Experience in the Exchange of Procedural Shape Models using ISO 10303 (STEP)

Michael J. Pratt[*]
LMR Systems, UK

Junhwan Kim[†]
National Institute of Standards and Technology, USA

Abstract

The international standard ISO 10303 (STEP) is being extended to permit the exchange of procedurally defined shape models, with additional parameterization and constraint information, between CAD systems. The transfer of parameterized assembly models is an additional objective. Most of the essential new resources have already been published by ISO, and the remainder are well advanced in the standardization process. Because these are new capabilities, at present not quite complete, there are at present no commercial STEP translators making use of them. However, several proof-of-concept trials have been performed or are in progress, using development versions of the STEP documentation. This paper reports in some detail on one of those trials, and comments on the experience gained. The conclusion is that the standardized exchange of CAD models containing 'design intent' information has been successfully demonstrated, but that the development of translators for that purpose is not an easy task. One particular problem area is pinpointed, where further research is needed to find ways of improving the efficiency of such exchanges.

CR Categories: H.2.5 [Information systems]: Heterogeneous databases—Data translation; H.5.3 [Information systems]: Group and organization interfaces—computer-supported cooperative work; J.6 [Computer applications]: Computer-aided engineering—Computer-aided design (CAD).

Keywords: standard, product data exchange, construction history, design intent

1 Introduction

The earliest parts of the international standard ISO 10303 [Int 1994] for the exchange of product data in electronic form were published in 1994. Since then the standard has become widely used for the exchange of computer aided design (CAD) models between different systems within companies, and also between companies up and down the engineering supply chain. The most widely used part of the standard at present is the application protocol ISO 10303-203 (AP203: 'Configuration controlled design of mechanical parts and assemblies'), which provides for the exchange of wireframe, surface and boundary representation (B-rep) solid models, together with associated administrative data. As mentioned in earlier papers [Pratt 2004a; Pratt et al. 2005] such models cannot be effectively edited in a receiving system after a transfer because all of the information collectively known as 'design intent' is lost in the exchange. Design intent is considered to include the construction history of the model together with any parameterization schemes and constraint sets imposed upon it. AP203, whose development was started well before these kinds of information were widely used in CAD systems, makes no provision for the capture and transfer of design intent, which is why the models it exchanges have become known as 'dumb' models. On receipt after an exchange such a model may have additional detail defined upon it, but it is not in general possible to change the basic properties (e.g., dimensions) of the exchanged model itself.

The new parts of ISO 10303 mentioned above are as follows:

- ISO 10303-55: 'Procedural and hybrid representation' [Pratt et al. 2005];
- ISO 10303-108: 'Parameterization and constraints for explicit geometric product models' [Pratt 2004b; Pratt et al. 2005];
- ISO 10303-109: 'Kinematic and geometric constraints for assembly models [Pratt et al. 2005];
- ISO DIS 10303-111: 'Elements for procedural modelling of solid shapes' [Pratt et al. 2005];
- ISO DIS 10303-112: 'Modelling commands for the exchange of procedurally represented 2D CAD models'.

The first three of these have already been published by the International Organization for Standardization (ISO) as parts of ISO 10303; the last two are (at the time of writing) at the Draft International Standard (DIS) stage.

This paper does not address the exchange of assembly models, and all 2D profiles or sketches involved in the tests performed have been defined in terms of explicit geometric elements rather than procedurally. The only further mention of ISO 10303-109 and -112 will therefore be at the end of the paper where some information is given about other proof-of-concept translation tests that have been performed or are in progress.

In what follows, ISO 10303, whose official title is 'Industrial automation systems and integration — Product data representation and exchange', will usually be referred to briefly by its informal name of STEP (STandard for the Exchange of Product model data). The individual parts of the standard listed above will be cited as 'Part 55', 'Part 108', and so on. A single reference is given to ISO 10303 in the bibliography, and this covers all published and DIS parts of the standard. A brief overview of the standard is given in [Pratt 2001] and a more extended one in the book [Owen 1997].

An early suggestion for a method of exchanging CAD models in terms of their construction history was made by Hoffmann and Juan [Hoffmann and Juan 1992]. Their EREP (Editable REPresentation) was a specification for the representation of sequential feature-based design processes, which supported parameterization and constraints. It was suggested that during a design session a CAD system could be made to generate an EREP model in addition to its own internal model. It should then be possible to export the EREP model and process it using a different CAD system to

[*] e-mail: mike@lmr.clara.co.uk
[†] e-mail: junhwan.kim@nist.gov

generate an equivalent model there. A trial implementation was made, but attention apparently later became focused on associated problems such as persistent naming [Capoyleas et al. 1996] and the solution of constraint systems [Bouma et al. 1995]. These are outside the scope of the STEP work, which has adopted a strategem (described later) for avoiding the persistent naming problem and addresses only the representation and transmission of constraint data, leaving the solution of constraint sets to individual CAD systems.

The closest recent parallel to the work reported here is that by Rappoport et al. [Rappoport 2003; Spitz and Rappoport 2004; Rappoport et al. 2005]. The primary differences are that the present research is aimed towards the development of an International Standard, and that it is based on the STEP philosophy of a standardized intermediate neutral representation. In the STEP approach the concept of 'feature rewrites' introduced in the cited papers occur during the preprocessing and postprocessing phases of the overall translation, rather than in a centralized processor. This may have the advantage of allowing the rewrites to be specifically tailored to the CAD systems concerned.

An alternative approach to the exchange of procedural models has been used at KAIST in Korea [Choi et al. 2002; Mun et al. 2003]. This is based on the capture and transfer of the journal file created by a CAD system, which contains a record of every action of the system user. However, the work of the team concerned is now directed to extending the STEP-related work described in this paper.

The basic assumptions made in this work are as follows:

1. Any design intent not present in the model in the sending system cannot be transmitted. This implies that the work carried out does not involve general mechanisms for such things as automated feature recognition or other implicitly defined model characteristics.

