Implementors Issue Log

Updated October 24, 2000

Note: In November of 1999, this log was updated to reflect when issues were logged. The dates assigned to issues created 1/1996 and prior were assigned 1/1996 since it was too difficult to research the date.

Table of Contents

Issue: 001 Scope

What does scope (in Part 21) mean? (Bernd Ingenbleek 1/1996*)

Discussion:

#1 = LINE( );
#2 = SCOPE
#3 = ...
#4 = ...
END_SCOPE LINE ( );

The intention in Part 21 is that #3 and #4 are depending on the existence of the LINE instance

END_SCOPE EXPORT (#3, #4) LINE ( );
is also allowed.

Nigel Shaw's view is that this complexity is not required. Nigel Shaw will find out what the conclusion at the ProSTEP round table was concerning this issue. At present, one cannot claim to have an implementation conforming to Part 21 if one cannot parse the SCOPE statement.

There is no statement that a processor reading the physical file needs to check if these files are correct. No EXPRESS rule-checking is required. The requirement is that any file output by a tool is correct

Status: Closed, SEDS
The Implementors feel that this capability is currently too open and confusing. The use of SCOPE is currently not recommended by both PDES Inc. and ProSTEP. The committee requests that WG11 define how SCOPE was intended to be used.

WG11 response is that this will be resolved in edition 2 of Part 21.

Issue: 002 Scope

Is a physical file that references a SCOPE statement correct? (Bernd Ingenbleek 1/1996*)

Discussion:

In general: What shall post-processors do if they encounter an error in a STEP Physical File they are reading. There seems to be a statement in the ATS documents according to which the postprocessor shall report the error and process as much of the SPF as it can. It shall not stop nor produce errors nor make any attempt to correct any error.

In the AP document one may limit any implementations of that AP to Part 21 not using the SCOPE statement. As for the SCOPE case: State that the file does not conform to Part 21. Inform the source of the file that the file is wrong. (Ask WG11 for advice to fix the problem with SCOPE.)

One has to be able to read all entities in a conformance class, but one does not have to be able to instantiate all entities in a conformance class, i.e., one may have to read a circle and be allowed to approximate it by a polyline (but one then has to state that this polyline has been a circle with a particular radius).

Status: Closed, SEDS
The Implementors feel that this capability is currently to open and confusing. The use of SCOPE is currently not recommended by both PDES Inc. and ProSTEP. The committee requests that WG11 define how SCOPE was intended to be used.

WG11 response is that this will be resolved in edition 2 of Part 21..

Issue: 003 Integers

Part 11 states that INTEGERS are a specialization of REAL numbers (Bernd Ingenbleek 1/1996*)

Discussion:

Thus, integers are reals. Integers are not allowed to be followed by a " . " (dot). All reals have to be followed by a " . " (dot). What is to be done?

Syntactically, one cannot have a SELECT between a REAL and an INTEGER in EXPRESS. Unresolved in Part 21 is which path to give in a SELECT tree if there are several paths possible. This happens in Part 46 (e.g., for text).

Status: Closed, SEDS

WG11 response is that this will be resolved in edition 2 of Part 11.

Issue: 004 ARM vs AIM

How can the population of an AIM be checked if the application specific terms are only used in the ARM? (Bernd Ingenbleek 1/1996*) How can ARM-level rules be checked at the AIM level?

Discussion

The ARM is informative only. ARMs cannot be implemented because they are not free of redundancy.

Several ways to look at the correlation of the Application (Software Product), ARM and AIM:

```
        GUI             Processor
    ARM -------->Mapping ----------->AIM ------> Physical File


                    Processor
    GUI -------->ARM-------------->AIM ------> Physical File


        GUI             Processor
    ARM -------->AIM ------------> Physical File
```

Harald Scheder:

Preparing the development of a STEP-based product data management infrastructure at BMW we are performing a series of tests and prototype developments. Beside the usual problems which we are facing there are at least two where we did not find an easy answer for.

May be there is someone out in the community who is able to give us a hint for the following issues. Moreover I think that we are not the only ones dealing with those problems and to my understanding both problems are open issues within the further STEP development in ISO.

A: It is clear and nobody has any doubt about the fact that the AIM schema of an AP is the bases for exchanging data between different CAx-systems.

However this approach is questioned if data have to be exchanged between two database systems. Talking about application data stored in a database system it is getting common understanding that therefor the AIM schema is not appropriate. Since the applications deal with application objects which fit much better to the ARM schema than to the AIM the following approach provides is the most promising solution.

```
  +--------------+  +--------------+
```

```
! Application 1 !   ! Application 2 !
+---------------+   +---------------+
        I         I
        I         I <--'data sharing'
        I         I
        +--------------+     +------------+     +-------------+
        ! Application  !     !   STEP-   !     ! STEP-FILE !
        !  database   ! <=> ! Interface ! <=> !  Part 21  !
        ! (ARM-based)  !     !(ARM<=> AIM)!     ! (AIM-based) !
        +--------------+     +------------+     +-------------+
```

                    'data exchange'

B: A further open issue is the multi language support in STEP.

As a global acting company, BMW needs to store some product information (i.g. PART_NAME or CHANGE_DESCRIPTION) in German and English language in parallel. I assume that other companies have the same problem. There have been some discussions within the SC4 to approach this problem via adopting the SGML solutions. But to my knowledge the importance of the issue is not recognized adequately and the issue is seen to be only a low level requirement.

Since the issue is of great importance for industry, it could become a major obstacle for using STEP as basis for product data management systems in global acting companies. The question is how to get to a quick, feasible, and reasonable solution avoiding inventing private workarounds?

Alan Wilson:

I would agree (as most would) that data sharing is much more efficient than data exchanges using a neutral file. Data sharing, however, implies that many of the details of the data are agreed upon. Details such as units of measure for linear dimensional data (mm, cm or m?), coordinate systems (which axis is Up and where is the origin in reference to the part?), and so on. These details about the data are not usually explicit in the ARM. If they were, the ARM could become the AIM.

Now to look at other situations to the above were you still need the STEP file:
If you buy a STEP based application-3 from a vendor who did not participate in the design of the application database, then there is a need for the STEP file to exchange data between application's 1 and 2 to application-3.

If you want to archive the data for later use in the future and application's 1 and 2 are to be discarded, then there is a need for the STEP file to load data into future STEP based applications.

This has been one of my hot issues since I first became involved in STEP in 1986. STEP is writing an

international data exchange standard but expects everyone to read English. That is not a very international point of view. (I can agree that English is OK for keywords that computers will read when parsing STEP files.)

The simple answer to the question above is that this is an ARM issue. If industry has a need to provide product information in multiple languages, then add that requirement to the ARM. The AIM could then be constructed to allow lists of strings instead of a single string for the product information. A more general solution would provide an AIC or a new resource for a multi-language list of descriptive information. This list could be used in any AP and anywhere there is a need for multi-language capability.

A concern here is that product keywords in the STEP file (such as a product name "bolt") must have a consistent computer readable form in addition to any human readable forms. It is impractical (and sometimes impossible) for an application to search multi-language lists looking for a string it recognizes to determine the product name.

Washington DC Discussion:
Bernd Ingenbleek gave a presentation on his question of ARM to AIM mapping.

Bernd stated: What about inverted ARM-AIM mapping? Does the inverted mapping exist? If it exists where is it documented?

The consequence of not having the inverse mapping is the ARM or AIM has to be thrown away.
If the inverse mapping exists but is not documented explicitly, every Implementor does it by himself, in a different way, probably with a reasonable number of errors because of complexity.
If there was an explicit inverse mapping, then confusing redundant information would be in the AP; one of ARM or AIM should be thrown away.

A lengthy discussion followed with the following comments made. ARMs are informative so they should not be implemented (not free of redundancy) The pictorial ARM is informative, the text is normative. An inverted mapping does not exist. The AIM contains more information than ARM due to the mapping to the IRs. Cannot find all info in the AIM without help, may need to look at the ARM and follow the mapping. The AIM has no information on process. The AIM defines the standard representation of the information. The ARM is an aid to help interpret information into AIM. Going ARM to AIM requires using integrated resources, which will give you extra data. Integrated resources exits to keep STEP integrated. The ARM is part of the process in developing the AIM - don=t need it after the standard is adopted. ISO does not have a standard picture format. The AIM is the Standard. The ARM is documented as to how the standard AIM came to be. Given a physical file which follows all requirements of ARM but required data is not defined in AIM (since it is not in mapping table) then the physical file is incorrect.

6

Agreement there is a problem -- not necessarily a known solution

Status: Closed. Issue has been sent to quality committee since it involves changing how APs are documented.

Issue: 005 Vertex Loop

The vertex loop was added to 203 IS version. The vertex loop is basically a face pointing to a loop which then points to a vertex. How its suppose to be used? (George Baker 1/1996*)

Discussion:

Peter Wilson:

Think of a cone. In a BREP, this can be represented as two faces, one a circular disc for the bottom and the other a conical face. There has to be an edge loop at the cone/plane boundary. In the minimum topological case, the point at the top of the cone can be represented by a vertex loop. That is, the boundary of the conical face is represented by two loops. One the edge loop at the base and the other the vertex loop at the cone apex. There are, of course, an infinite number of topological structures that could be used.

George Baker:

Thanks for the help. I still have a few more questions though. Since the vertex loop has no "curve", the loop has no direction. Is it always to be considered an outside loop or in the case of a non-manifold application can it be considered an inside loop? In the example you described, if I consider the vertex loop as an inside loop, then I know that the cone is bounded on the one side but not the other.

Can a vertex loop be used anywhere on a face or must it be used at poles only? I noticed in some of the PDES, Inc. plugfest files that one vendor put the vertex loop on the sphere, torus, and cone. Performing the Euler test on these solids, the sphere and the cone check out, but the torus doesn't add up. Is this valid?

Some one mentioned that the vertex loop could be used for any face that defined the natural outer boundaries for the underlying surface. If this is the case, and I don't have to worry about the Euler test, I could send out just about any solid with nothing but vertex loops on them.

Is there some place where implementation issues are kept? I know IGES has a recommended practices guide where the use of an entity like this could be documented a little better than just a three line description in the specification.

Peter Wilson:

Essentially a vertex loop is like an edge loop, except that its underlying domain is a point, not a

(complex) curved path. It can be used anywhere in a face.

Consider "drawing" on a face -- perhaps as a preliminary to creating a "subface" that is then extruded into/out of the solid. The first drawing action is to put the pen on the face; this creates a vertex loop. Moving the pen along creates a curve (and hence the vertex loop "changes" to an edge loop).

Felix Metzger:

In math, they use a much simpler model without such degenerate cases: Every shell is decomposed into a "Triangulation" where every edge has a different start and end vertex, where every face has exactly one loop, and where it does not happen that one edge of the loop of any face coincides with another one of the same loop. They do this just for the topology. For the geometry they allow that two different faces are associated with two different curve_bounded_surfaces where the underlying surface can be the same.

The advantage is that you have much less special cases to consider. Such a Triangulation can be constructed from a Part 42 model by adding edges and vertices.

For Peter's example of a cone, for instance, he has one shell (the whole cone surface), two faces (the basis plane disk and the conic surface), one edge (the circle which is the border of the basic disc), one vertex (a point on this circle) and one thing which is a vertex, a loop of edges, etc at the same time, called a "Vertex Loop" (the tip of the cone).

If you add an edge between the vertex and the tip, and you add another vertex somewhere on the circle, and you add another edge between this new vertex and the tip, you have a true Triangulation.

I personally would have preferred such a simple model in the standard, but I missed to submit comments against part 42 on this topic, because we noticed this problem too late.

Phil Rosche, you could use such a Triangulation as your intermediate data structure for your implementation: your internal data structure would then be a true Triangulation where you map any STEP model onto it by adding edges and vertices, and your application would then be based on this internal structure. I believe this was much easier.

Jim Jones:

Is it valid to use a vertex loop on a torus in a manifold solid? The example AP203 file I have seen contains a manifold_solid_brep consisting of 1 closed_shell, 1 face_surface (referencing a torus), and 1 face_bound (a vertex_loop). Realizing that a torus has a genus of 1, Euler's equation, V - E + 2*F - L - 2*(S- Gs) = 0, would be: 1 - 0 + 2*1 - 1 - 2*(1 - 1) = 2

It appears that this representation is invalid as it does not satisfy the required version of Euler's equation. Am I missing something here? Can anyone give me an example of a case where the vertex loop, at a location other than a pole, can be used in a valid manifold solid?

Peter Wilson:

The correct topological elements minimally required for a BREP torus are as follows, but first think of the torus and draw two circular curves on the surface (to represent edges). One is a small circle corresponding to a cross-section through the minor diameter and the other a large circle round the "equator" of the torus. Thus we have two circles that are perpendicular to each other, intersecting at a single point (a vertex).

The topological elements are: a vertex (as above), two edges (matching the two circles), one edge-loop consisting of the two edges, one face, and one shell. As you noted, the genus of a torus is one. Euler's formula then is:

$V - E + 2*F - L - 2*(S - G) = 1 - 2 + 2 - 1 + 2(1 - 1) = 0$ as required!

Now put another vertex somewhere on the toroidal face, but not on either of the edges. We now increment both the number of vertices and the number of loops by one, all other elements remaining the same. As the term $(V - L)$ appears in the Euler formula, the formula remains satisfied.

This last is not a particularly practical application of a vertex loop, so let me give you one. Suppose we have a BREP solid and want to designate some position on one of its surfaces as a datum point in a dimensional inspection application. Topologically (or logically if you prefer) this location could be "marked" by a single isolated vertex (and hence vertex loop). I don't know whether current STEP has such an application, but the topology was designed so as not to prevent such a usage.

There is certainly a minimal topology forced by the Euler equation for any geometrically bounded shape. Additional topological constructs can be added for the convenience of other applications. (This is what Felix was suggesting with his triangulation (i.e. simplex) message. Mathematicians most certainly use the topological forms within STEP and do not necessarily restrict themselves to triangulations). We tried very hard to produce a general topological model.

Kevin Weiler:

The problem related to the questions on obtaining valid Euler equations with tori and single vertex loops actually has little to do directly with either tori or single vertex loops.

The problem is that in order to obtain a valid Euler equation, the definition of a face must include the restriction that it is mappable to a plane. That's why a sphere with only a single vertex loop as a

boundary works with the equation given but using only a single vertex loop as a boundary on a torus' surface doesn't -- because in the latter case, the result doesn't produce a valid face which is mappable to a plane.

To create a minimal topology for a torus that also satisfies the equation, you must make the surface mappable to a plane. In this case, create a circle using a single edge with the same vertex at both ends -- the circle must go completely around the torus either through the minor or through the major diameter --it doesn't matter. What you get is a single face with two loops which is mappable to a plane. One loop consists of the edge from one side, the other loop is the same edge as viewed from the other side, on the surface.

For what it's worth, the usefulness of satisfying the Euler equation is highly over-rated.

First of all, it imposes a procedural restriction on the representation -- that is, while it doesn't restrict the domain (staying within the manifold domain apparently intended here), it does restrict how one represents things in the domain. For example, a sphere without any boundary (or a torus with a single vertex loop) just won't do, even though people can conceive of such situations as something they'd like to do directly without the extra baggage imposed by the "face- mappable- to- a- plane" restriction implied by use of the Euler equation. You can create the object you want, but you can only do it by adding stuff you don't want (the extra edge boundary in the torus example) -- a procedural restriction.

Second, and perhaps even more important, even accepting the first drawback, it is a necessary but not *sufficient* condition. That is, one can create numbers of vertices, edges, and faces that look good in the equation but create a nonsense object which doesn't adhere to the other "silent" constraints that are not embodied in the equation. For example, that each edge is used twice, once on each side in opposite directions, by one or two faces. Mantyla's paper on Euler ops, as well as many others, have identified some of the constraints that must be applied if one is create a valid model.

Third, there's the problem of knowing what the genus is if you want to use the equation for validity checking. Perhaps the best use of the equation is to compute the genus, given a valid model. I've seen little need for even that in practical applications, however. Filling numbers of components into the Euler equation doesn't do very much for ensuring validity. One would be better off identifying the information model, listing the constraints, and dropping any mention of the Euler equation -- trotting out the Euler equation seems to give people a false sense of security without doing much of anything. If a more comprehensive set of constraint checks were included, then the Euler equation check might be a reasonable part of a validity check; it's interesting, but it doesn't do nearly enough all on its own to be worthwhile.

