences for retract height, tool change position, and other information specific to a particular machine or process. Inspection options include the number of points to use for each feature at each level of inspection intensity (low, medium, or high).

The *Tool Catalog* describes the kinds of tools found in the facility, some or all of which may be found on the particular machine. This includes the tool type (e.g., end mill, twist drill), nominal dimensions (e.g., length and diameter), material from which the tool is made, materials which the tool can cut, number of flutes, maximum rotational speed, and maximum number of reworks.

The *Tool Inventory* lists the tools available for use on the particular machine. In addition to the information in the Tool Catalog, each instance of a tool in the Tool Inventory includes the unique tool identifying number, the location in the machine carousel, the actual dimensions, the time in service, and the number of reworks.

Tool Use Rules are expressions written in terms of tool and material variables (diameter, number of flutes, etc.) that are evaluated during operation planning to determine feed rate, spindle speed, horizontal stepover, vertical pass depth, and coolant use.

Machining Features are the basic features to be machined, as described earlier.

The *Shop Process Plan* is a sequence of run-setup operations, as described earlier.

The *Workstation Process Plan* is a sequence of machining and inspection operations to be performed at the Workstation level in the control hierarchy, as described earlier.

Fixture information includes the geometric description of the fixturing devices.

The *Workpiece* description includes the shape of the part before processing. This is the “as is” description.

The *Design* description includes the shape of the part after processing. This is the “to be” description.

Setup information describes the files needed by the Workstation and Task Controllers in each setup, and the transformations relating coordinates of features, fixture, workpiece, and design to world coordinates.

Each type of data is contained in a file globally accessible in the Data Repository shown in Figure 2. Currently, this is simply a networked file system. Files in the Data Repository are STEP Part 21 exchange files. For each file type, there is an EXPRESS schema providing a model of data of that type. Four types of data are prepared using STEP standards. The remaining five types do not have STEP standard representations. For consistency, they are defined in the FBICS in EXPRESS, following the STEP philosophy. The file types are listed in Table 2.

Data Type	**EXPRESS Schema**
Shop Options	a special-purpose schema written for the FBICS system
Task Options	a special-purpose schema written for the FBICS system
Tool Catalog	a subset of the ISO 13399 model with probes added
Tool Inventory	Tool Catalog with a tool_instance entity added
Tool Use Rules	expressions plus a special-purpose schema written for the FBICS
Machining Features	STEP AP 224
Shop Process Plan	ALPS plus run_setup
Workstation Process Plan	ALPS plus three-axis machining and inspection operations
Fixture	STEP AP 203

Workpiece	STEP AP 224
Design	STEP AP 224
Setup	a special-purpose schema written for the FBICS system

Table 2. Data representation for FBICS files.

The EXPRESS schema used in the FBCS for STEP AP 203 is the AP 203 schema provided by STEPTools Inc. with functions and “where” clauses removed. STEPTools, Inc. is a commercial venture that provides tools for dealing with STEP methods, models, and data. The EXPRESS schema used in the FBCS for STEP AP 224 is a schema provided by Len Slovensky, owner of AP 224. In STEP terms it is an Application Resource Model (ARM) type of model. The Tool Catalog and Tool Inventory are a subset of the model built in EXPRESS by the NIST Manufacturing Systems Integration Division and contractors of that division[15], with probes and a tool_instance entity added.

4. SCENARIO

To understand the interaction of control modules in the FBICS and the use of STEP data throughout the feature-based machining and inspection process, it is helpful to follow an example scenario. Figure 3 shows a sample part (a clevis) that is to be machined and inspected during this scenario. Manufacturing begins with the definition of all the data required to specify the part and the resources required to make it, as listed in Table 1, in the file formats listed in Table 2. This would typically be done using computer-aided design/manufacturing (CAD/CAM) software. However, since commercial programs that generate files in the various FBICS formats are not available, this data is generated by hand using a text editor.

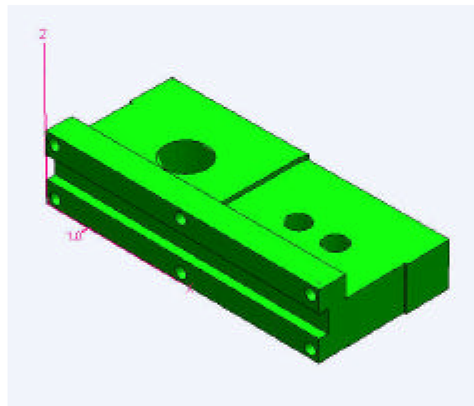

Figure 3. Sample part (a clevis) referenced in the scenario.