2. Even if it is present in a CAD model, any aspect of design intent that is not accessible via the applications programming interface (API) of a particular CAD system cannot be exchanged unless it can be inferred by indirect means.

3. STEP does not consider details of the behavior of the receiving system if the transmitted model is modified there, but only assumes that it is as far as possible intuitive for the system user.

4. The minimal criteria for a successful parametric model exchange are the correct transmission of parameters and constraints and their appropriate interpretation in the receiving system.

The exchange of parametric construction history models using STEP is based on the use of a dual model, consising of a primary procedural model and an associated secondary model of the B-rep or some closely related explicit type. The secondary model can be used in the receiving system as a check on the validity of reconstruction there, and to resolve ambiguities, e.g., to determine which of several valid solutions of a nonlinear constraint system was chosen in the original model. Spitz and Rappoport [Spitz and Rappoport 2004] similarly use pure geometric representations of individual features for verification purposes.

It is believed that the work described here illustrates several advances on previously published methods:

1. The transfer of parameters and mathematical relations between them has not been previously reported. Any attribute in the part model can be treated as a parameter in an exchange, so that it is possible (for example) to transfer a relation defining the number of hole instances in a circular hole pattern in terms of the radius of the pattern.

2. The method can handle several different types of constraints (algebraic, logical, dimensional) in a very general and easily extensible manner.

3. The method handles multiple simultaneous constraints in the model in addition to sets of independent constraints.

4. The method conforms to the international standard ISO 10303 through the use of new parts of that standard.

Having said this, there are some restrictions. The tests concentrate on the transfer of sketch-defined features, and all geometric constraints are two-dimensional. STEP permits the use of 3D constraints, which are mainly used in practice for inter-feature relationships and for constraining parts in an assembly. These capabilities have not so far been tested, but they are no different in principle from their 2D counterparts.

The methodology used for translator development has been based on a careful analysis of all the different types of information present in a procedural CAD model, and of the different usages made of each of those types of information. The structuring of information into optimal 'units of creation' has also been given detailed attention, as discussed below. Finally, semantic differences between CAD systems have been analyzed in the interests of maximizing interoperability and minimizing information loss in the exchange of CAD models.

2 Design intent and its representation in CAD systems

The most essential aspect of 'design intent' is the constructional history of the model. If this is recorded as the primary aspect of the model's representation, it is possible to replay that history, with modifications if desired, in the certainty that the designer's original methodology will be followed. Other important contributions to design intent are the presence of parameters in the model, representing values that it is permissible to change, and the presence of constraints, defining relationships that must be preserved in any change. Design intent, as defined in this paper, therefore corresponds to the way in which facilities provided by a CAD system are used in order to achieve intended design aims. A distinction is made between this and *design rationale*, which is concerned with the reasoning underlying the way those facilities are used (see Section 3.6).

Further details of some important aspects of design intent information are given in the following subsections.

2.1 Construction history

The construction history of a model is a procedural representation of that model, expressed in terms of the operations used to build it. Modern CAD systems provide users with a range of high-level constructional operations which shield them from having to work at the level of individual geometrical and topological elements. In CAD system terminology the configurations they create are referred to as 'features', though this is misleading. Strictly, a feature has some associated application semantics [Shah and Mäntylä 1995], but with the present level of CAD technology the intended design functionality of the features created in a design process is present only in the mind of the designer; it is not captured by the CAD system. These operations should therefore be regarded merely as 'shape macros' which, in B-rep terms, construct relatively complex subgraphs of the overall topological structure. However, because the use of the word 'feature' is prevalent in the CAD design context it will be used in the remainder of the paper, subject to the proviso made in this paragraph.

A construction history, then, is primarily a sequence of operations that create shape features. Part 111 of STEP, which provides represention of shape features, is based on a survey of major CAD systems. Its intention is to capture a range of the most widely used feature types. In general, the ordering of elements in a STEP exchange file is immaterial — in the exchange of a B-rep model, for example, the order in which elements are added to the model as it is rebuilt after a transfer does not matter provided all necessary elements are present in the file. As is well known, the ordering of operations in a construction sequence is crucial, however; a different ordering will in general lead to a different model. Part 55 of STEP therefore defines special structures for the capture of operation sequences.

Each operation is defined, as in the CAD systems themselves, in terms of the shape feature created (expressed in descriptive geometrical terms) and the size attributes of that configuration. For example, in the case of a simple rectangular pocket feature the pocket will be characterized by four planar walls and a planar floor. The values of its size attributes will define its length, width and depth. Implicit in the operation of the creation procedure will be the facts that opposite pairs of wall faces are parallel, that adjacent wall faces are perpendicular and that the walls are perpendicular to the floor, regardless of the values of the size attributes. The dimensional attributes themselves have no independent existence as elements of the model. In this case we refer to the constraints and dimensions of the feature as being *implicitly* defined.

Many feature creation operations require additional supporting information, though this varies from one CAD system to another. Typically, a round hole feature will need a centreline. In STEP terms this may be an unbounded line defined by a point and a direction, or some equivalent construct. The line and its defining point and direction need to be present explicitly in the exchange file for the hole feature to be reconstructed following a transfer. These elements do not occur in operation sequences, because provided they are present their place in the exchange file does not matter. They are therefore transmitted in 'traditional' STEP unordered mode.

In CAD systems, supporting elements for feature creation operations are usually stored within the data structures representing the features concerned. STEP, by contrast, transmits them at the model level rather than the feature level. The reasons for this are partly to maintain upwards compatibility with previous practice in STEP, and partly to keep the transferred model at the most general possible level so that it is compatible with a wide range of CAD modelling methodologies. There is also the distinction that a CAD system data structure is designed for efficiency and a neutral data structure such as STEP for informational completeness. It is therefore hardly surprising that they differ significantly, and this divergence leads to some problems for the STEP translator writer, as will be discussed below.