Edward Clapp:

After thinking about the exchange last week, I'd like to make an attempt at some clarification and

perhaps throw some fuel on the fire.

When K. Weiler talks about faces being "mappable to a plane", he does NOT mean "homeomorphic to a unit square". The kind of mapping he means is a graph-theoretical one that's described in Mantyla's book, "An Introduction to Solid Modeling".

Personally, I think that using the concept of being "homeomorphic to a unit square" is more intuitive, especially to the members of the STEP community. Historically, if I've got it right, the Euler formula was first proved for polygonal faces. Certainly, if you look at references in texts, they typically talk about faces as being planar polygonal ones. It then follows "naturally" for faces which are homeomorphic to them. A face consisting of a sphere or torus isn't and so the Euler formula doesn't apply.

When P. Wilson says "The correct topological elements minimally required for a BREP torus are as follows", he's making an assumption that faces are homeomorphic to a unit disk (it gets boring saying "unit square" all the time;-). According to my version of AP203, he's right:
    IP5: A face_surface has surface genus 0.

(I assume this means "homeomorphic to a unit square", not having been able to find a definition of the term "surface genus" in AP 203, but did find an assertion that "The portion of the surface used by the face shall be embeddable in the plane as an open disk, possibly with holes.")

K. Weiler makes an important point that needs reiterating: it's not a good idea to limit faces to be "basically" planar in this fashion. The topology is then an artifact of the representation scheme rather than of the user's intents. (This is akin to having trimmed surfaces not work with surfaces containing seams or poles. These aren't things the user wants to know about.)

While P. Wison is correct when he says "We tried very hard to produce a general topological model", it strikes me that we failed here. An information model for BREP objects shouldn't have restrictions like IP5.

BTW, we probably will want to examine other changes when we start to worry about nonmanifold capabilities.

Thanks, George, for raising the issue. Whatever the outcome, this does show the value of trial implementations in standardization work.

Status: Closed. Resolved
A vertex loop was added to be the minimum topology for closed bodies (sphere, torus).

Issue: 006 Conformance Classes

[This issue has been subdivided by the Implementors workshop into A-F. Each item will be covered in the status section separately.]

Ted Goosen 1/1996*
A) How many conformance classes can be exported in a part 21 file?

B) AP203 will need at least 2 to be complete (geometry & configuration management).

C) It was pointed out that since STEP places no constraints on how a user defines a product, there will be a definite need to allow multiple classes in single part 21 files.

D) AP203 conformance class prevents mixed model product definition exchange.

E) Do STEP files handle external references?

F) Since part 21 only allows one schema per part 21 file, How will the various AP's reference the same information. Will it be duplicated for each AP part 21 file or will a AP202 part 21 file reference the product definition in an AP203 part 21 file??

Discussion

Mitch Gilbert:

>How many conformance classes can be exported in a part 21 file?

As many as there are defined for an AP should be possible. Conformance classes define requirements for system capabilities, not for the contents of a physical file.

>AP203 will need at least 2 to be complete (geometry & configuration management).

Untrue. The conformance classes of AP203 are defined so that each of the geometry conformance classes contains the CM conformance class.

>It was pointed out that since STEP places no constraints on how a user defines a product, there will be a definite need to allow multiple classes in single part 21 files.

OK, I don't understand this comment.

>AP203 part 21 ?? conformance class prevents mixed model product definition exchange.

AP203's conformance classes say nothing about part 21.

>Could STEP handle external references?
If an AP defines requirements for external references, then they are handled in the AIM.

>Since part 21 only allows one schema per part 21 file, How will the various AP's reference the same information. Will it be duplicated for each AP part 21 file or will a AP202 part 21 file reference the product definition in an AP203 part 21 file??

This is probably an issue against Part 21.

Ted Goosen:

FIRST ISSUE-
How many geometric conformance class will be allowed in a pt21 file ? It was pointed out that since STEP places no constraints on how a user defines a product, there will be a definite need to constrain the way multiple geometric classes are placed in single part 21 files.

The question is two fold 1) can you place more than one geometric conformance class in a part 21 file, and 2) should it be allowed or not-allowed ?

SECOND ISSUE-
Since a part 21 file allows only one schema, can a part 21 file perform external referencing ? For example: will a part 21 based on 203 conformance class 1 be able to external reference a part 21 based on ap202 ?

Linas Polikaitis:

>How many conformance classes can be exported in a part 21 file? AP203 will need at least 2 to be complete (geometry & configuration management).

I hope that you are well aware that any conformance class defined within AP203 beyond the first MUST support all data associated with AP203 conformance class 1, as well as the geometry for that specific class. Therefore, 'to be complete', AP203 only does require 1 conformance class within a Part 21 file!

Dave Sanford:

I don't understand the issue as I have not been involved with the mechanics of implementation since the IS versions have been published. I understand the need for a single file with entities spanning multiple conformance classes. What is a "mixed model product definition exchange" and how does part 21 prevent this? What has this to do with multiple conformance classes? What has external reference to do with any of it? How and in what manner does Part 21 allow only one schema? Might that schema be the union of several AIM's? The problem needs to be outlined with more precision to ensure that participants in a discussion share a common understanding.

I suspect a related issue is a case where a single data element is desired to be sent in multiple part 21 files? How is that sameness communicated since identity is supplied at run time in building a part 21 file - names are not persistent. It seems imperative to me that the information exchanged in any implementation form be equivalent to any other form. The information is not different because I share via an SDAI based access vs. a part 21 based exchange. Consider a set of parts all designed in the same coordinate system and in some STEP based data base having multiple representation all sharing access to a single representation_context. How can I send each part in a separate part 21 file, and maintain the fact that there is only one instance of the representation_context? This problem can be generalized to any entity.

Nigel Shaw:

There seems to be general confusion over the role of a conformance class so I will try and put my understanding down. The same problem arises with UoFs in 214 as these seem to be the nearest thing in 214 to conformance classes 2-6 in 203.

To start with the basics: what is conformance? This is the match between a system and the standard. A system producer may claim conformance to some aspect of the standard and someone else may test that claim.

What is a conformance class? Effectively this is a pre-defined option within the standard that a vendor can choose to implement (and thereafter claim to conform to).

Thus to start with, as has already been stated by others, it is systems that conform, not files. However a conforming system will output a syntactically and semantically correct file (which it is very convenient but not strictly correct to describe as conforming!).

Thus we can have a situation where a system conforming to 3 conformance classes of an AP puts out a file containing data defined as belonging to 1, 2 or 3 of the class definitions. In all cases the file is correct. The mixtures which make sense depend on the (EXPRESS) data model and the data itself.

So what is the value of conformance classes? For one thing, they limit the number of combinations that implementers can choose from. they also provide a procurement handle from the user perspective and

provide a basis for reasonable conformance testing.

However... it is not clear to me that we have the right set of conformance classes yet or that the criteria for choosing them are known. The 201 developers could not reach consensus on classes. 203 has at least one conformance class that probably will not ever be implemented and will therefore fade away. (Similarly, the implementation class 2 of part 21 may never reach critical mass.) Furthermore many CAD models contain a mixture of data (product geometry in 2d and 3d and construction geometry) which cuts across the conformance classes defined in 203. There is not a defined class which explicitly matches this requirement (and need not be one). PDES, Inc. have also found that CC1 of 203 is in fact very large.

214 takes a different approach and defines large conformance classes (CC1 contains most of the geometry) with an assumption that all implementors will implement all of it. As it currently contains CSG, this is unlikely to happen. Instead, systems are already looking to use the UoFs defined for geometry in a similar way to the conformance classes of 203.

In conclusion, the current set of conformance classes are a first-pass (or guess?) at what is needed. Feedback from implementation may lead to a better set. It should also lead to recommended approaches for dealing with mixed models.

Status: A-Closed,B- Closed,C- Closed, D-SEDS,E-Closed, F-Closed, SEDS- Resolution in Part 21 amendment.

A) Answered by Mitch Gilbert. "As many as there are defined for the AP"

B) Answered by Mitch Gilbert. No.

C) Part of or same as D..

D) This is an AP203 issue and a SEDS report will be filed by Larry McKee.

E) Yes, STEP can handle external references.

F) Forwarded to WG11 (maybe WG10) to resolve. The committee felt that Part 21 needs to look at the SDAI mechanism of linking AP's and consider a physical mechanism to pass this in a 21 file. Also they should look at his problem within the context of multiple AP's in one 21 file or linking multiple 21 files.WG11 response is that this will be resolved in edition 2 of Part 21.

Issue: 007 Model Tolerance

Model Tolerance (Ismail Deif 1/1996*)

Discussion:

Ish Deif:

This issue/qustion is in regards to model tolerances, and how they are represented in STEP. AP203 defines the entity GLOBAL_UNCERTAINTY_ASSIGNED_CONTEXT, which can contain a list of UNCERTAINTY_MEASURE_WITH_UNIT entity references.

1- Can the list contain tolerances for length, angle, etc...

2- If different objects within the same overall model can have different tolerances, for example, say that in an assembly, different parts have different sets of tolerances, then must a new REPRESENTATION-CONTEXT be defined each time the tolerance set changes? Is this allowed, or must the whole model (i.e., the whole Part 21 file) have one set of tolerances?

Felix J. Metzger:

I propose to find a good term instead of "tolerances for length, angle, etc..." before starting the discussion, otherwise it is confusing as it has been shown.

Nigel Shaw:

I will try to answer your questions.

>1- Can the list contain tolerances for length, angle, etc...

Yes. That is a major reason why there is a SET of uncertainty_measure_with_units.

>2- If different objects within the same overall model can have different tolerances…

This needs to be looked at carefully. You have used a mixed terminology which needs care. A model as used in CAD-speak may be several things in STEP:

1) It could be a single representation (with associated context) in which case you cannot vary the uncertainty value as it is associated with the context.

2) A model may be a representation constructed from several other representations by means of mapped_item. In that case, each representation (including the final one) can have a different uncertainity. (How this is to be handled and what the consequences are is not clear.)

3) An assembly is defined as a product structure with associated representations. The representations may be: a) for components only, b) for the assembly constructed as in 2 above, c) for the assembly and the components but with duplicated geometry (or reference by defining the assembly and one or more components in the same context)., d) missing altogether! In cases a,b,c uncertainty may be specified as per 1 and 2 above.
Note: There are other possibilities beyond a-d.

Finally, I come to your last comment:

>   must the whole model (i.e., the whole Part 21 file) have one set of tolerances?

The concept of a file is completely independent of model. Starting at a CAD system, it is easy to view them as related. However STEP could be used between EDM systems or even between company processes where one file could contain zero one or many models. It is attractive from the CAD viewpoint to suggest having only one uncertainty per file (or one set of units?) but from the wider perspective such restrictions do not work.

Ish Deif:

In reply to Felix, I would say that the tolerances I spoke of were synonymous with the STEP UNCERTAINTY_MEASURE_WITH_UNIT, in other words, if two measurements of the specified unit fall so close to each other that the difference between them is less than the UNCERTAINTY (tolerance), then they can be considered the same.

Nigel's reply seems clear, but I'll try to rephrase it, to see if I understand it correctly. Its application to a design context would be that any part of a model can be constructed to a different set of tolerances (uncertainty), and that each such part would then require a different representation context, in order to indicate that it has such a different set of tolerances.

In addition, he seems to indicate that the same model can be represented in different ways, each with its own set of uncertainty measures. Nigel, could you supply an example?

Felix Metzger:

OK. Example: I am in the shop buying apples. In Europe, I have to do the measurement of mass myself attaching the label to the bag. My measurement yields 1,345 kg. The Lady at the checkout does not

believe that I put all apples into the bag before doing the measurement. She is doing it again and her result is 1,352 kg. Due to the MEASUREMENT UNCERTAINTY of 0,020 Kg -- this is the term I am proposing.

p.s.: I do not understand the text given for the entity "uncertainty_measure_with_unit" fully.

Status: Closed. Worked by accuracy team. Currently one value is supplied as a gap tolerance. It forms a connectivity region around and entity (sphere for point, cylinder for curve, etc).

Issue: 008 Cooperative Use of APs

Cooperative use of APs (Fritz Reuter 1/1996*)

Discussion:

Fritz Reuter:

I suppose that it is of general understanding that there never will be a single AP covering the needs of a company. If it were, it would be outdated at the time when starting such thing. Therefore there will be a general need of configure several application protocols to cover the company's needs.

At discussions within an interoperability group at ProSTEP, we discussed the importance of ARM's versus AIM's. There we found that the importance of the ARM has been neglected from the user's point of view. In this context, the user is not meant as an implementor of software. When looking through AP's searching for items you need to configure two or more AP's according to your needs in the company, the user naturally looks into the ARM. There he supposes to find the respective UOF's he is looking for. If he would look into the AIM, he probably would not find that he is looking for.

Due to the actual situation that the AIM is considered to be the most important part and the normative part - we consider now the ARM to be of the same importance. This will result in the following:

> that the ARM needs to be also a normative part

> changes in the AIM need to be reflected in changes of the ARM, i.e. ARM and AIM have to be consistent. The features possible in the AIM need to be made visible also in the ARM in order to be consistent.

There may be more consequences. As I said, this is not a view of an implementor which may be different. I would appreciate your comments.

Neal B. Appel:

I am not sure what you mean by "neglected from the user's point of view". It is true the in the AP context, the user of the AP information is not viewed as an implementor of software, but rather a user of the information the AP supports.

It may be that the AIM is "considered" by some to be the most important part and the normative part of an AP, but in fact the ARM and the mapping table are both normative and very important. The ARM is

really in two parts:

- an informative part, which is the data model found in the appendix
- the normative part, which is the definitions of the application objects and attributes, as well as the cardinalities of the application object associations. This portion is found in sections 4.2 and 4.3 of an AP.

These application objects, attributes, and associations are the industry requirements that the information user knows and understands.

The mapping table, also normative, tells how information in the form of the application objects and attributes is put into the AIM. It also contains rules (called mapping rules) which constrain this population. Any STEP conforming application must conform to the mapping table reference paths and rules as well as AIM structures and rules.

The industrial user who wants to know how the requirements are reflected in the AIM needs to look no further than the mapping table.

The mapping table is the relationship between the ARM and the AIM. The only time there are changes in the AIM that are not reflected in the mapping table is when the mapping table is unable to reflect these user requirements. (For example, if there should be exactly two measurements, the mapping table notation cannot say this, while the AIM EXPRESS rules can easily do so.)

The development of an AIM is in the direction of ARM to AIM, and not AIM to ARM. Changes in the AIM should only occur because of ARM requirements. If there are inconsistencies, then the work is flawed and needs to be fixed before moving to DIS or IS.

Perhaps I do not fully understand the problem. Is there a concern in the way 4.2, 4.3, the mapping table, and the AIM relate together or is there something fundamental that is missing? Is this a concern with the architecture, or is there something wrong with the STEP development process?

Gerry Graves:

Mr. Reuter's comment about the importance of the ARM, from a user's perspective, is correct. In our current methodology, it is the closest we come to "speaking the user's language." Users cannot be expected to traverse a complex mapping table, nor to recognize their information requirement as expressed in an AIM.

The problem alluded to in his message is two-fold.

First, if a future user of STEP wishes to "configure" an information exchange or sharing application, he

should be able to select a STEP construct (e.g., an AP, an ARM UoF, or an AIC) to satisfy his requirements. However, if multiple constructs are required that are NOT defined in a SINGLE STEP part, the user will likely face interoperability problems. One cause of this is the lack of consistency among AP mapping tables and the resulting AIMs, where similar ARM concepts exist.

Second, if a user desires to define a STEP AP for a new requirement, he is not required to review (or even guided to) existing APs that might support his requirements. Future AP development should begin with a clearly-defined user scenario that can be compared with existing APs. Only in this fashion will we meet the future needs of users in a timely, cost-effective fashion that promotes interoperability.