At the top level of the FBICS, shown as the Shop Controller in Figure 2, the user submits a request to plan the part. The request includes the files containing the STEP data definitions for the part and resources available. At the top-level prompt, this looks like:

```
SHOP => plan_part(data/clevis2/out2_224.stp, ON,
data/clevis2/in1_224.stp, pl1, fe1, fi1, se1)
```

The first argument to the plan_part request is the STEP Part 21 file containing the AP 224 features for the part to be made. The second argument means that the file that is the third argument already exists. If the second argument of “OFF” the file will be generated automatically. The third argument is the AP 224 description of the raw stock, in this case a 6.5-inch by 3.0-inch by 1.5-inch block of aluminum. The remaining arguments are the prefixes to use for files generated for process plans, feature assignments, fixturing, and setup.

The Shop Controller plans a series of setups and a series of tasks for each setup. Multiple setups may be required to access all the features given the kinematic configuration of the machine and the tooling available. The output of the planning phase of the Shop Controller is data, including a process plan for the Shop Controller and one or more process plans for the Workstation Controller, for the setups required to machine and inspect the entire part. In the FBICS, the process plan is described using the EXPRESS schema for ALPS, and written in STEP Part 21 physical file (ASCII) format. An example of the Shop-level process plan for the clevis part is shown in Listing 1.

The Shop Controller executes this process plan, and sends commands to the Workstation Controller to execute the plan for each setup. Data used by the Workstation Controller includes descriptions of each setup, as shown in Listing 2, and process plans containing references to the individual features able to be machined or inspected at each setup, as shown in Listing 3. Currently, FBICS fixturing is done manually, but is indicated architecturally as a “man in the loop” box labeled “Fixture Controller” in Figure 2.

```
#10 = PLAN('pl1_shop','version 1',(),(),'SHOP',’','',(#20,#30,#40,#50,#60,#70,#80));
#20 = START_PLAN_NODE(#10,1,$,$,(#30),());
#30 = PRECONDITION_SPLIT_NODE(#10,2,$,$,(#40),());
```

```
#40 = RUN_SETUP(#10,3,$,$,(#50),(),1,'se1_1.stp',());
#50 = RUN_SETUP(#10,4,$,$,(#60),(),1,'se1_2.stp',());
#60 = RUN_SETUP(#10,5,$,$,(#70),(),1,'se1_3.stp',());
#70 = PRECONDITION_JOIN_NODE(#10,6,$,$,(#80),());
#80 = END_PLAN_NODE(#10,7,$,$,(),());
```

Listing 1. *Shop process plan for complete clevis part, calling for three independent fixturing setups.*

```
#10 = CARTESIAN_POINT_SETUP(0.,0.,0.);
#20 = CARTESIAN_POINT_SETUP(6.5,1.5,3.);
#30 = CARTESIAN_POINT_SETUP(0.,0.,0.);
#40 = DIRECTION_SETUP(1.,0.,0.);
#50 = DIRECTION_SETUP(0.,0.,1.);
#60 = AXIS2_PLACEMENT_SETUP(#30,#50,#40);
#70 = AXIS2_PLACEMENT_SETUP(#100,#90,#80);
…
#190 = BOX_SETUP(#10,#20);
#200 = FILE_NAMES('se1_1.stp','data/clevis2/out2_224_1.stp','fi1.stp',
    'pl1_1.stp','fe1_1.stp','data/clevis2/in1_224.stp',$,$);
#210 = SETUP_SPEC(#200,#70,#110,#150,#60,#190);
```

Listing 2. *Setup description for the first fixturing, which includes coordinate geometry and bounding box. Ellipses indicate information similar to that already provided, omitted for brevity.*

```
#10 = PLAN('pl1_1','version 1',(),(),'WORK','a part','generated by FBICS', (#20,#30, …));
#20 = START_PLAN_NODE(#10,1,$,$,(#30),());
#30 = PARAMETERIZED_SPLIT_NODE(#10,2,$,$,(#50,#70, …),(),0,.SPLIT_TIMING_SERIAL.);
#40 = PATH_JOIN_NODE(#10,3,$,$,(#270),());
#50 = FINISH_MILL(#10,4,'',$,(#60),(),1,(),'END-MILL-0.5-2',
    0,3437,17.189240808531,.T.,.F.,0.25,0.25);
#60 = INSPECT_FEATURE_GEOMETRY(#10,5,'',$,(#40),(),1,.MEDIUM.,(),
    'PROBE-2.5-0.12', 0,10.);
…
#150 = TWIST_DRILLING(#10,14,'',$,(#160),(),1,(),
    'DRILL-0.1966-2',5,5200,5.1116,.T.,.F.,0.1966);
…
#410 = END_PLAN_NODE(#10,40,$,$,(),());
```

Listing 3. *Workstation process plan for clevis part, first setup.*

The Workstation Controller traverses its process plan (such as shown in Listing 3) and sends descriptions of individual machining or inspection operations to the Task Controller. The Task Controller takes each operation description, associates tool usage rules and other user preferences for process parameters (e.g., retract height) to each feature accessible in this fixturing setup, and generates NC code for machining or DMIS code for inspection. The machining or inspection code is stored to the Data Repository, and requests to the machine tool controller are made to run the programs.

