A Shape Modelling API for the STEP Standard

Michael J. Pratt [1]

*National Institute of Standards and Technology,
Manufacturing Systems Integration Division, 100 Bureau Drive, Mail Stop 8262,
Gaithersburg, MD 20899-8262, USA. E-mail pratt@nist.gov*

Bill D. Anderson

*Advanced Technology Institute, Product Data Technology Group,
5300 International Boulevard, North Charleston, SC 29418, USA.
E-mail anderson@aticorp.org*

Abstract

The international standard ISO 10303 for the exchange of product models and associated data between different CAD and other engineering systems was first issued in 1994. This paper reports on current work on extending the standard to enable the capture and transfer of parametrized CAD models with geometric constraints, a capability not provided in the initial release. This will allow the transmission of 'behavioural' information with the exchanged model. Two complementary approaches are being worked on. The first aims to add supplementary data to the types of explicit models that can currently be exchanged. The second is more radical; its objective is to transfer CAD models in procedural form, i.e., expressed in terms of the sequence of operations used to construct them. The paper concentrates on the second approach, which is characteristic of the primary model representation used by many modern CAD systems. It is shown that the requirements for a standard in this area are virtually identical with those for a standardized API for CAD modellers. Previous work in the latter area is surveyed, to determine whether there exists a suitable basis for the ISO 10303 work, and progress and technical problems are reviewed.

Key words: CAD models, Data exchange, Parametrization, Geometric constraints, Procedural modelling, Standard interfaces, System integration.

[1] Also affiliated with Center for Automation Technologies, Rensselaer Polytechnic Institute, Troy, NY 12180-3590, USA.

1 Introduction

The international standard ISO 10303, informally known as STEP (STandard for the Exchange of Product model data) is becoming increasingly recognized by industry as an effective means of exchanging product-related data between different CAD systems or between CAD and downstream application systems. The first version of the standard was published in 1994 [9,14] after some ten years of work by members of the relevant standards committee, ISO TC184/SC4 (whose responsibility is 'Industrial Data'). Much of this effort went to developing an infrastructure for the standard that will ensure its extensibility for the foreseeable future.

ISO 10303 covers a wide variety of different product types (electronic, electromechanical, mechanical, sheet metal, fiber composites, ships, architectural, process plant,...) and life-cycle stages (design, analysis, planning, manufacture,...). This range is continually expanding as new parts of the standard are issued. These parts are referred to as ISO 10303-*nnn*, where *nnn* is the part number, and each is a standard in its own right, despite being a component of a larger whole and interdependent on other parts. Currently the overall standard is composed of about 20 parts, though many more are in development.

At present ISO 10303 exchanges usually employ the well-known neutral file approach, in which transfer between two systems is a two-stage process. In Stage 1 data is translated from the native data format of the originating system into the neutral ISO 10303 format, and Stage 2 is translation from the neutral format into the native format of the receiving system. However, the standard makes a separation between the information model (written in a language called EXPRESS [17], which is part of the standard) and its physical implementation, and this permits ISO 10303 models also to be used in other ways. For example, ISO 10303-22 has recently been released, defining a standardized data access interface for repositories containing ISO 10303 neutral information. Other forms of data access and data sharing are in prospect for the future.

The implementable parts of ISO 10303, each applicable to a particular life-cycle stage of a particular product class, are known as Application Protocols (APs). However, the APs themselves are constructed on the basis of a set of Integrated Resources (IRs), defining fundamental constructs that can be specialized and applied for a wide variety of purposes. The present paper is concerned with the exchange of CAD models, and the most relevant IRs in this context are ISO 10303-41 ('Fundamentals of product description and support'), 10303-42 ('Geometric and topological representation'), 10303-43 ('Representation structures'), 10303-44 ('Product structure configuration') and 10303-108 ('Parametrization and constraints for explicit geometric product models'). The first four of these are already part of the standard, and Part 108 is currently under development, as will be described below.

Because of the nature of the international standardization process, the technical content of the initial release of ISO 10303 had to be frozen well before its final 1994 publication. Unfortunately the period concerned coincided with a time of rapid development of CAD systems, so that the present version of the standard reflects CAD technology as it was several years ago. Recent CAD developments have included the widespread use of parametrized feature-based models incorporating geometric constraints. However, the standard cannot at present handle the exchange of models possessing these characteristics; all that can be transferred is the basic geometry and topology of shape models of the boundary representation [6] and closely related types. At present a high rate of success is being achieved by industry in exchanging models of these kinds between different CAD systems, but the fact that parametrization, constraint and feature information is lost in the exchange makes the transferred models difficult to modify for downstream purposes in the receiving system.

Parametrization is an expression of the design freedom built into the model by its designer, who also incorporates constraints to ensure continued functionality of the modelled part or product when that freedom is used for design modification. Features are high-level shape constructs that make it possible for the designer to avoid having to work at the low level of individual curve and surface elements of the shape model. The designer's choice of parametrization and constraint schemes constitutes an important part of what is known as *design intent*. The classes of features used in design also embody design intent, and they may additionally have significant links to manufacturing or other applications.

None of the information described in the previous paragraph can currently be transferred by ISO 10303. Accordingly, work is under way to extend the standard for its capture and transmission.

2 The ISO TC184/SC4 Parametrics Group

The ISO 10303 Integrated Resources (IRs) are the responsibility of Working Group 12 (WG12) of ISO TC184/SC4. The Parametrics Group is a task force within WG12, whose primary goal is to bring the capabilities of the ISO 10303 standard into line with those of modern CAD systems. Current systems use a combination of two fundamental approaches to shape modelling. The first of these is referred to as *explicit modelling*. It is characterized by the use of boundary representation models described in terms of faces, edges and vertices, whose connectivity or topology is fully detailed, and whose associated surfaces, curves and points are specified in the model. The second modelling approach is *implicit* or *procedural*. A procedural model is represented in terms of the sequence of operations used in constructing it. This kind of model, in its pure form, therefore contains no explicit geometry or topology, since the elements of its detailed shape representation are not called into existence until the constructional sequence is actually performed. Most CAD systems generate models whose native internal storage format is some combination of these two approaches. A commonly occurring example is the swept volume. This is usually defined in terms of a closed 2D profile, made up from explicitly defined curve segments, and a sweep operation, either linear in a specified direction or rotational about a specified axis. The combination of the explicit profile and the procedurally specified operation is a hybrid representation of the volume swept out by the profile during the sweep motion. Note that the surfaces of this volume have no explicit representation — generation of that level of detailed information requires the specified constructional operation to be performed by a CAD system, in which event the original implicit or procedural model is converted into an explicit one. This process is sometimes referred to as *evaluation*.

