Open Data Exchange with HP PE/SolidDesigner

Surface and solid data can be imported from HP PE/ME30 and exchanged with systems supporting the IGES, STEP, and ACIS formats. Imported data coexists with and can be manipulated like native data.

by Peter J. Schild, Wolfgang Klemm, Gerhard J. Walz, and Hermann J. Ruess

HP PE/SolidDesigner supports the coexistence of surface data with solid data and provides the ability to import and modify surface and solid design data from a variety of CAD systems. Backward compatibility with HP PE/ME30 preserves the investment of existing HP customers. Using improved IGES (Initial Graphics Exchange Standard) import capability, both surface and wireframe data can be imported. Surface data and solid data can also be imported and exported using the STEP (Standard for the Exchange of Product Model Data) format. Once imported, this data can coexist with HP PE/SolidDesigner solid data. It can be loaded, saved, positioned, attached to, managed as part and assembly structures, deleted, and used to create solids. Attributes such as color can be modified. If the set of surfaces is closed, HP PE/SolidDesigner will create a solid from those surfaces automatically.

HP PE/SolidDesigner 3.0 also allows solid parts and assemblies to be exported to ACIS-based systems using Version 1.5 of the ACIS® SAT file format. This feature provides a direct link to other ACIS-based applications.

From PE/ME30 to PE/SolidDesigner

HP PE/ME30 is a 3D computer-aided design (CAD) system based on the Romulus kernel. To preserve the investment of existing customers it was required that the transition from HP PE/ME30 to HP PE/SolidDesigner be as smooth as possible. Therefore, an HP PE/ME30 file import processor is a integral component of HP PE/SolidDesiger.

In HP PE/ME30, 3D objects are built from analytic surfaces like cylinders, cones, spheres, planes, and toruses. The intersections of these surfaces can be represented as explicit analytic curves such as straight lines, circles, and ellipses, or implicitly by describing the surfaces involved and providing an approximation of the intersecting arc. Parabolic and hyperbolic intersections are represented implicitly.

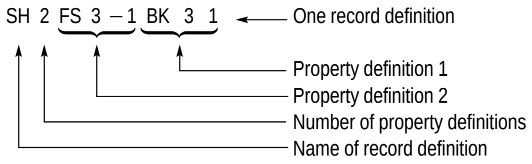
HP PE/ME30 Native File Organization

HP PE/ME30 supports the Romulus textual transmit format. The transmit file is not intended to be read by humans but the general structure can be examined. The file contains only printable characters, and real values are represented as text strings. The full format of a transmit file consists of six different sections. These will be described using the example of a single cylinder positioned at the origin of HP PE/ME30's coordinate system with base circle radius 10 and height 20.

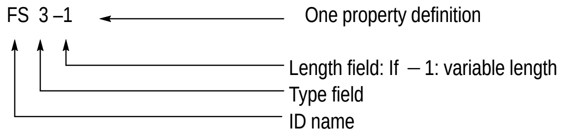
The first section, the header section, describes the environment, the machine type, the user login of the file creator, and the time and date when the model was created.

```
@* AOS
@* Machine type HP-UX
@* Transmitted by user_xyz on 27-May-94 at 13-06
```

The second section contains index and counting information related to the schema described in the third section. The schema defines the data structures used to represent the objects. It consists of a collection of record definitions. The following is an example of a record definition:



The following is an example of a property definition:



In the second section of the transmit file the number of record types, the numbers of record instances and property instances, the name of the schema, and its version and update number are supplied. The record instances and property instances contain the concrete data describing the model. The semantics and the sequence of data entities have to conform to the format specified by the corresponding record definition and property definition entities.

The information in the cylinder example file says that 11 instances of record definitions are supplied to describe the schema for the instance of the cylinder. For the actual object, 23 record instances built out of 115 property instances are used.

---

* A kernel is the heart of a modeling system. Currently, three kernels are used in various CAD systems. These are Romulus from Shape Data, Parasolid, an extension of Romulus, and the ACIS Kernel from Spatial Technology.

```
1
11
23 115
ROMDSCHMA 7 4
16
```

The third section, the schema section, contains the definition of the data structures used to represent the model. This section consists of the subset of record definitions from the HP PE/ME30 internal data structure schema that are needed to represent the model. The schema sections of files representing different models will be different. The schema section for the example cylinder is:

```
BY  19 UP 4 –1 SE 3 –1 TX 5 –36 FI 4 –6 CI 4 –12 PI 4 –8
GI 4 –1 SI 4 –3 TI 4 –3
RA  2 1 RN 2 1 ZI 4 –2 FN 1 1 CN 1 1 PN 1 1 TN 1 1
SN 1 1 ZN 1 1 NM 1 1
SH  2 FS 3 –1 BK 3 1
FA  8 UP 4 –1 AK 3 –1 RV 1 1 SF 3 1 SX 2 1 VR 3 –1
HA 2 –3 SL 3 1
VR 4 PT 3 1 BE 3 1 BV 3 1 FE 3 1
ED 2 CU 3 1 RV 1 1
CU 3 UP 4 –1 AK 3 –1 TR 3 1
TR 6 UP 4 –1 AK 3 –1 BK 3 1 EQ 2 –7 TS 3 2 TY 1 1
PT 4 UP 4 –1 AK 3 –1 CO 2 –3 GP 3 1
GP 5 UP 4 –1 AK 3 –1 BK 3 1 CO 2 –3 PX 3 1
SF 7 UP –1 SD 3 –3 AK 3 –1 BK 3 1 EQ 2 –7 SU 3 –5 TY 1 1
UA 3 OW 3 1 CL 1 1 II 1 *1
```

The fourth section contains, for each record type defined in the schema section, the number of data objects used for the transmission of the model. The sequence of numbers is identical to the sequence of record definitions used in the schema section. In the cylinder example, the object consists of one body built of one shell built of three faces. Four vertices, four coedges, two curve geometries, two edges, two points with two geometric point definitions, three surfaces, and one attribute are needed to represent the cylinder object. The file contents are:

```
1 1 3 4 4 2 2 2 2 3 1
```

The fifth section, the data section, contains the data structure instances. The contents of all records needed to represent the object are found in this section. To every record an integer record label is assigned. This number will be used in other record instances to point to the instance. In general the instances in the file appear in the order in which they are referenced by other entities. The data of an entity instance is not split. If forward references are contained in the instance definition the next instances can be found in exactly the same sequence as referenced. Because this rule applies recursively, newly referenced entities can be found first in the physical file sequence. If all references of an entity are resolved completely the next reference of the next higher level will be resolved. For the cylinder, the data section is:

```
1
1 1 25 Color 1 2 0 3 3 F0 4 F1 5 F2 2 14 E0 15 E1 2 18 P0 19 P1 0 0 0 0.000001
0.00000000001 0 3 3 2 0 0 0 0 25 1 1 1 16777215 2 0 1 3 0 0 0 22 0 1 6 0 2 22
0 0 0 0 6 0 0 0 0 0 –1 0 1 6 18 10 6 10 18 0 0 0 20 20 0 0 0 3 0 10 0 18 10 14
0 14 0 0 16 16 0 0 0 7 0 0 0 0 0 –1 10 0 0 2 4 0 0 1 23 0 1 7 0 2 23 0 0 0 0 6
0 0 20 0 0 –1 0 1 7 19 11 7 11 19 0 0 0 21 21 0 0 0 3 0 10 20 19 11 15 1 15 0
0 17 17 0 0 0 7 0 0 20 0 0 –1 10 0 0 2 5 0 0 0 24 0 2 8 9 0 2 24 0 0 0 0 7 0 0
0 0 0 1 10 0 2 8 18 12 8 12 12 14 1 9 19 13 9 13 13 15 0
```

The sixth and last section contains, for each top-level object transmitted in the file, the corresponding root entity and its name. In the cylinder example only one object is transmitted. HP PE/ME30 supports user-named objects, but in this example an HP PE/ME30 default name, B0, has been used for the cylinder.

```
1
B0
```

Analyzing the Transmit File

Because the information content of an HP PE/ME30 file cannot be understood by simply looking at the file, several internal analysis tools are used to extract the information. Statistics showing the number of different curve and surface types give a first hint of the complexity of the file. A graphical presentation of the data instances of a file can be generated.

```
|_BY____|_UA____|–>
        |_SH____|–>
        |_FA____|_SF___
                |_VR____|_PT___|_GP___
                        |_ED___|_CU___|_TR___
        |_FA____|_SF___
                |_VR____|_PT___|_GP___
                        |_ED___|_CU___|_TR___
       |_FA_____|_SF___
                |_VR____|–>
                        |_ED___|–>
                |_VR____|–>
                        |_ED___|–>
```

This reference structure can be read easily. The (cylinder) object in the file is a body (BY) which consists of a shell (SH) and three faces (FA). Shell and faces share the same hierarchy level. Each face consists of a reference to a surface (SF) and a start vertex (VR). Each start vertex is based on a geometric definition of a point (PT) and serves as the anchor vertex of an edge loop. A loop is not represented explicitly in the HP PE/ME30 exchange file. The implicit connection is done by a reference from a start vertex to the next and previous vertices in the loop. The edge (ED) entity represents the topological direction of the edge with respect to the loop. The curve (CU) entity is an intermediate instance on the way to the curve's geometry (TR).

If complete information from the data section is needed a translation tool is available that maps the data section to a format much more useful for human readers. The following extract describes how one of the faces and the corresponding surface component of the cylinder example are represented. The mapping from the data section to the readable format is also supplied.