PDES, Inc., along with ProSTEP and others, is addressing the interoperability problem. The recent work of WG10 is encouraging. But interoperability is a near-term problem for our community, and the solution lies in most every aspect of STEP, including AP development, Integrated Resources, Testing, Implementation, and use.

Matthew West:

Thank you for your comments on Cooperative use of AP's. As Gerry said:

>Interoperability is a near-term problem for our community, and the solution lies in most every aspect of STEP,

Please be sure that this is recognized in WG10. There are at least two initiatives under way in WG10 currently.

> - An experiment to investigate the practicality of Cooperative use of APs at the present time.

> - A paper on Core Data Models, what they are, and the role they could play to ensure APs are developed consistently and that data from them can be integrated.

I am aware of other work planned by WG10 members that may address these issues in addition to these.

You are encouraged to contribute to and comment upon the work of WG10.

Julian Fowler:

>I suppose that it is of general understanding that there never will be a single AP covering the needs of a company.

In general, yes. However, there may be cases where a single AP does satisfy a company's total product

data communication requirements. Think, for example, of a small manufacturing company making specialist products for a small number of customers. This company's needs might be covered completely by AP203.

>Therefore there will be a general need of configure several application protocols to cover the company's needs.

Again, yes. It may be better to identify that the company has several different needs, each of which is satisfied by a different AP. There will, of course, be overlaps and inter-relationships amongst these requirements, and therefore amongst the APs.

Note also that the idea of "configurations" of APs has several aspects, including:

"standard time" configuration, i.e., creating new APs as standards using existing APs (or components of existing APs)

"contract time" configuration, i.e., creating a contract between exchange partners that identifies a combination of APs (or components of APs)

"compile time" configuration, i.e., the creation of an information system implementation) that supports communication using combination of APs (or components of APs)

"run-time" configuration, i.e., the selection by a user at the time of exchange of the combination of APs (or components of APs) to be used as the basis for communication.

STEP currently only provides guidance in the first of these areas (through the AP integration process). None of the others have as yet been raised as formal requirements within the ISO activity.

>Due to recent discussions within an interoperability group at ProSTEP we discussed the importance of ARM's versus AIM's.

What is meant here by "ARM". A STEP AP contains as documentation elements:

information requirements (clause 4), a normative statement of the application objects and assertions that pertain to the scope of the AP

a graphical presentation of these requirements, as an informative annex. The latter is correctly referred to as the "ARM".

"There we found that the importance of the ARM has been neglected from the user's point of view. In this context the user is not meant as an implementor of software."

I find this a very interesting statement. The intent of the ARM within the STEP architecture is to be a *user's* statement of requirements, defined in the terminology of the application domain. The ARM is not intended to be a data *structure* model. The whole purpose of the AP is to enable communication between users who share the concepts and knowledge described by the ARM.

"It's simply the reason, when looking through AP's searching for items you need to configure two or more AP's according to your needs in the company, the user naturally looks into the ARM. There he supposes to find the entities respective U0F's he is looking for. "

Yes, but with care ... it is important to remember that the ARM is described in the terminology of a specific application domain. It is always possible that two separate but related domain may use the same term for different concepts. Bill Burkett illustrates this using the term "light": for an electrical engineer, this means an electro-luminescent device for the conversion of electrical energy to electro-magnetic energy in the visible spectrum (i.e., a light bulb). To a brewer, however, "light" is a type of beer! More seriously, reviewers and users of AP203 have found it enlightening to discover that different companies in the same industry sector have significantly different understandings of terms such as "part", "assembly", etc.

"If he would look into the AIM, he probably would not find that he is looking for."

Agreed. The AIM is not aimed (no pun intended) at the user. The AIM is the standard data structure specification that enables exchange of data between computer applications.

"Due to the actual situation that the AIM is considered to be the most important part and the normative part - we consider now the ARM to be of the same importance."

Where in STEP is it stated that the AIM is the "most important part"? Yes, the AIM is the normative requirement for implementations; however, the ARM (and the scope statement, and the AAM) are equally (if not more) important in providing a complete statement of the information (the "domain ontology") that is to be communicated.

"This will result in the following: > that the ARM needs to be also a normative part."

Clause 4 already is normative. This is the key definition of the requirements that the AP satisfies. Making the graphical (or lexical) presentation of the ARM normative would not add very much. A much greater contribution to the AP development process, and the quality and usability of results, would be to:

> a) Select a single language for the statement of the information requirements (ARMs currently may be presented using IDEF1X, NIAM, or EXPRESS-G)

b) Identify and select a single methodology for the development of ARMs, i.e., for the discovery and documentation of the information requirements within an application domain

c) Develop (and possibly standardise) an implementation architecture for STEP, that shows the relationships between the various elements of an AP from the viewpoint of an implementor (or a user of an implementation). Part 10 of ISO 13584 describes such an architecture for Parts Libraries -- how come we don't have a similar document for STEP?

"c) - changes in the AIM need to be reflected in changes of the ARM, i.e. "

The ARM/clause 4 and the AIM *have* to be consistent. The interpretation methodology and the procedures of qualification are designed such that this should be so. An inconsistent ARM and AIM are indications of an incorrect and/or immature AP.

The point about visibility I take to refer to the differences in the "level of detail" found in ARM and AIM, e.g., where "geometry" in AP203 ARM maps to most of Part 42! A more detailed mapping between ARM and AIM may be beneficial for the reasons stated here; however, it should not be supposed that this will result in a "simpler" (one-to-one) mapping.

"There may be more consequences."

This issues raised in the message relate very closely to current work in WG10 that addresses both the documentation of the current STEP development methodology, and the issue of "Co-operative use of APs".

PS. Having read Neal Appel's and Gerry Graves' replies to Fritz's initial message since drafting this reply, I concur with their statements! :-)

Dave Sanford:

I have been watching this set of messages over several days, with responses from Gerry Graves, Neal Appel, and Julian Fowler. The thoughts behind this are all fine and I hope I am not being over sensitive, but my reading leads me to believe that the responses to date fail to sufficiently acknowledge how much the current process really does in terms of providing a focus on user needs and the ability to share the satisfaction of those needs across a range of applications.

The AIM is developed, imperfect as it is, precisely to provide the commonalty between AP's that we see discussed. As Julian points out, it might be possible to extend the methodology in the future to base implementation on the ARM through some formalization of the ARM methodology and the ARM to AIM mapping. In today's world, implementation is based on the AIM. We are limited by the tools at hand. If there were to be only one isolated AP, the job would be done when the ARM was developed.

The AIM is interpreted for the purpose of determining the commonalties in various AP's and expressing those commonalties in a unified manner through use of integrated resources. Indeed, where common uses are found, they are documented as AIC's. The relationship back to the ARM is documented in the normative Mapping Table as Neal also points out.

It would be nice if ARM's could be implemented as views against the results of integration and interpretation, but machine navigable mapping mechanisms could not be found. Indeed, the ARMS, prepared as they are by experts in domains other than data modeling, are often not well formed data models to begin with. If this were overcome, they would still, of necessity, be developed in the lexicon of diverse applications, with little ability to recommend commonalty other than through some fact finding process such as the AIM interpretation process. This is addressed also by Julian in his response.

In short, the developers of the current STEP process, with the tools that were and are available, developed the current process with exactly the goals in mind which Mr. Rueter is requesting. It is easy to conceptualized in generalities that the world would be improved if ARMS were somehow implementable and interoperable, but it is quite another thing to define a methodology by which that might happen. The methods of STEP as we have them, imperfect as they are, represent a technically sound effort at the rather ambitious task of semantic integration. The number of AP's which have been and are in the process of being interpreted is a testament to the success of that activity.

The normative presence of section 4 in the AP's and the ARM to AIM Mapping as well as application_context in the AIM are present precisely to ensure, both in development and implementation, that there is a well defined relationship to the user requirements and that these user requirements are fulfilled and the results sharable. STEP has a very demanding and intentional customer focus.

I am sure that the AP process can be improved on. As both Julian and Matthew West point out, improvements to the methodology are being considered in WG10. In the meantime, STEP as we have it today is useful and does satisfy in a traceable manner a well documented set of user requirements. The processes of integration and interpretation are part of the AP process precisely to find and document the commonalties between applications as a usable foundation for sharing.

Tom Kramer:

These are comments on the issue of ARMs and AIMs in APs, which has been discussed in the SC4 email group recently.

Accomplishing ARM to AIM mappings which preserve the semantics of the ARMs, in my view, is generally not possible. This is because most ARMs (I've looked in detail at about ten) contain entities which have no counterparts in the integrated resources (IRs). The rule that only entities which are defined in the integrated resources may be used in AIMs is a very bad one, in my opinion. I have been

amazed for years that the STEP effort has been able to live with it.

One or both of two responses to AP developers who have entities not in the IRs are usually given:

1. Put the new entity into the appropriate IR.

This is a poor response because:

      a. there may be no appropriate IR.

      b. there may be an appropriate IR, but the owner will not acknowledge that it is appropriate or may oppose the change for some other reason.

      c. if there is an appropriate IR, it will probably take a year or two to change it if everyone agrees the change is desirable.

2. Use an entity in an IR that can be construed to be a generalization of the ARM entity.

In some cases the generalized entity will be appropriate, but in many cases (in my opinion), using the generalized entity will entail a lot of stuff of the following sorts:

      a. using additional entities (which probably have "relationship" as part of their name), so that the exchange file contains ten entity instances where the ARM logically needed only one (other exchange methods would be similarly inconvenienced).

      b. using a lot of WHERE rules to limit the values of attributes of the generalized entity. This is often roughly equivalent to using some attributes to identify what should really have been a subtype of the generalized entity and other attributes to actually give the data for that subtype. It also obscures the fact that, semantically, different subtypes are recognized.

      c. writing implementation instructions so that implementors can all understand the usage the same way. If this is not done, the usage of the generalized entities will probably differ from implementor to implementor, so that the data may be easily exchanged but the meaning will be lost.

My recollection is that good examples of the above can be found in existing APs which have AIMs, but I have not done the detail work to identify them here.

Dave Sanford has pointed out that the AIM is intended to "provide the commonality between AP's". I agree that this is the intent, but what is happening (in my opinion) is that the appearance of commonality is being created without functional commonality.

To make an analogy in the C programming language, suppose a team of programmers (the STEP community) got together and wrote up a bunch of header files (the integrated resources) defining certain structures. They then declared that data to be shared between systems could be defined only as instances of those structures. An application would be permitted to define whatever structures it needs, but not to record data to be shared in terms of those structures. Instead the application must make a mapping from what is really needed to the structures in the approved set, and use this mapping to encode and decode what is actually intended. Any C programmer can see that this is absurd on the face of it; no fixed set of structures is going to fit all applications. A clever programmer can live with this by making mapping rules for what is really intended by certain usages of the approved structures. This procedure makes life easier for only one group, the folks who store and transmit data. They can set things up to handle data in the approved structures quite efficiently. Life is harder for everyone else because they have to deal with another layer of encoding and decoding. Upper management can point and say "We can share data among all our applications", but if that is true, it is true in spite of the way things are done, not because of it.

This map analogy maps back to STEP, as follows. In the above situation, one method for C applications to deal with data exchange under the above rules would be as follows. In this method:

1. Sender and receiver of data at the application semantic level agree to use the same header file (the ARM in STEP).

2. Sender and receiver of data at the data handling semantic level agree to use only data encoded in a particular way (the AIM) in terms the approved structures (the IRs).

3. a system of encoding (and decoding) data from the application semantic level to data in terms of the approved structures is devised. (In STEP, this is a set of software each application would have to produce.)

4. In the actual transfer of data files, steps A to E below occur. Transfer of data other than by files (SDAI) would be performed similarly.

A. An application level data file is created in terms of the application header file (ARM) by a sending application.

B. A sending mid-level data handler uses the mapping rules to re-encode the data from the original data file to a derivative file encoded in terms of the approved structures.

C. Lower level data handlers transport the derivative file from one place to another.

D. A receiving mid-level data handler decodes the derivative file from the approved structures

to the application structures and generates a third file which should be the same (functionally) as the one made in step A.

E. A receiving application uses the file created in step D.

Note that this gives the application the option of using the encoding and decoding or shortcutting it by simply sending data encoded according to the agreed application header file directly to the receiving application.

In STEP, the analogous situation will be enabled if ARMs are made normative and required to be stated in EXPRESS. I suggest STEP should do this. Then users will have a choice of using database systems which can only deal with IR entities and going through the encoding and decoding or using database systems (or other information sharing techniques) which can deal with data which has the ARM semantics.

If the rule which says only IR entities may be used in AIMs is not changed, I think the method just described may be a reasonable way to work around it. I think it would be enormously better to do away with the rule. The proposed work-around is likely to obscure the real commonalities among applications. It seems preferable to put AIMs in terms of common entities where the semantics are common and in terms of different entities where the semantics are different.

Gary K. Conkol:

To all the STEP community who cares about the users needs and interoperability between users:

The Testing project (including the people we have in WG6 and others) has gone over these issues in painful detail for many years. The focus is on three areas including validation, conformance, and interoperability.

Validation:

There are two categories here. The first is "internal" validation which is concerned with the correctness of the express and mapping of the ARM-AIM. This part has little to do with the users except to validate that the end result is true to the ARM. "External" validation refers to the ability of the AP to adequately address the actual user requirements in the eyes of the user. This is a hole in the current development process. The only place it is addressed is when the AP is reviewed for a planning project or acceptance as a work item. There is no formal process by which the ARM is reviewed by the user community outside of the developers themselves, who, in their own right, are experts. The result is that the acceptance of APs by the end users varies widely. Some APs contain validated user requirements that are readily accepted in the field while others spend a lot of time developing items that are not used in the field.

This gap has been reconfirmed as the ATSs (part 300 series) are reviewed for completeness. This requires an ARM / domain expert and there is currently no formal way to make this a part of the process. The Testing Project has proposed that external validation be addressed via a new form of validation committee. We have started discussion with WG3 as well.

We have found that the users want to be involved and there is a convenient mechanism which brings us to the next subject of interoperability.

Interoperability:

IT, here defined as the exchange between users, has always been hard to test. We had a breakthrough a few years ago when we relaxed the constraint of defining interoperability outside the user environment. We allowed the actual person testing interoperability to define their own criteria for success. This was combined with a reentrant procedure which forced the tester to converge on a set of acceptance criteria similar to defining which conformance classes to seek. This procedure worked well in the field and allowed the tester to define the needs of the user in terms that could be addressed by the standard developers and be repeatable.

The implementation of this approach in the STEP environment is being addressed through a guide being developed by the Interoperability and Acceptance Testing Methodologies group. This document, still very much a draft, contains a chart which cross references CAD/CAM applications found in actual software to the APs and their conformance classes. This helps answer the users question as to which AP to use for a given application. We have seen that as the number of APs and conformance classes increase, that the end user has difficulty is selecting the right AP parts to accomplish their purpose.

The other area is conformance testing, which, we believe, is addressed adequately in concept within STEP. Time will tell how the approach will work with the implementors.

Overall, we support the intent of the AAM and ARM to represent the users needs but see that we need to close the development loop and ask the users if the "as built" AP is valid. We also observe that interoperability between users has the same requirements that have been discovered through preceding standards. These two items combine to support the idea of a core application model with generic test criteria. We have continually found that the AAM and ARM are very useful in the field and generally accepted by the user community regardless of how they are actually transferring data currently.

Interestingly enough, the AEC core model of late mimics the core CAD/CAM model that serves as the base for most companies. In this light, it is predictable that the APs will develop along the same lines.

I would welcome discussion on this as we are currently trying to restructure the testing functions to be less resource constrained.

Kais Al-Timimi:

Assuming we all agree that STEP is all about PRODUCT data models, then we should think in terms of COMPANY APs rather than INDUSTRY SECTOR APs, as currently is the case now. Why? Because companies have products, industry sectors do not.

No one could dispute the fact that no single off-the-shelf system could satisfy any company's needs. As a general rule, companies buy off-the-shelf systems to provide some 80-90 or whatever percent of their needs and find ways to fill the remaining gap in one way or another.