2.2 Parameters

Parameters represent values that may be changed in a part model to generate different members of a family of parts. They include the dimensional attributes of feature creation operations mentioned above. Since these have no independent existence as elements of the model they are here referred to as *implicit parameters*. Explicit parameters, by contrast, are model elements in their own right. Part 108 of STEP defines representations for their transmission in an exchange file. The primary uses of explicit parameters in CAD modelling are (i) for specifying dimensional relationships in the 2D sketches often used as the basis for created features, (ii) for specifying dimensional relations between features, and (iii) for positioning part models in assembly models.

CAD systems differ in the way they deal with parameters. Most have separate data structures for sketches in which explicit sketch parameters are represented. In some cases separate model-level parameter data structures are provided, and used as the basis for the generation of tabular displays of parameters that may be changed in the model. Some systems, under some circumstances, will generate explicit parameters coresponding to parameters that were initially implicitly defined.

Another important aspect of parameters is that they may have relationships defined between them. For example, it may be desired to model a family of rectangular blocks in which the length of a block is always twice the width. More complex algebraic relationships may also be defined. Further, it is possible to define parameters whose values do not correspond to any physical quantities in a model, as in the case where the length and width of a block are required to be given by t^2+1 and $t+3$, t being a parameter that is not directly associated with any specific dimension in the model. All CAD systems provide capabilities of these kinds, and Part 108 of STEP makes provision for their capture and transmission, though in that document they are treated as specialized constraints. The topic of constraints in general is covered below.

2.3 Constraints

As with parameters, constraints have implicit and explicit forms. Implicit constraints were mentioned above; they occur automatically as the result of creation operations. Explicit constraints, by contrast, are modelling elements in their own right, which make reference to other modelling elements and constrain them to satisfy specified relationships. For example, a sketch of a rectangle with rounded corners is made up of four line segments and four circular arcs. This collection of geometric elements may be supplemented by explicit constraints requiring (i) opposite pairs of sides of the rectangle to be parallel, (ii) adjacent pairs of sides to be perpendicular, and (iii) lines and arcs to be tangential where they adjoin. Some CAD systems will add constraints to ensure that the end points of lines and arcs are coincident where they adjoin, but others achieve this result in different ways.

It was mentioned above that constraints involved in feature creation are usually implicit. Explicit constraints have similar application areas to explicit parameters: (i) for specifying geometric relationships such as parallelism, perpendicularity or tangency between geometric elements of 2D sketches, (ii) for positioning and orienting features with respect to each other, or with respect to datum elements defined in the model, and (iii) for positioning and orienting part models in an assembly model.

Part 108 of STEP [Pratt 2004b] defines entities representing explicit constraints. Apart from constraints specifying explicit mathematical relationships between parameters, a wide range of 'descriptive' constraints is provided, expressing geometric relationships such as parallelism or tangency. The latter basically record no more than the nature of the constraint and the elements that are subject to it. All CAD systems implement geometric constraints of these types, their semantics are widely understood, and so it is best if their precise mathematical formulation is left to the systems concerned. Some descriptive constraints have dimensional subtypes. For example, the parallelism constraint, applying to lines and planes, has a dimensional subtype whose interpretation is ‘parallel at a specified distance’; it only makes sense to assert the dimensional aspect of the constraint once the logical condition (parallelism) has been established.

CAD systems usually store explicit constraints in the datastructures of the individual sketches or features they relate to. However, Part 108 of STEP is part-oriented rather than feature-oriented, and explicit constraints may occur anywhere in an exchange file, in the same manner as the supporting information for feature operations mentioned earlier.

2.4 Sketches

CAD systems provide self-contained datastructures for sketches, as mentioned above. Parameters and constraints associated with sketches are usually stored in these datastructures. Part 108 of STEP provides representations for sketches, though it distinguishes between 2D sketches defined in neutral coordinate systems (such as might be stored in a library of sketches for multiple re-use) and the results of transforming them into 3D model space.

In the case of a sketch-based feature, the creation of the defining sketch is usually treated as a single operation, because although a sketch may contain many elements its parametric variation is determined by computation of a new solution to its constraint system rather than by a replay of its construction history.

2.5 Datums

Model elements are often positioned or dimensioned with respect to datums, which may or may not be geometric elements of the model. The centreline of an axisymmetric hole feature is an example of an element that is not a constituent of the B-rep of the model containing the hole. Despite this, it may be used as a reference element in positioning the hole in the model, perhaps with respect to a datum coprresponding to the axis of another hole. Most CAD systems store details of datums with the features or sketches that make use of them. Part 108 of STEP provides for the definition of datums under the general heading of ‘auxiliary geometric elements’, but as with parameters and constraints they are not associated in an exchange file with specific sketches or features.

3 Problems encountered in writing trial STEP translators for construction history models

This section outlines some of the primary problems encountered in writing translators for the exchange of CAD models of the construction history type with parameters and constraints. We start with a few general remarks.

The basic principle of the exchange of a feature-based construction history CAD model is that each successive creation operation is mapped onto a corresponding operation or combination of operations in the receiving system. This allows a natural decomposition of the overall process into a sequence of transfers of simpler components, which may be optimized for the transfer of each of those components. Ideally, the performance of the transferred sequence of operations in the receiving system will result in a correct reconstruction there. This process has the property that the partial models generated at various stages of the overall reconstruction process will also match the corresponding partial models that were created during the original design process. This is significant for the transfer of user-selected elements, as will be explained below.

In the tests performed, the translators read and wrote model information through the applications programming interfaces (APIs) of the CAD systems concerned. The success of such translations therefore depends crucially on the completeness of the functionality of those interfaces. In comparable experiments performed elsewhere it was found that not all CAD systems provided adequate access to the data required [Stiteler 2004]. The APIs of most CAD systems provide the translator developer with an entry point to the data structure of a represented model which, at the highest level, gives access to lists of features and sketches used in defining the model.