For this component from the data section:

```
................................................
  ..........5 0 0 0 24 0 2 8 9 0 2 24 0 0 0 0 7 0 0
0 0 0 1 10 0 2 ..................................
```

the corresponding translated part is:

```
5 = FA (Face owning (anchor) vertex), the properties are :
         UP is  EMPTY                ... List of permanent universal attributes
         AK is  EMPTY                ... Backpointer from element of feature
         RV :   INTEGER = 0      .  .. Sense of face, edge geometry
         SF :   POINTER = 24     .  .. Surface of face
         SX :   REAL    = 0      .  .. Hatching pitch
    1    VR :   POINTER = 8          .. Anchor of face
    2    VR :   POINTER = 9          ... Anchor of face
         HA is  EMPTY                ... Hatch direction
         SL :   POINTER = 2          ... Shell of face

24 = SF (Surface of face), the properties are :
         UP is  EMPTY                .. List of permanent universal attributes
         SD is  EMPTY                ... Surface supporting this surface
                                          definition
         AK is  EMPTY                ... Backpointer from element of feature
         BK :   POINTER = 0          ... Backpointer from assembly or body to
                                          token
         1   EQ :   REAL   = 0       ... Geometry definition
         2   EQ :   REAL   = 0       ... Geometry definition
         3   EQ :   REAL   = 0       ... Geometry definition
         4   EQ :   REAL   = 0       ... Geometry definition
         5   EQ :   REAL   = 0       ... Geometry definition
         6   EQ :   REAL   = 1       ... Geometry definition
         7   EQ :   REAL   = 10      ... Geometry definition
             SU is   EMPTY           ... Surface supported by this suface
             TY :   INTEGER = 2 (CYLINDER)   ... Geometry type
```

Import Module

The HP PE/ME30 to HP PE/SolidDesigner import interface is linked directly to the HP PE/SolidDesigner code. In HP PE/SolidDesigner's user interface it simply adds a button to the external filing menu. If a file name is specified, the processor is activated. Internally, several C++ classes are added to HP PE/SolidDesigner to represent the schema and instance entities of the HP PE/ME30 file. For every supported HP PE/ME30 record definition entity a class derived from a generic record instance object is defined. The most important member function of each of these classes is the convert function. This function performs the mapping of the HP PE/ME30 file object to the corresponding HP PE/SolidDesigner entity.

The three main components of the HP PE/ME30 to HP PE/SolidDesigner processor are a lookup table, a schema manager, and a set of classes to represent the supported HP PE/ME30 file entities.

The lookup table is part of the interface to an HP PE/ME30 file. The main task of this table is to manage the mapping of HP PE/ME30 file entities to already created corresponding HP PE/SolidDesigner entities. A lookup table is generated for every open HP PE/ME30 file.

A schema manager is initialized if a new HP PE/ME30 file is opened. It contains the schema section information found in the newly opened file. For every open file a corresponding schema manager is available to control the interpretation of the entities of the file.

The record instance class builds the third basic data structure of the processor. Record instances are generic containers to store all of the data objects that can be expressed by valid record definitions. The constructor of the record instance class calculates the entity type from the reference number and then allocates memory and reads in the properties from the file corresponding to the property definitions of the schema. For every supported HP PE/ME30 entity a separate C++ class is derived from the record instance class, but the generic constructor is used for all subtypes. The main differentiator between the classes is the convert function.

Conversion Process

The convert function of the record instance class itself is not called by the conversion process. Rather, every derived class implements its specific conversion function (in this sense the convert function is purely virtual in C++). The individual conversion function converts itself to an HP PE/SolidDesigner entity.

Conversion and the creation of new derived instances of the record definition class constitute a recursive process. If during an active conversion an unresolved (not already converted) reference is found the corresponding HP PE/ME30 file entities can be found as the next entities in the physical file (see the description of the data section). The conversion module then creates a new derived instance of the record instance class and forces the translation of this entity to a HP PE/SolidDesigner entity that can be used to complete the conversion of the current entity. The algorithm is as follows:

```
A reference to an HP PE/ME30 file entity is found:

Already "converted"? (lookup table search)
   YES:  Use the available conversion result
   NO:   Create the new derived class of record instance
         Call the convert function
         Attach the conversion result to the lookup table
         Delete the instance to free the memory used
         Use the newly generated conversion result to continue the
           conversion.
```

Nonanalytic Intersection Curves

The conversion for intersection curves is not done on the fly, but by a postprocessor after the rest of a body is converted completely. The convert routine for an intersection track simply collects the two intersecting surfaces and all available additional information found in the file to represent the intersection. The completion of the intersection curves is done by the convert function for HP PE/ME30 bodies. After a first intermediate topology of the new HP PE/-SolidDesigner body is calculated and all analytic surfaces and analytic curves are attached to the created body, the calculation of the intersections begins.

The topology of the intersection between two surfaces in HP PE/SolidDesigner is not always the same as in HP PE/ME30 because different constraints on topology and geometry exist in the two modelers. For instance, it may be necessary to represent the single segment found in HP PE/ME30 as a sequence of different curves. In such cases the original topology has to be modified and some edges may be split. To find the appropriate intersection in HP

PE/SolidDesigner is mainly a selection process. In many cases two surfaces intersect at not only one but several distinct sections.

Consider the intersection of a cylinder with a torus in the case of perpendicular axes. Four possible intersection curves may be part of the model (see Fig. 1). In the HP PE/ME30 file additional help points are supplied to allow the correct selection. The direction of the intersection curve (the tangent to the curve) is not guaranteed to be the same in HP PE/SolidDesigner as in HP PE/ME30. Therefore the correct fit to the model is calculated and the resulting direction is reflected in the topology of the imported model.

Quality and Performance

To test the quality of the HP PE/ME30 import processor a large HP PE/ME30 test library has been compiled. It now contains more than 2300 examples of parts and assemblies. All of the test cases used during HP PE/ME30 development and support are included along with new user models consisting of recently acquired data from internal and external HP PE/ME30 users. An additional test matrix subtree was developed by creating base parts with critical features. In particular, all possible surface-to-surface intersections and various special cases have been generated.

The regression test procedure is to import HP PE/ME30 models from the test library part by part and perform the HP PE/SolidDesigner body checker operation on each. The loading time and the body checker result are collected in a reports file. A reports file can be analyzed by a shell script to supply a statistical summary of the current quality of the HP PE/ME30 interface. Because of the large amount of test data a complete test takes a long time. Therefore, an intermediate test is available. The complete test performs the basic load and check test on all currently available test models of the library directory. The intermediate test examines the reports file of the latest complete test and repeats all reported problems. It also repeats a random selection of the successful tests. At this time over 99% of the complete test conversions are classified as successful.

The performance of the import process for HP PE/ME30 files is mainly dependent on three variables: the size of the schema, the number of entities, and the number of intersections that have to be calculated:

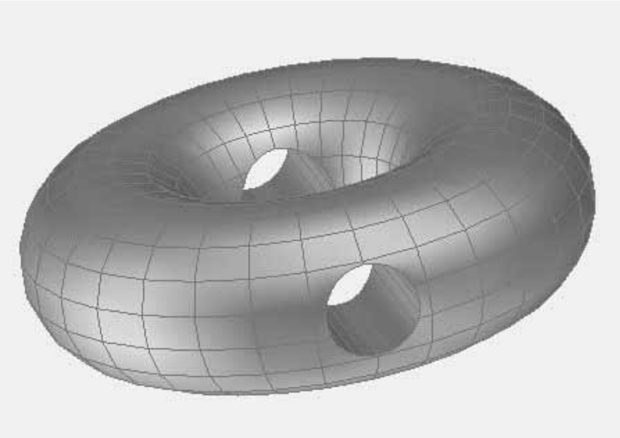


Fig. 1. Intersection of a torus and a cylinder.

$$\text{Load Time} \approx \text{Size} \times k1 + \text{Entities} \times k2 + \text{Intersections} \times k3,$$

where $k1 < k2 << k3$. The size of the schema section does not vary very much between different files and is normally relatively small compared to the size of the data section. The number of entities and the file size are strongly related. The calculation and selection of the nonanalytic intersection curves fitting the model is a relatively expensive component of the processor because a completely new representation of the data structure has to be generated.

Data Exchange Using IGES

An important task in computer-aided design is the transfer of the completed model to downstream applications and other CAD applications. These applications vary from finite element analysis and numerically controlled (NC) manufacturing to visualization and simulation. HP PE/SolidDesigner currently uses IGES 5.1 (Initial Graphics Exchange Specification) for file-based data exchange.

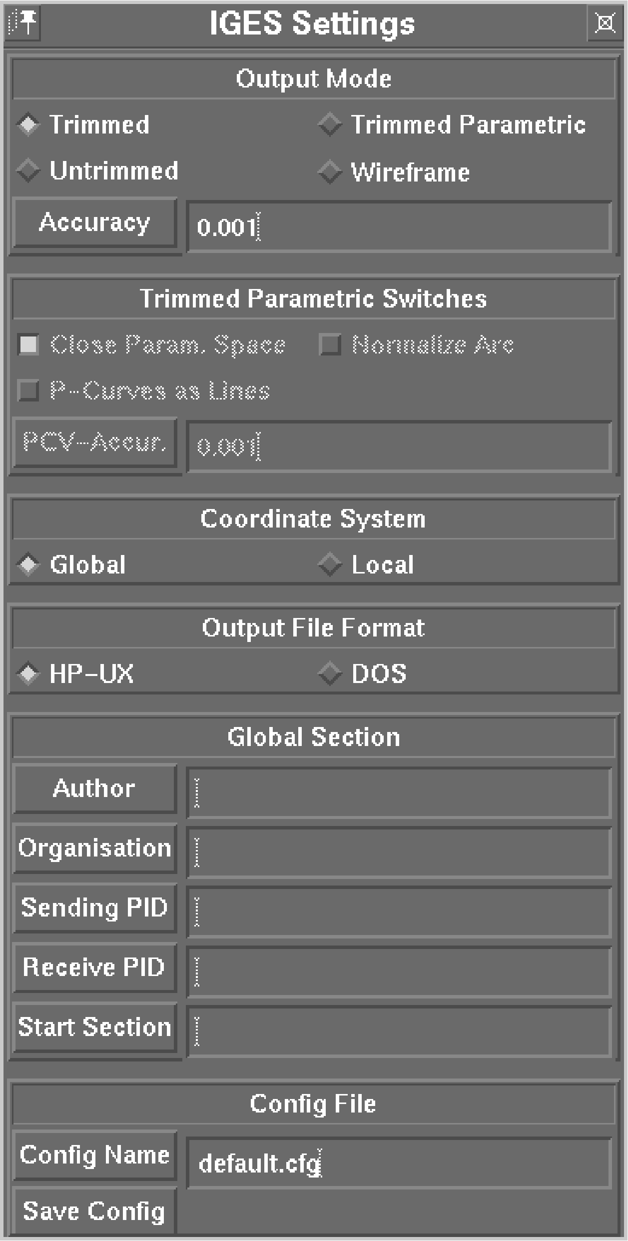
Because of the broad variety of receiving systems an IGES interface must be flexible so that the contents of the output file match the capabilities of the receiving system. It must be possible to transfer whole assemblies keeping the information on the parts tree, or only specific parts of a model, or even single curves or surfaces. This is achieved by a mixture of configuration and selection mechanisms.