If we think of company APs to cater for a company's, requirements, the 100%, then the off-the-shelf part can come from AICs and IRs. On this basis, it would make sense for standardisation process to focus on providing as wide a suite of AICs and IRs as possible and to make it easy for companies to build their own APs.

There are three attractions to this approach,

> 1. It is much easier for ISO to agree specifications for "small generic" applications than larger non-generic ones.
>
> 2. It is much faster for companies to develop something they want rather than wait for ISO.
>
> 3. It is much easier for companies to evolve their APs. So they can start small and build as they go up the learning curve.

These mean faster "time-to-market" and more flexible STEP data models. Very important factors if STEP is to succeed. There is a very wide interest in STEP now amongst users (the ones who would be paying for the bill of future STEP projects). They say we like what we hear about STEP, but when can we have it???

Yes there are reservations about proliferation of bad data models, but that is part of the learning process. The challenge for WG10 is come up with answers to help companies do a good job on their APs.

Keeping the momentum going is more important than any thing else now. There is a real risk that investment in STEP projects would dry up if users continue to hear that they cannot benefit from STEP until their AP becomes standard - maybe in 3 to 5 years time - assuming theirs, is a main stream application.

Mitch Gilbert:

OK, I admit, I am EXTREMELY confused by this entire discussion. I think that we need to revisit just what it is that we are doing in SC4. How in the world would a company be able to standardize an application protocol for its own business? I could see how a company might use a STANDARD AP, or even build an integrated information system based on the cooperative use of multiple standard APs given the future vision of product development and maintenance software applications written with SDAI interfaces. I could even envision a company using the STEP methodology to assist them in their development of a company specific integrated product data repository.

The MAIN objective of STEP right now, however, is the EXCHANGE of information. By EXCHANGE I don't mean transfer of files. I mean (and I think that STEP means) the ability of computers to communicate without the development of point to point solutions; that is enabling the computer software applications to speak a common language using a common vocabulary. If every company uses its own non-standard AP, then how will two companies, each with their own non-standard AP ever be able to exchange information?

"If we think of company APs to cater for a company's, requirements, the 100%, then the off-the-shelf part can come from AICs and IRs."

This will only lead to the same problems experienced by IGES data exchange exacerbated exponentially due to the difference in scope of the two efforts. The IRs provide a common vocabulary, the AICs provide some common idioms and the AIMs provide the language and the entire story for a particular application area. Without the story and the language how can we communicate?

"There are three attractions to this approach,"

These attractions may also be viewed as detractions:

"1. It is much easier for ISO to agree specifications for "small generic" applications than larger non-generic ones."

The more generic you get, the less specific communication you are able to achieve. We can exchange data, but we won't know what to do with it once we get it.

"2. It is much faster for companies to develop something they want rather than wait for ISO."

Of course it is, but I always thought that it was our job to develop an International Standard, after all, STEP does stand for the STANDARD for the Exchange of Product model Data. The only implication of this statement is that in order for companies to achieve communication, some company would have to develop and publish their own COMPANY AP, and hope it becomes an AD HOC STANDARD. I know of one such company that has pulled this off to some degree in a limited environment, but I am skeptical that this approach would work in any integrated product data environment.

"3. It is much easier for companies to evolve their APs. So they can start small and build as they go up the learning curve."

Again, the APs may evolve faster, but the dependency on AD HOC standardization is too great. Our objective is to make the job of doing business less expensive in the long run by fostering communication among heterogeneous computer software applications. International Standards facilitate this capability.

"These mean faster "time-to-market" and more flexible STEP data models. Very important factors if STEP is to succeed."

What "time-to-market"? It means that a company could possibly find a publisher to publish a document, but is that "time-to-market" in a standards environment? I would like to know what these users are hearing about STEP? STEP is not a turn-key panacea with which people can plug and play and eliminate the job of developing or purchasing data communication software. STEP is first and foremost and INTERNATIONAL STANDARD!!!!! This international standard allows us to reach consensus on requirements for communication of information, and use the principles of information engineering to achieve a consistent, semantically integrated set of manageable communication protocols. Of course the people developing STEP are not perfect, and will make mistakes. There is always room for improvement, and that is the focus of
WG10.

"Keeping the momentum going is more important than any thing else now. There is a real risk that investment in STEP projects would dry up if users continue to hear that they cannot benefit from STEP until their AP becomes standard - may be in 3 to 5 year time - assuming theirs, is a main stream application."

The job of ISO subcommittees and working groups is the development of International Standards. I don't think that any group working on an AP who desires to be able to foster communication with an industry or group of industries comes to the table without being prepared to work within the ISO rules as much as none of us like them.

If the users want to benefit at an early stage of standards development, then it is up to them to push their vendors to begin some work on AP implementation early in the AP development phase, perhaps at CD. This is risky for the implementor, however, as there is no guarantee that their work won't be a throw away at that stage of the standard's development.

The STEP architecture does provide a bit of a degree of stability, however, due to the use of the IRs and AICs in AIM development. There is an assurance that those bits will not change drastically. Of course that depends on your definition of "drastic".

In summation, I am of the opinion that while the ideas expressed in the discussion to which I am replying are interesting, they are not what the STEP effort is about. Of course if a company that feels the STEP architecture and AP methodology will help them with developing internal integrated data repositories, or communicating among their own "homegrown" applications they are free to develop their own "Company AP". This AP will probably be very useful within their company. I don't think that it is the job of ISO TC184/SC4 to work in that environment, however. The job of ISO TC184/SC4 is to develop INTERNATIONAL STANDARDS.

Matthew West:

I feel compelled to contribute to the exchange between Kais and Mitch because I think both are right.

As Mitch says (or I understood from my reading), it is very important to understand in what you are doing just what is compliant use of a standard, and what is not. It is not that there is anything necessarily wrong with non-compliant use of the standard, but there are risks and potential disadvantages.

On the other hand, Kais is saying (again from my interpretation) that the benefits of the formal standards take too long to deliver. Thus we need to look for ways of delivering short term benefits from pre-standardisation efforts, and/or internal use of STEP based tools etc to solve specific interfacing problems. (After all in most companies some 90% of the communication will be internal, and can be governed by internal standards).

If I step back from these two (in my opinion consistent) views, I am reminded why my company is prepared to pay for me to contribute to the development of STEP. It is not for the purpose of developing a standard, it is for the purpose of developing business opportunities for Shell to make money or reduce costs.

To me this means that we have to achieve the long term aim of standards, but in such a way that short term benefits can also be achieved. I would certainly not be able to obtain funding based on a statement that in 3-5 years you get a standard and then you can start implementing it.

Jim Mays:

I have seen nothing to date in DoD that would indicate that we will be working on a "company AP". That would defeat our objectives of information exchange. We want to buy technical data from a variety of international sources in ISO formats. We will also want to use that data in competitive procurements on EDI/EDIFACT networks. Use of commonly accepted and understood product models is the only way to make electronic commerce or flexible business systems (virtual enterprises) possible.

Point-to-point solutions are best done between specific partners, not in the ISO arena.

Dave Sanford, is it your position that the processes of integration and interpretation assure that when two information objects from different application domains are named and described differently, but in fact represent the same or almost the same object that they will be consistently mapped to the AIM?

Guy Pierra:

Last week a very interesting discussion was initiated by Mr Reuter's mail on APs nteroperability. The starting point was "the role of ARM Vs AIM". Two months ago a related discussion took place on the AEC exploder using the concept of "CORE model" as starting point. Before the Washington meeting, a workshop on AP interoperability took place, the focus was on "data integration". During the Washington meeting, a joint WG10/WG2 meeting and a joint WG2/AEC workshop were organized. The main concern was "interfacing ISO 10303 and ISO CD 13584".
All these deeply related discussions show that we are facing a new challenge: to design an improved SC4 methodology that provides not only (as the previous one successfully did) for data exchange but also for data integration.

Such a change clearly requires to re-consider all the foundations of the previous methodologies. I would suggest, in this mail, to re-start the discussion from its early beginning:

    - WHAT is the goal of SC4 (and of the methodology to be re-defined)
    - WHY is some methodology selected (in other words, what are our basic assumptions)
    - HOW do we proceed and/or should we proceed

1 - *WHAT* IS THE GOAL OF SC4 (AND OF THE METHODOLOGY TO BE RE- DEFINED)

I would propose, as a starting point for discussion, the following goal.

"The goal of the methodology is to enable a number of different groups of experts, each group having a specific domain expertise AND some data modeling expertise to develop efficiently a number of different ISs such that:

    - each IS is complete, consistent and efficient for the exchange of product/part data that constitutes its scope, and

    - the data exchanged through these different ISs may be integrated (i.e., gathered together is the same "tank") efficiently

This goal (if agreed) emphasizes the two dimensions of the needed methodology:

    1) A managerial dimension: to CO-ORDINATE a large number of different groups with DIFFERENT expertise to achieve both specific goals (their own AP or VEP) and a common

goal (data integration).

2) A technical dimension: to define a framework that provides for "AP interoperability" or more generally "Data Integration"

2 - *WHY* IS SOME METHODOLOGY SELECTED, IN OTHER WORDS ON WHICH ASSUMPTIONS IS THE PRESENT METHODOLOGY BASED

As far as I know, the assumptions on which are based the present methodology have never been made explicit. I think that it might useful to try to state them, in order to see whether, or not, they are still valid, and whether, or not, they achieve consensus. This is the goal I will try to achieve below.

Two kinds of (implicit) assumptions may be considered that address (respectively) the managerial and the technical issues.

2.1. MANAGERIAL ASSUMPTIONS

For my (subjective) point of view, the present methodology appears to be based on the following implicit managerial assumptions:

2.1.1 AP DEVELOPERS SKILLS

ASS_1: AP developers DO NOT HAVE the skills needed to develop "high quality" EXPRESS data models, they can only define informally their requirements.

> => In the present methodology AP developers are neither required NOR ALLOWED to develop the ARM as an EXPRESS schema. They describe it textually with some additional INFORMATIVE graphical information. Therefore, as underlined by Mr. Reuter, "the importance of the ARM has (really) been neglected".

If this kind of assumption was true five years ago, I think that we may now consider the following alternative assumption.

ASS_1_BIS: AP developers SHALL HAVE the skills needed to develop EXPRESS data model of their requirements: ARM shall be specified through a NORMATIVE EXPRESS schema.

> => Qualification and integration are no longer intended to INTERPRET the requirements but to VALIDATE and INTEGRATE (data integration perspective) the different schema.

2.1.2 A-POSTERIORI VS A-PRIORI INTEGRATION

ASS_2: "Even if they were allowed to (and advised to) reuse already defined resources to express their requirements, AP developers would define new resources for the same requirement"

> => AP developers are not allowed to express their requirements in terms of pre-existing resources. The decision whether or not two requirements are the same is done during integration.

If we observe what is done now, for instance in AEC, we might consider a completely different assumption:

ASS_2_BIS: "If AP developers are allowed to re-use existing resources (defined in IR, AIC or other APs) to express their requirements, they will try to re-use existing resource as much as possible. They will create new resources only for requirements that are not addressed by the existing resources".

> => Qualification and integration teams only check that this assumption has been ensured.

2.1.3 Co-operation between data modeling experts and application area experts

The co-operation between AP developers and qualification/integration teams seems to be based on the following assumption:

ASS_3: "It is easier for data modeling experts to understand and to identify overlap between different application requirements, than for application area experts to understand data modeling practices".

Due to both the dissemination of knowledge about data modeling over the last few years in the SC4 community, and the multiplicity of application domains covered by new APs, we may consider, now, the alternative assumption.

ASS_3_BIS: "It is easier for application area expert to master data modeling practices and to identify overlap between cognate application APs, than for data modeling experts to understand and compare requirements from quite different application areas".

Of course such an assumption may be more easily ensured by developing explicit and agreed guidelines about what are "good data modeling practices" (such that the ones developed by Matthew West).

2.2. TECHNICAL ASSUMPTIONS

2.2.1. REQUIRED SIZE OF THE EXCHANGE RESOURCES

Once again, for my (subjective) point of view, the basic assumption that seems to found in the present STEP mapping methodology might be stated as follows:

ASS_4: "Encoding a large set of different requirements (i.e. semantics) into a small set of constructs (i.e., vocabulary) improves the sharing".

This assertion is obviously wrong, or, at the minimum, not agreed as proved by the mail from Tom Kramer, who uses a nice metaphor based on C structures.

I think that one major reason why this (implicit) assumption has been used is that the reference to IRs mixes, in fact, two very different uses. (1) The *USE* of already interpreted abstract data types (e.g., geometry, topology, measure, ...). These constructs HAVE an operational semantics (i.e., a lot of associated operations). To USE the same construct enables to SHARE the same operations (reading, storing, processing). (2) The *SPECIALISATION* of abstract structures that take only an operational semantics through their specialisation (e.g., the product_definition_relationship: NO operation is related to this structure that may stand for, e.g., whole/part structure, container/contained, spatial relationship or what ever). In Object Oriented methodology (see, e.g., B. MEYER "Object oriented software construction"), these two uses are clearly distinguished. The STEP methodology does not make this distinction.

As suggested by Tom Kramer, we may consider to replace this assumption by the following.

ASS_4_BIS: "Similar semantics shall be encoded by similar entities, different semantics shall be encoded by different entities".

2.2.2. LEVEL OF INTERPRETATION OF THE EXAMPLE RESOURCES

A crucial question is the suitable level of abstraction/interpretation of the resources (IRs in the present methodology) that may be used at exchange time to express very different requirements.

The corresponding assertion (explicitly stated by Bill Danner two years ago during a joint WG2/PMAG meeting) is the following:

ASS_5: "The more abstract the resources, the better they are"

For instance, and unlike any other OO method I know, the STEP resource don't distinguish the whole/part structure (a bolt_and_nut CONSISTS_OF a bolt and a nut) and the container/contents relationship (a box CONTAINS a set of nuts). The ABSTRACT product_definition_relationship may be specialized to express both. The questions are: (1) WHAT IS THE BENEFIT TO MAP TWO DIFFERENT SEMANTIC RELATIONSHIPS ONTO A COMMON ABSTRACT ENTITY? (2) ARE WE SURE THAT, IN DIFFERENT APs THE MAPPING WOULD BE THE SAME? As underlined by the mail from Gerry Graves, the answer to the second question is NO. I think that answering to the first question is not straightforward.

38

Therefore, I would propose that we consider the following alternative assertion.

ASS_5_BIS: "The more interpreted are the resources, the more they contribute to data integration"

Using this assertion would lead, for instance, the AEC community to define (e.g., as 10303-10X) their CORE model as a shared set of resources (possibly defined by reference or specialization of the already existing IRs), and then to USE this set of resources (without any further interpretation) in their different APs.

3 - HOW SHOULD WE PROCEED, IN OTHER WORDS, WHAT SHOULD BE THE METHODOLOGY

I just outline, below, what might be a methodology based on the "bis" assumptions (this methodology shares a lot of commonalty with the process defined for ISO CD 13584). I follow the distinction, done by Felix Metzger, between INTERSECTION interoperability (the same semantics shall be expressed the same way) and UNION interoperability (I want to gather in the same "tank" -e. g., database- data coming from different "pipes" -e. g., AP conforming exchange context).

3.1. INTERSECTION INTEROPERABILITY

We might consider the following process:

1 - AP requirements are expressed by means of the EXPRESS ARM schema, that may use resources from IRs, from other EXPRESS-based IS (e. g., ISO 13584, IEC 1360-2) AND from existing APs. AP developers are required to review existing (related) SC4 standards BEFORE defining (from scratch, or by specialisation) new entities.

2 - The EXPRESS ARM schema ARE INTENDED TO BE USED for exchange purpose.

3 - During AP phases, experts of the corresponding discipline are required to define (as one part of ISO 10303-1XX) their CORE model.

4 - During qualification/integration phases, a validation process takes place, some constructs being possibly moved AP to IRs.

3.2. UNION INTEROPERABILITY(DATA INTEGRATION)

I would propose the following equation:

Data integration = intersection interoperability

+ persistent names
+ characterisation of specific features

3.2.1. USE OF THE ENCAPSULATION PRINCIPLE

Unlike in STEP, in usual software engineering practices, integration of large systems is done through the distinction of:

- the INTERFACE of each unit, that support the relationship between units

- the BODY of each unit, private, that "implements" the interface, and may not be referenced from outside.