Both forms of representation have advantages and disadvantages. For example, explicit models provide ready access to detailed geometric information that is important for downstream applications such as manufacturing. On the other hand, in their pure form they are difficult to edit, e.g., in a design optimization process, because they lack any record of how they were originally built, and therefore contain no design intent information. A procedural model is very easy to edit, by modifying its procedural description (for example, by updating the dimensional parameters in some constructional operations) and then rerunning it. The constructional history embodies the design intent that is lacking in the explicit model. But on the debit side, as already pointed out, such a model contains no detailed geometric information and so is difficult to use for applications subsequent to design. The balance between the two approaches exhibited by current CAD systems reflects their developers' notions of the optimal tradeoff between these and other advantages and disadvantages.

From the point of view of ISO 10303, the situation described above implies that the standard should provide facilities for transferring both kinds of model, and hybrids between them. Currently, all that can be transferred is explicit models with no information concerning parametrization, constraints or features. The Parametrics Group is therefore working on two fronts. One effort is devoted to enhancing the standard's explicit model exchange capabilities to include the information that currently gets lost. The other is concerned with developing an approach for the exchange of procedural models in a standard form. These two efforts are briefly described below. The remainder of the paper then gives more detail of the work in the procedural modelling area, which has interesting implications for the future of integrated systems that incorporate a geometric modelling capability.

2.1 ISO 10303-108

It was mentioned in the Introduction that Part 108 of ISO 10303 is currently under development. This new resource will provide facilities for representing parametrization and geometric constraints as they apply to explicit shape models. The topic of features has not yet been addressed in the Part 108 context, but when the two basic mechanisms mentioned above are in place rapid progress will be possible in this area also. One of the key problems faced in this work is that of making it upwardly compatible with existing parts of the standard, i.e., of retrofitting significant capabilities without making any changes to its earlier released parts. This is a strongly imposed requirement, since alterations to the existing standardized parts would involve multiple CAD vendor companies in significant and unprofitable activity in modifying their existing ISO 10303 translators.

An important aspect of this work is that it will provide information in an ISO 10303 model concerning the behaviour of the model in a receiving system. This is totally lacking in the current version of the standard, because existing ISO 10303 models are purely descriptive; they provide a snapshot of the state of the shape model at the moment of transfer, with no behavioural information. It was mentioned earlier that such information would be very valuable in giving guidance to the user of the receiving system as to ways in which it is and is not permissible to modify the received model.

2.2 New ISO 10303 procedural modelling resource

The provision of procedural representations encapsulating model behaviour will be a new departure for ISO 10303. What is needed is a standardized means of representing constructional operations for CAD models. Fortunately, existing commercial CAD systems are similar in the range of operations they provide in their systems, but it is certainly not a trivial task to standardize them. In the remainder of the paper some of the difficulties are described, a survey is given of previous attempts to define standardized interfaces to CAD system functionality, and progress in the ISO 10303 context is reviewed.

2.3 Requirements analysis for procedural modelling interface

This section gives a very brief summary of the requirements for a standardized procedural modelling representation in ISO 10303. Most modern CAD systems provide a feature-based approach to design, in which complete substructures of a model are created by single high-level operations. The initial approach will concentrate on trying to capture a range of such operations in a neutral form. There is some difficulty in this because no ISO 10303 resource currently provides representations of parametrized design features, and so it will be necessary to define notional feature entities to provide a conceptual model for the creation operations. However, some ISO 10303 Application Protocols, notably AP214 ('Core data for automotive mechanical design processes') and AP224 ('Mechanical parts definition for process planning using machining features') do contain non-parametric feature definitions. These cannot be referred to from a resource document, but it will nevertheless be important that the Parametrics definitions are compatible with them.

At a much lower level, we must also be compatible with ISO 10303-42, the basic resource for the representation of explicit models. This implies that we need to specify operations for creating every one of the modelling entity types specified in that resource. This includes

- Topological entities such as faces, edges and vertices;
- Geometrical entities such as points, lines, circles, cones and non-uniform rational B-splines (NURBS);
- Higher-level constructs built from these entities;
- Associated local coordinate systems, transformations etc.

Since model creation often includes backtracking and modification, operations for modifying and deleting both high- and low-level entities will also be needed. The provision of operations based on ISO 10303-42 will ensure a basic level of interoperation between the new procedural resource and the present explicit modelling resource. For compatibility with current CAD technology, the new resource will also need to handle the entities defined in the emerging Part 108 mentioned earlier; this will provide an additional capacity for operations involving parametrization and geometric constraints.

The use of standardized operations for the exchange of procedural models is regarded by the ISO TC184/SC4 Parametrics Group as just one application of what will in fact be a standardized general-purpose applications programming interface (API). It is envisaged that this exchange can be made largely in terms of sequences of high-level operations, though the use of low-level operations will sometimes be necessary for making detail changes in designs. On the other hand, it will be highly desirable if the same set of operations can also be used for purposes other than model exchange, and these are likely to require more intensive use of low-level operations. For example, in generating data for machining a face of an object, the application system will need to determine the details of the individual boundaries of the face and to create a sequence of tool contact paths on it. It will also, in general, need to perform tool interference calulations to ensure that while the tool is cutting at one point it is not generating unintended gouges at another. All of these operations will involve detailed low-level queries and construction operations.