An analysis of the IGES translators of many different systems showed that it is possible to classify them in four main categories:

- Wireframe Systems. These systems are only capable of importing curve geometry. This is typical for older CAD systems or 2D systems with limited 3D capabilities.
- Surface Systems Using Untrimmed Surfaces. These systems are capable of importing untrimmed surfaces and independent curve geometry. This is typical for low-end NC systems that need a lot of interaction to create tool paths and define areas.
- Surface Systems Using Parametrically Trimmed Surfaces. These systems are able to handle trimmed surfaces. Trimming is performed in the parametric domain of the surfaces. Periodic surfaces are often not handled or are incorrectly handled. Each surface is handled independently. This is typical for surface modelers and sophisticated NC systems.
- Topological Surface Systems and Solid Modelers. These systems are able to handle trimmed surfaces using 3D curves as trimming curves. They are able to handle periodic surfaces, nonplanar topology, and surface singularities. Connection between adjacent trimmed surfaces is maintained and the normal to the trimmed surface is important for inside/outside decisions. This is typical for advanced surface and solid modelers.

HP PE/SolidDesigner's IGES interface is designed to work in four output modes: wireframe, untrimmed, trimmed parametric, and trimmed. Each output mode represents one of the categories of receiving IGES translators. This has the advantage of giving as much information about the solid model as possible to high-end systems (trimmed, trimmed parametric), without burdening low-end interfaces with too much information. For some modes (trimmed parametric) more configuration parameters allow fine tuning to specific systems to maximize the transfer rate. Each mode has a specific entity mapping that describes which IGES entities are used to describe the model (see Tables I, II, and III). Users can specify additional product related data and arbitrary comments for the start and global sections of the IGES file directly via the IGES output dialog box. Specific configurations can be saved and loaded so that the configuration has to be determined only once for each receiving system. Fig. 2 shows the IGES dialog menu.

To allow maximal flexibility in what is translated, the user is allowed to select assemblies, parts, faces, and edges and arbitrary combinations. All selected items are highlighted and the user can use dynamic viewing during the selection process. If the user selects assemblies, the part tree is represented with IGES entities 308 and 408 (subfigure definition and instance). Shared parts are represented by shared geometry in the IGES file.



Output Mode: ● Trimmed ○ Trimmed Parametric ○ Untrimmed ○ Wireframe

Accuracy 0.001

☐ Close Param. Space ☐ Normalize Arc ☐ P-Curves as Lines

PCV-Accur. 0.001

Coordinate System: ● Global ○ Local

Output File Format: ● HP-UX ○ DOS

Author

Organisation

Sending PID

Receive PID

Start Section

Config Name default.cfg

Fig. 2. HP PE/SolidDesigner IGES output dialog menu.

Table I
Curve Mapping

HP PE/SolidDesigner	IGES 3D Entity
Straight	Line (110)
Circle	Circular arc (100) with transformation
B-spline	Rational B-spline curve (126)
Intersection curve	Rational B-spline (126)
Parameter curve	Rational B-spline (126) or line (110)

Trimmed Mode

The trimmed mode is the closest description of the internal B-Rep (boundary representation) data structure of HP PE/SolidDesigner. It uses the IGES bounded surface entities 143 and 141 as the top element of the model description. Each selected face of the part maps to one bounded surface (entity 143) containing several boundaries (entity 141). Trimming of the surfaces is performed by 3D model space curves. To fulfill the requirements of the IGES specification of entities 141 and 143 some minor topological and geometrical changes of the HP PE/SolidDesigner internal model have to be made. Vertex loops are removed, propedges on toruses are removed, and intersection curves are replaced by B-spline approximations.

Because the IGES bounded surface entity 143 does not have any information about topological face normals, the surfaces are oriented so that all geometrical normals point to the outside of the part (Fig. 3). Thus, enough information is put into the IGES file that a receiving system can rebuild a solid model from a complete surface model.

Untrimmed Mode

The untrimmed mode contains basically the same information as the trimmed mode. For each face the untrimmed surface plus all trimming curves are translated. But instead of explicitly trimmming the surfaces with the appropriate entities, surface and trimming curves are only logically grouped together. This usually requires manual trimming in the receiving system, and is only suited for some special applications.

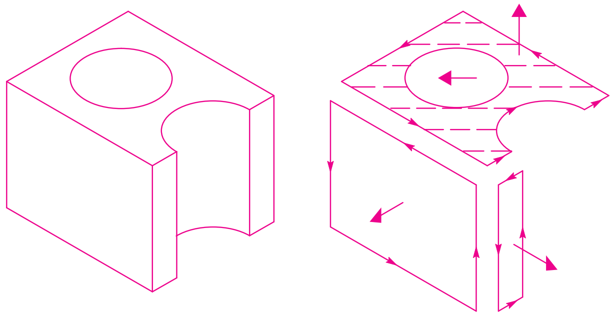


Fig. 3. (left) Solid model. (right) Surface model with normals.

Table II
Surface Mapping

HP PE/SolidDesigner	IGES 3D Entity (trimmed and untrimmed)	IGES 3D Entity (trimmed parametric)
Plane	Plane (108)	Ruled surface (118)
Cylinder	Surface of revolution (120)	Ruled surface (118)
Sphere Torus Cone Spun B-spline	Surface of revolution (120)	Surface of revolution
B-spline surface	B-spline surface (128)	B-spline surface (128)
Parallel swept B-spline	Ruled surface (118)	Ruled surface (118)

Trimmed Parametric Mode

The trimmed parametric mode uses the IGES trimmed parametric surface entity (144) and the curve on parametric surface entity (142) as representations of a trimmed surface. These entities have been established in the IGES standard for a longer time than entities 143 and 141 or the trimmed mode. For this reason they are more commonly used. The main difference from the trimmed mode is that the trimming is performed in the parametric domain of the surfaces. Each surface must have a parametric description that maps a point from the parameter domain D (a rectangular portion of 2D space) to 3D model space:

$$S(u,v) = (X(u,v),\ Y(u,v),\ Z(u,v)) \quad \text{for each } (u,v) \text{ in D.}$$

$$D = \{\text{all } (u,v) \text{ with } u_{min} \leq u \leq u_{max},\ v_{min} \leq v \leq v_{max}\}.$$

The following conditions apply to D:

- There is a continuous normal vector in D.
- There is a one-to-one mapping from D to 3D space.
- There are no singular points in D.

Furthermore, trimming curves in 2D space must form closed loops, and there must be exactly one outer boundary loop and optionally several inner boundary loops (holes). Fig. 4 illustrates parameter space trimming.

These restrictions make it clear that there will be two problem areas when converting HP PE/SolidDesigner parts to a parametric trimmed surface model: periodic surfaces and surface singularities.

On full periodic surfaces like cylinders, HP PE/SolidDesigner usually creates cylindrical topology. There will not necessarily be exactly one outer loop. Furthermore, 3D edges can run over the surface seam (the start of the period) without restriction. This leads to the situation that one edge may have more than one parametric curve (p-curve) associated with it. Also the p-curve loops may not be closed even if the respective 3D loop is closed. Fig. 5 illustrates this situation.

HP PE/SolidDesigner avoids this problem by splitting periodic surfaces along the seam and its antiseam. The seam and antiseam are the isoparametric curves along the parameters u_{min} and $u_{min}+u_{period}/2$. Thus, one face may result in

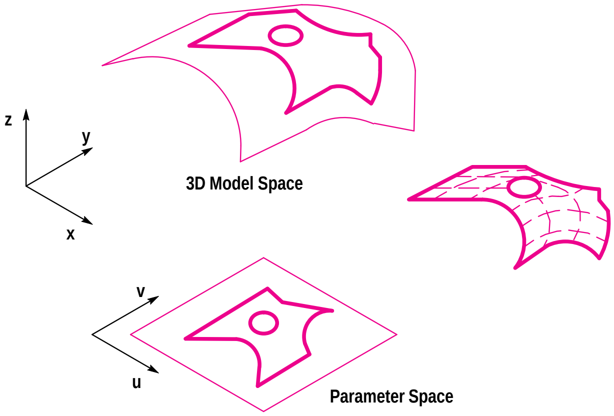


Fig. 4. Trimming in parameter space (p-space).

