Extension of STEP for the Representation of Parametric and Variational Models

M. J. Pratt[1]

The International Standard STEP (ISO 10303) is intended to facilitate the exchange of data between CAD systems. The first release of the standard, which occurred in 1994, allows the transfer of geometric product models in terms of geometry and topology alone. Since most CAD systems now allow the creation of parametric, constraint-based and feature based models, it is necessary to extend STEP to take into account these newer capabilities. What follows is a discussion of some of the considerations which have arisen in the early stages of this work.

Introduction

The topic addressed here is the problem of developing enhancements to the international standard ISO 10303[Int94], informally known as STEP. At present STEP is not capable of transferring parametrized geometry, constrained geometry or feature-based representations[Eas94]. Since most major CAD systems currently generate product models having these characteristics, an effort has recently started to extend the standard to accommodate them. For readers unfamiliar with the details some information on the organization of the STEP development work is given in Appendix 1. For the moment, it is sufficient to say that the information actually exchanged by STEP in a particular application context is specified by an Application Protocol (AP), and that at a lower level Integrated Resources (IRs) are provided which are are in general referred to by multiple APs. The IRs fall into the 40-series parts of the overall standard, and the APs into the 200-series. Refer to the Appendix for further details.

Since the early days of CAD there have been two approaches to product

[1]National Institute of Standards and Technology, Gaithersburg, MD, U.S.A.

modeling. One is the procedural approach, epitomized in former years by constructive solid geometry (CSG). The product description generated by a procedural modeler is a sequence of instructions for creating a model of the product; it therefore embodies a history of the process of model construction.

The other approach does not capture the constructional history of the model, but rather records explicit details of it at any one time during its construction. The boundary representation approach to solid modeling is an example of this approach. At any time during the modeling process what is stored by such a modeler may be regarded as a 'snapshot' of the model at whatever particular stage of its construction has been reached. No constructional history is recorded by, or can be inferred from, such a model.

In what follows the two types of models will be referred to as *implicit* and *explicit*. The justification for this is that in the first case it is not possible to obtain detailed information about the model until the constructional procedure has been followed through, while in the second case the full details are known although the constructional history is lost.

In its present form, STEP is almost exclusively concerned with the transfer of explicit product models of the boundary representation type. Furthermore, in the released parts AP201, AP203 and in other APs still in preparation, no provision is made for the transfer of parametrised entities or of models based on the use of constraints or features. Some future APs, notably AP214 and AP224, will allow the representation of feature-based models, but in the latter case these are machining features and in the former their intended application area in the overall product realization process is left indeterminate. No generic mechanism currently exists in STEP for the representation of features.

Although its current emphasis is on explicit models, STEP does make some concessions to the implicit or procedural approach. Part 42, for example, provides for the representation of CSG models built from primitive volumes by the use of Boolean operations. However, this is at present only possible using numerically rather than parametrically dimensioned primitives, because of the lack of parametric capabilities in the standard. Furthermore, none of the currently released parts of the standard actually make use of this latent capability. It is also worth noting that certain specialized geometry definitions may also be regarded as procedural, notably offset curves and surfaces, which are customarily defined in terms of a base entity and an offset distance. It may therefore be seen that STEP as it currently exists is something of a compromise, with implicit and explicit representations mixed up in a not very logical manner, the emphasis being strongly on the explicit forms. An alternative implicit or procedural means for the exchange of product models has been proposed by Hoffmann and Juan[HJ92].

Despite the apparently fundamental differences in approach between implicit and explicit modeling, most of the practical CAD systems of the last twenty years or so have in some sense provided both types of capability. Any procedurally-based system has needed the capability for generating explicit versions of its models, if only for the fundamental purpose of creating graphical renderings of them. Conversely, any explicit modeling system has provided the means for storing the set of modeling commands actually invoked by the system user in creating the explicit model. These commands are an implicit description of the model, and thus, in a very real sense, any CAD system may be regarded as a hybrid. There may be greater or lesser degrees of coupling between the two forms of the model. In the worst case there is no coupling at all. Thus, in early CSG systems, the explicit model was not generated until the implicit modeling process was complete, whereas later a higher degree of coupling was achieved through incremental evaluation of the explicit model during the construction of the implicit model. Similarly, in early boundary representation systems there was no coupling between the explicit model and the file of commands compiled; to change the model it was necessary to edit the command file and then re-run it from the beginning. Nowadays it is on many cases possible to 'roll back', i.e. undo the effects of the most recent commands, make a modification to some previous command and then roll forward again to generate the modified model.