The construction of a complex CAD model entirely in terms of creation operations for its low-level constituent entities would be a very tedious task, which is the reason why higher-level operations are being provided. But conversely, if a particular CAD system does not implement one of the high-level capabilities in the standard, then it is in principle possible to achieve the same result using the fundamental or atomic low-level operations. In fact the interface design poses the technical challenge of determining the appropriate granularity of the constructional resource in terms of its balance between high- and low-level capabilities.

High-level operations include examples such as the sweep operations mentioned earlier (these are regarded as feature operations in most systems), the Boolean operations of constructional solid geometry (CSG) [6], and operations for rounding sharp edges (or sets of sharp edges) in a model. The provision of such operations motivates a requirement for query operations on the entities in the model. If the designer invokes a high-level operation it is often not clear *a priori* what precise changes will occur in the model. The number of new faces and edges created, and also the details of their geometry, will depend upon local geometric conditions in the region of the model being modified. Subsequent modelling operations may require a detailed knowledge of the structure of the region where the change has occurred, and for this reason it is desirable to provide query facilities so that this structure may be ascertained. Thus, finally, we arrive at the following list of required operations:

- Creation
- Deletion
- Modification
- Query

These should be applicable to all the low-level basic entities defined in ISO 10303-42 and also to the more complex constructs generated by the higher-level creation operations (some of whose results — but not the operations themselves — are already defined in ISO 10303-42).

At this point it may be noted that there have been several efforts in recent years to define standardized procedural interfaces (i.e., APIs) for CAD modellers. The objective has been to provide a means of linking external application programs to CAD systems in a modeller-independent manner. This will have several perceived advantages, permitting

- Plug-and-play CAD modeller capabilities in modular integrated product realization (design and manufacturing) systems;
- Release of CAD users from dependence upon their CAD supplier for the supply of application software interfacing to their system;
- Alternatively, release from the requirement for writing in-house application software tailored to a particular CAD system;
- Encouragement of 'third-party' application software developers, who can write systems that will interface without modification to any one of a range of CAD systems that have implementations of the standardized interface.
- Exchange of model information with a CAD system in 'conversational' mode, in which the interacting system can specify exactly what it needs and the CAD system can respond with the precise information required. The alternative is file transfer, which is akin to mailing and receiving a letter. Here the content is under the total control of the sender; what is sent may hold far more information than is needed for a specific application purpose, or possibly far less.

It is very significant that the necessary functionality identified for such a standardized CAD system API has proved to be identical in all major requirements with that identified earlier in this section for the ISO 10303 procedural modelling capability. Consequently, it will be possible to capitalize on previous work on standardized APIs in the development of the new capability. The resulting resource will then serve a much wider purpose than pure CAD model exchange. A critical survey of existing proposals for standard CAD system APIs is reported on in the next section.

2.4 Proposals for standard CAD interfaces

Six proposals have been examined, as detailed in what follows. Only the first two of them have been submitted for any formal standardization process.

2.4.1 ISO 13584-31

ISO 13584 (Parts Library) [8] is another ISO standard under development in the same ISO committee as ISO 10303. It is concerned with the standardized means of representation and information access for standard parts in computer-based libraries. Parts Library provides a means for the parametrized representation of families of parts, to avoid the need for separate representations of each member of what may be an extensive collection. In particular, ISO 13584-31 (Geometric programming interface) defines a procedural interface for the generation of parametrized product shape models. To some extent it meets the requirements specified earlier for the ISO 10303 procedural modelling capability, but it lacks certain significant capabilities, as listed below:

- Its model creation capabilities are based on a very limited subset of ISO 10303-42;
- Although its models are parametrized, it does not allow the definition of geometric constraints;
- It provides few query operations and no modification operations apart from parameter changes — for Parts Library use these are not necessary since only final designs will be represented;
- The only solid modelling capability provided is for constructive solid geometry (CSG) representations, whereas all current CAD systems generate solid models of the explicit boundary representation type [6].

ISO 13584-31 was recently published as an International Standard. Its procedures are specified in language-independent terms. A FORTRAN language binding is provided, and a Java binding is in preparation.

2.4.2 CAM-I Applications Interface Specification

CAM-I (Consortium for Advanced Manufacturing International, Inc.) is an international industrial research organization based in Texas, USA. As early as 1979 CAM-I foresaw the need for a standardized procedural or programming interface for CAD solid modelling systems, as a means for accessing their internal functionality, and to facilitate the creation of integrated design and manufacturing systems. They developed their Applications Interface Specification (AIS) to satisfy this need, and since that time the AIS has gone through several versions, some of which have been extensively tested in practical implementations [15,19]. The latest version [2] was designed to be compatible with ISO 10303-42. It has undergone some testing, and has also spent a 3-year period as an ANSI Draft Standard for Trial Use. However, despite the long history of the AIS, CAM-I has consistently found it difficult to obtain CAD vendor commitment to the idea of implementing it.

The AIS is a better fit for the new ISO 10303 procedural modelling resource than ISO 13584-31, because its coverage of the entities in ISO 10303-42 is much more complete. Also, unlike the Parts Library resource it is defined in an object-oriented manner, which is felt to be an advantage. To balance against these positive points it has the following shortcomings, as identified in a recent analysis carried out by the Advanced Technology Institute for NIST:

- It provides an interface specifically to solid modellers, and not to other types of CAD modellers (though this will not require major extension, as ISO

10303-42 provides resources for all types of CAD models used in practice);
- The AIS needs enhancement to handle the parametrization and constraint information defined in ISO 10303-108;
- Because it concentrates mainly on low-level operations on individual geometrical and topological entities, significant further extension will be needed to cover certain types of high-level constructional operations (e.g., feature-based operations, edge rounding) that are commonly provided in CAD systems but currently catered for neither by ISO 10303-42 nor by the AIS.
- Not all Part 42 curve and surface types are covered by the AIS.

The AIS is specified in a language-independent form, and a C language binding is also provided.