In the past, differences in the internal numerical tolerances used to judge coincidences etc. in CAD systems gave rise to major problems with geometry/topology incompatibilities in the STEP-based exchange of B-rep models [Gu et al. 2001]. These were largely overcome in the years following the initial publication of STEP, and currently there is a high success rate in the transfer of B-rep models. However, the introduction of design intent information into exchanged CAD models gives further scope for the occurrence of accuracy problems. In particular, constraints and dimensions may be satisfied by the accuracy criteria of the sending system but found to be unsatisfied by the more stringent criteria of a receiving system. So far, this has not been found to be a major problem in the tests performed. One can also take the view that the transfer of construction history information should alleviate accuracy incompatibilities, because the received model is always reconstructed according to the accuracy criteria of the receiving system, and mismatches are largely avoided. One potential area for mismatch remains, however, in the handling of elements selected by the user from the screen of the sending system. This will be discussed below, in Section 3.3.

3.1 Operation granularity

CAD systems differ in the complexity of the model substructures that are created by a single operation. We will use the term *granularity* in this context. An operation with coarse granularity in one system may need to be reformulated as a sequence of operations with finer granularity in another. The idea of a ‘unit of construction’ is a useful one; it is an operation or group of operations that results in the creation of a new geometric configuration in the model, but which may require different but equivalent sequences of one or more operations in different CAD systems. Two examples follow:

- Some CAD systems include positioning and orientation information in the basic definition of a feature (coarse granularity) while others allow the creation of the feature as an operation in its own right and then require the use of additional operations to position and orient it in the model (finer granularity).
- Some CAD systems allow the creation of underconstrained sketches or features, and permit later fine-tuning of the model in terms of lower-level operations. But other systems, with coarser granularity, only allow the creation of fully constrained constructs, possibly through the use of default options. The identification of such defaults, and their appropriate capture in an exchange file has proved to be one of the more difficult aspects of the work described.

The proof-of-concept tests described here attempt the automatic identification of units of construction with the same number of degrees of freedom in both the CAD system and the ISO 10303 neutral file. Degrees of freedom include dimensions and other parameters that may be defined either implicitly or explicitly. It is usually necessary to match a set of finer granularity system creation operations to a single ISO 10303 operation of coarser granularity, or vice versa. A one-to-one mapping would be ideal, but is rarely possible. The foregoing remarks apply both in the preprocessing phase (translation from the sending system to the ISO 10303 exchange file) and the postprocessing phase (translation from the exchange file to the receiving system).

The four possibilities for matching units of construction are clearly the following:

identity: there is a perfect match between operations;

aggregation: the translation must combine two or more finer-level operations to match an operation of coarser granularity;

decomposition: the translation must decompose a coarse granularity operation into two or more operations of finer granularity;

complex: it is necessary to use some combination of aggregation and decomposition.

To illustrate, we consider the case of a protrusion feature with a rectangular cross-section. Most CAD systems provide several means for the creation of such a feature, including

1. Creation of a block primitive and use of a Boolean union operation. These two operations map exactly onto operations that can be represented in STEP, and thus we have two cases of identity matches. The constraints on the form of the protrusion are implicit, being inherent in the definition of the block primitive.

2. Extrusion of a rectangular sketch defined on the part surface. The CAD system may provide rectangle creation as a single coarse-granularity operation in which the geometric constraints of the rectangle are defined implicitly, or may require the user to create a quadrilateral and impose the necessary constraints to make it a rectangle. In the latter case the constraints will be created explicitly in the CAD system by separate fine-granularity operations. At present STEP requires the constraints to be explicit, which implies the need for decomposition in the preprocessing phase if the sending CAD system is of the first type, and aggregation in the postprocessing phase if the receiving system is of the first type.

3. Creation of a general B-rep hexahedron, imposition of appropriate parallelism and perpendicularity constraints on its faces, association of explicit parameters with its dimensions, and use of a Boolean union as in Case 1 above. This method is possible in principle, and though it will rarely be used in practice it is useful for illustrative purposes. The protrusion will be editable in the receiving system because of its associated design intent information. However, in this case the block is defined entirely in terms of fine-granularity elements, and its mapping into the exchange file, for example as a block primitive, would require a high level of aggregation, together with the initial difficulty of recognizing automatically that the hexahedral B-rep does indeed have the form of a rectangular block. Alternatively, the exchange could be restricted to the transfer of the fine-granularity elements, without the added ‘block’ semantics. This would result in the exchange of the correct shape, but without the additional feature recognition process an important element of design intent would be lost.

As far as is known, feature recognition as suggested in Case 3 above has not yet been attempted for aggregation purposes in the the context of CAD model exchange, and it was ruled out of scope for the present work. If the model to be transferred contains explicit constraints and parameters these would provide a good basis for the use of the ’hint-based’ apporach to feature recognition [Han et al. 2000].

In general, translation should be performed at the coarsest possible level of granularity, because this preserves the highest level of user intent. However, the translation software must be endowed with considerable intelligence to enable it to determine the most appropriate units of construction. In many cases, STEP feature definitions have a more general specification than the corresponding CAD system features, which allows the possibility of mapping the feature plus several additional constraints from the sending system to a single feature in the exchange file. This maximizes flexibility for interpretation of that feature in the receiving system. In postprocessing, it is best to select the coarsest granularity compatible option from the feature library of the receiving system. Any remaining differences in the representations can then be taken into account using additional finer-level elements.

3.2 Feature support information

Most modern CAD systems used in mechanical engineering allow design in terms of features. However, there are wide variations between systems regarding what is and what is not regarded as a feature, and this can lead to semantic mismatches. For example, a datum may be treated as a geometric element in one system but as a feature in its own right in another. Generally, CAD systems store design information as a collection of feature representations, each feature having associated with it all the supporting information needed to define that feature.