Table III
Model Mapping

Entity	Trimmed	Trimmed Parametric	Untrimmed	Wireframe
Parts and Assemblies	308+408	308+408	308+408	308+408
Faces	Entity 143	Entity 144	Entity 402	
Loops	Entity 141	Entity 142	Entity 102	
Edge+Base Curve	Curve Entity	Curve Entity	Curve Entity	Curve Entity
Base Surface	Surface Entity	Surface Entity	Surface Entity	None

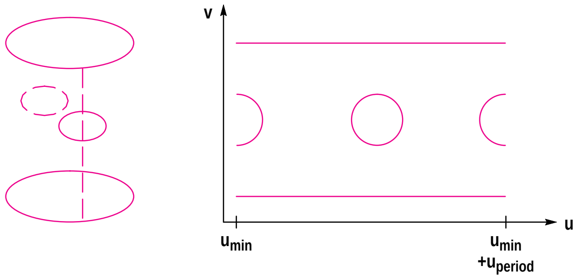


Fig. 5. Cylinder topology in 3D and p-space.

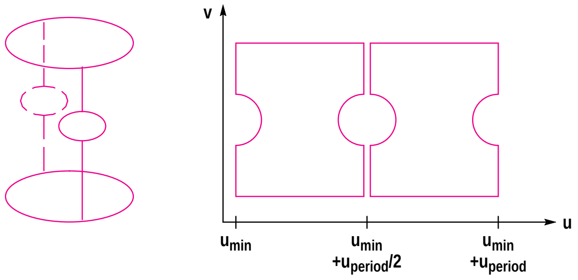


Fig. 6. Periodic surfaces in 3D and p-space after splitting.

two or four parametrically trimmed surfaces (u- and v-parametric surfaces (toruses)) in the IGES model. Fig. 6 illustrates this situation.

Another problem with parametric trimmed surfaces are surface singularities. Singular points are points where the surface derivatives and normal are not well-defined. For such points there is not always a one-to-one mapping from 2D parameter space to 3D model space. This means there is an infinite set of (u,v) points in parameter space that result in the same 3D model space point. Such singularities are easily created by rotating profiles around an axis where the profile touches the axis. Examples are cones, spheres, degenerated toruses, triangular spline patches, and so on (see Fig. 7).

HP PE/SolidDesigner is designed to handle singularities as a valid component of a model. They are marked with a vertex if they are part of a regular loop or with a special vertex loop if they are isolated from the remaining loops. However, it is not possible to express singularities in trimmed parametric surfaces legally in IGES.

To resolve this issue we reduce the singularity problem to the problem of the valid representation of triangular surfaces. The splitting algorithm just described is applied so that all singularities are part of a regular loop. Thus, we are always faced with the situation illustrated in Fig. 8.

Each singularity of a face is touched by two edges, one entering and one leaving the singular vertex. Knowing how triangular surfaces are handled in potential receiving systems, we offer four ways to export this kind of geometry. These are the four possible combinations of closed or open parameter loops and avoiding or using singularities.

Some systems do not need closed p-space loops, while others strictly expect them. If the closed option is chosen, the endings of the p-curves are simply connected with a straight line.

Geometrical algorithms usually become unstable near singularities. Some systems are not prepared to handle this situation and will fail. To avoid this, it is possible to shorten the parameter curves when entering or leaving a singular vertex and connect them at a numerically safe distance. This distance is measured in 3D space and is also configurable. It usually varies between 0.1 and 0.001. This will result in a surface where the region around the singularity is cut out. Fig. 9 illustrates the four possible singularity representations.

Wireframe Mode

For the wireframe mode HP PE/SolidDesigner also avoids the cylindrical topology, because in some cases information about shape would be lost (e.g., a full surface of revolution). After applying the face splitting algorithm all edges of the selected faces and parts are translated. No surface information is contained in the resulting IGES file.

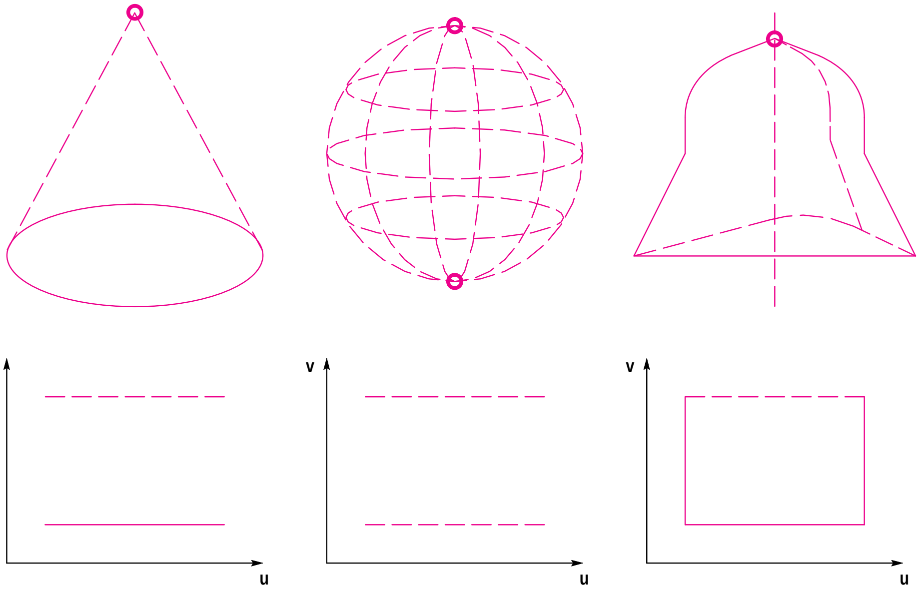


Fig. 7. Examples of surface singularities in parameter space.

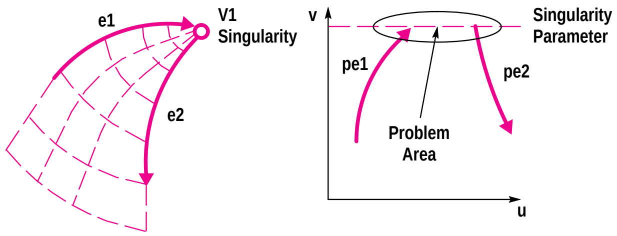


Fig. 8. Triangular surface situation.

Extracting Solid Information from Surface Models

IGES surface data from solid modelers often contains all surfaces of a closed volume or a connected face set. However, the connectivity between adjacent faces is lost. If the surface model fullfills some specific requirements it is possible for the receiving system to recompute this missing information. The following describes these requirements and shows how connectivity between faces can be reestablished. This method can be used to create a solid model from HP PE/SolidDesigner IGES output.

Automatic comparison of all boundary curves on coincidence or reverse coincidence would be a very time-consuming and numerically unstable task. However, it is common for the endpoints of the trimming curves of adjacent faces to be coincident within a very small accuracy. This makes it possible to identify trimming curves that share common start points and endpoints. If the two faces of these trimming curves have the same orientation one can try to connect the faces to a face set. For this task one must try to find a geometry for a common edge that fulfills the following accuracy constraints (see Fig. 10):

- The curve is close enough to surface 1.
- The curve is close enough to surface 2.
- The curve is close enough to curve 1.
- The curve is close enough to curve 2.

The first candidates for such a curve are the original trimming curves, curve 1 and curve 2. If either satisfies all four requirements it is incorporated into both face descriptions and the connection is established. If neither curve can be used, one can try a combination of the two, or reduce the receiving system's accuracy.

This method fails if the face orientation is inconsistent or if adjacent faces do not share common start points and endpoints.

Importing IGES Wireframe Data

IGES wireframe data can be easily imported into HP PE/-SolidDesigner, since HP PE/SolidDesigner's kernel supports wire bodies. The modified wire data can be saved in HP PE/SolidDesigner's data format. Possible uses for this capability include migration from old-line systems to HP PE/SolidDesigner, interaction with different sources and suppliers, and communication with manufacturers.

In HP PE/SolidDesigner a wire is defined as a set of edges connected by common vertices. A body consisting only of wires is called a wire body. IGES 3D curve data is used to generate the edges of a wire body. This includes lines, circles, B-splines, polylines, and composite lines. IGES surface data such as trimming curves of trimmed surfaces are also used to generate edges. To simplify later solid model generation the axis and generatrix of a surface of revolution are also transformed into edges for the wire body. Since only edges have to be generated for a wire body, there are no accuracy problems as described above for IGES surface importation. On the other hand, information on B-spline surfaces is lost.

Wire data imported from an IGES file is collected into an assembly. The assembly gets the name of the IGES file. Any substructure of the IGES file like grouping in levels is transformed into parts within the assembly. Thus, hierarchical information contained in the IGES files is maintained within HP PE/SolidDesigner. The generated parts can be handled like any other part in HP PE/SolidDesigner. To distinguish

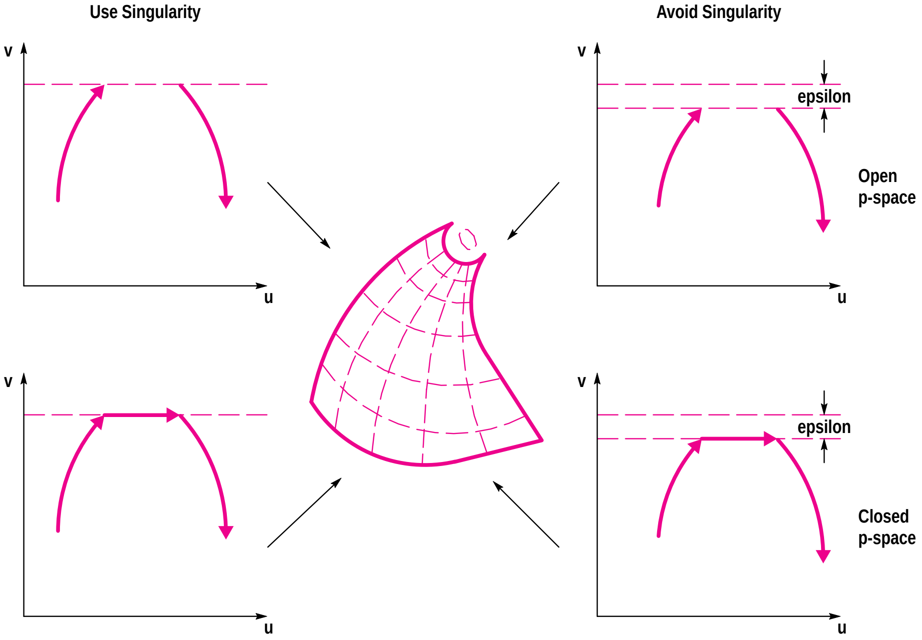


Fig. 9. Four possible singularity representations.

