The Globalization of Manufacturing in the Digital Communications Era of the 21st Century: Innovation, Agility and the Virtual Enterprise

Proceedings of the Tenth International IFIP WG5.2/5.3 Conference
PROLAMAT 98

#49

Extension of the Standard ISO 10303 (STEP) for the exchange of parametric and variational CAD models

M. J. Pratt
National Institute of Standards & Technology
Manufacturing Systems Integration Division
Gaithersburg, MD 20899, USA
Tel: +1 (301) 975-3951, Fax: +1 (301) 975-4482
E-mail: pratt@nist.gov

Abstract

The first release of the International Standard ISO 10303 (STEP) was made in 1994. It permits the exchange between dissimilar CAD systems of 2D drawings and of product models of the boundary representation type, including degenerate forms such as surface models and wireframes. However, the standard cannot currently capture and exchange the parametrization and constraint information associated with the models generated by most modern CAD systems. The loss of this information during a model transfer makes it difficult or impossible to edit the model in a receiving system subject to the intentions of the original designer. The paper outlines the consequences of this loss of information, and describes the current status of work aimed at enhancing the ISO 10303 standard to enable its retention. Some of the technical problems arising in this work are discussed, and possible solutions are outlined.

Keywords

Parametric models, constraint-based models, CAD data exchange, design intent, ISO 10303, STEP

1 INTRODUCTION

In 1994, ten years after work on it began, the first release occurred of the International Standard ISO 10303 (ISO 1994a, Owen 1997), informally known as STEP (STandard for the Exchange of Product model data). The length of time taken only partially reflects the fact that the development of an International Standard is an inherently slow process, with built-in delays for the achievement of international consensus. Time was also necessary for the creation of a significant infrastructure for what will eventually be a very wide-ranging standard. In fact, the scope of ISO 10303 is electronic data exchange for any form of discrete product, and it covers all stages of the product life-cycle. This ambitious coverage necessitated major restructuring of the standard as it developed and as the problems associated with its scale became manifest. STEP is currently under active expansion to encompass new product ranges and life-cycle stages, and its architecture is still evolving.

A key element of any product model is its shape representation. The forms of representation currently handled by STEP include 2D drawings and 3D models of the wireframe, surface and boundary representation solid types (Hoffmann 1989). However, for good reasons it was not possible to make major changes in the technical content of STEP for some considerable time before its initial release in 1994. Thus, major developments in computer aided design (CAD) shape modeling technology during the late 1980s and early 1990s were not taken into account in the first release of the standard. The capabilities in question allow the creation of models with parametrization, geometric constraints and form features. Information relating to any of these capabilities is currently lost in the STEP transfer of CAD models, since the standard does not provide for its capture and transfer. This paper describes an effort currently under way to provide extensions for STEP in the interests of preventing this loss of information.

2 PARAMETRIC AND VARIATIONAL MODELS

Parametrization is the association of named variables, or expressions involving named variables, with certain quantities in a model. In a shape model these quantities are usually dimensions. The use of parameters allows the definition of **families of parts**, from which individual members or **instances** can be generated by the assignment of specific values to the parameters.

Parametrization provides information about the freedom that is available in a design. On the other hand, there has to be some restraint on that freedom in the interests of design functionality. For this reason, geometric constraints are used in many CAD systems. These are relationships, imposed by the designer, between shape elements in the model. Examples include a perpendicularity constraint between two planar faces, or a tangency constraint between a line and a circle.

The word **variational** is sometimes used to denote the type of model that exhibits both parametrization and constraints. In a variational model, changing a parameter value will often trigger the solution of a set of simultaneous constraint equations as the system seeks a new configuration of the model that accommodates the change whilst preserving functionality as expressed by the constraints.

Most modern CAD systems also allow **design by features**. In this context features, like constraints, are associated with design functionality. They may be thought of as "shape macros" that generate local shape configurations on the part for specific functional purposes, e.g., cooling fins or bearing housings. Constraints may be used to define relationships between geometric elements in feature definitions, or to position and orient features appropriately with respect to other elements in a model.