I would suggest to analyze what might be the interface between different APs, and to put the corresponding resources into the IRs. Afterwards, the AP developers would have a lot of degrees of freedom to design their own "body": when an UoF is only used in one AP it is useless to map it onto any IR, a new entity shall be defined. If, later on, this resource is required by another AP, it might be referenced directly from the AP ARM schema (see ASS_2_BIS)

3.2.2 PERSISTENT NAMES

Different APs (or libraries) are intended to be provided as different files. Therefore persistent names - and human readable descriptions - shall be provided for those entities that are intended to be used for reference between these files (i.e., the interface)

To support this capability, a concept, called the semantic (data) dictionary has been developed, together with IEC, in ISO CD 13584-42. The corresponding resources should be considered for integration in ISO 10303 IRs, and for referencing by AP developers.

This mechanism, also called the BSU mechanism; basically consists of the following

- absolute identification (i.e. permanent names) for each part/product "supplier" (i.e., organization responsible for)

- within the context of each "supplier", absolute identification of each part/product he is responsible for

- within the context of each part/product, absolute identification of each entity instance (e.g., placement) intended to be referenced from outside. (In the context of STEP, further structurization might be considered)

3.2.3. CHARACTERIZATION OF SPECIFIC FEATURES

When data conforming to different APs, but relating to the same product, are to be integrated in the same "tank", the role of each set of data shall be characterized unambiguously (life cycle, discipline, ...) both for human beings and for computers. Resource constructs that enable such a characterization should be provided in the IRs (In ISO CD 13584, such a characterization is done through view_- logical_name, view_control_variable, and is_view_of relationship).

The reason why I outlined some different methodologies was only to point out that different alternatives *are* possible.

Nevertheless, due to the lack of consensus about what should be a new methodology, as proved by the recent e-mail exchange, I suggest that it might be easier to try to agree first on the underlying assumptions that may found a new methodology.

Sorry for the any possible errors above in trying to identify the present assumptions...but trial and error are also a way to progress!.

P. S. The resources for multi language support required by Mr Scheder, and discussed by Mr Alan Wilson *are* provided in ISO CD 1#3584-42 (language_resource_schema). This interoperability problem *should* also be addressed by the future methodology.

Dave Sanford:

I feel I must respond to Guy Pierra's mailing on this topic. I have taken several days to so, frankly, because I was so astonished that the current process could be so misunderstood and misrepresented.

Guy makes several good suggestions, but also several rather poor ones and mixes in several supporting phrases which simply are not, I feel, completely accurate, particularly the suggestion that the current process does not address integration.

Guy suggests in his second paragraph that we need "to design an improved SC4 methodology that provides not only (as the previous once successfully did) for data exchange, but also for data integration." I find this statement almost unbelievable. The current methods of STEP are certainly predicated on and have achieved numerous successes in the area of semantic integration. In fact, much of the balance of Guy's note goes onto to discuss the integration assumptions of that methodology. Having sat in on integration for nearly two years, I can recall the incompatibilities that existed between the geometry in part 42, and part 46 and the total inability to share information between shape representation and drafting presentation except through the efforts of the current integration process.

In terms of utility of the current STEP processes, I would like to point out the success of the current AP

225 team. I do not wish to drag these people into this discussion, but merely pass on my interpretation of what I was told about their experience by Mr. Wolfgang Haas. Their team had participated in the AP process on a previous occasion, and came to their new project with a culture of using the existing methodology to get a job done. With this approach, they were able to plan and fund for integration support in their project. They were able to bring a completed AP to CD status in a flow time of 15 months and feel it might have occurred a little faster if some aspects of their project had been different.

I am the first to advocate reconsidering our methods at all times, but only in the context of improving them, not wholesale replacement. STEP is far too mature a project, with far too many resources invested to consider "restarting from the beginning". For all the work that exists in the development of ISO 13584 CD's, it is minuscule next to the work to date in ISO 10303.

Secondly, I suggest that the forum for improving SC4's methodology is WG10, not the STEP community as a whole. For the further details in Guy's memo about the underlying assumptions of the current methodology and how they might be changed, I respectfully suggest that these are more properly deferred to the WG10 forum. To continue such a discussion on the SC4 exploder does a disservice to those not completely familiar with STEP. STEP is not in a methodology morass in need of starting from the beginning. We are rather more mature than that, and in need of constructive incremental improvements.

Matthew West:

"I suggest that the forum for improving SC4's methodology is WG10, not the STEP community as a whole. For the further details in Guy's memo about the underlying assumptions of the current methodology and how they might be changed, I respectfully suggest that these are more properly deferred to the WG10 forum."

I didn't agree with everything Dave wrote, but I did agree with this.

earlier he said

"Guy suggests in his second paragraph that we need "to design an improved SC4 methodology that provides not only(as the previous once successfully did) for data exchange, but also for data integration."I find this statement almost unbelievable. The current methods of STEP are certainly predicated on and have achieved numerous successes in the area of semantic integration."

There is a basic misunderstanding here I think. Guy is talking about integrating data. Dave is talking about integrating data models.

What is understood (I believe in WG10) is that the current STEP methodologies are designed to achieve data communication and not data integration. Some level of data integration is achievable with

the current methodology because of the features Dave points out, but it is by hard work rather than by design.

Specifically there is no single semantically complete data model which could be the basis for data integration between a number of APs which might have different views and constraints from their different perspectives.

"I am the first to advocate reconsidering our methods at all times, but only in the context of improving them, not wholesale replacement. STEP is far to mature a project, with far to many resources invested to consider "restarting from the beginning". For all the work that exists in the development of ISO 13584 CD's, it is minuscule next to the work to date in ISO 10303."

I am only reading here from Dave's quote of Guy's note above, but I read IMPROVEMENT not wholesale replacement as far as the methodology is concerned. Guy is merely suggesting that in looking for improvement it is necessary to go back to the beginning and look at the assumptions that the current methodology is based on. Doing this does not mean that we throw away what we already have, it means knowing why we might make any improvement.

I appreciate Guy's note was challenging, and not always well presented (would the rest of us like to try discussing these topics in French or German?). I think we have to be careful though about mis-interpreting what others say. We can waste a lot of energy punching at shadows.

Status: Closed-Forwarded to WG10

Issue: 009 External Mappings

External Mappings (Ismail Deif 1/1996*)

Discussion:

Ismail (Ish) Deif:

When adhering to Part 21 conformance class 1, and when external mapping must be used, why must the external mapping traverse the inheritance hierarchy all the way back to the earliest ancestor of the entities in question?

The following example shows what I consider needless output (note, I'm showing each simple record on a separate line for clarity):

```
#77 = (BOUNDED_CURVE()
    B_SPLINE_CURVE(2,(#67,#68,#69),.UNSPECIFIED.,.F.,.F.)
    B_SPLINE_CURVE_WITH_KNOTS((3,3),(0.0,1.0),.UNSPECIFIED.)
    CURVE() GEOMETRIC_REPRESENTATION_ITEM()
    RATIONAL_B_SPLINE_CURVE((1.0,0.70710678118654802,1.0))
    REPRESENTATION_ITEM('bspl'));
```

In the above, the entities BOUNDED_CURVE, CURVE and GEOMETRIC_REPRESENTATION_-ITEM serve no useful purpose, that I can see, other than to be there. Also, REPRESENTATION_-ITEM only contributes the name.

I would think that the following alternative would suffice:

```
#77 = (B_SPLINE_CURVE('bspl', 2,(#67,#68,#69),.UNSPECIFIED.,.F.,.F.)
    B_SPLINE_CURVE_WITH_KNOTS((3,3),(0.0,1.0),.UNSPECIFIED.)
    RATIONAL_B_SPLINE_CURVE((1.0,0.70710678118654802,1.0)));
```

Where the lowest common ancestor (lca) can be a complex entity. It would seem that as long as there is a traversable representation of the schema, the latter record can be just as interpretable as the former.

Felix J. Metzger:

The answer is, because this encoding (external mapping as described in ISO 10303-21:1994(E)) can

be defined by a simple algorithm.

If somebody was proposing another algorithm WHICH IS ABLE TO DEAL WITH ANY POSSIBLE CASE, we can discuss it. But one example does not tell a whole newly proposed algorithm.

In addition, the external mapping is also used for archiving and interoperability purposes, where the information shall be given as explicit and as simple as possible.

Ismail (Ish) Deif:

There really is no new algorithm. Rather, the proposal is to combine the internal mapping algorithm with the external mapping algorithm, where the internal mapping algorithm applies to the lowest common ancestor (lca), and everything beyond that follows the rules of external mapping.

The only new element introduced by this proposal is the identification of the lca.

In addition, the external mapping is also used for archiving and interoperability purposes, where the information shall be given as explicit and as simple as possible.

This is only true if conformance class 2 of Part 21 is adhered to. However, in conformance class 1, MOST entities will probably be internally mapped anyway, so the benefits that you are pointing out will not accrue.

Status: Closed. Unpersuasive.

Issue: 010 Property Definition

Property Definition (Ismail Deif 1/1996*)

Discussion:

Ismail (Ish) Deif:

Question: Is any CAD system AP 203 implementation attempting to use PROPERTY_DEFINITION entities to define "attributes" of a CAD model? Why or why not?

Status: Closed. Answer appears to be No. Unpersuasive.

Issue: 011 Uncertainties and Context

Uncertainties and Context (Ismail Deif 1/1996*)

Discussion:

Ismail (Ish) Deif:

Prologue:

> 1- Assumption: I understand the entity UNCERTAINTY_MEASURE_WITH_UNIT to mean something akin to "tolerance" of measurement. In other words, if two points fall within the value of the measurement uncertainty, they are considered to be the same point.

> 2- Situation: I have a model that contains two solid bodies, each with a different tolerance (uncertainty). In other words, the solid modeler allows me to specify a different uncertainty for each solid independently, and for the purposes of my model, each solid does have a different value of uncertainty (tolerance) associated with it.

> 3- Goal: In accordance with AP 203, I want to output both solids within the context of the same "product", where I understand "product" to roughly correspond with a CAD "part". This output is to be in the form of an ASCII file, as specified by Part 21.

Issue-

To specify that the two solids rely on different uncertainty measures, I must have:

> a- Two different UNCERTAINTY_MEASURE_WITH_UNIT entities, one for each tolerance value.

> b- Two different GLOBAL_UNCERTAINTY_ASSIGNED_CONTEXT entities, and related GEOMETRIC_REPRESENTATION_CONTEXT entities.

> c- Two different shape representation (ADVANCED_BREP_SHAPE_REPRESENTATION) entities, since that is the only way to relate the solids to their respective geometric contexts (and from those, to their uncertainties).

The problem arises if this model (product) is actually part of an assembly. In this case, I must define a SHAPE_REPRESENTATION_RELATIONSHIP between the SHAPE_REPRESENTATION of the

assembly (which is another product), and the SHAPE_REPRESENTATION of this model (product).

When I have two or more SHAPE_REPRESENTATION entities associated with the same product, must both be related to the SHAPE_REPRESENTATION of the container assembly product, or is it sufficient to related only one of them?
UNCERTAINTIES AND CONTEXTS (part 2)

This issue is a corollary to the first UNCERTAINTIES AND CONTEXTS issue. It arises because the Parasolids solid modeler allows for the specification of individual tolerance values (uncertainties) for individual faces, edges and vertices.

Can different uncertainties be specified for different faces, edges, vertices of a solid?

Status: Closed. Worked by Accuracy Team

Issue: 012 Model degradation

Model degradation strategy (ProSTEP Agreement 1 1/1996*)

Discussion:

The pre-processor of a CAD/CAM system maps the system model onto the STEP physical file in its highest possible mode. It does not convert the system model into a lower valued geometric model.

The post-processor of a CAD/CAM system has to be able to read also higher-valued models and to convert them into its native model.

It is accepted that a tool provided by a third party may be used as an intermediate stage if this reduces the potential degradation of the data.

Status: Closed. Accepted. Models should be output using the most intelligent structures available.

Issue: 013 Bounded Surfaces

Mapping of bounded surfaces (ProSTEP Agreement 2 1/1996*)

Discussion:

It is strongly recommended that topological bounding is used for all appropriate representations which allow for both topological bounding and geometric bounding. In particular, this is the case for bounded surface models which should be given as shell_based_surface_model and not using curve_bounded_-surface.

Status: Closed, Accepted

Issue: 014 Mapping Documentation

Common view of mapping documentation (ProSTEP Agreement 3 1/1996*)

Discussion:

The following recommendations are given to get a common view among the system vendors to allow a successful data exchange

If any native model is discussed at the Round Table, EXPRESS should be used as the common language everybody knows.

Naming of system data structures in EXPRESS: There should be a common use of terminology. The name should consist of a prefix of three or four characters typifying the system and an entity names which is a STEP entity name when the system data structure is similar to that STEP entity or another name when it is different.

Status: Closed. Unpersuasive.

Issue: 015 Processor Documentation

Processor documentation (ProSTEP Agreement 4 1/1996*)

Discussion:

For the documentation of the processor capabilities, a pixit like definition should be used including the valid combinations of complex entities.

Status: Closed, Accepted

Issue: 016 Polyline

Handling of polyline (ProSTEP Agreement 5 1/1996*)

Discussion:

STEP polylines when closed shall refer to the identical point as the last and the first point.

STEP polylines are open when the first and last point are not identical

This decision shall apply to similar more general cases where the information is not explicitly captured in the entity

Note: The concept of identity will be that used in the full International Standard ISO 10303-11, i.e., the same instance. In the physical file this implies the same entity number (#nnn).

Jim Jenkins 6/2000-

As far as a closed polyline goes, I think the standard should state polyline closure is defined by the same instance for start/end and put the onus on the preprocessor to get it right and save all of the postprocessors the onus of determining closure by accuracy/uncertainty value. I haven't seen polylines in too many vendors' STEP files so I think less processors will be impacted if we put the onus on the preprocessor.

Status: Closed

Current input suggests that closure should be designated by the start and end being the same point instance as suggested by ProSTEP. The onus should be on the sender to specify this as only they know the intent. May also be recommended to become an Implementor Agreement.

Issue: 017 Circular Arc

Handling of circular arc (ProSTEP Agreement 6 1/1996*)

Discussion:

If a system does not support circle (or ellipse) in its private schema, but rather represent the circle by a 360 degree circular arc, then such full 360 degree arc shall be mapped onto STEP circles (or ellipses)

System A   *STEP*
Circle   circle
circular arc 360 degrees   trimmed curve based on circle

System B   *STEP*
circular arc 360 degees   trimmed curve based on circle

System C   *STEP*
NURBS curve   (1) circle
(marked as circle)   (2) B-spline with curve_form = circle
                                              (3) trimmed curve based on circle

System D   *STEP*
NURBS which when geometrically   B-spline
analysed turns out to be a circle

On receipt of STEP data; If a system cannot handle 360 degrees arcs but can handle circle, such arcs should be mapped onto circles.

Status: Closed. Accepted.

Issue: 018 Surface Intersections

Surface intersections (ProSTEP Agreement 8 1/1996*)

Discussion:

The situation is equivalent similar in faced based surface models and B-rep models.

Face based surface models

It has to be distinguished between truly connected faces and not connected faces.

1. Connected face set

In the proper representation of a face-based surface model, the two joining faces share the same edge_curve which points to a surface-curve pointing to one 3D curve and two pcurves.

2. Not connected face set

If faces are not connected each of them has its own edge_curve pointing to a surface curve. The surface curve points to a pcurve and a 3D_curve.

 3. BREP models

The oriented_edges of two faces which are connected in a BREP model share the same edge_curve. The edge_curve shall always be a surface curve providing a 3D_curve and pcurves if the related surfaces is of parametric type. The surface curve may point to surfaces instead of pcurves if the surface are of analytic type.

Status: Closed. Accepted

Issue: 019 Scope

Use of SCOPE in physical files (ProSTEP Agreement 9 1/1996*)

Discussion:

It is left to the system vendors if the pre-processors write flat files without any scope or with scope. All post-processors are able to read any scope.

Changing to: Do NOT output SCOPE.