Ideally, once a product model has been transferred from System A into System B, it should be possible to continue modeling with it in the receiving system just as though it had originally been created there. To enable this to be so regardless of the underlying nature of the sending and receiving systems is a major challenge to the developers of STEP, and whether it is even possible is an open question. However, we can take some comfort from the fact established earlier that all CAD modelers are in some sense hybrid, and thus they all share some common ground.

Current Directions for the Representation of Parametric and Variational Models in STEP

There are several teams working within ISO TC184/SC4 (see Appendix 1) who have an interest in parametric and constraint-based modeling. The ISO Parametrics Group hopes to coordinate the efforts of these teams to achieve a uniform approach across all the various parts of the STEP standard. The Parametrics Group is composed of two subgroups. One is concerned with a specific short-term issue, namely the urgent requirement to extend STEP so that it can handle the parametric, variational and feature-based information generated by modern CAD systems. The emphasis at present is primarily on explicit models, since it is felt that these are characteristic of the native models generated by such systems. Only the short-term issues will be covered here; the interests of the other ('long-term') Parametrics subgroup lie in such topics as knowledge representation and non-geometric constraints.

One other Working Group (WG) within ISO TC184/SC4 has aleady done some significant work on parametric modeling, and that is WG2, the developers of the future ISO 13584 Parts Library standard. This group has a requirement for parametric models that can be instantiated with appropriately chosen parameter values to represent any member of an entire family of parts. The approach taken has been an implicit one, part representations being specified in the form of a sequence of parametrized operations on geometric elements[Pie94]. For purposes of data transfer it is proposed that the implicit model description is transmitted together with a 'current instance', i.e. an example member of the family.

Numerous further SC4 teams have expressed an interest in the work of the Parametrics Group, with a view to using the capabilities it develops in new or revised parts of the standard. There is strong interest in the AEC area, for example, and potential application areas in mechanical engineering include all those where form features are likely to be used, including tolerancing, various manufacturing processes, assembly and inspection.

The Parametrics short-term subgroup is currently working closely with WG3/T1, the Shape Representation team, to develop the initial basic variable parameter and constraint capability in an extended STEP context. The declared aims of this work are to cover the capabilities of current CAD systems and to meet the needs of the WG2 Parts Library team. However, since this requires an explicit approach on the one hand and an implicit approach on the other it may be necessary to follow two different paths initially before achieving convergence in the future.

There are numerous liaisons with other groups in the STEP community, since extensions are now being made to various parts of the standard that have already been released. Ongoing discussions concern what changes need to be made in several parts of the STEP to facilitate provision of the new Parametrics capabilities, and the best ways of achieving upwards compatibility with the current version.

External liaisons will also be very important. Several projects are currently running, mainly in the USA, Germany and Japan, which will generate results useful to the ISO Parametrics work. It will also be necessary to circulate Parametrics proposals around the CAD vendor companies to obtain informed technical feedback on compatibility with the capabilities of their various systems.

The work of the Parametrics Group has only recently started, and efforts are currently under way to develop representations for parametrized entities and for constraints. Some of the relevant issues are discussed in what follows.

Parametrized models

The Generic Resource Part 42 of STEP provides definitions of a variety of geometrical and topological entities for use in building product model representations. At present these require all defining dimensions to be of type LENGTH_MEASURE, where LENGTH_MEASURE is an entity of type REAL defined in Part 41. In order to provide a parametric capability compliant with the current version of STEP it has been found necessary to define *parametric* geometric entities paralleling each of the types defined in Part 42, and to construct a schema defining variables and algebraic expressions for use in assigning variable dimensions to them. However, it has been determined that a comparatively small change in Part 41, to allow LENGTH_MEASURE to be either of type REAL or type VARIABLE, will allow the Part 42 geometric entities as currently defined to be used in a parametric manner. This provides a simple solution, without the need to double the number of geometric entities, but at the cost of changing an existing part of the standard. This is one example of the type of tradeoff which has to be made in revising STEP.