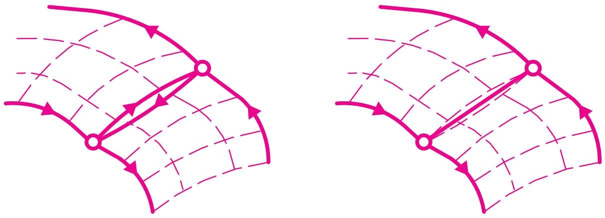


Fig. 10. Finding a common edge for adjacent faces.

wire parts, they can be colored. The options of HP PE/SolidDesigner's show menu work for the parts as well as the settings of the part container. A wire part can become the active part. The edges and vertices of a wire part are displayable, all browsers work with wire parts, and wire parts can be moved or become members of an assembly.

To build a solid model from a wire body, the edges and vertices of the wire body can be used to position a workplane. Then edges of the wire body can be selected and projected onto the workplane. The resulting profile can then be used to create a solid, for example by measuring an edge length needed for an extrude operation.

Fig. 11 shows an example of an IGES wireframe model with four parts and the resulting solid model. Automatic generation of solids from wires could be implemented but freeform surface information would probably be lost. The real benefit of wireframe import is for reference purposes.

STEP-Based Product Data Exchange

Manufacturing industries use a variety of national and industrial standards for product data exchange. These include IGES for drawing and surface exchange (international), VDA-FS for surface exchange (mainly the European automotive industry), and SET for drawing and surface exchange (France and the European Airbus industry). This variety of different incompatible standards causes a lot of rework and waste of valuable product development time which cannot be afforded if companies are to survive in the competitive marketplaces of tomorrow. Today's standards, originated in the early 1980s, are no longer satisfactory for product data description and exchange. Standards like IGES or VDA-FS, which are limited to surface or engineering drawing exchange, do not adequately handle other explicit product data categories such as product structure or assemblies or geometric solid models.

Industry trends today are characterized by internationalization of manufacturing plants which are spread over the continents of the globe, and by lean production in which many parts are subcontracted or bought from local or international suppliers. National standards and incompatibilities between existing standards are obstacles to these trends and will have to be replaced by international standards.

Large companies in the aerospace and automotive industries in the U.S.A. and Europe have now taken the offensive towards the implementation and use of STEP (*St*andard for the *E*xchange of *P*roduct Model Data) as an international standard for product data exchange and access, starting in 1994. Companies such as BMW, Boeing, Bosch, General Motors, General Electric, Daimler-Benz, Pratt&Whitney, Rolls Royce, Siemens, and Volkswagen have been using STEP prototype implementations in pilot projects with promising results.

Ultimately, STEP is expected to meet the following requirements for an international product data exchange standard:

- Provides computer interpretable and standardized neutral product model data. Neutral implies compatibility with any CAD or CIM system that best fits the design or manufacturing task.
- Implements the master model concept for product data. The entire set of product data for a product with many single parts is kept in one logical master model which makes it possible to regenerate the product as a whole at a new manufacturing site. This means that product assemblies, including administrative data and bills of material, are handled.
- Provides completeness, conciseness, and consistency. This requires special data checking and validation mechanisms.
- Provides exchangeable product data without loss. The product data must be exchangeable from one CAD or CIM system to another without loss of data.

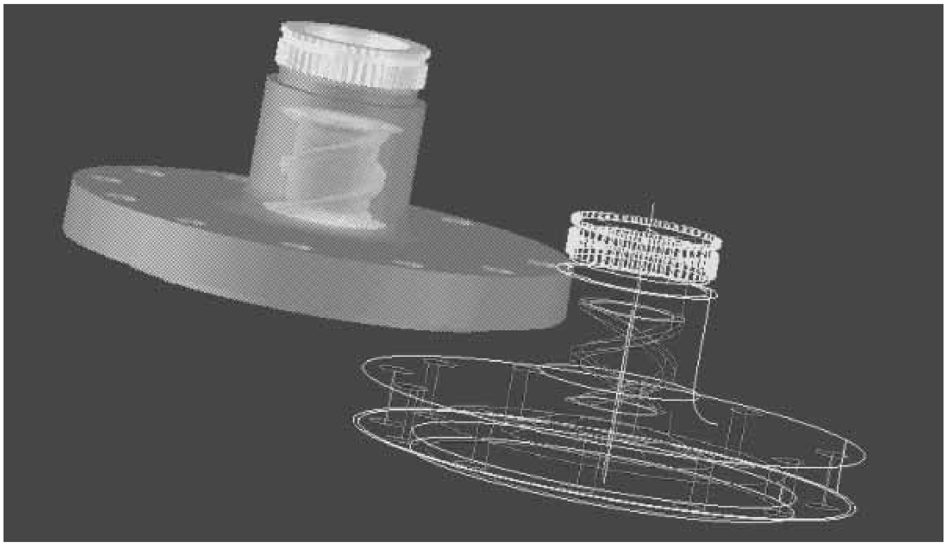


Fig. 11. Imported wire-body and the solid model constructed by HP PE/SolidDesigner.

- Provides long-term neutral data storage and interpretability. Product data is an important asset of a manufacturing company. The product data should be retrievable and interpretable by any CAD or CIM system after a long period of time, say 10 years or more. This is a significant challenge.

These requirements cannot be satisfied immediately. The STEP program also has shorter-term priorities for standardizing specific subsets of the product data. These include:

- The complete 3D geometric shape in the form of a 3D boundary representation solid model (B-Rep solids)
- Surface model and wireframe model data
- Product structure and configuration data.

Another priority is product documentation. An important goal is consistency of the engineering drawing with the 3D product geometry.

STEP Overview

STEP, the Standard for the Exchange of Product Model Data, is the ISO 10303 standard. It covers all product data categories that are relevant for the product life cycle in industrial use. STEP describes product data in a computer interpretable data description language called *Express*. The STEP standard is organized in logically distinct sections and is grouped into separate parts numbered 10303-xxx (see Fig. 12).

The resource parts of the standard describe the fundamental data and product categories and are grouped in the 1x, 2x, 3x, and 4x series. The Express data description language is defined in part 11. All other product description parts use the Express language to specify the product data characteristics in the form of entities and attributes. In addition to the product description parts there are implementation resources which are given in part 21, the STEP product data encoding scheme (the STEP file), and part 22, the Standard Data Access Interface (SDAI), which provides a procedural method for accessing the product data. There are different language bindings for part 22, such as C or C++ programming languages. The 3x series parts specify conformance requirements for STEP implementations.

Examples of STEP-standard resource parts are the fundamentals of product description and support (part 41), the geometrical shape (part 42), the product structure (part 44), material (part 45), the product presentation (part 46), tolerances (part 47), and form features (part 48). The application-specific resources are grouped in the 1xx series. Examples are drafting resources (part 101), electrical (part 103), finite element analysis (part 104), and kinematics (part 105). On top of the resource parts and application resources are the

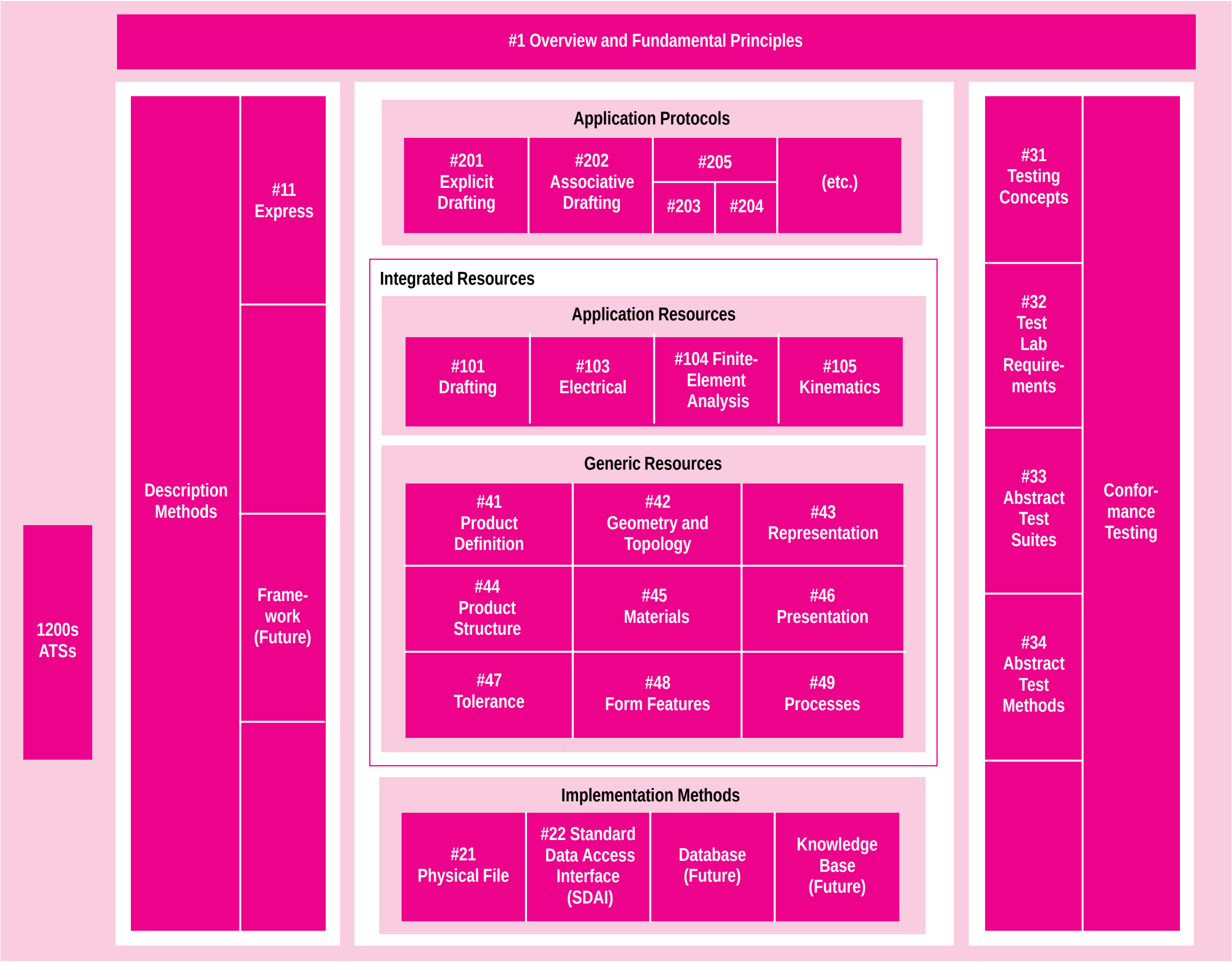


Fig. 12. Architecture of ISO 10303, Standard for the Exchange of Product Model Data (STEP).