STEP, by contrast, is part-oriented rather than feature-oriented. This is partly for historical reasons and partly because of the need to cater for all types of systems, including any that may not be feature-based. Details of information supporting feature definitions is therefore present in the exchange file, but it is not identified in any way as being associated with specific features. Such information may include, for example, an explicitly defined line used as the centreline of a cylindrical hole, or an explicit direction specifying the direction of extrusion of a sketch. In these two cases the hole and the extrusion, respectively, will refer to the supporting elements, but there will be no references in the reverse sense. The translation process frequently requires these inverse references, and currently they can only be found by searching the entire exchange file until the feature referencing the supporting element in question is identified. At present this process is made more efficient by the generation of ephemeral data structures recording the inverse relationships as they are found. However, this has been found to be one of the most computing-intensive aspect of the translation, and it is felt desirable to amend the STEP resource that defines design features to make such searches more efficient.

One area where the part-oriented data structure of STEP causes problems is in the handling of explicit constraints. Such a constraint occurs as an instance in the exchange file, and it references the model elements that are the subject of the constraint. But it does not make any reference to the feature or sketch to which those elements belong. For translation into a feature-based CAD system, the translator must therefore identify the feature or sketch concerned indirectly, by a search process that compares the elements involved in the constraint with the elements of features and sketches represented in the file. For example, if an instance of a constraint in the exchange file references a line instance, and a sketch instance references the same line instance, then clearly the constraint belongs to the sketch, despite the fact that it is not directly referenced by it. The most robust exchanges will be those in which all constraints can be assigned to features or sketches in this way.

The proposed change to STEP is simply the addition of a new entity to Part 111, the STEP resource defining design features, whose instances will provide on the one hand a pointer to a construction operation and on the other the relevant sets of explicit parameters, explicit constraints and their supporting elements relating to that operation. Then by scanning all such instances in the exchange file the translator can create a structure that lists all sketches and features and the constraint elements that belong to them, and use this for the correct allocation of elements to the sketch and feature data structures of the receiving system. The use of this new entity will be optional; for example, it would not be appropriate to use it for translating a STEP file into a CAD system that was not feature-based. Subject to agreement by the relevant ISO technical subcommitee, this new entity will be added when the International Standard version of Part 111 is published. It will not represent a technical change in the way that models are represented and transmitted, but will provide a redundant additional construct to aid in the setting up of temporary data structures needed for the translation process.

3.3 Identifiers and user-selected elements

In some CAD systems each feature, topological element and geometric element has an identifier that is unique in the model. However, in most systems the data structures are based on individual feature or sketch elements, and identifiers are only unique within those subunits of the overall model. A STEP exchange file is model-based, individual instances of STEP entities being referred to uniformly by identifiers of the form #*n*, where *n* is an integer unique to the instance concerned.

During the CAD design process the system user frequently selects model elements by picking from the screen. Such selected elements may be used as datums, or may be the basis of further creation or modification operations. An example is provided by the selection of an edge, or a set of edges, to be blended or filleted in a subsequent operation. In the CAD system concerned the selected element is referred to by its internal system identifier, but such identifiers can be correctly interpreted only in the context of the system where they are created. In the preprocessing phase no attempt is made to associate the system identifiers of the sending system with STEP instance identifiers, and in the postprocesing phase the STEP instance identifiers will similarly be discarded as system identifiers are generated for the reconstructed elements created in the receiving system. That being so, some system-independent means is needed for indicating elements in the exchange file that correspond to user-selected elements in the sending system.

The method adopted in STEP exchanges is to write the selected element explicitly into the exchange file, but to mark it as a selected element so that it can be correctly interpreted in the receiving system. The reconstruction of the procedurally defined model in that system will give rise to an element that corresponds to the element selected in the sending system, and that element may be determined by matching all model elements of the appropriate type against the explicitly transferred selected element. In the absence of a perfect universal persistent naming method [Capoyleas et al. 1996] this has been found to be the most robust method of dealing with the selected element problem. Admittedly, geometric accuracy problems may in principle cause the matching process in the receiving system to fail, but so far this has not been found to happen in practice.

A similar approach to the handling of user-selected elements has been adopted by Rappoport et al. [Rappoport et al. 2005], who point out that the matching process is often complicated by the fact that different CAD systems use different topological structures in representing the same shapes. The cited paper deals with the matching process for vertices and edges. In the latter case matching may require identification of a pair of edges with different end-points but lying (to wihin some numerical tolerance) on the same curve.

Another area where identifiers need to be handled carefully is in the transmission of mathematical relations between the values of dimensions or other parameters. Again, each CAD system has its own internal method for allocating identifiers to dimensions and parameters. These are both treated in STEP as mathematical variables with associated semantics, and if they occur explicitly in the exchange file they are referenced in terms of their instance identifiers rather than by any other form of identifier (they may optionally have an additional name associated with them in the form of a text string, but this is not intended to play any part in the exchange process).

CAD systems generally store mathematical relationships as strings, in the manner of scientific programming languages. STEP, by contrast, for reasons of upwards compatibility with earlier parts of the standard, represents them in a parsed form in terms of sequences of entity instances defining individual operators and operands. Translation between the two forms presents little difficulty; the major requirement is the careful recording in a temporary data structure, for both the preprocessing and postprocessing phases, of correspondences between system names of variables or parameters and the identifiers of their representations in the STEP exchange file.

Part 108 of STEP provides two types of mathematical relationships, the *assignment*, where the value of one variable is required to be equal to the value of an expression involving other variables, and the *relationship*, which specifies a more general type of relation involving two or more variables. Both equality and inequality relationships are provided in the latter case.

3.4 The interplay between implicit and explicit data

This topic is related to that of the granularity of units of creation, previously discussed. To illustrate the connection, we will consider a CAD system that provides an operation of high granularity that creates, dimensions and positions a new feature on the model in a single operation. Then all the defining information, including the positioning information, will be input as arguments to the creation operation. This information will therefore be present implicitly in the model, to use the terminology introduced earlier. If transferred directly into a STEP exchange file it will occur as values of attributes of the feature instance in the file.