2.4.3 DMAC OLE for Design and Modeling

DMAC is the Design and Modeling Advisory Council [2], a consortium of CAD and application software vendors committed to the use of PC platforms and Microsoft technology. OLE is Object Linking and Embedding, a proprietary Microsoft means for constructing compound documents, in a very general sense. This is achieved by the use of Microsoft Component Object Model (COM) technology. The principle is that an object generated by one application can either be linked to another object (by reference) or embedded in it (by copying). In the present context, the OLE for Design and Modeling interface [4] allows the linking of a 3D CAD model into a document, and its subsequent interrogation from within that document. Here the 'document' will in general be generated by another CAD program. It should be emphasized that the CAD model in the original system (the server) is not affected by this process. However, its characteristics may be interrogated in the second system (the client), and subsidiary datastructures relating to it may be created there. Essentially, then, this is a one-way interface, permitting query operations only, and providing no means for the creation or manipulation of the model in the server system. From the ISO 10303 point of view the disadvantages of OLE for Design and Modeling are

- The one-way nature of the interface;
- The proprietary nature of the communication mechanism, which makes the interface unsuitable for adoption as an international standard.
- Uncertainty as to whether the present version can handle parametrization and constraints; feature-based capabilities are also apparently excluded.

DMAC claims that there is an advantage in not relying on neutral file technology but always accessing the required data in its native format, which gives

[2] In the name of this organization 'modeling' is spelt with one 'l' in the American manner rather than two in the European manner used elsewhere in this paper.

greater reliability. This is undeniable, though it invalidates OLE for Design and Modeling as a candidate for the ISO 10303 requirement. Nevertheless, there is advantage in studying the nature of the information accessible via this interface, since it has been developed and tested with a range of commercially available CAD and other application systems, and should therefore accurately reflect their capabilities.

At the time of writing DMAC claims two interfaces ready for implementation, one concerned with geometry and topology, the other specifically with curve geometry (presumably this is intended to handle curves specified in a piecewise manner, e.g., NURBS curves, and the two-dimensional profiles commonly used in 3D constructional operations). Several other interfaces are under development. Significantly, one deals with persistent identifiers (see Section 3.1). Another relates to assembly geometry and structure. Access to these interface specifications is currently restricted to DMAC member companies, though in the past the documentation was freely available. At that time models with parametrization and constraints could be handled, but it is not clear whether that capability is still included in current versions. However, there are several specifications described as 'dormant', i.e., not currently being worked on. They are publicly available, and one of them deals with features. None of these interfaces have been put forward as formal standards, though the geometry/topology interface has been tested in various applications. Some of them are detailed at the referenced Web site [4]. The DMAC interfaces cannot be considered to be language-independent because of their basis in Microsoft OLE/COM.

2.4.4 ENGEN Data Model (EDM) Construct Module

ENGEN (Enabling Next GENeration mechanical design) was a project jointly funded by DARPA (Defense Advanced Projects Research Agency) and PDES, Inc. (see the web site www.aticorp.org/pdesinc), an industry/government consortium managed by the Advanced Technology Institute, whose primary goal is to accelerate the development and deployment of the STEP standard. The formal project duration was 1995 – 1998; the work reported here can be regarded as an extension of the ENGEN project into the area of feature-based construction history, as part of the ISO 10303 Parametrics work.

ENGEN developed EXPRESS information models for adding parametrization and constraint information to neutral format CAD models similar to those of ISO 10303. However, there was some departure from the precise details of the standard in the interests of achieving demonstrations of the transfer of this information with limited financial and manpower resources.

In fact the transfers achieved to date [1] have concentrated on explicit models with parametrization and constraints, using resource models akin to abridged versions of ISO 10303-42 and 10303-108. Some work has also been done on the transfer of procedural models restricted to the CSG type, but current extensions are in the area of feature-based model transfer.

The earlier exchange demonstrations made use of the ENGEN Data Model (EDM), closely related in style, as mentioned above, to an ISO 10303 integrated resource. Although the EDM concentrated on providing the means for representing explicit CAD models, some thought was also given to the transfer of procedural models. For this purpose a skeleton 'Construct Module' was written, specifying a small number of 2D constructional operations. This module was not worked out in detail. Its main interest in the present context lies in the way the operations were represented.

The EXPRESS information modelling language (ISO 10303-11) cannot currently represent instantiable mathematical functions as procedures with defined inputs, outputs and actions[3]. It is therefore not possible to transfer constructional procedures in this manner without somehow extending the language. To avoid this problem, ENGEN defined the EDM constructional functions in an entity-based manner. The example below illustrates this approach.

Consider the way a line entity is defined in ISO 10303-42 (a slight simplification has been made in the interests of clarity):

```
ENTITY line
  SUBTYPE OF (curve);
  pnt : cartesian_point;
  dir : vector;
END_ENTITY;
```

The interpretation of this EXPRESS language specification is straightforward. It defines a class of unbounded lines characterized by point and direction attributes. The language provides subtype/supertype relationships, and this entity also inherits from further up the hierarchy a `representation_context` attribute relating to a local coordinate system. In an instantiation of the line, for example in an exchange file, the `pnt`, `dir` and `representation_context` attributes will be replaced by references to instances of a specific cartesian point, a specific vector and a specific representation_context occurring in the file, to define a specific line. The line instance in the file is purely descriptive, capturing a static relationship between a line, a point and a vector, and the coordinate system in which they exist. If the line model is transferred to a receiving system and (for example) the position of the point is then edited so

[3] Such functions and procedures are defined in the language, but only for ephemeral use in validity checking of entity instances at translation time, and not for permanent transfer with the CAD data.

that it no longer lies on the line, the data will become inconsistent because no design intent has been transmitted.

Now consider the following EDM definition (again, slightly simplified):

```
ENTITY constructed_line
  SUBTYPE OF (constructed_curve);
  pnt : cartesian_point;
  dir : vector;
END_ENTITY;
```

This looks very similar to the previous entity definition, but in fact its semantics are intended to be totally different. In the EDM, transmission of an instance of this entity is regarded as an instruction to the receiving system to construct a line passing through a point and having a direction already defined in the model being generated. Further, the relationship between the point, the vector and the line are to be treated as constraints once the line has been generated, so that if either of the attributes is subsequently edited in the receiving system the line will change accordingly. This provides an example of what are known as *implicit constraints*, i.e., constraints that are not explicitly imposed on elements in the model but are inherent in the operation of a constructional procedure.