application protocols (AP) which use the underlying resources in a specific application context, such as mechanical design for discrete part manufacturing, and interpret the resource entities in the application-specific context. STEP implementations for CAD or other computer-aided systems are based on application protocols. Application protocols are under definition for application areas like basic drafting, associative drafting, mechanical design, electrical design, shipbuilding, piping, architecture, and others. Here, we highlight just two examples, AP203 and AP214.

AP203: Configuration-Controlled 3D Design. AP203 was developed under the leadership of PDES Inc. It covers the major requirements for U.S.-based industries such as the aerospace industry for government and industrial manufacturing contracts. The product data covered in AP203 includes geometric shape (B-Rep solid models, surface models, wireframe models), product structure, and configuration management. AP203 is the underlying STEP specification for many CAD and CIM system implementations.

AP214: Core Data for Automotive Mechanical Design. AP214 has been developed by the automotive industry and covers product data categories relevant for the design and manufacturing of automotive parts and products. AP214, initiated in Germany and internationally supported, is still under finalization in parallel with its industrial implementation in CAD and CIM systems. The implementations have been coordinated and harmonized in the European ProSTEP consortium and the implementation is focused initially on the geometrical product descriptions (solid models, surface models) and product structure. However, all other kinds of product data categories relevant for mechanical design in the automotive industry (e.g., form features, materials, tolerances) are within the scope of AP214 and are going through the standardization process.

Initial Release

The initial release of STEP parts focuses on the most urgently needed kernel definitions of the standard, which cover the geometrical shape description, including all topological information, the product structure, and the configuration management data. Basic product documentation in the form of low-level engineering drawings is also covered. The parts included in the initial release are parts 1, 11, 21, 31, 41, 42, 43, 44, 46, 101, 201, and 203. The first two application protocols to become standards are AP201: Explicit Drafting and AP203: Configuration-Controlled 3D Design.

Upcoming releases of STEP will cover the next priorities in the area of drafting, such as AP202: Associative Drafting, materials, tolerances, form features, and parametrics, and other application protocols such as AP204: Mechanical Design Using B-Rep Solid Models and AP214: Core Data for Automotive Mechanical Design.

HP Involvement in STEP

HP has been working on the standardization of product model data since 1989 and has focused on the emerging international standard STEP for 3D product data. The product data focus has been on 3D kernel design data, completeness of topology and geometry, B-Rep solid models, and product structure and assemblies, as well as on associative drafting documentation. HP is an active member in organizations that have an impact on the ISO STEP standard, and contributes to STEP through national standards organizations in the U.S.A. (e.g., NIST, ANSI) and Europe (e.g., DIN in Germany). Of particular interest are the organizations PDES Inc., PRODEX, and ProSTEP.

PDES Inc. HP has concentrated on three major areas of PDES Inc.'s STEP activities: mechanical design of 3D product data, associative drafting for CAD data, and electronic data definition and exchange.

The mechanical design initiative of the U.S. aerospace and aircraft industries, the automotive industry, and the computer industry resulted in STEP application protocol 203. HP, a PDES Inc. member in the U.S.A. and an ESPRIT CADEX member in Europe, contributed to the 3D geometric design definition of AP203 in a joint effort of PDES Inc. and CADEX. The AP203 3D geometries cover solid models, surface models, and wireframe models and are shared by other application protocols, thereby promoting interoperability between different application areas.

HP has also been actively supporting the U.S. initiative to define a good-quality standard for associative drafting documentation in STEP. Associative drafting, covered by AP202, is considered an integral portion of the product data for contractual, archival, and manufacturing reasons. For example, government contracts and ISO 9000 require that product data be thoroughly documented. This includes engineering drawing data of a product in addition to the 3D product data and the configuration data. Electronic design and printed circuit board design data are also covered in STEP.

PRODEX. In 1992 participants in the ESPRIT CADEX project demonstrated publicly the first B-Rep solid model transfer via STEP for mechanical parts in Europe. To develop this new technology the PRODEX project was founded in 1992 with the goal of developing STEP data exchange for CAD design, finite element analysis, and robot simulation systems. Twelve European companies joined the project. So far, the project's achievements include the definition of a STEP implementation architecture, the development of a STEP toolkit, and the development of STEP preprocessors and postprocessors.

Product data exchange between the different vendors is ongoing and shows very promising results for CAD-to-CAD data exchange, CAD-to-finite-element-system exchange, and CAD-to-robot-simulation-system exchange. The STEP standard has been further fostered by a joint effort with the ProSTEP project to develop AP214, in cooperation with the U.S., European, and Japanese automotive industries.

ProSTEP. ProSTEP is an automotive industry initiative for a highway-like STEP product model data exchange. In 1992 the German companies Bosch, BMW, Mercedes-Benz, Opel (GM), Volkswagen, and Siemens launched an initiative to bring the major CAD vendors together with the goal of implementing the first harmonized set of STEP product data exchange processors (product data translators) for industrial use in the automotive industry. The approach taken was to compile the user requirements, to build on the results and experiences of the ESPRIT CADEX project, and to launch at the ISO level a STEP application protocol, AP214, which covers the core data for automotive mechanical design.

The following CAD/CIM systems are involved in the project and have STEP data exchange processors either available or under development: Alias, AutoCAD, CADDS/CV-Core, CATIA, EUCLID3, HP PE/SolidDesigner, EMS-Power Pack, I-DEAS Master Series, SIGRAPH STEPIntegrator, SYRKO, Tebis, ROBCAD, and others.

The initial focus in ProSTEP for STEP products is on design data exchange for 3D geometry: B-Rep solid models, surface models, and wireframe models. For migration from legacy systems, wireframe data needs to be supported, at least for data import. Communication with applications like numerical control (NC) programming systems today typically requires surface model data, although in the future more solid model data will be used. Initially, the HP emphasis is on bidirectional product model exchange (input and output) of 3D B-Rep and surface models.

STEP Tools Architecture

In STEP implementation projects, standardization has been extended beyond the product data to the STEP implementation tools. The CADEX, PDES Inc, PRODEX, and ProSTEP projects have all taken this approach.

A standardized STEP tool architecture provides the following benefits. These include shareability of tools between different implementors, shortened development time for STEP processor implementations (software development productivity gain), increased likelihood of compatibility between STEP implementations (differences in STEP definition interpretations are minimized), parallel development of tools (concurrent engineering), extendability of tools to track new standardization trends, increased flexibility (new STEP models require fewer code changes), and centralized maintenance of tools.

Fig. 13 shows the PRODEX STEP tools architecture. The functional blocks of a STEP toolkit or STEP development set are:

- STEP Standard Data Access Interface (SDAI),
- STEP Express compiler
- STEP file scanner/parser
- STEP file formatter
- STEP data checker
- STEP conversion tool.

The main interface to the STEP application is the STEP Standard Data Access Interface, which provides a computer programming language for dynamic access to the STEP data. Application-specific mapping and conversions are implemented on top of this interface.

The Express compiler conveys the product data descriptions contained in an Express schema (the metadata of the data model) to the toolkit. It contains an Express file reader and compiles the file contents to the internal representation of the data model. The SDAI is the recipient of the product data metamodel and uses the metamodel as a reference for the product instance data, which is imported through the STEP file scanner/parser.

The STEP file scanner/parser reads (scans and parses) the STEP instance data contained in a STEP data file and uses the currently valid metamodel for checking the syntax of the imported instance data.

The STEP file formatter formats the data to a part-21-conformant STEP file which is read from the SDAI by using the current valid metadata (e.g., a specific application protocol such as AP203).

The STEP data checker is a validation tool that checks the instance data currently in the SDAI based on the corresponding metadata model, which is also contained in the SDAI. The checking covers consistency checks like references between entities (e.g., existence dependency), and rule checking, which is covered in the metamodel. The checking is optionally applicable to the data in the SDAI. It is very helpful during the development of processors, for checking new metadata models, or for checking the first data imported from a new system.