It was earlier mentioned that not all systems adopt this approach. In a system with lower granularity, the operation used to create the feature and the operations used to position and orient it in the model may be separate. Suppose that a dimensional constraint is created for positioning purposes. In this case a dimension which is implicitly represented in the first system is represented explicitly in the second system, as an entity instance in its own right. Then in an exchange of models between the first and the second system one or more implicitly defined items of information must be made explicit. Conversely, for exchange in the reverse direction explicit information must be made implicit. The place where the conversion is made depends upon the granularity of the representations of the two systems with respect to the granularity of the STEP representation. Even here matters are not totally clear-cut, because STEP often provides alternative ways of representing the same configuration, as illustrated in Section 3.1 above. The recommendation made there was that the highest possible level of granularity should always be used for maximal preservation of design intent.

Matters become more complex when implicit information is hidden in the sending system. For example, the creation of a constant radius blend feature will usually lead to the designer's chosen blend radius being stored by the system as an attribute of the feature representation. On the other hand, some systems provide the capability for defining a default value for blend radii, and in this case the value of the default radius may have to be accessed in a different manner through the system API. In either case, however, the value of the radius is accessible, and a corresponding explicit dimension can be created in the receiving system if that is appropriate.

3.5 Differences in modelling methodology

Attention is restricted in this section to the topic of explicit geometric constraints, to provide some illustrations of differences in CAD system modelling methodology that have to be taken into account in a successful inter-system model exchange.

An explicit geometric constraint has a specification describing its semantics, and refers to two or more constrained geometric elements. Some CAD systems only allow binary constraints, in which the number of geometric elements involved is limited to two. In this case if the designer selects $N > 2$ geometric elements, then $N - 1$ separate binary constraints are created.

Geometric constraints are often *directed*, in the sense that one or more model elements is constrained with respect to one or more reference elements. In such a case the configuration may only be modified by editing the reference elements(s), when the constrained elements will automatically change so that the constraint in question remains satisfied. Undirected constraints also exist, in which all pairs of members of a set of elements are required to satisfy a specified constraint; for example, a set of planes may be constrained to be parallel to each other but not with respect to any reference element.

The API of a CAD system, when queried for the geometric elements involved in a constraint, may return either the system names of those elements or direct pointers to the elements themselves.

Some CAD systems do not allow the definition of 3D constraints within a part model. In these cases, the effect of a 3D constraint may be achieved indirectly, for example, by the creation of a datum element based on an element of one feature that is used in the definition of a second feature. While this implies a 3D relationship between the two features concerned, it may not be represented explicitly as a constraint in the CAD system. In an ISO 10303 exchange file, however, the possibility exists for expressing this relationship via an equivalent explicit constraint.

As far as the STEP neutral file is concerned, all geometric constraints must reference the underlying geometry of constrained topological elements (points, curves, surfaces). Sometimes it is necessary to refer to the defining elements of geometrical entities rather than the entities themselves. For example, if it is desired to constrain a cylindrical surface to be perpendicular to a plane it is necessary to formulate the constraint in terms of the axial direction of the cylinder rather than the actual cylindrical surface. This approach allows the total number of constraint types to be reduced because specialized constraint types do not have to be defined for each specific type of geometrical entity. On the other hand, the referencing of defining elements of constrained curves and surfaces rather than the curves and surfaces themselves creates more difficulties in the implementation of translators because the references to constrained elements are indirect.

A problem frequently arises when a datum acting as reference element for a constraint is translated from the sending system to a STEP file. Datum elements are usually defined by the CAD system with default dimensions, though in some cases no dimension is specified for them. For example, a datum plane may be displayed by a CAD system with default dimensions of 100×100 units, and it may be represented internally with that precise size and the topology of a face, or as an unbounded plane that is displayed as finite for easier understanding by the designer. In either case the pre-processor should create appropriate geometry to enable the post-processor to reconstruct the relationship involving the datum correctly from the point of view of the receiving system. Initial experience suggests that the most appropriate type of geometry to transfer for a datum is the most general – for example, unbounded lines, planes, etc. rather than bounded ones.

CAD systems may also re-interpret the user's input in some cases. For example, in creating a sketch the user may select a plane as the reference element for a constraint. The CAD system will then often represent the reference element as a compatible line rather than the chosen plane, reducing everything to 2D terms. Such reinterpetations can cause dimensionality problems in a STEP exchange; STEP regards a positioned sketch in model space as a 3D construct composed of 3D elements, and therefore will require the reference element of the constraint to have dimensionality 3.

Finally, we give an example of how the same geometric situation may be represented in different ways. It is required to constrain a line in a 2D sketch to be vertical. Many CAD systems allow this to be achieved in several ways:

- by simply subjecting the line to a **vertical** constraint;
- by using a **same-coordinate** constraint, requiring the x-coordinates of the positions of the end-points of the line to be equal;
- by constraining the line to be **parallel** to some other line that is vertical;
- by constraining the line to be **perpendicular** to some other line that is horizontal.

STEP permits the constraint to be transferred in the second, third and fourth of these forms, and a constraint originally expressed in the first form will need to be reformulated appropriately for transmission. Similarly, a particular receiving system may not implement the **same-coordinate** constraint, for example, and in that case the postprocessor needs to be provided with the intelligence to output one of the corresponding forms in order to preserve the design intent.

3.6 Design rationale

This topic was not addressed in the tests described, but it is mentioned here as being important for the future. It has been pointed out by Ohtaka [Ohtaka 1999] that a constructional history, even if it can be transmitted effectively, may be difficult to work with in a receiving system. One reason for this is because the history alone lacks any information about the designer's motivation in choosing a particular design methodology (this motivation is referred to as *design rationale*). In its absence, it may be impossible to understand why the designer used his chosen approach to the construction of the model, why certain features are present, why certain constraints have been imposed, and so on. Another reason is that the construction history of a complex model may be large and have many embedded levels of detail, and consequently be difficult to understand and modify simply for that reason. Both difficulties could be overcome through the provision of design rationale information with the construction history transfer, and it is very desirable that such information is captured for long-term data archiving. However, no effective method has yet been found for capturing such information automatically during the design process, and the best that can be done at present is to make provision for the insertion of design rationale in text form at appropriate points in the history. That will require the designer to input the text as he proceeds (possibly by speech recognition rather than via the keyboard), and problems may still result from differences in the manner of description of the design process, which is likely to vary significantly between designers.