5. SUMMARY

The FBICS is a feature-based control system that interfaces with controllers to legacy machine tools programmed in RS-274 for machining or DMIS for inspection. It provides the ability to program machining and inspection using manufacturing features as defined in ISO standard 10303 (STEP), Application Protocol 224. The purposes of the FBICS are to demonstrate feature-based control in an open-architecture control system; to serve as a testbed for solving problems in feature-based manufacturing; and to test the usability of STEP methods and models. The FBICS philosophy is to use STEP standards where they exist, and to create data formats consistent with the STEP approach where they do not exist. A prototype system has been built that addresses simple 3-axis machining and inspection.

6. REFERENCES

1. Kramer, Thomas R.; *Issues Concerning Material Removal Shape Element Volumes (MRSEVs)*; International Journal of Computer Integrated Manufacturing; Vol. 7, No. 3; 1994; pp. 139-151; (also published as NISTIR 4804; National Institute of Standards and Technology; March 1992)

2. Catron, Bryan; Ray, Steven R.; *ALPS - A Language for Process Specification*; International Journal of Computer Integrated Manufacturing; Vol. 4, No. 2; 1991; pp 105 -113

3. Slovensky, Len; *Industrial Automation Systems and Integration - Product Data Representation and Exchange - Part 224: Application Protocol: Mechanical Product Definition for Process Planning Using Machining Features*; Committee Draft; International Organization for Standardization; 1995

4. Kramer, Thomas R.; Jun, Jau-Shi; *Software for an Automated Machining Workstation*; Proceedings of the 1986 International Machine Tool Technical Conference; September 1986; Chicago, Illinois; National Machine Tool Builders Association; 1986; pp. 12-9 through 12-44

5. Kramer, Thomas R.; Jun, Jau-Shi; *The Design Protocol, Part Design Editor, and Geometry Library of the Vertical Workstation of the Automated Manufacturing Research Facility at the National Bureau of Standards*; NBSIR 88-3717; National Institute of Standards and Technology (formerly National Bureau of Standards), January 1988

6. Kramer, Thomas R.; *Process Plan Expression, Generation, and Enhancement for the Vertical Workstation Milling Machine in the Automated Manufacturing Research Facility at the National Bureau of Standards*; NBSIR 87 - 3678; National Institute of Standards and Technology (formerly National Bureau of Standards); November 1987

7. Kramer, Thomas R.; *A Parser that Converts a Boundary Representation into a Features Representation*; International Journal of Computer Integrated Manufacturing; Vol. 2, No. 3, May-June 1989; pp. 154-163; (also published as NISTIR 88-3864; National Institute of Standards and Technology; September 1988)

8. Kramer, Thomas R.; *The Off-Line Programming System (OLPS): A Prototype STEP-Based NC-Program Generator*, proceedings of a seminar Product Data Exchange for the 1990's; New Orleans, Louisiana; NCGA; February 1991; Vol. 2

9. Kramer, Thomas R.; *A Library of Material Removal Shape Element Volumes (MRSEVs)*; NISTIR 4809; National Institute of Standards and Technology; March 1992

10. Albus, James S.; *A Theory of Intelligent Systems*; Control and Dynamic Systems; Vol. 45; 1991; pp. 197 - 248

11. Albus, James S.; McCain, Harry G.; Lumia, Ronald; *NASA/NBS Standard Reference Model for Telerobot Control System Architecture (NASREM)*; NIST Technical Note 1235, 1989 Edition; National Institute of Standards and Technology; April 1989

12. Albus, James S.; *A Reference Model Architecture for Intelligent Systems Design*; NISTIR 5502; National Institute of Standards and Technology; September 1994

13. National Center for Manufacturing Sciences; *The Next Generation Controller Part Programming Functional Specification (RS-274/NGC)*; Draft; NCMS; August 1994

14. CAM-I, Inc., *Dimensional Measuring Interface Standard*, Revision 3.0, ANSI/CAM-I 101-1995, 1995

15. Jurrens, Kevin K.; Fowler, James E.; Algeo, Mary E.; *Modeling of Manufacturing Resource Information*; NISTIR 5707; National Institute of Standards and Technology; July 1995