Status: Closed, SEDS
See issue 1 & 2. WG11 response is that this will be resolved in edition 2 of Part21.

Issue: 020 Layers and Groups

Use of Layers and Groups (ProSTEP Agreement 10 1/1996*)

Discussion:

Groups and layers are currently misused to hide semantics (e.g. groups for assemblies). The goal is to provide the functionality to map this information to other mechanisms provided by STEP. This can be done only by user interaction during the processor run.

Pre-processor: For the pre-processor the following approach will be implemented by the ProSTEP partners:

Groups: The mapping of groups needs directive whether they should be converted to STEP-groups, STEP layers or STEP assemblies (not now). This may be done either for all groups or individually.

Layers: the mapping of layers needs directive whether they should be converted to STEP-groups or STEP layers. This may be done either for all groups or individually.

Pre-processor limitations and arrangements:

If layers or groups have a flat structure in the system, they may be mapped to STEP-layers or STEP-groups.

If layers or groups have a structure (e.g. tree structure) in the system, they will be mapped only to STEP-groups.

If the names of layers or groups are integers in the system, the names should be mapped to strings representing the integers.

Post-processor: The following handling was agreed for post-processors:

STEP-groups: The mapping of STEP-groups to the system needs directive whether they should be converted to groups or layers. This may be done for all STEP-groups in the physical file (not individually).

STEP-layers: The mapping of STEP-layers to the system needs directive whether they should be converted to groups or layers. This may be done for all STEP-layers in the physical file (not individually).

57

Post Processor limitations and arrangements:

STEP-groups may have a tree or flat or any other structure in the physical file, they may have names or not. This may lead to problems for the interpretation in the post processor. The agreements could be found:

If the structure in the receiving system is flat, no common solution for adapting another structure could be found. The post-processor shall provide a private solution which is documented in the user manual.

If the systems allow only tree-structures, all entities which are in more than one group should be doubled or dropped from the other groups. The user should be informed through the protocol.

In a STEP-group member is not allowed to be a member of a native group because of its type, the entity should be dropped from the group. The user should be informed.

If the mapping conflicts because of other group relationships, a private solution should be provided and documented in the user manual.

If the name of layers or groups is limited in the receiving systems, new names should be generated by the system. The user should be informed about the mapping of these names in the protocol.

 Michael Endres:

The issue 020 on groups and layers of the implementors issue log is derived from an agreement of the Agreement Log of the ProSTEP Round Table. This agreement dates from 1994. One of the main problems in understanding the approaches of STEP to groups and layers are the different terminologies on groups and layers used by the Integrated Resources, AP214 ARM and AIM and within the CAD area (as discussed at the ProSTEP Round Table):

Terminology of STEP Integrated Resources/AICs

layer (presentation_layer_assignment):
- visibility control for styled_items
- no nesting

group:
- collection of elements of different types
- nesting

  Common CAD terminology

----

layer/level:
- visibility control for elements of different types
- no nesting

 group:
- collection of elements of different types
- nesting

In the first CD of AP214, the layer in the ARM was defined according to the common CAD terminology. In the AIM, only the STEP functionality was provided. Therefore the ARM requirements were not mapped consistently to the AIM for the layers. The distinction of the concepts between groups and layers in STEP was therefore not consistently defined in the ARM.

Due to the discussions and implementations over the last few months, the agreement 10 has became obsolete and was rejected in the forum as is (i.e. issue 020 of the issue log of the implementors forum). A new implementation practice and agreement will evolve from further investigations and implementations.

Status: Closed. Withdrawn.
Issue has been withdrawn for re-formulation and re-submittal.

Issue: 021 Implementors Agreement

International Agreement (ProSTEP Agreement 11 1/1996*)

Discussion:

The ProSTEP Round Table recognizes that:

There is a need to have agreement on an international basis for issues raised during the implementation phase of the STEP standard.

Such agreements must be common to the different groups such as the ProSTEP Round Table, that have or are being established internationally.

The Round Table requests its chairman to work with the relevant bodies to ensure a harmonized and effective international process for resolving the issues. The Round Table requests ProSTEP association to make appropriate resource available for this work.

Status: Closed. Accepted
The Implementors agree with this position and ISO is working on the appropriate mechanism.

Issue: 022 Units

Use of Units (ProSTEP Agreement 12 1/1996*)

Discussion:

ProSTEP partner system vendors will use one global assigned unit entity (for length and angle) for one physical file.

This agreement is restricted to shape_representation. Consider a file created by an Engineering Data Management system which includes data from several systems which may use different units.

Status: Closed. Unpersuasive.

Issue: 023 Sphere Topology

Minimum topology for a complete sphere (ProSTEP Agreement 13 1/1996*)

Discussion:

To represent a complete sphere in a STEP B-rep, some topology is required.

The recommended minimum topology is a single vertex loop located at the North pole of the sphere as determined by the parametrisation of the underlying spherical surface. Those systems that add a seam to the sphere may continue to do so.

Status: Closed. Accepted
See Issue #5. Accepted.

Issue: 024 Part 21

Part 21 implementation classes (ProSTEP Agreement 14 1/1996*)

Discussion:

The IS version of ISO 10303-21 (the STEP physical file) introduces a choice of two implementation classes. The difference between the classes is in how supertype/subtypes are mapped to the file.

Implementation class 1 uses the same form of mapping as was given in the DIS.

Implementation class 2 requires the use of the external form of mapping for all entities involving supertypes/subtypes.

The Round Table members agreed to only use implementation class 1.

Status: Closed. Accepted.

Issue: 025 Angular Units

Use of angular units (ProSTEP Agreement 15 1/1996*)

Discussion:

Many angles can be expressed exactly in degrees with a limited number of decimal digits. The same angle expressed in radians would usually require a representation with an infinite number of digits and is therefore truncated to the maximum number of digits appropriate for the floating point representation used. But as soon as the angle is used in computation, it must usually be converted to radians. All trigonometric functions in FORTRAN, C, C++ use radians. Many other formulas also need radians (e.g. computation of arc length). There is no advantage in using degrees. The conversion to and from radians adds further inaccuracies.

Recommendation: Use the system's internal units (degrees or radians) on output and always use the global_unit_assigned_context. This approach avoids unnecessary conversions

Status: Closed. Accepted

Issue: 026 Part 21 and Schemas

Schema identification in the file header (ProSTEP Agreement 16 1/1996*)

Discussion:

In the full International Standard version of ISO 10303-21, a facility was introduced to enable an object identifier to be given with the schema name.

It was agreed to use this facility to allow unique identification of different schemas and also to make the distinction between DIS and IS versions of the file format.

An object identifier will be provided for all schemas agreed for use by the Round Table. The identifiers provided by ISO 10303 will be used where possible.

Status: Closed, SEDS
WG11 response is that this will be resolved in edition 2 of Part 21.

Issue: 027 Pcurve in Class 2

PCURVE in Class 2 (Jeff Shultz 4/1996*)

Disucssion:

Jeff Shultz of UG:

I have been implementing Class 2 in our AP203 translator and have come
across a problem. BOUNDARY_CURVEs (say from a CURVE_BOUNDED_SURFACE)
are subtypes of COMPOSITE_CURVE_ON_SURFACE.
COMPOSITE_CURVE_ON_SURFACEs require the parent curve of each curve segment
to be a PCURVE, a SURFACE_CURVE or a COMPOSITE_CURVE_ON_SURFACE.
But, the supertype of COMPOSITE_CURVE_ON_SURFACE is also COMPOSITE_CURVE
which requires all parent curves of the segments to be BOUNDED_CURVEs. I would think
then that I should use BOUNDED_PCURVEs or BOUNDED_SURFACE_CURVEs for my
boundary curves. Unfortunately, neither BOUNDED_PCURVE nor
BOUNDED_SURFACE_CURVE is included in the AP203 schema. The syntax checker
catches parent curves that are not bounded but it seems to ignore the rule about
SURFACE_CURVEs needing to be PCURVEs or SURFACE_CURVEs.

Have I missed something major here or is there a problem here? If this is
a real problem, how are these being handled currently?

Jeff Shultz More on Class 2 & boundary curves:

Last week I sent a message describing a problem with boundary curves and
AP203 not including BOUNDED_PCURVEs or BOUNDED_SURFACE_CURVEs. I have
since spoken with Shantanu and he has found the solution. He agrees that
it would be much cleaner using BOUNDED_PCURVE or BOUNDED_SURFACE_CURVE
but the rules can be satisfied by instantiating complex instances that are ANDs
of the following :

1. BOUNDED_CURVE, CURVE, PCURVE

2. BOUNDED_CURVE, CURVE, SURFACE_CURVE

My question now is this: before I incorporate this into our preprocessor, is everyone going to be able to
read this complex entity or should I be looking for an alternate solution?

Ismail Deif :

I sent him a reply warning him off of PCURVEs because they currently fail the checker, due to a contradiction in the AP rules. There is a SEDS report out against this.

Another issue is, do we really need to have COMPOSITE_CURVE_SEGMENT point to a complex record of (BOUNDED_CURVE, CURVE, SURFACE_CURVE), just so that it points to a bounded curve, or would it be sufficient that it point to a SURFACE_CURVE, as long as the latter points to a bounded curve?

After all, the intent for parent_curve of a COMPOSITE_CURVE_SEGMENT was that it define the geometry of the segment. Since a SURFACE_CURVE defines it indirectly, we could say that the curve that ultimately defines the geometry must be a BOUNDED_-CURVE.

Larry McKee:

There has been a lot of traffic on this. The rules in 203 come from the geometry AICs so I am forced to defer to Bill Anderson.

Bill Anderson:

There are two AICs that were used for the Class 2 geometry in AP 203:Part 507 - Geometrically Bounded Surface and Part 510 - Geometrically Bounded Wireframe. It is 507 that will need to address the issue (Jochen Haenish is the current model owner) and I will forward the email to him on this topic. In the meantime, it seems that the complex instance is the proper way to proceed.

Ray Goult:

I would like to begin with a bit of background from part 42 requirements.

A curve_bounded_surface requires a composite_curve_on_surface to define the boundaries. The composite_curve needs composite_curve_segments these are required to satisfy 2 conditions for their referenced parent_curve attribute:

> (a) parent_curve must be of type bounded_curve

> (b) (from constraint function on composite_curve_on_surface) parent_curve shall be either (i)

pcurve, (ii) surface_curve, or (iii) comp_c_on_s.

Note that (a) and (b) together can only be satisfied by some sort of complex instance.

In part 42, we deliberately introduced the complex subtypes bounded_surface_curve and bounded_-pcurve to make these definitions possible.

Considering just one of the possibilities in more detail I am not at all sure that, without using bounded_-pcurve, it is possible to create a thing which is simultaneously a bounded_curve and a pcurve, note in particular that:

a trimmed_curve with pcurve as basis_curve is not itself a pcurve

a pcurve which references (via definitional_representation) a trimmed_curve is not a bounded_-curve although it will of course have the properties of a bounded_curve (The inheritance rules in EXPRESS only work in one direction a subtype inherits all the properties BUT having the right properties does not imply being of that type.)

In order to create the corresponding test cases in part 305, I made extensive use of the bounded_-surface_curve (including complex instances of bounded_surface_curve AND intersection_curve) in order to define the boundaries.

Long term I think the correct solution is for part 203 to support both bounded_pcurve and bounded_-surface_curve. I note that the new draft of AIC 507 supports these types and, as far as I can see, the curve_check function in this AIC will permit their use (in a sense this function will repaat a check which has already been done by the WRs on the special bounded_ subtypes.)

Status: Closed. AP 203 will incorporate the latest AIC information in its next release.

Issue: 028 Processor Usage

Give users visibility into intermediate forms (Emery Szmrecsanyi 4/1996)

Discussion:

Emery Szmrecsanyi:

We need to try to protect STEP from some of the bad press that IGES has gotten due to publicity of bad results. In both cases this could have been reduced if users had visibility into the intermediate forms used in the translation. This would mean that the user would be able to verify any conversions (e.g. curve to NURB) prior to the converted form being translated.

Status: Closed. Passed as input to CAx-IF forum. Model quality techniques help in preventing this.

Issue: 029 Annotation

Large Number of Shape Reps Required for Annotation (Keith Hunten 4/1996)

Discussion:
Keith Hunten- There is an inordinate number of shape_representation instances and mapped_items required to do annotation in STEP. This can be seen in APs 201 and 202. Something needs to be done to make this more efficient. This issue will also be submitted as a SEDS by Keith.

Status: Closed.
There is a focus on annotation for draughting and 3D models in the STEP testing forums. This may lead to guidance or fixes to remedy this situation. In any case, the SEDS will manage it resolution.

Issue: 030 Complex Instances

Order of Entities in Complex Part 21 Instances (Helmut Helpenstein 9/1996)

Discussion:

Helmut- Currently Part 21 requires that entities in a complex instance be listed in alphabetical/ lexographical order. This may be fine in some cases where it is not obvious what the order would be. In cases where the order is obvious, the obvious order should be used.

Status: Closed, SEDS
WG11 response is that this will be resolved in edition 2 of Part21.

Issue: 031 Implicit ANDOR

Implicit ANDOR Compiler Combinations (Helmut Helpenstein 9/1996)

Discussion:

Helmut- Currently a number of compiler generate data structures for all possible permutations and combinations of SUBTYPEs for implict ANDOR relationships. This requires huge amounts of disk space and virtual memory. The compiler should be restricted to look at the data and only generate the combinations which have populations.

Status: Closed. SEDS
WG11 response is that this will be resolved in edition 2 of Part 22.

Issue: 032 Advanced BREP

Additional entities in advanced BREPs (Michael Endres 9/1996)

Discussion:

Michael Endres:

There is a need to change the BREP AIC to allow for surface of revolution, offset curves and offset surfaces as this is the current practice in CAD systems. The sending system will be responsible for the integrity of the model.

Larry McKee:

The PDES, Inc. STEPnet activity has suggested adding surface, seam and intersection curves as well.

Status: Closed. SEDS

Issue: 033 SDAI Iteration

SDAI requires serial processing of aggregates (Helmut Helpenstein 9/1996)

Discussion:

Helmut- SDAI forces you to iterate through an aggregate one member at a time. It needs a capability to allow for non-sequential processing.

Status: Closed, SEDS
WG11 response is that this will be resolved in edition 2 of Part 22.

Issue: 034 Non-manifold Solids

Should STEP allow for non-manifold solids? (Michael Endres 9/1996)

Discussion:

Larry McKee:

What the implementors forum is looking for is industrial requirements for STEP to provide representations of non-mainfold solids. AP 214 has heard rumblings that such a capability may be necessary. In STEP AP 203, non-manifolds are representable as a vanilla shape_representation. This capability was provided as there may be a need to create aspects of BREPs which may turn out to be non-manifold or outside the capabilities of the current STEP shape AICs. The question here is do we need AICs for non-manifold solids to support production applications.

Mitch Gilbert:

I'd like to clarify one point that Larry made about AP 203. The AP 203 shape_representation was never intended to be used to represent a product shape by a non-manifold solid model. Of course if you leave it open to anything, which must be done for the representation of a shape_aspect, then you may really do anything. The catch here is that the shape_representation
will not be exchanging information to mean that a product's shape is specified using a non-manifold. You can also represent a shape_aspect by a point, but I wouldn't say that AP 203 supports the representation of product shape by points.

As far as non-manifolds are concerned, Part 508 defines a shape_reprsentation that is a topologically bounded non-manifold surface model. This is a step down constraint-wise from a brep model, but may be sufficient for industry.

Another point is that if industrial requirements are what is being sought, maybe the implementors forum isn't the best place to seek them. I would canvas the AP teams currently developing APs, and start a discussion on the WG3 list. There will be a better chance that industrial requirements are specified there.

Jim Jenkins:

Here's a response I got from Parasolid.

I guess that eventually STEP should be able to support non-manifold bodies. Parasolid can currently represent them topologically but functional support is by no means complete - indeed it is really too

sparse to be of much use as yet - when it is likely to be completed is unclear. At V6 we started work on support for general (and in particular non-manifold) bodies but since then not much has been done. I believe that ACIS supports them too. I guess eventually we will
have to finish off providing full support in Parasolid too. There is not much point our pushing for them to be in STEP until we can actually do anything useful with them ourselves. So I guess my overall feeling is that one day they will need to be in STEP but from our perspective there is probably no pressing demand immediately.