There is a rich mine of "behavioral" information in these aspects of a CAD model. With STEP as it currently exists, this information is lost in any model exchange, and what is transmitted is simply a "snapshot" of the model at some particular time in its history, expressed as far as its shape is concerned by nothing beyond geometry and topology. Editing the model is a frequent requirement in design optimization, or for redesign in response to feedback from downstream applications such as manufacturing planning. If this happens after a STEP transfer, important aspects of the original design intent have been lost. Parametrization information is not present, and hence there is no indication concerning intended design freedom in the model. Neither is there any indication of which faces are grouped into features, or of the way elements of the model should be constrained with respect to each other. Consequently, modifications made to the model are likely to violate the functionality of the modeled product, because no guidance is available as to what are acceptable changes.

On the other hand, if STEP can be enhanced for the transfer of the information that is currently lost, the received models will no longer be mere snapshots; they will contain dynamic elements that cause them to behave in the receiving system according to the original design intent. Thus they will be editable by varying parameters, the constraints will be maintained, and feature-based operations will be available to avoid the necessity of manipulating individual low-level geometric or topological elements of the model. Significant operator time will be saved by this added convenience in redesign capability, whereas at present it is necessary to try to guess the design intent and use trial and error methods.

3 EXPLICIT AND HISTORY-BASED MODELS

Two methods are available for the representation of models with parametrization and constraints:

- The first is based on an **explicit model**, fully expressed in boundary representation (Brep) form (Hoffmann 1989). Parameters are associated with dimensional elements in the model, and constraints are explicitly specified between particular elements such as faces or edges.
- The second uses an **implicit** or **history-based representation**, in which the primary representation of the model is in terms of the sequence of operations used to construct it. In this case there is no explicit information about the model shape at all – that information does not become available until after the

specified operations have been performed, the result of this being an **explicit** or **evaluated** model.

Most existing CAD systems use a hybrid approach combining elements of both methods. For example, an explicit 2D profile may be defined in terms of line segments and circular arcs, with explicit constraints to ensure perpendicularity, parallelism and tangency between geometric elements as appropriate. A solid may then be created by means of an **extrusion** operation, acting on the profile. The volume defined is that swept out by the area of the profile as it is translated through space. Note that the volume is defined by one explicit entity (the profile) and one operation (the extrusion). The only explicit geometry in the solid is therefore the boundary of one end face of the extrusion. All the other faces do not exist (or only exist **implicitly**) in this definition. It is not until the extrusion is actually performed and all the new faces and edges generated that a fully explicit model results. Release 1 of ISO 10303 is capable of transferring the fully explicit evaluated model, but that model contains no information as to how it was generated. It is therefore not possible in a receiving system to rerun the operation, for example with a different value of the extrusion length parameter.

As stated earlier, the current version of ISO 10303 cannot transmit models of either of the types described above. The Parametrics Group within ISO TC184/SC4, the standards committee developing and maintaining the standard, has been created to provide the necessary enhancements for some future release of ISO 10303. The two different modelling paradigms to be accommodated require different approaches. These are outlined below.

3.1 Explicit model approach

For this approach, ISO 10303 already provides the basic shape model that is required. Additional resources must be provided, however, to allow the association of parameters with dimensions in the model and the assertion of constraint relationships between geometric elements of the model. Work is already in progress on developing these new capabilities. A parametrization mechanism now exists, and a range of constraint types has been defined. These include parallelism, perpendicularity, coaxiality, tangency, symmetry and incidence (all of which are **logical constraints**, having no associated dimensional value), together with several **dimensional constraints** including distance, angle and radius. The following minor problems remain to be tackled:

- The semantic distinction between a dimension and a dimensional constraint has to be made very clear, and the treatment of dimensional constraints has to be made compatible with the representation of dimensions that currently exists in STEP.
- Several possible mechanisms are available for the representation of mathematical expressions to relate values of parameters in a model. The ultimate choice is important, but there are many conflicting considerations to be taken into account.

Work is proceeding well on this new ISO 10303 resource. The concepts involved are easily captured in the information modelling language EXPRESS used by the standard (ISO 1994b, Schenk and Wilson 1994). Further, the parametrization and constraint information can be provided in such a way that it supplements models of the type currently exchanged using ISO 10303. The new entities defined have pointers into a geometry/topology model, but require none in the reverse direction. It is therefore not necessary to change the existing geometry and topology resources of ISO 10303, and the resulting enhancement will be upwardly compatible with Release 1 of the standard.