The ISO TC184/SC4/WG12 Parametrics Group has reviewed this type of representation and would prefer to reject it in favour of the procedural method. However, discussions with CAD system vendor companies indicate that some of them prefer the entity-based approach because it is more compatible with their present ISO 10303 translator implementations.

One problem with the entity-based approach is that the static and dynamic entities both have a very similar appearance, even though they have very different semantics, and it is felt that this may give rise to confusion. Another reason is that Supplement 2 of the ISO/IEC Directives (which specify the style of ISO standards documents) recommends that standardized API specifications should be procedural, with a language-independent basic form plus one or more bindings to standard programming languages. Furthermore, there has been a recommendation from the Object Management Group (OMG) Manufacturing Domain Task Force that the API specification should be written procedurally in CORBA IDL[4] [10]. However, a strong preference by implementers for the use of the entity-based representation may well be decisive in the absence of overwhelming reasons for adopting the alternative. The matter is at present undecided, and will be resolved by intensive consideration of

[4] The Interface Definition Language of the OMG Common Object Request Broker (CORBA) Architecture. The OMG is currently developing a Request for Proposals for a “CAD Services” API, to which the STEP Parametrics Group will respond.

specific examples.

If the procedure-based representation is chosen, there is a good reason why the API operations should be formulated using the EXPRESS syntax for functions — this will enable their use in the formulation of instantiable constraint relationships in the model, along with the 'standard' functions defined in EXPRESS, which include sine, cosine, exponential, etc. However, there is an equally good argument for developing an IDL version in parallel with the EXPRESS version, since this will allow the use of the API in the general OMG CORBA environment.

To summarize, the disadvantages of the EDM Construct Module for adoption by ISO 10303 are as follows:

- Only a very rudimentary set of 2D constructional operations has been specified;
- The operations are defined in an entity-based rather than a procedure-based manner.

The EDM Construct Module is essentially a place-holder in a potentially much expanded document, and it covers no more than a miniscule range of the overall set of requirements for the new ISO 10303 procedural modelling resource. The primary reason for including it in this survey has been to illustrate the possibility of using an entity-based approach rather than a library of procedure specifications for defining the syntax of the interface.

2.4.5 Hoffmann's EREP (Editable Representation)

The EREP [7] is under development at Purdue University. It is intended to provide a universal procedural interface for the construction of feature-based CAD models. It will also serve to preserve design intent in models transmitted in terms of the constructional operations it specifies. In fact the EREP regards a CAD model as being built entirely by a sequence of feature creation, modification and deletion operations, which may be captured and used as a procedural description of the modelled product shape. This is both an advantage and a disadvantage. On the credit side, it deals in detail with the definition of features and their various modes of attachment to a CAD model under construction, while none of the other proposals surveyed does this. On the debit side, it does not concern itself with the constructional details of some of the explicit elements used in feature construction, e.g., the 2D profiles mentioned earlier that are often used in the generation of swept volume features.

Although EREP models are fundamentally procedural in nature, they also incorporate important explicitly defined modelling elements as mentioned above, and they are therefore hybrid from the point of view of ISO 10303. Other explicit constructs include dimensions and constraints, which are used in feature definitions, and for the relative positioning and orientation of features in the model. The EREP also handles the constrained placement of parts in assemblies. It makes no claim to be compatible with ISO 10303.

Since it is based exclusively on the use of high-level feature-based constructional operations, we may conclude that the EREP specification is not a candidate for adoption as the basis for the new ISO 10303 procedural modelling resource. However, it will certainly prove valuable as a reference for the development of that part of the interface that deals with feature-based operations and the representation of feature-based assemblies.

A further noteworthy aspect of the EREP is that, as in the ENGEN Data Model, features are transmitted as entities and not in terms of the constructional procedures that generate them.

2.4.6 Djinn

Djinn is a solid modeller API specification developed by a group of researchers in the United Kingdom [5,13]. It is aimed at providing access to a wide range of solid modeller capabilities in a way that is independent of the nature of the underlying modelling system. The desirability of this is shown by the fact that ISO 10303-42 and the CAM-I AIS both provide capabilities for handling models of the boundary representation (Brep) and constructive solid geometry (CSG) types, but these provisions are made in parallel, so that not all of them apply to all modellers. In particular, operations and queries addressing low-level geometrical and topological elements of a boundary representation model (e.g., edges and their associated curves) cannot be performed on a CSG model since it does not contain representations of such entities.

To overcome this problem, the developers of Djinn have devised a conceptual representation for solid models that provides mappings onto both Brep and CSG data structures. In fact their representation will also handle general non-manifold situations (including solid models with internal structure) that many commercial CAD systems cannot yet deal with. Since their representation is compatible with both the primary approaches to solid model representation, the Djinn group has been able to use it as the basis for defining a procedural interface that is independent of the type of the underlying modeller.

The Djinn work is not aimed at developing a standard. Rather, it is in the nature of a research project, and its documentation identifies further research issues that need to be addressed at a lower level of detail to provide more complete coverage of CAD modeller functionality. Thus Djinn is not a candidate for adoption as the basis of the new ISO 10303 resource, for the following reasons:

- Politically, it would be impossible to get agreement in ISO TC184/SC4 for the adoption of a new underlying model representation, even though it is in principle compatible with the representations specified in ISO 10303-42.
- Djinn does not handle parametrized models, geometric constraints or features;
- In the sense that it has indicated new representational problems that need to be solved, the Djinn work is incomplete.

Although the interface does not cover feature operations in its present form, some work on extensions in this area has been reported [12].

3 Technical issues and opportunities

The work of the ISO TC184/SC4 Parametrics Group in extending the product data exchange standard ISO 10303 for the transfer of parametrized models has been described. An important consequence of this work will be that models can be transferred in the future that contain specifications of allowable behaviour under modification operations in a receiving system. Two approaches have been mentioned, one based on the enhancement of the current explicit type of exchange model, and the other based on the transfer of procedural models defined by sequences of constructional operations.