Another example relates to the need for a variables/functions schema. Such a schema has been developed in compliance with the standard as it currently exists[Int95], but it is fairly complex. The possibility is now being investigated that comparatively small changes in the EXPRESS information modeling language, the foundation of the whole edifice of STEP, may provide what is needed in a much less complex way. Since EXPRESS is already part of the current standard, any changes will need to be upwards compatible with the previous version.

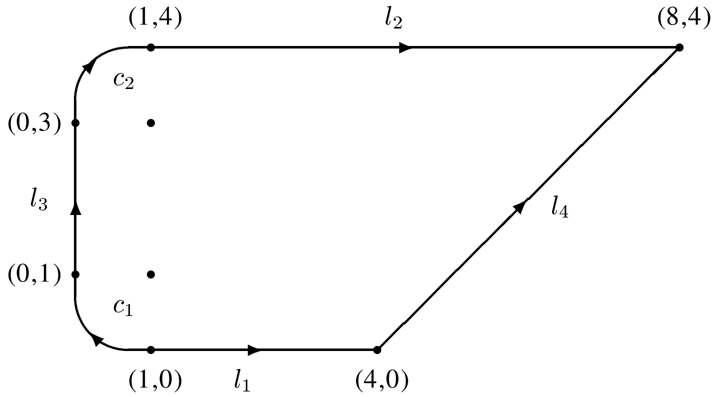
Changes of the kind envisaged will also affect Part 21, which specifies the manner in which STEP information is actually stored in a physical file. Once again, upwards compatibility must be the aim.

It will be seen that the decisions to be taken are of a delicate nature. What is desired is a compromise between representational power, conciseness, and degree of compliance with the existing released version of the STEP standard.

Constraints

It appears that the provision of constraint representations in STEP will give rise to fewer interactions with the existing parts of the standard, since these are essentially new entity types rather than variants of existing ones. The following discussion is based upon the simple 2D profile shown below. The geometry of the profile is composed of the entities listed below (they are not given in STEP format):

l_1: line from (1,0) to (4,0)
l_2: line from (1,4) to (8,4)
l_3: line from (0,1) to (0,3)
l_4: line from (4,0) to (8,4)
c_1: circular arc with center (1,1), start (1,0), end (0,1)
c_2: circular arc with center (1,3), start (0,3), end (1,4)



Some constraints which might be applied to the profile include

1. l_1 and l_2 are parallel
2. l_1.*start* and l_2.*start* are 4 units apart
3. the (oriented) angle between l_1 and l_4 is 45°
4. the (oriented) angle between l_2 and l_4 is 45°
5. c_1.*center* is equidistant from l_1 and l_3
6. c_2.*center* is equidistant from l_2 and l_3

7. c_1 and l_1 are tangent at c_1.*start* and l_1.*start*

8. c_1 and l_3 are tangent at c_1.*end* and l_3.*start*

9. c_2 and l_2 are tangent at c_2.*end* and l_2.*start*

10. c_2 and l_3 are tangent at c_2.*start* and l_3.*end*

11. l_1.*end* and l_4.*start* coincide

12. l_2.*end* and l_4.*end* coincide

13. l_1.*start* is fixed at (1,0)

14. l_3 is perpendicular to l_1

There is of course redundancy in this set, and in a practical situation only a subset of these constraints would be imposed. Since the emphasis here is primarily on the provision of mechanisms, the list is intended to be illustrative rather than to provide exhaustive coverage of all types of 2D constraints that will be needed in practice.

As STEP currently exists, only cases 1, 11 and 12 can be represented. Since a line is represented in terms of a point and a direction, the parallelism of l_1 and l_2 can be captured implicitly by making them share the same direction entity in the STEP file. An argument is given below as to why this probably not a good idea. The coincidence of end-points as expressed in Constraints 11 and 12 will be captured by making the specified bounding vertices of the edges in question lie at the same point, which is a standard connectivity mechanism in boundary representation modeling.

It may be noted that Constraints 2, 3, and 4 in effect specify *dimensions*, and any provision in this area should therefore be compatible with other STEP capabilities for the representation of dimensions. At present these occur mainly in the context of 2D drawings. The 'oriented angles' mentioned in Constraints 3 and 4 give the counterclockwise change in direction required to rotate from the direction of the first line specified to that of the second. If we regard perpendicularity as a special case of an angular dimension, as seems reasonable, then the same remark also applies to Constraint 14. In this case it will be logical to deal with parallelism as an angular dimension constraint in the same way.