3.2 History-based model approach

For history-based models the situation is very different. Such models are defined procedurally, by specifying the sequence of operations needed to construct them. Standardization of this type of model therefore requires the compilation of a list of constructional operations covering the capabilities of commercial CAD systems. Each operation must be defined in terms of its input(s), its output(s) and its functionality. For example, an operation may be provided for generating a line parallel to another given line, in a two-dimensional context. In this case the input is a reference to the given line, and the value of a distance. The output is a reference to the constructed parallel line. The functionality is a description of the relationship between the input and the output. In this example, it must define the relation between the "senses" or directions of the first and second line, and indicate how the two possible output lines are distinguished, i.e., how it is determined which side of the original directed line the constructed line is to lie. In a slightly more general case, the distance associated with the constructed parallel line may have a parameter associated with it; in the final model, variation of this parameter can then be used to generate members of a family of parts.

ISO 10303 currently possesses very little capability of the type described above. The only such resources provided in the current release of the standard are

- A limited constructive solid geometry (CSG) capability (Hoffmann 1989), for defining product shapes in terms of Boolean operations on previously existing volumes. These volumes may include the usual CSG primitive shapes (e.g., blocks, cylinders, cones, etc.,) and also previously existing boundary representation volumes (Hoffmann, 1989). The primary limitation here is that all the volumes involved must have fixed dimensions, so that the creation of parametrized families of parts is impossible.
- Several geometric entity types that are essentially procedural in their definitions. These include parallel offset curves and surfaces (defined in terms of a base curve or surface and an offset distance), and also volumes defined by extrusion or rotation of specified two-dimensional profiles. As in the case of the CSG resource, there is no parametrization capability.

Although Boolean operations, extrusions and rotations are frequently used constructional operations that are widely implemented in current CAD systems, there are many other such operations used in practice that have no counterparts at present in ISO 10303. Furthermore, the lack of a parametrization capability is a severe shortcoming of the standard as it currently exists. For these reasons, it will be necessary to develop a new ISO 10303 resource, significantly extending and generalizing the existing limited history-based modelling capability described above. Some of the considerations involved in this task are discussed below. The hybrid capability characteristic of most CAD systems will require the history-based model representation to be compatible with the resource for explicit parametrization and constraints described earlier in Section 3.1.

Existing proposals for standard CAD modeler APIs

Many CAD systems provide an application programming interface (API) allowing external software to interface with the system and make use of its internal functionality. The interface definition often takes the form of a library of callable procedures or subroutines. There have been several efforts directed towards the development of a standardized API, one that can be implemented with a wide variety of different CAD systems. This will allow an external software application to interact with any system possessing such an interface with no need for modification.

The new capability required in ISO 10303 demands the specification of just such an API. This will define a set of operations that can be mapped onto the native API of the receiving CAD system, giving access to its constructional functions and also to its query facilities. The latter are sometimes needed for determining low-level details of the outcome of one operation before the next operation can be appropriately specified. An example might be the generation of a blended edge on the model, which may result in the creation of a composite surface, built up from regions of several different simple surfaces. An interrogation is needed in such a case to determine the number and specification of the simple surfaces involved. Subsequent modelling operations may depend on the results of such queries.

A survey of proposals for standard CAD modeler APIs has been undertaken to determine whether any of them form a suitable basis for the new ISO 10303 resource. The documents surveyed include the following:

- Part 31 of ISO 13584 – a resource of the Parts Library standard (ISO 1996), developed by the same ISO subcommittee that is responsible for ISO 10303. This document defines an API suitable for specifying history-based models of families of standard parts, for use in conjunction with publicly accessible libraries of standard parts. It provides only for parts with simple geometry, and is purely history-based, i.e., it does not allow for the use of "hybrid" models using a combination of explicit and history-based representations. It does however allow the representation of parametrized models.
- The CAM-I Applications Interface Specification (CAM-I 1994) – developed by the organization Consortium for Advanced Manufacturing, International (CAM-I). This provides for the creation of hybrid models with the full range of ISO 10303 geometry (including non-uniform rational B-splines or NURBS), but excludes parametrization and explicitly defined constraints.
- OLE for D&M (DMAC 1996) – a "Design and Modeling" interface based on the Microsoft Object Linking and Embedding (OLE) technology. This is

under development by a consortium of CAD system vendors oriented towards PC-based systems. It covers the whole range of facilities needed for explicit models in ISO 10303 (including parametrization, constraints and features), but provides only query capabilities. It will therefore not be possible to use this interface in regenerating a history-based model in a receiving system, since no constructional facilities are available.