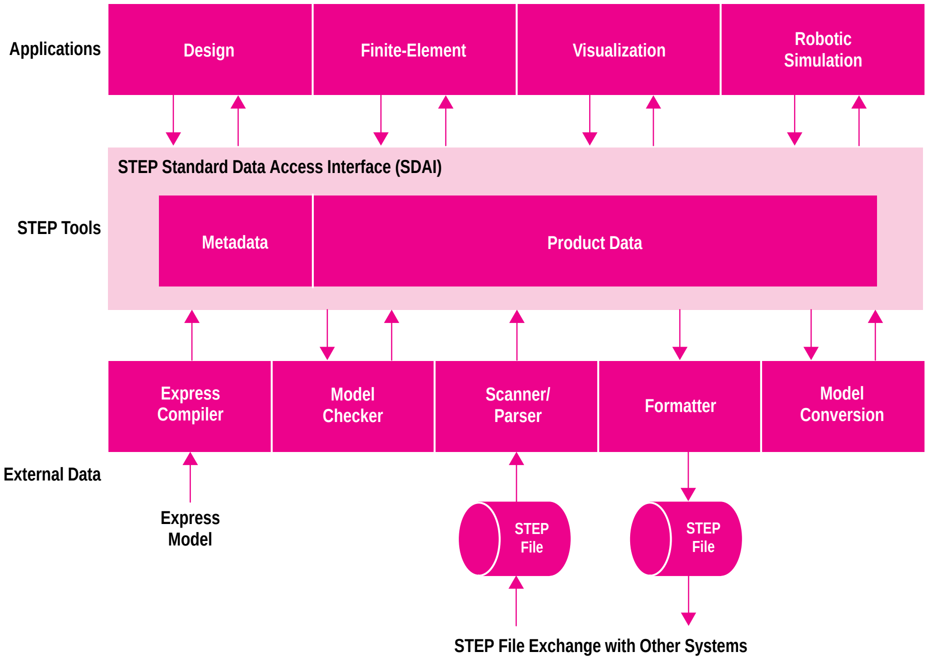


Fig. 13. PRODEX STEP tools architecture.

The STEP conversion tool is a pool of conversion functions (a library) that includes all kinds of geometrical, topological, and other model conversions. The focus is on geometrical conversions which are heavily used for data exchange between systems with different geometric modeling concepts. For example, one CAD system might use rational polynomial representations for its inherent geometric representation of curves and surfaces (e.g., NURBS, nonuniform rational B-splines), while the other might use nonrational representations (e.g., NUBS). In this case an approximation to the nonrational representation has to be applied, at the price of increasing the amount of data. For another example, a surface modeling system might export trimmed surface data with curve representations in 2D parameter space, whereas the receiving system might handle only 3D space curves. In this case the 2D parameter curves have to be evaluated and converted to 3D trimming curves in 3D space.

By using a STEP toolkit the requirements for the implementation of a STEP processor might be reduced to just the native data interface to the STEP tools, which consists of the data output to the SDAI (for the STEP preprocessor) and the data imported from the SDAI (for the STEP postprocessor).

The main task in linking a CAD system to the toolkit consists of defining and implementing the mapping between the system internal representation and the standardized entity representation in the schema of the standard (e.g., an application protocol).

HP PE/SolidDesigner STEP Implementation

The target application protocols for HP PE/SolidDesigner are initially AP203 and AP214, in which both solid and surface models are supported. In addition to the HP PE/SolidDesigner internal data models, the solid and surface models of other CAD systems are of major interest. With the introduction of STEP, B-Rep solid model data exchange comes into industrial use, representing a new technology shift. HP PE/SolidDesigner has its focus on solid models and is best suited for STEP-based bidirectional solid model exchange. However, surface models are also supported.

In addition to the geometric specifications, product information and configuration are covered in the implementation. In this article, the geometric and topological mappings are discussed. The assembly, product structure, and administration mappings are not covered.

STEP Preprocessor (STEP Output)

The preprocessor exports the HP PE/SolidDesigner model data in a STEP file. The preprocessor takes care of the mapping of the HP PE/SolidDesigner model to the STEP model.

The internal geometrical and topological model of HP PE/SolidDesigner is in many respects similar to the STEP resources of part 42 of the STEP standard. Hence the mapping is often straightforward. On the other hand, there are data structure elements that are not mapped to the STEP model.

HP PE/SolidDesigner uses the following geometric 3D elements:

- Analytics: 3D surfaces such as planes, cones, cylinders, spheres, and toruses, and 3D curves such as lines, arcs, circles, and B-splines
- Nonanalytics: typically 3D elements such as B-spline curves and surfaces, and linear and rotational swept surfaces.

The topology used for the exchange of solid models is based on the manifold topology of STEP part 42. The elements used are manifold solid boundary representations, closed shells, faces, loops, edges, and vertices. The link between the topology and the geometry is given by references from faces to surfaces and from edges to curves. The geometrical points are referenced by vertices.

The HP PE/SolidDesigner STEP surface models are also based on topological representations. Special elements are used for surface models, such as shell-based surface models and closed and open shells. The other underlying topolocial elements are the same as in the solid models. The geometric representations of the surfaces are typically the same as in the solid model representations.

STEP Postprocessor (STEP Input)

The HP PE/SolidDesigner postprocessor supports the import of B-Rep solid models and surface models along with the necessary product structure data. The postprocessor is

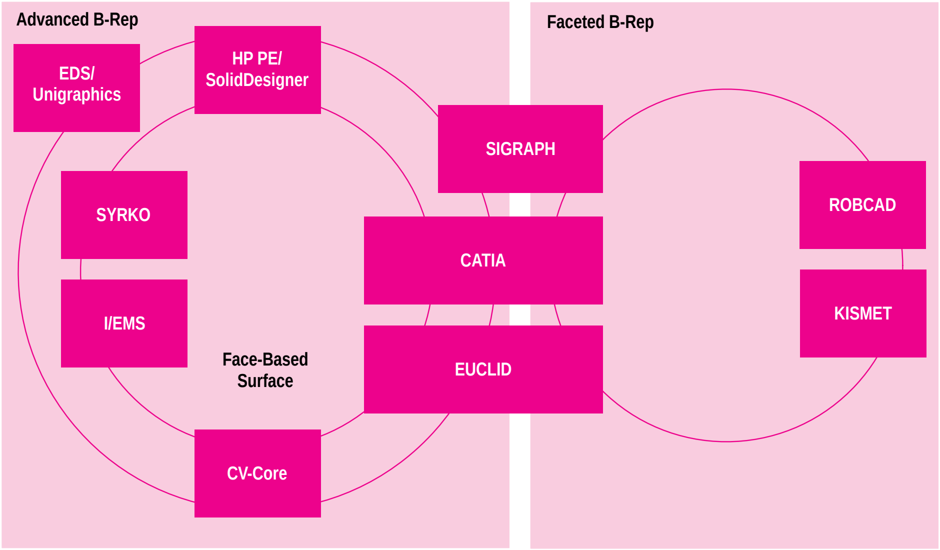


Fig. 14. Data exchange cycles between different CAD systems, including robot simulation systems, in the ProSTEP project.

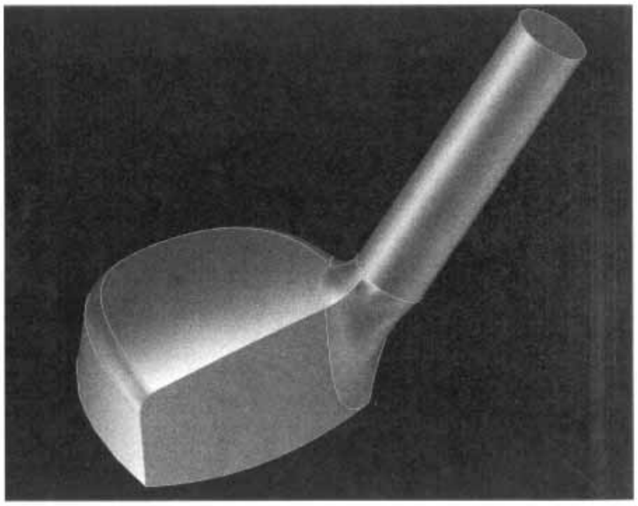


Fig. 15. Golf club solid B-Rep model imported into HP PE/Solid-Designer from CATIA (CAP-Debis).

capable of covering at least the functionality of the preprocessor so that it is possible to store and retrieve HP PE/Solid-Designer data in a STEP file representation (this is called the *short cycle test*).

The STEP postprocessor imports STEP files from other systems based on specifically supported application protocols. Postprocessing is one of the most difficult tasks in data exchange, especially when the data imported comes from a system that is very different from the receiving system. Potential problems arise in postprocessing if the sending and receiving systems have different accuracies, use different modeling techniques to generate the data, have different or missing surface connectivity, use different algorithms or criteria to determine surface intersections or connectivity, or use different model representations for similar model characteristics.

When surface models are imported, it cannot be guaranteed that they can be migrated to solid models even with user interaction. However, in special cases imported surface models can be migrated to solid models without problems. In many cases imported surfaces provide boundary conditions for the solid model. In most cases the data can be used as reference geometry to check interference or provide dimensions for the solid models. For example, an imported surface set might represent the surrounding boundary geometry within which the final mechanical part has to fit without interference.

Importing surface models into HP PE/SolidDesigner is considered important and critical since many other CAD systems, especially legacy systems, often support only surfaces or wireframe models, not solid models. Therefore, the postprocessing of STEP surface models needs to cover a broader scope than the preprocessing. Sometimes, different surface representations are used in different application protocols, such as AP203 and AP214. Hence, different external representations may need to be mapped to one internal representation in HP PE/SolidDesigner.

In the initial implementation of the HP PE/SolidDesigner postprocessor, topology bounded surface models are supported. These provide the most sophisticated description of the connectivity of the individual surfaces used in a solid model. Geometrically bounded surface models are supported as a second priority.

The Accuracy Problem

When importing CAD data from other systems the accuracy of the data plays a key role and determines whether a coherent and consistent CAD model can be regenerated to represent the same kind of model in the receiving system.