3.7 Nature of the tests performed

This paper being in the nature of a survey, and covering a wide spectrum of issues, it is not possible in a limited space to go into fine detail regarding the tests performed. More detailed information will be given in a forthcoming journal paper.

The primary systems used in the exchanges were SolidWorks and ProEngineer (both registered trade-marks of their respective vendor companies). However, other major CAD systems were also examined, to ensure that the approaches used were compatible with those systems and that the necessary information could be read and written via their APIs. The test parts used were fairly simple, and were chosen to exhibit a range of different aspects of parametric feature-based design. Two examples are given here.

3.7.1 Case Study 1

In Case Study 1 the base shape is the linear extrusion of a 2D sketch originally defined on a datum plane. The sketch contains geometric constraints and dimensions. Two additional features are defined upon this base shape, a circular protrusion and a circular depression. In the originating system (ProEngineer) the constraints defined are 5 point coincidences, 3 tangencies, 2 horizontal direction constraints, a ‘same *x*-coordinate’ constraint between the centre point of the arc R92 and the lower end-point of that arc, and 5 dimensions. The part is illustrated in Figure 1. The numbers in boxes are those of entity instances generated in the STEP exchange file.

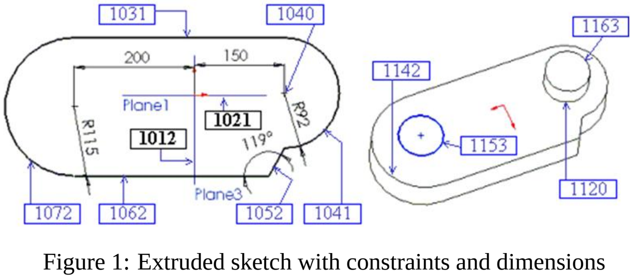

Figure 1: Extruded sketch with constraints and dimensions

In this case study the dimensions of the transferred model were editable in the receiving system, subject to the constraints imposed in the sending system.

3.7.2 Case Study 2

For Case Study 2 the base shape is a simple rotational shape with a circular pattern of four holes defined upon it, as shown in Figure 2. The position of the first hole is specified in terms of a datum plane and the distance between the axes of the part body and the hole. The pattern is then defined by a pattern creation operation, which automatically generates the angular dimension shown in the figure.

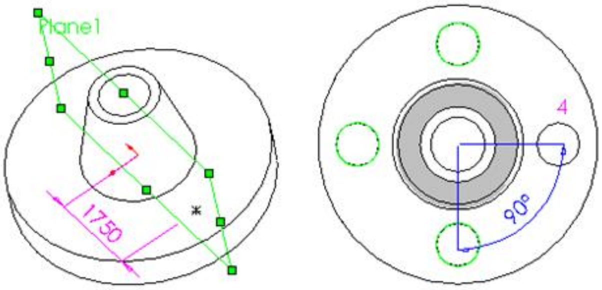

Figure 2: Rotational part with hole pattern

Two relations were defined between parameters for this part. The first simply expresses the diameter of its hole circle as twice the radius dimension (the value of the diameter is important in another context), and the second relates the number of holes in the circular pattern to the radius dimension. Variation of the salient dimensions of the part in the receiving system then allowed the generation of members of a family of parts as shown in Figure 3. The ’Master Model’ is the case shown in Figure 2, where there are four holes. The model contains ten independent parameters; all ten degrees of freedom were tested, and no incorrect results were generated. Dependent parameters were also correctly evaluated, subject to transmitted constraint relationships.

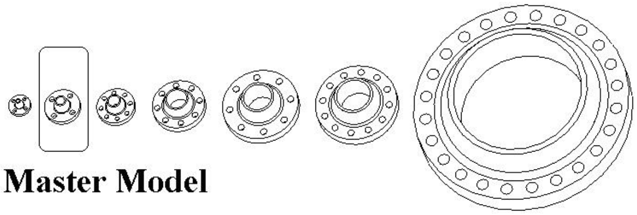

Figure 3: Members of a parametric part family

4 Other related tests

Other tests similar to those described here have been performed or are currently in progress, aimed at proving out different capabilities of the new STEP resources. A project coordinated by the organization PDES Inc. (http://pdesinc.aticorp.org) involved two major aerospace manufacturing companies and four of their supplier companies, and concentrated mainly on the transfer of pure construction history models. Some details are given in [Pratt et al. 2005], and the business case for exchanges of this type is made in [Stiteler 2004]. The rationale for pure construction history approach without parameterization and constraints is that many companies, in communicating designs with their suppliers, would prefer to suppress full details of the design intent in the original models because some of this information is regarded as proprietary. For data transfers within a company, however, the benefits of the more complete exchange are recognized.

The organization ProSTEP (http://www.prostep.de), based in Germany, is also engaging in tests of the new STEP facilities, though nothing has yet been published. The context here is automotive, and the Part 109 capability for representing parameterized assemblies is being evaluated. The data exchange experiments performed at KAIST in Korea have already been mentioned [Choi et al. 2002; Mun et al. 2003]; although they used a slightly different approach, there will soon be further demonstrations using Part 112 of STEP. Further related work, concentrating on the STEP-based exchange of sketches with parameterized geometry and constraints, has been reported from Troyes University of Technology in France [Charles et al. 2003].