Some other proposals have also been examined, most of them less well developed. It appears that a standardized API for history modelling in ISO 10303 can be created as a synthesis of the three listed above, and work on this task will shortly be started. The full survey mentioned above can be found on the World Wide Web (Pratt 1998).

Representation of constructional operations in ISO 10303

Once the set of operations to be provided in the API has been chosen, the question arises as to how they should be represented. Most of the existing known proposals for standard APIs represent such operations as language-independent procedure specifications, and provide bindings to widely-used programming languages such as FORTRAN, C++ or Java. On the other hand, ISO 10303 at present deals almost exclusively with entities, not methods.

To take an example, the **`line`** entity in ISO 10303-42 (Part 42 of STEP, the primary resource for geometry and topology) is defined as follows:

```
ENTITY line
    SUBTYPE OF (curve);
    PNT : cartesian_point;
    DIR : vector;
END_ENTITY;
```

This is slightly simplified – a rule has been omitted that requires the dimensionality of the point and the direction vector to be the same (either 2 or 3). The line as thus defined is unbounded, and the interpretation of its definition should be quite clear. Note that the EXPRESS language (ISO 1994b, Schenk and Wilson 1994) allows the definition of subtype/supertype relationships, and this entity inherits from further up the hierarchy a **`representation_context`** that defines its local coordinate system, which may be either 2D or 3D. Note also that any instance of this entity, with specific values for the **`PNT`** and **`DIR`** attributes, simply asserts a "snapshot" relationship between the line, the point and the direction.

For history-based modelling, an operation for constructing a line through two points could be modeled in EXPRESS as

```
ENTITY constructed_line_two_points
    SUBTYPE OF (constructed_curve);
    POINT1 : cartesian_point;
    POINT2 : cartesian_point;
END_ENTITY;
```

This is a different species of line definition, not currently provided by STEP, and assigned to the hypothetical new supertype **`constructed_curve`**. Clearly, the two points specified in an instance of this new entity are its defining data, and the definition is at first sight a fairly routine one. However, the intended *interpretation* of an instance of this entity in the receiving system would differ greatly from the semantics of the standard ISO 10303 line definition. If interpreted using the current approach, it would simply be part of the description of a static model. Then editing either of the points in the receiving system would in general give rise to an inconsistency in the snapshot model – if moved, they will probably no longer lie on the line. Interpretation as a constructional entity, on the other hand, will require the relationships defined by the entity to be treated as **constraints** in the receiving system, so that if either of the points is moved the line changes accordingly. In this case the model will have been transferred with a dynamic characteristic preserving an aspect of the designer's intent. When the entity is first passed to the receiving system, the appropriate native API function or functions should therefore be called to create a line in terms of the two specified points and to apply the geometric constraints implicit in the definition.

If constructs of this type are to be used in a history-based model, the sequence in which the implied operations are performed is important. A list of such entities has more the appearance of a simple collection of geometric elements than of a set of instructions for creating a model. It will therefore be necessary to specify a sequence for such entities, to ensure the correct order of evaluation.

However, the use of constructional entities as shown above is only one way of representing model history. It is the approach that is most consistent with current ISO 10303 modeling methodology. By contrast, most other proposals for standardized APIs for history-based modelling have defined the API as a library of procedures. Adopting this approach, we could specify the operation of constructing a line through two points in language-independent terms by

```
line = line_2_pnt(point1, point2),
```

which corresponds closely to the header of the Parts Library procedure for this purpose (ISO 1996). For the procedure-based approach, the input and output parameters of the procedure need to be specified with regard to their types, and a description of the intended functionality given. Some of this information is of course implicit in the entity-based approach, where attributes correspond to input parameters and the entity itself corresponds to the output parameter. In either case some description is necessary to clarify the semantics. For example, it is not clear from either the **`constructed_line_two_points`** entity definition or the **`line_2_pnt`** procedure heading whether the constructed line is unbounded or is bounded by the two points.