An analysis of the requirements for a standardized method for representing product shape models procedurally has shown that those requirements are essentially the same as those for a standardized applications programming interface (API) for a CAD modeller. Previous work on the specification of standard CAD system APIs has therefore been surveyed.

Much has been learned from the six standardized API proposals examined. None of them is a perfect fit for ISO 10303 context, for the following reasons:

- ISO 13584-31 was developed for a very specific purpose, is too limited in scope and is not object-oriented;
- The CAM-I AIS handles low-level operations well, but is lacking in the areas of features, parametrization and geometric constraints.
- OLE for Design and Modelling covers much of the required range of CAD modeller functionality but is purely a query interface;
- The EDM Construct Module is extremely limited in scope and uses an entity-based representation for constructional operations that is felt to have disadvantages;
- The EREP covers only feature-based operations, and is also entity-based,

but it will be a useful reference for the future since none of the other interface specifications deals with features;

- Djinn is essentially an interim result from a research project, which does not address features, parametrization or constraints. However, it has interesting aspects, and may be a harbinger of significant developments in the future.

Development of an initial draft of the ISO 10303 API is currently at a very early stage. It will proceed in parallel with work on ISO 10303-108, since operations will need to be defined for all the entities provided in that emerging resource.

Before concluding, it is appropriate to mention some of the technical challenges and opportunities that will arise in the course of this work. One challenge has already been pointed out, the problem of achieving the optimal balance between high- and low-level operations in the interface. Another is that of defining complex high-level operations such as edge-rounding (also known as blending or filleting) in a way that is compatible with a spectrum of CAD systems all implementing them in slightly different ways. A further important problem that needs to be tackled is that known in the technical literature as *persistent naming*.

3.1 The Persistent Naming Problem

During the design process, if the system records the operations used, that sequence of operations constitutes a procedural description of the model created. When the sequence is replayed the system should regenerate the same explicit model, and use it to draw the same picture on the screen. However, the designer may frequently pick elements of the model from the screen display for modification during the design process. The question is, how does the system identify, in the procedural model that specifies only a set of operations and contains no explicit geometric or topological elements, which explicit elements were picked by the designer? This is the essence of the problem — the necessity to identify an explicit element in a model that is fundamentally implicit or procedural. CAD systems often discard the explicit model used for screen display after a model is edited, and generate a new one from the procedural description, so the explicit model cannot be referred to directly during regeneration after an edit. At present the system sometimes has to 'guess' the appropriate element; if it guesses wrong, the model can (and often does) behave in unexpected and undesirable ways after an editing operation.

Various heuristic solutions to the persistent naming problem have been proposed (see [3,11], for example), but none of them will handle all cases. Some initial work on a more formal approach has also been reported [16], but here again there is no complete solution. It is believed that no CAD system vendor has completely solved this problem, but the Parametrics group will need to adopt an approach to it that is at least sufficiently complete for existing implemented element identification mechanisms to map onto it. The Parametrics Group is closely following academic work in this area, though of course the commercial system developers do not publicize the methods they use.

Early analysis indicates that the persistent naming problem in model exchange differs to some extent from the corresponding problem within a single specific CAD system. The ISO 10303 solution is likely to employ a synthesis of appropriate elements of published approaches together with other capabilities developed specifically for model exchange. The ISO 10303 naming mechanism will be worked out with the collaboration of CAD system vendors, and it should therefore be compatible with their various systems. This will ensure acceptable results when a received model is subsequently edited.

3.2 The Geometric Accuracy Problem

The geometric accuracy problem arises in the exchange of explicit CAD models using the current ISO 10303 approach. A boundary representation model contains two classes of information — topology, which specifies how the various model faces, edges and vertices are connected together, and geometry, which specifies the surfaces, curves or points they lie on, respectively. Unfortunately, any geometric computation is subject to errors. This frequently leads to a situation where, for example, the model topology indicates that two edges are connected whereas the geometry indicates that their corresponding end-points are in fact slightly different. All CAD systems specify geometric tolerances that determine the maximum extent to which 'slightly different' can be interpreted within the system as 'identical for practical purposes'. Unfortunately, different CAD systems work to different internal geometric accuracies. One system may consider two points to be coincident if they lie within 10^{-4} units of each other, while another may use 10^{-7} units for the same criterion. Then if a model is generated in the first system and transferred into the second, the receiver is likely to judge that the elements of the model are not properly connected to each other, and either the exchange will fail completely or the model will need to be enhanced or repaired before it can succeed [5].

The geometric accuracy problem was, until recently, one of the major hurdles to 100% success in the transfer of explicit CAD models. However, advances

[5] STEP provides a construct **uncertainty_measure_with_unit** that allows transmission with an exchange file of a real value that is intended to provide a measure of the accuracy of the transmitted model; for example, the minimum distance between two points for which they are considered to be distinct.

in translator technology and the availability of software tools for analysing 'model quality' and for repairing or 'healing' defective models have largely overcome it. It is nevertheless worth noting that geometric accuracy is much less of an issue in the transfer of procedurally defined models. Here what is exchanged is primarily the methodology for constructing the explicit model, and the actual construction is done mainly in the receiving system, subject to that system's accuracy criteria. There are exceptions in the case of hybrid model transfer, for example where the construction of a swept volume may be based on the exchange of an explicitly defined 2D profile. However, 2D geometry is less subject to error than 3D geometry, and the very fact that less explicit geometry needs to be transferred when there is a procedural basis for the exchange should lead to a smaller failure rate.

Another aspect of the numerical accuracy problem arises when constraint information is transferred with an explicit geometric model. It may be, for example, that a perpendicularity constraint has been imposed between two planar surfaces, but the actual angle between those surfaces as represented in the model differs slightly from 90° because of computational rounding errors. Such cases can be handled straightforwardly; in the receiving system, the model can be reconstructed in the usual way using geometrical and topological information alone, and the supplementary constraint information then used to adjust the surfaces concerned so that the perpendicularity constraint is satisfied to within the numerical tolerance of the new computational environment.