As has been shown, some specific types of constraints in the above list bear some relation to existing Part 42 capabilities. Others do not, and some of the new capabilities required seem to be

1. The capture of tangency between specified end-points of two geometric entities (Constraints 7, 8, 9 and 10). This could be dealt with by a combination of a coincidence and a parallelism constraint, though for reasons mentioned below this would be undesirable.

2. The statement that a point is equidistant between two lines (Constraints 5 and 6).
3. The ability to state that a vertex of the profile is anchored at a fixed point (Constraint 13); this effectively allows other vertices to be moved during design modifications, but not the one at (0,1).
4. A concept of construction geometry.
5. The ability to reference subfields of geometric entities. The EXPRESS language does not currently permit this, and the examples above show a clear requirement for this facility in the next version of the language.

It must not be forgotten that STEP should also be able to deal with constraints involving *functional* relationships between entities. It was mentioned earlier, in the section on parametric modeling, that discussions are already under way on the best way of achieving this.

Since only a few of the desired constraint types can be handled by STEP as it currently exists, it seems best to provide a new and consistent treatment for all constraints, even at the expense of some slight redundancy in the information exchanged. In any case, employing the existing Part 42 capabilities mentioned above would really be misusing them, since their intention is essentially to capture the a *static* product model, not to impose restrictions on how it can be modified in the future.

Several considerations on the design of a constraint modeling capability in STEP are discussed in the following paragraphs:

What kinds of entities should constraints apply to? It is necessary to decide whether the constraints should apply to topological entities (e.g. edges) or to the underlying geometric entities (e.g. lines, or at a lower level, directions). The second alternative is probably most appropriate, since it seems to reflect what most CAD systems do in practice. Nevertheless, it may under some circumstances be useful to apply constraints to topological entities, and possible requirements for this should be examined.

Parallelism and other constraints on angular dimension: It was mentioned above that one way of handling constraints on the parallelism of lines is simply to define the lines concerned in terms of the same direction entity. The use of this implicit mechanism is consistent with existing Part 42 capabilities, but neverthless it does not seem to be a good idea. A parallelism constraint is a special case of an angular dimension constraint, and there is no comparable mechanism for handling the more general case, which includes the frequently occurring case of perpendicularity. In any case, CAD systems may replicate direction vectors rather than check for the pre-existence of the appropriate direction vector when each new line entity is created. Thus parallel lines may refer to different (albeit equivalent) direction vectors, and unless translators are designed to check for this kind of thing the implicit parallelism relationships may be lost. For these reasons, and also in the interests of consistency in modeling constraints, the initial conclusion is that explicit constraint representations are best.

Next consider the representation of a tangency constraint; this could in principle be done in terms of a coincidence and a (parallel) directional constraint. The problem with this approach is that it does not generalize to cases where (for example) second or higher order derivatives are required to be continuous across a join. In fact there is a tie-up here with one of the enhancements requested by AP developers in Part 42 ('Continuity Constraints'). It therefore seems desirable to treat tangency as one particular case of a more general 'geometric continuity' constraint.

Another question regarding tangency is whether STEP should capture only *apparent* continuity (i.e. the visual smoothness of the join in the diagram), or should take into account also the sense of the defined lines and arcs? In the latter case, Constraint 7 in list above would not count as a tangency, since the *directed* tangents are in opposite directions.

Mode of constraint representation:

Several different possibilities for the capture of constraint information have been suggested. These include

Declarative, e.g. **parallel**(l_1, l_2) — or, perhaps better, **angle**$(l_1, l_2, 0)$, which provides a unified way of handling parallelism, perpendicularity and general angular relationships. Here 'parallel' and 'angle' have been written in the form of PROLOG predicates[CM94].

Relational, e.g. **directionfn**$(l_1) =$ **directionfn**(l_2). In this case **directionfn** is a vector-valued function to be evaluated. Alternatively, it would be simpler in this case to use l_1.*direction* = l_2.*direction* in terms of attributes, but sometimes function evaluation is going to be needed.