It appears that the procedure-based approach is preferable for history-based modelling, for the following reasons:

- It maps more directly onto the native API of the receiving CAD system. The functionality of that API provides access to the system capability needed for regenerating an explicit model following an exchange.

- It provides less possibility for confusion; there are apparent disadvantages in using a very similar entity-based format for two classes of modelling entities, one having semantics related to constructional operations and the other not.
- There is no obvious way of extending the entity-based approach to cover the query capabilities required in the API for history-based modelling.

However, majority opinion within the STEP development community may decide that the entity-based approach should be used for consistency with previous practice. This decision will need to be made in the near future, since the requirements analysis phase for the history-based modelling extensions to ISO 10303 are almost complete at the time of writing.

Persistent naming

Another problem that will need to be addressed in the context of history-based modelling is that of **persistent naming**. It may be illustrated by an example. Suppose the designer generates a geometric model of an object by means of the following operations:

- Create a rectangular block.
- Create a hemicylindrical slot across the top face of the block, subdividing that face into two unequal parts.
- Round the left-hand edge of the slot.

Now imagine that the model is transferred into another system, which is required to regenerate it from the transmitted constructional history. The receiving system should have no trouble in creating the block. If we assume that the slot was originally created by Boolean subtraction of a cylinder positioned in terms of absolute coordinates, then it should also be easy to reproduce that operation. However, the edge must then be rounded. In the generating system, the designer probably identified the edge to be modified by picking it from the graphical representation on his screen. In the receiving system, the corresponding edge has to be identified automatically in order to regenerate the model. How can this be done? It is not possible to rely on descriptors such as “left” and “right”, because these are view-dependent and a rotation of the model may interchange their meanings. Can the system determine the desired edge in terms of the surfaces that intersect there? No, because the two surfaces concerned may have more than one intersection, as is the case in this example, where the slot surface is defined by a cylinder having two intersections with the relevant planar face of the block. In fact, the provision of a means for determining the geometrical or topological entities originally picked by the designer in a manner suitable for automatic regeneration is a difficult and delicate problem, one that has not yet been fully solved by the CAD system developers themselves. The same problem may arise within a single system if a model is regenerated following a design modification.

Two seminal papers on the persistent naming problem are those by Kripac (1995) and Capoyleas et al. (1996). They suggest heuristic methods for persistent entity identification; these will work for much of the time but (as shown in the second paper in particular) there are many cases where more sophistication will be needed.

Raghothama and Shapiro (1997) provide what is believed to be the first formal analysis of the nature of this problem and suggest a solution that can be guaranteed to work within certain well-defined limitations.

Clearly, any standardized representation for history-based models developed in the context of ISO 10303 will have to address the persistent naming problem. It may not be possible to provide a complete solution, since this does not appear to exist in any practical system at present. However, some mechanism must be provided that will give correct results in most cases and also, of course, be compatible with the approaches taken by the CAD system developers. The fact that these commercially developed approaches are closely-guarded proprietary secrets adds an extra dimension of difficulty for the developers of the standard.

Other related problems of "indirect referencing" also need to be solved in the STEP treatment of history-based models. Consider a situation where a model of a car is to be exchanged. The car has four wheels. A procedural model is used to represent the shape of a wheel, and then this is used to create four instances of the wheel, one at each corner of the car. Each instance consists of a reference to the basic wheel model with a positioning and orienting transformation. Now it is desired to refer to a wheel-bolt hole in the front left-hand wheel of the car, and to set up an association between it and the corresponding bolt by linking their mating faces. However, in a history-based model there is no explicit face information; such details do not get generated until the operations specified by the model are actually performed, giving rise to an explicit model. This is just one further example of the need for a general mechanism enabling low-level explicit elements of a history-based model to be referred to in an unambiguous way.