3.3 Resolution of ambiguities in parametric models

When a transferred model is a hybrid, containing both explicit and procedural elements, it may be ambiguous, i.e., there may be more than one evaluated model corresponding to the hybrid model. This situation arises, for example, when one operation in the constructional sequence is the solution of a set of constraints that are required to hold simultaneously. Constraint equations are often nonlinear, and a system of such equations will then have multiple solutions. Some means is therefore needed of identifying, during evaluation of the model in the receiving system, which choice of solution was made by the user of the sending system. The proposed method is to send an explicit evaluated model together with any model containing procedural elements. Ambiguities arising during evaluation of the procedural model can then be resolved by reference to the explicit model, which is referred to as the *current result*. The current result may be regarded as a representative example of the family of parts defined by a parametric model; after transfer, it should be possible to edit the defining model to generate other members of the family.

3.4 Issues relating to engineering tolerances

Another ISO technical committee, ISO TC213, is currently developing a radical new approach to the representation, interpretation and use of engineering tolerances [10,18], based ultimately on group theory. There is a formal liaison between ISO TC213 and ISO TC184/SC4, and one of its active areas of work is investigation of the relationship between the geometric constraints defined in ISO 10303-108 and related constraints used by ISO TC213 in building toleranced features as assemblages of simple sufaces. Initial impressions are that Part 108 of STEP is compatible with the TC213 approach, and that a synthesis of the two may prove useful in developing a useful application-independent method of classifying features. Further work on this topic is in progress.

3.5 Long-term archiving

Procedural models have two significant characteristics that make them more suitable than current explicit ISO 10303 models for long-term archiving. Firstly, they do not contain the mass of detailed information that an explicit model does, and can therefore be stored much more compactly. Secondly, they are independent of the kind of changes in explicit model representation that may occur as new CAD systems are launched or new versions of existing ones are released to users. There is a legal requirement, for example, that design data pertaining to an aircraft is retained during the lifetime of that aircraft. With modern aircraft that lifetime can run into many decades, during which time CAD technology may change dramatically. This is an added incentive for the development of a standardized means of representing a CAD model in a way that says 'this is how it was built' rather than 'this is the end result, in all its gory detail'.

3.6 Structuring of the model history

Several aspects of model history dictate essential characteristics needed in a standard for the transfer of procedural model files:

- If the file is to be editable, it must also be (in some sense) human-understandable. This applies not only to the operations used in constructing the model, but also to the strategy used in constructing it, and the technical considerations underlying the occurrence of design features. These require the capture of *design rationale* information, which currently exists mainly in the head of the original designer or designers. Research over the past few years has not yet revealed an easy way of acquiring such information during the design process, but provision must be made for its incorporation in the file when it becomes available in the future.

- If the model is an assembly, the history file will not be a simple sequence of operations, but a collection of sequences. There may, however, be links between them, for instance in the case when geometry is queried from an existing part model and used in creating geometry of a mating part in the assembly. The structure of a model exchange file will need to take into account the logical structure implied by such relationships.
- It should be clear, at any point in the file, what the current operations are concerned with. That is, the file should ideally be 'commented', to indicate which part, and which region of that part, is being created by a particular sequence of operations. The necessity for this is precisely equivalent to the need for comment in a computer program.

3.7 Example of ISO 10303 API constructional function

Specification of the ISO 10303 API is currently at a very early stage. However, it has already been remarked that the API must provide functions for the creation of all the fundamental shape-defining entities defined in ISO 10303-42 and ISO 10303-108, and that it is desirable for the syntax of the function definitions to follow that of the EXPRESS language. This implies, for example, that a function is required for creating a line. If the procedure-based approach discussed earlier is adopted, the EXPRESS definition[6] of the function will be as follows:

```
FUNCTION line_create
           (name : label;
            pnt  : cartesian_point;
            dir  : direction
           ) : line;
EXTERNAL;
END_FUNCTION;
```

Invocation of this function will create a line using the canonical ISO 10303-42 line definition in terms of a point and a direction. What is returned is in fact a persistent name, but the type associated with that name is `line` in this case. The line itself will not be created until the history file is executed in the receiving system, and the persistent name can be used to refer to it in subsequent constructional operations in the exchange file.

[6] In fact the example given conforms to Version 2 of EXPRESS, currently under development, rather than Version 1 as specified in ISO 10303-11.

Function definitions in EXPRESS normally have a function body specifying how the intended functionality is to be achieved. However, for the purposes of the API this will not be appropriate — all CAD systems provide the necessary functionality, but they do it in different ways, certainly as far as high-level creation operations are concerned. It is not the business of ISO 10303 to specify how CAD systems provide their capability, and in the API functions the body is therefore replaced by the EXPRESS reserved word `EXTERNAL`, implying that the algorithm used to achieve the desired effect is specified by some external system rather than in the EXPRESS definition.

It was mentioned earlier that functions cannot be instantiated in ISO 10303 at present, and the mechanism currently proposed for transferring constructional functions in a model history file is the use of an 'intelligent string'. Each operation will give rise to such a string, which will contain a call to a function such as that defined above. Intelligent strings will be transferred between systems as specialized subtypes of the basic EXPRESS string data type, but will then require a further level of processing to decode their semantics.

Initially, the ISO Parametics Group has decided to concentrate on the provision in the API of certain high level construction operations, such as blending, shelling, and the generation of draft angles. Two prioritized lists of such operations have been agreed, as noted below, as an initial framework for this development. These lists identify common capabilities of major CAD systems. Several PDES, Inc. member companies would like to have trial implementations of the capabilities inherent in the Priority 1 list in the near future. The work is likely to be performed in collaboration with the corresponding European CAD vendor organization, ProSTEP, with US and European industrial CAD user participation.