Algebraic, e.g.

$$(l_1.end.y - l_1.start.y)/(l_1.end.x - l_1.start.x) =$$

$$(l_2.end.y - l_2.start.y)/(l_2.end.x - l_2.start.x),$$

which is equivalent to the two previous examples. It is probably a less desirable alternative, for two reasons. Firstly, its geometric significance is not intuitively apparent, and secondly it requires exceptions for special cases, for example when one or other denominator is zero. The second problem could be overcome by expressing the constraint in terms of projective coordinates, but the additional complexity makes the simplicity of the declarative and relational approaches seem increasingly alluring.

On the other hand, 'special-case' explicit algebraic constraints of the type

$$box.length = sqrt(box.height^2 + box.width^2)$$

are probably indispensable; it is true that they can also be expressed in a declarative form, but much is thereby lost in terms of comprehensibility.

Geometric. This category has been suggested for examples of the type 'The centre of circle c_1 is the perpendicular projection of the intersection of lines l_1 and l_2 onto line l_3.' This could be formalized in declarative terms, given a supply of predefined PROLOG predicates, as follows:

$$\begin{aligned}&\textbf{intersection}(l_1, l_2, p_1);\\&\textbf{normproj}(p_1, l3, p2);\\&\textbf{center}(p_2, c_1)\end{aligned}$$

Clearly an algebraic expression of this constraint is also possible, though it will be rather complicated. In relational terms we could use

$$c_1.center = \textbf{projfn}(\textbf{intfn}(l_1, l_2), l_3)$$

This is concise, but its meaning is not intuitively obvious.

Both the declarative and relational approaches have the virtues of conciseness, and both provide more comprehensible semantics than the algebraic formulation. There seems to be no virtue in identifying a separate category of 'geometric' constraints, since the other three representations are sufficiently flexible to cover situations that will arise in practice.

Different CAD systems doubtless represent constraints in different ways, and for purposes of data exchange it is important that the formulation used for them should capture their meaning or semantic content. The algebraic approach only does this at a very low level, in terms of relations (in STEP terminology) between individual attributes of entities; the 'engineering' intent of the constraints is lost. The declarative and relational approaches, on the other hand, are much more successful in expressing the meaning of constraints as they apply to entire modeling entities.

The issue of semantics is important, since however a constraint is represented in a STEP model it must be possible to translate it into whatever is the appropriate format for a receiving system. This requires that the translator must in some sense be able to 'understand' the nature of the constraint in order to reformulate it if necessary. In general that will be impossible for a constraint expressed algebraically in a STEP file, which can therefore only be passed on into the receiving system in algebraic form. This is not a problem in cases such as the box dimensions example given in 3) above, but for constraints involving relatively high-level geometric concepts such as 'parallelism' or 'tangency' it is highly desirable to capture those concepts in the information transferred. One possibility might be to require that any algebraically formulated constraint having one of a 'standard' set of geometric meanings should be transmitted together with a declarative or relational statement of that meaning.

These, then, are some of the considerations to be borne in mind when the modeling approach for STEP constraints is decided. Two further significant points are briefly discussed below:

1. All the examples given are two-dimensional. We will of course have to watch developments in the area of 3D constraints, and to try to accommodate new CAD capabilities as they arise.

2. One suggestion made during a recent ISO meeting is that constraints should be weighted or prioritized, so that in the event of conflicting constraints a 'best' compromise solution (in some sense) can be determined. At present there is probably little use for this capability, but it may become important in the work of the Long-term Parametrics Subcommittee when they come to discuss the handling non-geometric design constraints.

Features

The provision of feature capabilities in STEP will not be discussed in detail. It has been mentioned that feature representations are being developed in the context of two future APs, but these will be limited in extent. What is really needed is a generic capability in the standard for the flexible definition of feature classes, and this will become possible once mechanisms for parametrically defined entities and geometric constraints are in place. Features will therefore be addressed by the ISO Parametrics Group in the medium-term future.

Conclusions

Some preliminary considerations relating to the representation parametric and constraint information in the STEP standard have been presented. The whole discussion has been at a low level, but clearly any of the representations eventually chosen must be fitted into a hierarchical data structure and ultimately into one or more STEP schemas. Top-down analysis will therefore be needed to supplement the bottom-up approach whose initial stages are given here.

Acknowledgements

The author would like to thank members of the Parametrics Group and the Shape Representation Task Group of ISO TC184/SC4 for helpful discussions. In particular, Edward Clapp of Autodesk Inc. suggested an initial version of the constraints example analyzed earlier, and provided insightful comments upon it.