4 FEATURES

The topic of features has not so far been addressed in detail in the work described, but it is a natural extension once the basic mechanisms for parametrization and constraint representation have been agreed. A feature class may be defined in terms of parametrized dimensions and constraints between constituent lower-level elements. However, regarded as an entity in its own right a feature may also be positioned and oriented parametrically in a model, and constrained with respect to other model elements or features. There are no major conceptual difficulties in extending the present work to cover the exchange of feature information.

5 CONCLUSIONS

Requirements for extensions to the ISO 10303 standard (STEP) for CAD data exchange have been described. The new capabilities will allow the standard to handle CAD models with parametrization and constraints. There are two approaches to the representation of such models, one explicit and the other based on representation of constructional history of the model. The second gives rise to greater problems in the ISO 10303 context. The nature of some of those problems has been discussed, and possible solutions outlined. It should be emphasized that, even in the absence of these capabilities Release 1 of ISO 10303 (ISO 1994a) is meeting with increasing success in industrial use as initial teething problems are identified and overcome. The proposed new enhancements will substantially improve the utility of this form of data exchange.

6 ACKNOWLEDGMENTS

The author acknowledges contributions by other members of the ISO TC184/SC4 Parametrics Group towards developing the new ISO 10303 capability, in particular Noel Christensen, Tom Kramer and Akihiko Ohtaka.

7 REFERENCES

CAM-I (1994) Application Interface Specification (AIS), Version 2.1; Volume 1, *Functional Specification* and Volume 2, *C Language Binding,* Report R-94-PM-01, Consortium for Advanced Manufacturing International, Inc., Bedford, TX, USA.

Capoyleas, V., Chen, X. and Hoffmann, C.M. (1996) Generic naming in generative constraint-based design. *Computer Aided Design*, **28**, 1, 17–28.

DMAC (1996) OLE for Design and Modeling, Design and Modeling Advisory Council, http://www.intergraph.com/iss/technologies/jupiter/ole.htm

Hoffmann, C.M. (1989) *Geometric and Solid Modeling.* Morgan Kaufmann, San Mateo, California.

ISO (1994a) Industrial automation systems and integration – *Product data representation and exchange* – ISO 10303:1994. International Organization for Standardization, Geneva.

ISO (1994b) Industrial automation systems and integration – Product data representation and exchange – Part 11: *Description Methods: The EXPRESS language reference manual* – ISO 10303-11:1994. International Organization for Standardization, Geneva.

ISO (1996) Industrial automation systems and integration – Parts Library – Part 31: *Geometric Programming Interface,* ISO Draft International Standard 13584-31. International Organization for Standardization, Geneva.

Kripac, J. (1995) A mechanism for persistently naming topological entities in history-based parametric solid models, in *Proceedings, 3rd Symposium on Solid Modeling and Applications* (eds. C.M. Hoffmann and J.R. Rossignac), ACM Press, New York.

Pratt, M.J. (1998) Background Study for Procedural Modeler Interface, ISO White Paper TC184/SC4/WG12 N106, http://www.nist.gov/sc4/wg_qc/wg12/n106.

Owen, J. (1997) *STEP: An Introduction.* Information Geometers, Winchester, UK.

Raghothama, S. and Shapiro, V. (1997) Boundary representation variance in parametric solid modeling. Technical Report SAL 1997-1, Spatial Automation Laboratory, University of Wisconsin–Madison.

Schenk, D.A. and Wilson, P.R. (1994) *Information Modeling: The EXPRESS Way*. Oxford University Press.

8 BIOGRAPHY

Mike Pratt has an MA in physics from Oxford University, an MSc in aerodynamics and a PhD in mechanical engineering from Cranfield University in England. He spent much of his career at Cranfield, where he rose to the position of Professor of Computer Aided Engineering and headed the Department of Applied Computing and Mathematics. In 1991, having had enough of administration, he left Cranfield and moved to a senior research position at Rensselaer Polytechnic Institute in Troy, NY, USA. For the past four years he has been seconded to the National Institute of Standards and Technology in Gaithersburg, MD. His interests include all aspects of product modelling, and especially its uses in the integration of CAD with downstream applications. He is actively involved in development of the ISO 10303 standard, as leader of the STEP Parametrics Group.