Important aspects of this work have included discussions with CAD vendor companies and review of their user documentation. This has helped the Parametrics Group to understand widely available operations in major CAD systems, to grasp the (sometimes subtle) differences between the way in which they are provided to the system users, and thus help to ensure that a common core of constructional capability can be implemented in the API. The lists mentioned above are as follows:

Priority 1

Linear sweep of sketch (extrusion)
Rotational sweep of sketch
Boolean operations (union, intersection, difference)
Blending (including rounding, filleting, chamfering)
Rigid body transformation (translation, rotation) of features
Generation of patterns of features

Use of system-defined features from a library

Priority 2

Sweep of a sketch along a curve
Sweep of a sketch with distance-related scaling
Skinning (lofting of a sequence of sketches)
Shelling and (related) offsetting
Tapering/generation of draft angles
Sectioning of a solid by a general surface
Bending of a solid

Work on the Priority 1 list is under way, with the involvement of CAD system vendors who are reviewing the work as it progresses.

3.8 Final remarks

This paper has described the preliminaries to an important new development in the ISO 10303 standard. The specification of a standardized means for the exchange of procedural CAD models will also provide a standardized API for CAD modelling systems, as pointed out in the paper. Much work remains to be done, and significant technical challenges must be faced. However, it is believed that both CAD data transfer in particular and the integration of product realization systems in general will benefit greatly from these efforts.

Acknowledgements

The authors would like to express their gratitude to present and past members of the ISO TC184/SC4/WG12 Parametrics Group for their contributions to the work described, especially Aziz Bouras (Université Claude Bernard, Lyon, France), Jean-Marc Brun (Université Aix-Marseille II, France), Noel Christensen (Honeywell, USA), Ray Goult (LMR Systems, UK), Tom Kramer (NIST, USA), Akihiko Ohtaka (Nihon Unisys, Japan) and Chia-hui Shih (Pacific STEP, USA). The work of the Advanced Technology Institute reported here was partially supported by grants from DARPA and NIST, and work at NIST was carried out under the SIMA (Systems Integration for Manufacturing Applications) program.

References

[1] W. Anderson and S. Ansaldi. ENGEN data model: A neutral model to capture design intent. In G. Jacucci, G. J. Olling, K. Preiss, and M. J. Wozny, editors, *CDROM Procs. of IFIP PROLAMAT'98 conference, Trento, Italy, Sept. 1998.* Kluwer Academic Publishers, 1998.

[2] CAM-I. Application Interface Specification (AIS) Version 2.1, (2 vols.). Technical Report R-94-PM-01, Consortium for Advanced Manufacturing International, Inc., Bedford, TX, 1994.

[3] V. Capoyleas, X. Chen, and C. M. Hoffmann. Generic naming in generative constraint-based design. *Computer Aided Design*, **28**, 1, 17 – 26, 1996.

[4] Design and Modeling Advisory Council. *What is OLE for Design and Modeling?*, 1999. At http://www.dmac.org/ — this site also contains specifications and future specifications under development.

[5] Geometric Modelling Society (UK). *Djinn Geometric Modelling Applications Programming Interface.* Draft of proposed book, available at http://www.bath.ac.uk/ ensab/GMS/.

[6] C. M. Hoffmann. *Geometric and Solid Modeling.* Morgan Kaufmann, San Mateo, CA, 1989.

[7] C. M. Hoffmann and R. Juan. EREP: An editable high-level representation for geometric design and analysis. In P. R. Wilson, M. J. Wozny, and M. J. Pratt, editors, *Geometric Modeling for Product Realization.* North-Holland Publishing Co., 1992. (Proc. IFIP WG5.2 Workshop on Geometric Modeling in CAD, Rensselaerville, NY, 27 Sep - 1 Oct 1992).

[8] International Organization for Standardization. *ISO 13584 – Industrial Automation Systems and Integration – Parts Library*, 1998. (The ISO catalogue is at http://www.iso.ch/cate/cat.html – search on 13584 to obtain a listing of parts of the standard.).

[9] International Organization for Standardization, Geneva, Switzerland. *Industrial Automation Systems and Integration – Product Data Representation and Exchange*, 1994. (The ISO catalogue is at http://www.iso.ch/cate/cat.html; search on 10303 for a listing of parts of the standard).

[10] International Organization for Standardization, Geneva, Switzerland. *Geometrical Product Specification (GPS) – Model for geometric specification and verification*, 1999. (ISO Technical Specification ISO/TS 17450:1999).

[11] J. Kripac. A mechanism for persistently naming topological entities in history-based parametric solid models. In C. M. Hoffmann & J. R. Rossignac, editor, *Proc. 3rd ACM Symposium on Solid Modeling and Applications, Salt Lake City, 17 – 19 May 1995*, pages 21 – 30. ACM Press, New York, NY, 1995.

[12] A. E. Middleditch and C. M. P. Reade. A kernel for geometric features. In C. M. Hoffmann and W. F. Bronsvoort, editors, *Proc. 4th ACM Symposium on Solid Modeling and Applications, Atlanta, GA, May 1997.* Association for Computing Machinery, New York, NY, 1997.

[13] A. E. Middleditch, C. M. P. Reade, and A. J. Gomes. Set-combinations of mixed-dimension cellular objects of the Djinn API. *Computer Aided Design*, **31**, 11, 683 – 694, 1999.

[14] J. Owen. *STEP: An Introduction.* Information Geometers, Winchester, UK, 2nd edition, 1997.

[15] S. G. Pavey, S. R. Hailstone, and M. J. Pratt. An automated interface between CAD and process planning. In J. A. McGeough, editor, *Computer Aided Production Engineering.* Mechanical Engineering Publications Ltd., Bury St. Edmunds, England, 1986. (Proc. International Conf., Edinburgh, April 1986).

[16] S. Raghothama and V. Shapiro. Boundary representation deformation in parametric solid modeling. *ACM Trans. on Computer Graphics*, **17**, 4, 259 – 286, 1998.

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

[18] V. Srinivasan. A geometrical product specification language based on a classification of symmetry groups. *Computer Aided Design*, **31**, 659 – 668, 1999.

[19] P. R. Wilson, I. D. Faux, M. C. Ostrowski, and K. G. Pasquill. Interfaces for data transfer between solid modeling systems. *IEEE Computer Graphics & Applications*, **5**, 1, 41 – 51, 1985.