Let's define the term accuracy. There are different accuracy or resolution values that must be considered in geometric modeling and CAD systems. For 3D space, a minimum linear distance value (a length resolution value) can be defined, which is the absolute distance between two geometric points that are considered to coincide in the CAD internal algorithms; this represents the zero distance. We'll call this value

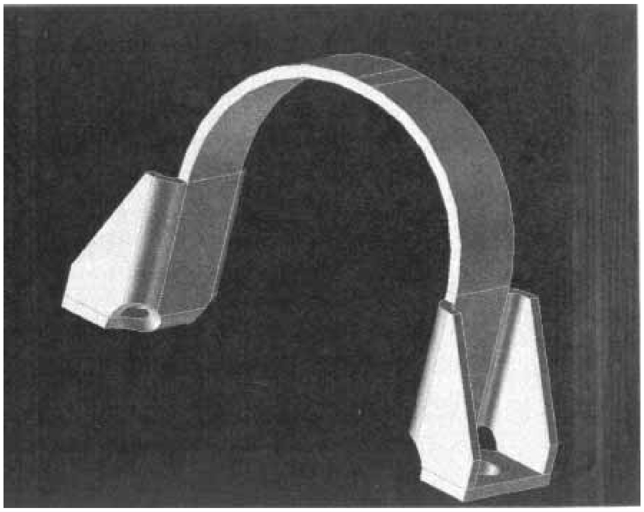


Fig. 16. Clamp solid B-Rep model imported from Unigraphics II (EDS).

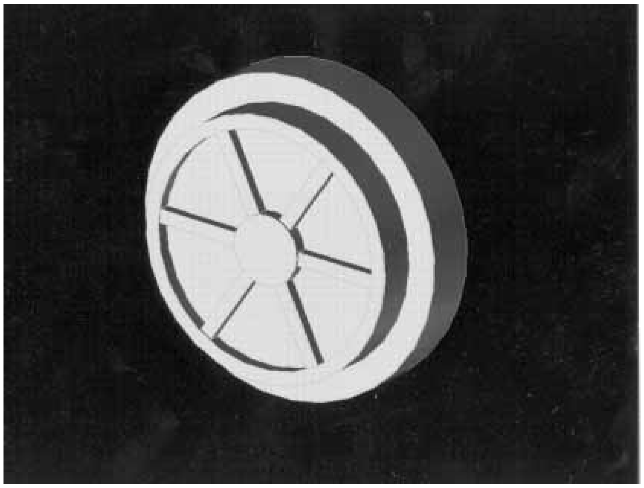


Fig. 17. Wheel solid model imported from SIGRAPH-3D (Siemens-Nixdorf).

---

* Often, CAD surface models are not consistent because the generating system lacks checking mechanisms or does not track connectivity. Very often, consistency and accuracy are the responsibilities of the user of the system rather than under system control.

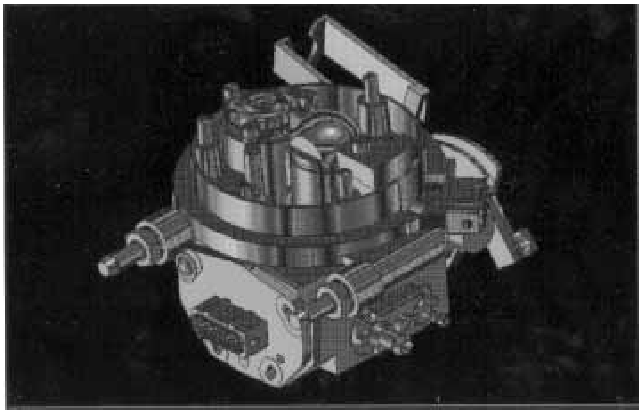


Fig. 18. B-Rep model imported from Unigraphics (EDS).

the *linear accuracy*. A typical value could be 10^{-6} mm which is highly accurate for many mechanical design applications. A similar value can be specified for *angular accuracy*, *parametric accuracy*, and so on. The discussion here is limited to linear accuracy.

If the sending system uses a higher linear accuracy (more precise data) than the receiving system, distinct geometric points will be detected to coincide in the receiving system. This might result in a change in the topology (which might cause further inconsistencies) or the geometry. If the sending system uses a lower linear accuracy (less precise data), the receiving system might complain that the topology is not correct or the geometry and the topology are inconsistent.

To prevent or at least minimize these kinds of accuracy problems it should be possible to adjust the accuracy in the receiving system to the accuracy values of the data to be imported. For example, if the sending system uses a different accuracy for the model generation process, say a linear accuracy of 10^{-2} mm, then the receiving system should adjust its internal algorithms to the same accuracy.

Experience with HP PE/SolidDesigner has shown that this kind of adjustable accuracy helps regenerate CAD models that were generated in different systems with different accuracies. Also, for data models composed of components with different accuracies, the components can be brought together on the assembly level to form a complete product model.

In the STEP implementation of AP214 an adjustable linear accuracy value is conveyed in the STEP file to tell the receiving system the appropriate accuracy value for postprocessing.

User Features

The user can select via the HP PE/SolidDesigner graphical user interface the objects (e.g., several B-Rep bodies) to put into a STEP file. For example, the user decides whether to send the data in a B-Rep solid model or a surface model representation. The user can choose some configuration parameters that help tailor the model data set for best communication to a specific target application. However, all data must comply with the STEP standard.

When importing (postprocessing) a STEP file the user can define some parameters that ease the processing of data. For example, the user might set the accuracy value before importing a data set that was designed with a specified accuracy, or might choose to convert the imported data to a different representation.

STEP Model Exchange Trials

Various STEP file exchanges have been performed within the last 12 months, not always with satisfying results. This has resulted in more development work by the exchange partners. This process of harmonizing the STEP preprocessors and postprocessors of different CAD vendors is considered to be of vital importance for the acceptance of the STEP standard and its application protocols. Within the ProSTEP project this process has worked particularly well. Other work has been done with, for example, AP203 implementors together with PDES Inc.

At this time, solid model data exchange can be said to be working very well, especially compared with what was possible with existing standards. STEP-based surface model exchange has also reached a level that was not possible with existing standards like IGES or VDA-FS, especially with respect to topological coherence, which is easily conveyed with STEP between many CAD systems. Of course, the wide variety of surface models, with the resulting accuracy and connectivity problems, will need to be addressed by the different CAD system vendors to optimize data transfer via STEP. In the meantime, STEP file exchange has matured to the point where STEP products are offered by various CAD vendors and system integrators.

Within the ProSTEP project one of the broadest ranges of STEP-based data exchange trials have been performed between HP PE/SolidDesigner and other CAD systems (see Fig. 14). Solid model industrial part data has been exchanged, for example, with CATIA (CAP Debis and Dassault/IBM), Unigraphics II (EDS), SIGRAPH Design and STEP Viewer (Siemens-Nixdorf), and others. Some of the successful results are shown in Figs. 15, 16, 17, and 18. Surface model industrial part data has been exchanged with CATIA, EUCLID, SYRKO (Mercedes-Benz corporate design system), and others. Some of the successful results are shown in Figs. 19 and 20.

Next STEPs

Future releases of the STEP standard covering product data categories such as materials, tolerances, form features, manufacturing process data, and others are expected in the next few months. The expected release of AP202, associative drafting, will allow documentation of the product data in engineering drawings. Work is ongoing towards the parameterization of product features, which needs further development in the STEP standard.

The expected finalization of AP214 will make it possible to convey the product data categories in STEP files and will help to reduce design and manufacturing development cycles for simple as well as complex products. This process will be supported by further extensive use of data communication networks in the various countries. The migration from existing standards is aided by several product offerings of IGES-to-STEP and VDA-FS-to-STEP data converters.

The STEP implementation technology based on the STEP Standard Data Access Interface will be broadened and used

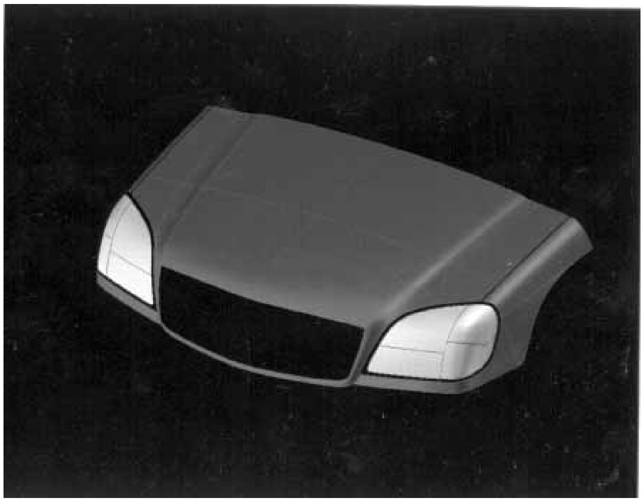


Fig. 19. Surface model imported from SYRKO (Mercedes-Benz corporate design system).

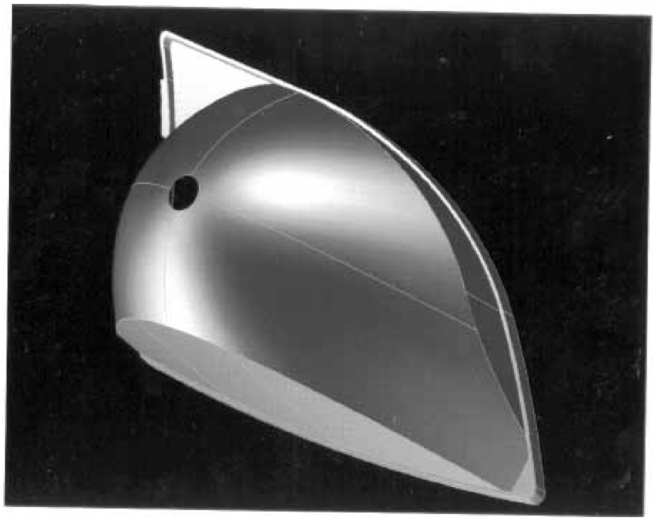


Fig. 20. Headlight reflector surface model imported from EUCLID (Matra Datavision).

in database access implementations to allow concurrent access by product design and manufacturing development.

However, for industrial use, the database technology and the STEP data access technology need to be extended and integrated. This process is expected to take several years.