Michael Endres:

Since Kobe, there have been some discussions in the AP214 community on the issue on non manifold geometry. Therefore I would like to give you an update on this discussion:

For AP214 it was agreed that compound breps are needed on the ARM level, especially for covering the nonmanifold topology cases. This is a strong requirement in the FEA domain.
Therefore it was proposed to rework the current ARM that Compound_b_rep_model is not only a mechanism for collecting all kinds of topologic models like advanced_brep or topologically_bounded_-surfaces (as defined in WG3 N509), but to contain all topologic elements likeEdge_curves, Shells, Faces, and B_reps individually to allow nonmanifold geometry as well as combinations of topologic models of different dimensionality.

Up to now the entity non_manifold_surface_shape_representation is misusing the term non manifold topology. This shape representation has nothing to do with the usual understanding of non manifold topology. The STEP topology is not explicitly supporting non manifold topology, e.g. more than 3 faces connected by a single edge, or one vertex lying in a face without been
member of an edge_loop.

Another update:

At the ProSTEP Round Table, there was a further discussion on the issue raised by me at the implementors forum in Kobe: The issue was the questionable restriction in AICs concerning the usage of offset_curves and offset_surfaces in the context of advanced_faces as well as the restriction of surface_of_revolutions with arc as generating curves. In general, it was complained that by implication of the AICs or even in Part 42 some constraints on objects are defined that might force some loss of information with the apparent aim of reducing out certain potential
problems.

The agreement of the Round Table is that the specific restrictions should be removed. The integrity of the data should be guaranteed by the sending system as assumed in other parts of ISO 10303.

On this issue there will be an official statement from Nigel Shaw and Klaus Troendle.

Bill Anderson:

At the time the AIC for non-manifold surfaces was developed, my thought is that probably no requirements had been identified for non-manifold solids. Perhaps AIC 508 can be extended by Jochen Haenish to incorporate that requirement. For the advanced face, the restriction to not allow offsets, likely was to get away from procedural curves/surfaces, which can raise other problems in data exchange; also I believe this was based on Ray Goult's experiences in European projects.

Ray Goult:

In the shape representation committee, we have NEVER had a clear statement of requirements for non-manifold models. It was within the potential scope for part 42 version 2 but no material has been offered. The question really is how non_manifold do you want to get??

Contrary to what one might expect from the names, there is already limited support for non-manifold models within the current AICs. In an advanced_brep_shape_representation (part 514), each item has to be a manifold_solid_brep but the complete model can be non-manifold if, for example, 2 of the B-reps touch over a common face or edge.

In the manifold_surface_shape_representation, the items are usually (manifold) shells but again a very non-manifold collection of these can be constructed.

The even more general capabilities are provided by AICs 508 and 507 where, in addition to the non-manifold collection of 'items' it is now possible for the individual 'item' to be non-manifold.

In 508, each item can be a connected_face_set which would, as a simple non-manifold example be 3 faces connected (T junction) along a common edge - this is not a shell. In 507, all you have is geometric_sets and virtually anything non-manifold is possible but (of course) you can't use topology!

Status: Closed. Unpersuasive.
There have been no production examples shown where the current STEP capabilities are inadequate to represent the data. A new issue can be submitted if such a situation arises.

Issue: 035 Weight Unit

Part 41 needs a specific weight unit (Nicolay Shulga 9/1996)

Discussion:

Nicolay- Part 41 needs a weight unit. The definition of mass unit in Part 41 is in question/poorly defined.

Ed Barkmeyer:

As someone half-trained in physics (like all good engineers), I am confused about the terms used in Issue 035.

1. The definition of "mass_unit" is not at all "in question/poorly defined". The text of clause 4.14.4.1 and 4.14.4.17 make clear the relationship between mass_unit and the corresponding notions in ISO 31 and ISO 1000. I believe the text could have been better organized, and it probably should have said that a "mass_measure" must be given in either (pfx) gram or some unit with a standard conversion to gram. But there is no "ambiguity" in Part 41.

2. "Specific weight" and "weight" are two entirely different things. "Specific weight" is a ratio_measure which relates to the behaviour of density of a substance under changing pressure and temperature. I assume that the term "specific weight" was not really intended.

3. "Weight" is a common term for the behaviour of a mass under gravitational force. The more general, and unambiguously defined, term is "force". It is true that Part 41 does not define a "force" unit. And the absence of a force unit encourages the common confusion of weight with mass. Mass in invariant under changes of location; weight varies with location (because gravitational force is dependent on the distance between the centers of mass of the two bodies). Most "weights" SHOULD be characterized as mass, but others should be characterized as force.

So I believe that the issue should be rephrased as:

Issue: Part 41 does not define a force_unit corresponding to ISO 1000. The proper characterization of the notion "weight" is sometimes "force" and not "mass". Part 41 should also define a "force_measure" to represent measurements in (ISO 1000) newtons or units with a standard conversion to newtons (such as "pound").

The force unit 1 newton = 1 kg*m/sec^2 and from this the obvious WHERE clause for the "force_unit"

78

in 4.14.4.n can be constructed.

Alternative issue: An AP which includes "mass_measure" and "mass_unit" should make clear whether the common term "weight" should be interpreted as "mass", and rendered into mass units, for exchanges under that AP.
Alternatively, the AP could create a context_dependent_measure for "weight" and state its interpretation.

I take no side on the importance of this change. My purpose is to clarify the issue. I believe that the importance of the issue could be better judged if the submitter had indicated what implementation (interpretation) problem was encountered with which AP.

Mitch Gilbert:

Is the unit newton a standard international unit in ISO 1000? If so, we need to consider it for the si_unit type and make a change to Part 41. However, what you described above, Ed, looks to me like a derived unit in Part 41. The derived_unit would be comprised of a mass_unit element (kg), a length_-unit element (m) and a time_unit (sec). Any unit of this type may be defined in an AP with the WHERE clause that you describe above to specify the elements that comprise the unit.

There is one slight problem with Part 41 in this area because the derived_unit has no attribute to capture the name of the unit. This issue is addressed by a SEDS report that has been considered in the current revision to Part 41 (a name attribute has been added to the derived_unit entity).

Ed Barkmeyer:

In response to my suggestion that the issue should be whether Part 41 needs a "force unit", Mitch asks:

"Is the unit newton a standard international unit in ISO 1000?"

Yes. In fact, you can find it in the SI_UNIT_NAME list in Part 41 clause 4.14.3.23.

"If so, we > need to consider it for the si_unit type and make a change to Part 41."

As you can see, it is sort of half there already.

"However, what you described above, Ed, looks to me like a derived unit in Part 41. The derived_unit would be comprised of a mass_unit element (kg), a length_unit element (m) and a time_unit (sec). Any unit of this type may be defined in an AP with the WHERE clause that you describe above to specify the elements that comprise the unit."

This is also a possibility, and it is the same for all SI units that are not among the fundamental 7 (See Part 41, 4.14.4.17, or ISO 31). SI_UNIT_NAME covers watt (energy), Pascal (pressure), and many other units for which Part 41 has no current type, whilst Part 41 explicitly models volume and area measures, which have the simplest derived units. So there is no current "systematic" model of ISO 1000 in Part 41. I believe the philosophy was "we will model other ISO 1000 units when we discover we need them."

It is not clear to me, however, whether the proposal to add a "force" unit is what Shulga is asking for in Issue 035. That is, we have identified two possible solutions which may both be solutions to the wrong problem.

I believe that the implementation issue may have to do with mapping weight to mass (my point 3), and not with representing force. One could argue in this vein that a weight in pounds is a "converted_unit" to/from kilograms.

(I think we need a clearer statement of the implementor issue before we can go much further with this. One ambiguous line is not a clear identification of a problem.)

Status: Closed. Submit as SEDS if required.

Issue: 036 AP Identities

APs need to be interoperable.(Ed Clapp 9/1996)

Discussion:

- APs need to be interoperable (which equals identical) in the areas where they overlap (e.g., BREP) or development costs will skyrocket; the vendors can=t afford to do the same thing differently anymore.

- Felix Metzger gave a history of AP development since the IPIM; he says APs were started so that conformance testing could be done on the whole thing

- AICs are being updated now but are already being used in an IS part -- a maintenance problem

- Potential resolution: where 2 APs are doing the same thing they must do it the same way

- if we had AICs for each area that is shareable, we could legislate that constructs are implemented the same way -- but AICs are taking a long time to get through the process

- write a position from the STEP Implementors= Forum on the problem and give suggestions on how to go about solving it:

IT IS THE POSITION OF THE IMPLEMENTORS= FORUM THAT STRONG AP INTEROPERABILITY IS REQUIRED. ASTRONG AP INTEROPERABILITY@ IS DEFINED BY: (1) HARMONIZATION OF ARM REQUIREMENTS OVER THE ENTIRE SUITE OF APs AND (2) ENSURING THAT COMMON REQUIREMENTS ARE INTERPRETED INTO EXACTLY THE SAME CONSTRUCTS.

Status: Open
Statement sent to SC4 chair. SC4 is aware and shares the concern. A projects have been alerted to the need for common structure as has the interpretation project. There are efforts underway to harmonize certain groups of APs.

Issue: 037 Schema Identification

Add AP number to schema name of APs. (Dave Mattei 9/1996)

Discussion:

Dave- STEP needs to add the AP number to the schema identification for APs. People currently identify more with the number than the description of the AP.

Status: Closed, Submit as SEDS.

Issue: 038 Symetrical Parts

How are Symetrical Parts (left hand/ right hand) handled in STEP? (Emery Szmrecsanyi 1/1997)

Discussion:

Emery Szmrecsanyi:

How are Symetrical Parts (left hand/ right hand) handled in STEP?
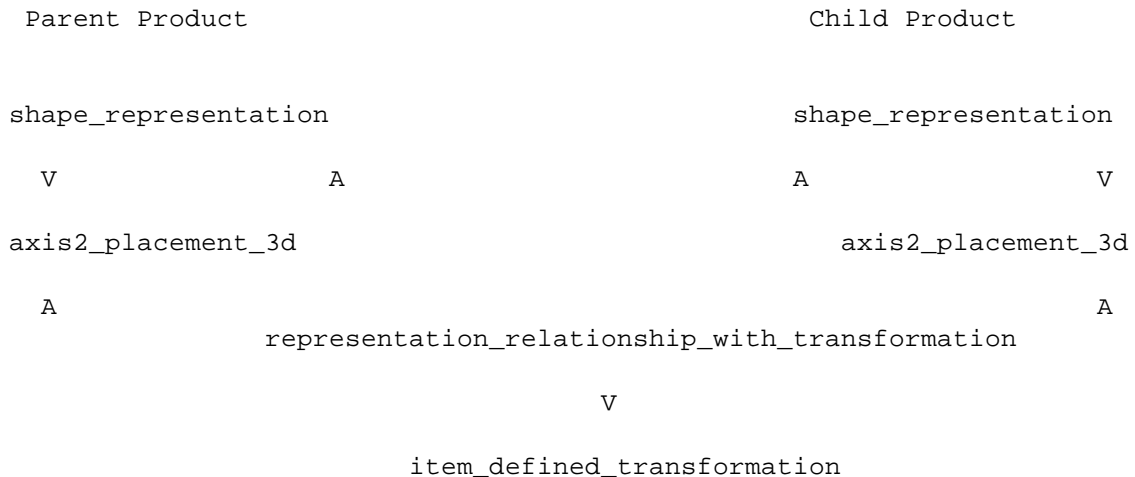
Larry McKee:

Some work has been done on this by Boeing. Larry will ask Dave Briggs for a copy.

From Dave Briggs:

Diagrammatic description of the Boeing method of representing mirroring in an AP203 Physical File.

```
1 - Normal Mapping of two right-handed Shapes using CDSR

    Parent Product                                    Child Product


  shape_representation                              shape_representation

    V                  A                      A                  V

  axis2_placement_3d                                  axis2_placement_3d

   A                                                                  A
                representation_relationship_with_transformation

                             V

                    item_defined_transformation
```

This is the Method normally used to position a Child with respect to a location within the Parent. Note that MAPPED_ITEM could also be used, but as we are dealing with Solids, and that CARTESIAN_TRANSFORM_OPERATOR_3D (CTO3D) is not allowed as one of the items within an ADVANCED_BREP_SHAPE_REPRESENTATION, we will not investigate this method.

2 - Mirroring via CDSR (NB - transformation can be at Parent ONLY)

```
 Parent Product                                    Child Product


 shape_representation     <                        shape_representation

   V                                              A                    V

 cartesian_transformation_operator_3d               axis2_placement_3d

   A                                                                   A
               representation_relationship_with_transformation

                                   V

                    item_defined_transformation
```
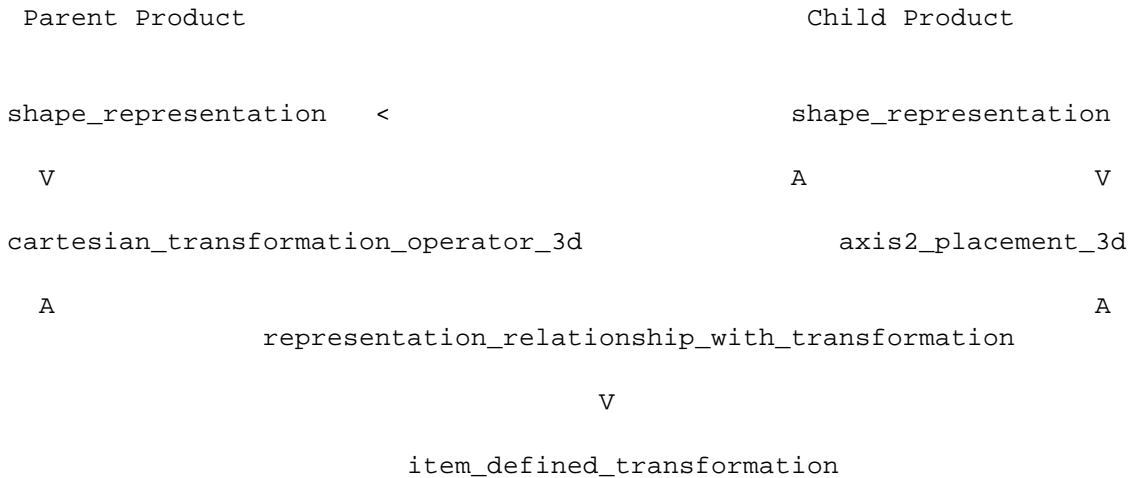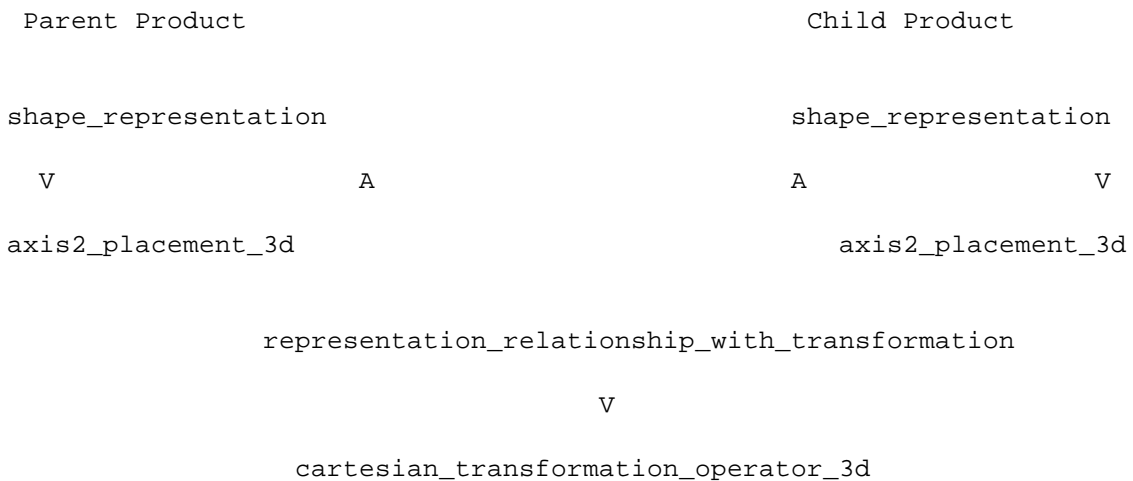
It has finally sunk in what Jeff Hunsaker from UG was trying to tell us at the last telecon. Originally, it was assumed that the CTO3D could be referenced by either the parent or the child, but since the child is always instantiated as a SUBTYPE of SHAPE_REPRESENTATION, then it cannot have the CTO3D in its set of Items, thus the CTO3D must always be part of the set associated with the Parent. Note that this will only work if the Parent node is a plain SHAPE_REPRESENTATION, i.e it has no Geometry associated with it that is not inherited via a relationship.