References

[CM94] W. F. Clocksin and C. S. Mellish. *Programming in Prolog (4th Edition)*. Springer-Verlag, 1994.

[Eas94] C. M. Eastman. Out of STEP? *Computer Aided Design*, **26**, 5, 338 – 340, 1994.

[HJ92] 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).

[Int94] International Organisation for Standardisation. *Industrial Automation Systems and Integration – Product Data Representation* *and Exchange*, 1994. (International Standard ISO 10303, informally known as STEP).

[Int95] International Organisation for Standardisation. *Industrial Automation Systems and Integration – Parts Library, Part 20: General Resources*, 1995. (Committee Draft ISO CD 13584-20).

[Pie94] G. Pierra. Parametric product modelling for STEP and Parts Library (V0.3). Working Document ISO TC184/SC4/WG2 N183, International Organisation for Standardisation, July 1994.

Appendix 1: ISO TC184/SC4 and STEP

The International Organisation for Standardisation (ISO) administers a range of committees, which in general are divided into subcommittees. ISO Technical Committee 184, Subcommittee 4 (TC184/SC4) is concerned with the development of International Standards for the digital representation of product data and manufacturing management data. This is the forum in which ISO 10303 (informally known as STEP: STandard for the Exchange of Product data) has been under development since 1984. The first release of the STEP standard[Int94] occurred in 1994. Earlier standards (for example IGES) were intended primarily for the exchange of pure geometric data between design systems, but STEP is intended to handle a much wider range of product-related data covering the entire life-cycle of a product.

ISO TC184/SC4 is also responsible for the development of ISO 13584 (Parts Library), a future standard for accessing libraries of standard part information for use by designers. Additionally, early work is in progress on a standard representation for manufacturing management data (MANDATE).

The development of STEP has been one of the largest efforts ever undertaken by ISO. Several hundred people from many different countries have been involved. The standard is being released in parts. Currently there are twelve of these, but many more are in preparation, dealing with specific product ranges (e.g. automotive, AEC, shipbuilding, electrical, ...) and different aspects of the product life-cycle (design, finite element analysis, process planning, ...).

The initial parts of STEP dealing with geometry transfer are two Application Protocols, AP201 (Explicit Draughting) and AP203 (Configuration Controlled Design). The first is concerned purely with 2D drawing information, while the second covers wireframe, surface and boundary representation solid models. The content of AP203 models is restricted to geometric and topological data, together with 'configuration' information relating to such matters as version control and release status.

STEP is designed to operate in the first instance as a 'neutral file' transfer mechanism. Each CAD system must be provided with a *preprocessor* and a *postprocessor*. Their functions are, respectively, to translate native data from the sending CAD system into the neutral STEP format, and to translate from the neutral format into the native format of the receiving system. This philosophy only requires the provision of $2n$ translators for exchange between any pair chosen from n systems, rather than $n(n-1)$ if 'direct' translators have to be written. As an alternative to file transmission, STEP information may be stored in a database, and a STEP Data Access Interface is being developed as part of the standard to allow the use of shared data access.

Many CAD vendors have developed or are developing STEP AP203 translators; some are already commercially available, while others are under test. Some third-party software vendors are also marketing STEP AP203 translators.

The currently released parts of the standard are

Part 1	Overview
Part 11	EXPRESS language (used in writing the standard)
Part 21	Physical file format
Part 31	Methodology and framework for conformance tools
Part 41	Fundamentals of product description and support
Part 42	Geometric and toplogical representations
Part 43	Representation specialisation
Part 44	Product structure configuration
Part 46	Visual presentation
Part 101	Application resources: draughting
AP (Application protocol) 201	Explicit draughting
AP (Application protocol) 203	Configuration-controlled design

The structure of the standard is fairly complex, but the lower numbers (100-series and below) define the infrastructure and a set of integrated resources. The actual data exchange standards are specified by the Application Protocols, and these are defined in terms of the lower-level resources. The EXPRESS language is an information modeling language, rather like a programming language, which is used for the formal definition of constructs in the exchange files.

Parts of the STEP standard still currently under development are freely available by anonymous ftp from the Solis information server at NIST (ftp.cme.nist.gov). Files which can be downloaded are in directory pub/step, listed under STEP Part numbers (e.g. part224).