5 Conclusion

The paper has described experiences in the testing of new STEP capabilities for the standard-based exchange of construction history CAD models with parameterization and constraints. Overall it has proved possible to transfer a range of different models, and to subject them to parametric variation in the receiving system. This is the primary criterion for success in such transfers; it demonstrates that design intent has been preserved in the exchange, a facility that was impossible until the recent development of Parts 55, 108 and 111 of STEP. However, certain difficulties and inefficiencies were identified in the course of the work, and one major potential improvement to Part 111 has been suggested that should significantly speed up the translation process in the future.

6 Acknowledgments

The first author is grateful for financial support from the US National Institute of Standards and Technology (NIST) under Contracts SB134102C0014 and SB134105W1299. As leader of the ISO TC184/SC4/WG12 Parametrics Group, he also acknowledges major recent contributions to the work of the group by Bill Anderson (Advanced Technology Institute, USA), Ray Goult (LMR Systems, UK), Soonhung Han (KAIST, Korea), Akihiko Ohtaka (Nihon Unisys, Japan), Vijay Srinivasan (IBM/Columbia University, USA), Tony Ranger (Theorem Solutions, UK), and Nobuhiro Sugimura (Osaka Prefectural University, Japan). He would also like to thank the many other people who made contributions during earlier phases of the work.

The second author gratefully acknowledges partial support through Korea Research Foundation Grant M01-2003-000-20351-0, funded by the Korean Government (MOEHRD).

7 Disclaimer

Certain commercial software systems are identified in this document. Such identification does not imply recommendation or endorsement by NIST; nor does it imply that the products identified are necessarily the best available for their purpose.

References

BOUMA, W., FUDOS, I., HOFFMAN, C. M., CAI, J., AND PAIGE, R. 1995. Geometric constraint solver. *Computer Aided Design* **27**, *6, 487 – 501*.

CAPOYLEAS, V., CHEN, X., AND HOFFMANN, C. M. 1996. Generic naming in generative constraint-based design. *Computer Aided Design* **28**, *1, 17 – 26*.

CHARLES, S., DUCELLIER, G., EYNARD, B., LI, L., AND RAKOTOMAMONJY, X. 2003. Standardisation des échanges de modèles géométriques 3D paramétrés non figés. *Revue Internationale de CFAO et d'informatique graphique* **18**, *4, 389 – 407*.

CHOI, G.-H., MUN, D.-W., AND HAN, S.-H. 2002. Exchange of CAD part models based on the macro-parametric approach. *International J. of CAD/CAM* **2**, *2, 23 – 31*. (Online at `http://www.ijcc.org`).

GU, H., CHASE, T. R., CHENEY, D. C., BAILEY, T. T., AND JOHNSON, D. 2001. Identifying, correcting, and avoiding errors in computer aided design models which affect interoperability. *Int. J. Computing and Information Science in Engineering* **1**, *1, 156 – 166*.

HAN, J.-H., PRATT, M. J., AND REGLI, W. C. 2000. Manufacturing feature recognition from solid models: A status report. *IEEE Trans. Robotics & Automation* **16**, *6, 782 – 797*.

HOFFMANN, C. M., AND JUAN, R. 1992. EREP — An editable, high-level representation for geometric design and analysis. In *Geometric Modeling for Product Realization*, North-Holland Publishing Co., P. R. Wilson, M. J. Wozny, and M. J. Pratt, Eds. (Proc. IFIP WG5.2 Workshop on Geometric Modeling, Rensselaerville, NY, Sept/Oct 1992).

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, GENEVA, SWITZERLAND. 1994. *ISO 10303:1994 – Industrial Automation Systems and Integration – Product Data Representation and Exchange*. (Information on ISO standards is given at http://www.iso.ch/cate/cat.html).

MUN, D.-H., HAN, S.-H., KIM, J.-H., AND OH, Y.-C. 2003. A set of standard modeling commands for the history-based parametric approach. *Computer Aided Design* **35**, *1171 – 1179*.

OHTAKA, A., 1999. Parametric representation and exchange: A sample data model for history-based parametrics and key issues. White Paper, ISO TC184/SC4/WG12/N295, International Organization for Standardization.

OWEN, J. 1997. *STEP: An Introduction*, 2nd ed. Information Geometers, Winchester, UK.

PRATT, M. J., ANDERSON, B. D., AND RANGER, T. 2005. Towards the standardized exchange of parameterized feature-based CAD models. *Computer Aided Design* **37**, *1251 – 1265*.

PRATT, M. J. 2001. Introduction to ISO 10303 — the STEP standard for product data exchange. *ASME J. Computing & Information Science in Engineering* **1**, *1, 102 – 103*.

PRATT, M. J. 2004a. Extension of ISO 10303, the STEP standard, for the exchange of procedural shape models. In *Procs. 2004 Shape Modelling and Applications Conf., Genova, Italy, June 2004*, IEEE Computer Society Press, F. Giannini and A. Pasko, Eds.

PRATT, M. J. 2004b. A new ISO 10303 (STEP) resource for modelling parameterization and constraints. *ASME J. Computing and Information Science in Engineering* **4**, *4, 339 – 351*.

RAPPOPORT, A., SPITZ, S., AND ETZION, M. 2005. One-dimensional selections for feature-based data exchange. In *Proc. 2005 ACM Solid & Physical Modeling Symposium, Cambridge, MA, USA*, ACM Press.

RAPPOPORT, A. 2003. An architecture for universal CAD data exchange. In *Proc. 2003 ACM Solid Modeling Symposium, Seattle, WA, USA*, ACM Press, 266 – 269.

SHAH, J. J., AND MÄNTYLÄ, M. 1995. *Parametric and Feature-based CAD/CAM*. Wiley.

SPITZ, S., AND RAPPOPORT, A. 2004. Integrated feature-based and geometric CAD data exchange. In *Proc. 2004 ACM Solid Modeling Symposium, Genova, Italy*, ACM Press.

STITELER, M. 2004. Construction History And ParametricS: Improving affordability through intelligent CAD data exchange. Tech. rep., CHAPS Program Final Report, Advanced Technology Institute, 5300 International Boulevard, North Charleston, SC 29418, USA.