3- Functionally Defined Transformation

```
  Parent Product                                    Child Product


 shape_representation                               shape_representation

   V                        A                       A                   V

 axis2_placement_3d                                   axis2_placement_3d


             representation_relationship_with_transformation

                                   V

             cartesian_transformation_operator_3d
```

It is presumed that the transformation would be applied to the coordinate system of the Child prior to it being mapped to that of the Parent.

Status: Closed, Accepted

Solution proposed is the preferred solution.

Issue: 039 Best Translation Practices

The implementors forum should attempt to create best translation processes which would describe the steps users should use to ensure the best possible translation. (Emery Szmrecsanyi 3/1997)

Discussion:

Emery Szmrecsanyi:

The implementors forum should attempt to create best translation processes which would describe the steps users should use to ensure the best possible translation.

Status: Closed. Done by others.
ProSTEP and PDES, Inc. currently have these being prepared. The implementors are referred to http://www.prostep.de/BP and http://www.stepnet.org/.

Issue: 040 EXPRESS Precision

Please review the attached description of what I call the EXPRESS "precision" problem. If it is at all coherent (remember that I'm not a mathematician), maybe it can be fixed somehow. (David Price 3/1997)

Discussion:

EXPRESS allows defining a REAL number with an optional "precision_spec."
I quote from section 8.1.2 of ISO 10303-11:1994 (E), substituting quoted strings for bold-face type:

Rational and irrational numbers have infinite resolution and are exact. Scientific numbers represent quantities which are known only to a specified precision. The "precision_spec" is stated in terms of significant digits.

A real number literal is represented by a mantissa and optional exponent. The number of digits making up the mantissa when all leading zeros have been removed is the number of significant digits. The known precision of a value is the number of leading digits that are necessary to the application.

"Rules and restrictions:"

a) The "precision_spec" gives the minimum number of digits of resolution that are required. This expression shall evaluate to a positive integer value.

b) When no resolution specification is given the precision of the real number is unconstrained.

--- END OF QUOTATION ---

Now, quoting from section 11.11.18 (Validate real precision) of ISO/CD 10303-22:

This operation checks whether REAL-valued attributes are of the minimum precision. ....

Result: logical_value;
        TRUE if all REAL-valued attributes are of at least the declared precision; FALSE if at least one REAL-valued attribute violates the declared precision; and UNKNOWN if the result cannot be determined.

--- END OF QUOTATION ---

Finally, quoting from section 12.2.1.1 (Numeric comparisons) of ISO 10303-11:1994 (E):
NOTE - precision specifications are not considered when comparing two real numbers.

EXAMPLE 84 - Given:

a : REAL(3) := 1.23
b : REAL(5) := 1.2300;

the expression a = b evaluates to TRUE.

--- END OF QUOTATION ---

In my opinion, there are two different ways to interpret this information as it applies to a "C" language implementation, both of which are either useless or wrong.

The descriptions mix up the uses of the words "precision" and "significance," both of which are well-defined mathematical terms. I choose to use the word "precision" in its strict mathematical sense because mathematical "significance" cannot easily be applied to the floating-point hardware implementations of digital computers. For example, the number zero has no significant digits, yet the measurement 0.000 (which has no significant digits) may well have three digits of precision (if it represents a measurement to the nearest thousandth of a (choose your unit), i.e. the item is between 0.000 and 0.001 but nearer to 0.000).

Regardless of the actual precision of a measurement, however, when such a number is stored in a "C" language "double" variable, any notion of its precision is irretrievably lost. The number is "zero" period. It is impossible to determine its actual precision.

Consider also the number 100 (one hundred). It has only one significant digit. However, the equivalent numerical value 1.00E+2 has three significant digits. When converted to a "C" language "double" variable in IBM C/370, the corresponding value is 0x4264000000000000. This value has 14 hexadecimal digits of precision (equivalent to either 15 or 16 decimal digits of precision), but its number of significant digits is unknown (and can be anywhere between one and sixteen).

Therefore, the only viable way to interpret the EXPRESS definition is not as it applies to the numbers themselves, but can only be as it applies to the underlying hardware. That is, a declaration of a REAL(6) variable could possibly be interpreted to mean that the underlying hardware supports six or more digits of precision. Although this interpretation is a valid one, it is essentially useless because it is known in advance that the precision of a "C" language "double" variable is exactly 14 hexadecimal digits on IBM System/370 processors and is exactly 53 binary digits on IEEE 754 implementations. Surely the "Validate Real Precision" function cannot usefully mean that any precision of up to 16 is allowed on mainframes and anything more is disallowed (15 for IEEE 754), without regard for the actual numerical

values of the variables.

There is a different problem that could usefully be solved by the application of a "precision" rule to EXPRESS REAL-valued attributes. This is the "radix-change" problem which results from the conversion of numbers in the decimal numbering system to the hexadecimal system for mainframes or the binary system on workstations. An example is the number 0.1 (one tenth) in decimal. When converted to a "C" language "double" variable in IBM C/370, the result is 0x0.1999999999999999.... This is a repeating fraction that would require an infinite number of digits for exactness. Since the floating point hardware is fixed at 14 hexadecimal digits, this value is rounded off to 0x0.1999999999999A, a value that is obviously slightly larger than one tenth. Now when this value is converted to decimal for display to human beings, it can be shown as:

 0.100000000000000006 with 18 digits of precision shown, or as   0.10000000000000001 with 17 digits of precision shown, or as   0.1000000000000000   with 16 digits of precision shown.

Only the last of these appears to be the "right" value to a human being. The radix change that is introduced by the conversion from decimal to hexadecimal or binary requires "rounding" for all irrational numbers and for all rational numbers whose denominator is not one and which contains any factor that is not a power of two.

This rounding affects the results of computations that approach the limits of precision of the floating-point hardware. The result is that the expected results of computation appear to not be met.

An example (using the maximum available precision of the IBM System/370 floating-point hardware) is: $1 / 3 * 3 <> 1$.

This seemingly incorrect result occurs because $1/3 = 4055555555555555$ when stored as a "double" variable. When this value is multiplied by 3, the result is 40FFFFFFFFFFFFFF (exactly, as far as the hardware can tell). Now this result value, when converted to decimal is:
    0.999999999999999986 when expressed as 18 digits of precision   0.99999999999999999 when expressed as 17 digits of precision   1.000000000000000   when expressed as 16 digits of precision.

If this value is used in further computations, the inaccuracy can grow and the result can drift even farther away from the true result. The ability to direct EXPRESS to perform computations (including comparisons) at, say, 10 digits of precision can go a long way toward solving "problems" such as this. If these values were REAL(10) values, $1 / 3 * 3$ would actually be stored as exactly one.

Ed Barkmeyer:

David Price writes: "In Part 11 subclause 12.1 there are directions for "Real number rounding". There is

a list of steps that would seem to round -1.559 to -1.5 instead of -1.6 for REAL PRECISION(2) which is not what the implementor expected. Step A) yields -1.559 Step B) yields k - v-1.559 Step D) yields -1.5 as it says to ignore the 59 Step H) says we're done leaving -1.5 Is this an error?"

Yes, this is an error. Please submit a SEDS.

For no apparent reason, clause 12.1 tries to specify asymmetric rounding for positive and negative numbers. But the algorithm given does not conform to any of the standard rounding algorithms in either IEC 559, Floating-point arithmetic for microprocessors (commonly known as IEEE 754), or ISO/IEC 10967-1, Language-independent arithmetic. It appears that the
algorithm is trying to specify "round-to-nearest" and misidentifies the troublesome case in which the value is equidistant from both values at the target precision. As observed, -1.559 is NOT equidistant from -1.5 and -1.6.

The quick fix is to treat rule (d) as if it were identical to rule (c). While this does not produce the intent indicated by the NOTE, it is a valid round-to-nearest algorithm.

To get (d) to be a valid round-to-nearest algorithm and conform to the intent of the NOTE, one must replace (d) with:

(d) if the real value is negative, do the following:

> -- if the digit at k is in the range 6..9, add 1 to the digit at k-1, ignore the digit at k and all digits after it. Go to step e;
> -- if the digit at k is in the range 0..4, ignore the digit at k and all digits after it. Go to step h;
> -- if the digit at k is 5 and there are no non-zero digits after it, ignore the digit at k and all digits after it. Go to step h;
> -- if the digit at k is 5 and there is at least one non-zero digit after it, add 1 to the digit at k-1, ignore the digit at k and all digits after it. Go to step e.

My personal feeling is that there is no reason whatever for this complexity, and it doesn't match the statistical round-to-nearest algorithm for decimal arithmetic in IEEE 854, or the decimal rounding algorithms of either COBOL or PL/I (the only two ISO programming languages with decimal arithmetic rules). Express -- the inept programming language -- strikes again.

Status: Closed, SEDS
WG11 response is that this will be resolved in edition 2 of Part 11.

Issue: 041 Defining New Conformance Class

Can we define a new conformance class other than ones defined in APs ? (Y. Udagawa 6/1997)

Discussion:

If YES, how can we make the class in public ? Will ISO allow the addition of the new class? Or shall we use it in local (domestic) ?

This issue can divided into two concerning AP203 specific and ISO10303 in general.

(A) As for AP203, there will be a need to use multiple models (i.e. wireframe, surface and solid models) in a combined manner. Each shape model in the combined model is related through shape_representation_relationship and representation_relationship_with transformation entities as specified in section 5.2.1.2 of AP203. Can we define this kind of conformance class (subset) ? And can we put it in public as a standard of some kind ?

(B) In general, ISO 10303 allows to define a new conformance classe in addition to ones defined in APs ? How will it be documented ? Will ballots need to define a new conformance class ?

Status: Closed
Currently, there is an implementors agreement under evaluation which would allow AP 203 and 214 to have instances of shape_representation where the contents do NOT match any of the currently defined AICs. This would mean where a CAD model is a combination of wireframe and solid it would be acceptable to make this a shape_representation.

New conformance classes can be added to APs through TCs (if a subset of an existing class) or by amendment/ new editions. These should be proposed through SEDS.

Issue: 042 Use of Surface Entities

Use of surface entities. (Y. Udagawa 6/1997)

Discussion:

Cylindrical_surface, conical_surface and toroidal_surface are included in CC2 of AP203, whereas they are excluded from CC3, CC4 and CC6. There is a need to support multiple shape models (1) to manage model degradation (in other words, to keep upwards compatibility) and (2) to handle sophisticated model which is represented by a combination of wireframe, surface and solid models. Thus, I think Cylindrical_surface, conical_surface and toroidal_surface
entities should be included in CC3, CC4 and CC6.

Status: Closed. Combine with #41.
Will be handled as a part of issue 41.

Issue: 043 Use of Kanji in Part 21

Use of Kanji in Part 21. (M. Palmer 6/1997)

Discussion:

M. Palmer:

I have received the following question from Hitachi regarding the use of "two-byte Japanese code", I presume e.g., Kanji, for text descriptions in an AP 227 file.

"7. Some members of Japan Plant CALS/STEP would like to use the two-byte Japanese code as 'description' in any entities. Please give me some comments."

I remember all the discussions about this in IGES, but I am not up to date on the use of "non-English character sets" (or whatever is the correct term) that is supported (or supportable but not implemented) by STEP.

Please enlighten me. Can Kanji be used in a Part 21 file for text strings?

Y. Udagawa:

I think encoding two-byte codes is well specified in Part 21. Roughly, it says, (1) two-byte character should be represented in two 8-bytes as specified in ISO 10646, and (2) each byte should be encorded in two (or four) hexadecimal characters. And in ISO 10646, Kanji (or strictly CJK) should be represented between 4E00-9FFF.

There are no problem in theory. But when we try to implement Part 21, the following two problems occur.

(1) Very few computers (or operating systems) support ISO 10646. We have no way to support ISO 10646 in practice. Most Japanized-computers now we are using use Extended Unix Code (EUC), or Shift-JIS code. We can translate from EUC to Shift-JIS and vice versa in an easy way. But EUC and Shift-JIS are totally different from ISO 10646.

(2) Someone (such as folks in AP227) says Part 21 should allow to use Japanese character as it is.

My opinion to the problems:

(1) Part 21 restricts ISO 10646 to encode two-byte character. The restriction is too severe for most Japanese implementors. I recommend to include ISO 2022, which virtually allows to use EUC and Shift-JIS codes, in addition to ISO 10646. As you know, IGES Version 5 accepts an modified version of Shift-JIS (virtually ISO 2022) called FC2001.

(2) As for the problems to use Japanese character as it is in Part 21 file, honestly, I could not understand where the requirement comes from. End users never see the Part 21 file as texts. They look at a kind of STEP viewer to make sure a translation of given model. The only folks to see Part 21 file I can expect are implementor of STEP-related software. (This problem may come from only for representation purpose for implementors at a debugging phase, I
believe.)

Before discussing the problem, I suggest to make clear who benefits and how benefits by supporting Japanese character as it is.

Status: Closed.
Per Mr. Oku in Lillehammer 6/99, ISO 10646 can represent all character sets. There still is an issue on how these are locally used. This portion of the problem is more appropriately a Part 21 issue.

Issue: 044 Solid Model History

Need for construction history for solid models. (L. McKee 11/1999)

Discussion:

L. McKee:

The STEP standard needs to support the transfer of solid model construction history information to allow for incremental modification of the received solid. The current information structure results in a unitary solid which is difficult to use for collaboration.

Status: Open.

Issue: 045 STEP Meta Data

There is a need for the equivalent of HTML META tags for STEP files (J. U'ren 02/2000)

Discussion:

Jim U'ren

Finding things on the web can be difficult. Most search engines are like drinking from a fire hose: more than you can handle; can't get what you want; Don't know where the hose has been and if you should be drinking from it. Jim said that STEP files and data will have limited use if they are not available and findable via the web.

The web has a set of META tags in HTML which are extremely underutilized. NASA is working with others to propose and standardize through W3C and ISO a standard for Core Meta Data. This is minimum TAGs and object should carry with it to help identify and classify it.

Status: Closed.
Forward to WG11/Quality.

Issue: 046 STEP and XML

How can STEP and XML be used together? (A. Peltzman 06/2000)

Discussion:

A. Peltzman

There is a need to discuss how STEP and XML might be used together. Part 28 is being developed to identify bindings of Express and XML. These will have effects on implementation. It may be possible that XML replaces SDAI.

Status: Open.

Will be added to Big Issues list for near term monitoring.

Issue: 047 New Chair

There is a need for a new chair for the implementors forum (L. McKee 06/2000)

Discussion:

L. McKee

My assignment in IBM is changing from a focused PDES, Inc. assignment to one that is more focused on internal implementation with minimal PDES, Inc. involvement. I can no longer guarantee any participation at ISO meetings. As such, I will have to terminate my chairperson role.

Status: Closed.

The US is nominating Martin Hardwick. Martin was approved by the forum and confirmed by SC4.

Issue: 048 Design Intent/ Parametrics

When will Design Intent/ Parametrics be ready to be implemented? (T. Briggs 10/2000)

Discussion:

T. Briggs

There is more than solid model history missing from STEP solid model functionality. When will design intent information (e.g. parameteric constraints) be ready for implementation?

L. McKee

This depends on Part 108. There was some work done in projects like ENGEN. The formalization of this is through Part 108. Will request an update from Mike Pratt's team.

Status: Open.

Request an update from Mike Pratt's team. Add to Big Issues list for tracking.