Recommended Practices for External References

with References to the PDM Schema Usage Guide

Release 2.1

January 19, 2005

Contacts:

Jochen Boy
PROSTEP AG
Taunusstraße 42
80807 Munich / Germany
jochen.boy@prostep.com

Phil Rosché
ATI/PDES, Inc.
5300 International Blvd.
North Charleston, SC 29418 USA
rosche@aticorp.org

Andreas Trautheim
PROSTEP AG
Dolivostraße 11
64293 Darmstadt / Germany
andreas.trautheim@prostep.com

Table of Contents

1 External References Test Scenario 3

1.1 Motivation 3

1.2 Nested External References 3

1.3 Possible scenarios 4

1.4 Scope of this document 5

2 Master File Structure 5

2.1 Placement of the file name 5

2.1.1 On import 5

2.1.2 On export 6

2.2 Important Notes: 6

2.3 Document Format Properties 7

2.4 File Example 8

2.4.1 Master file containing assembly structure and part master data 8

2.4.2 Leaf node geometry files 8

3 References to the PDM Schema Usage Guide 8

List of Figures

Figure 1: Relationship of generated files 3

Figure 2: Reference mechanism with nested external references. Leaf nodes (P3, P4, P5) are geometry files. 4

Figure 3: Instance diagram for external references 5

Figure 4: Physical File excerpt for Figure 2 7

Figure 5: Structure Diagram for Document Format Properties 8

1 External References Test Scenario

1.1 Motivation

As CAx-PDM interaction becomes an increasingly common part of the business process, guidelines are needed for how the information regarding exchanged models can be provided in a way that both CAx and PDM systems can deal with it. The method presented here uses the so-called “External References” approach. Therefore, the model is split during the export process into several individual files, where the assembly structure is contained in one file, and the individual parts in a single file each. The assembly structure in the master file links to its components in the separate geometry files.

The ‘master’ file shall use the actual PDM Schema or the IS version of AP214-CC06 as its file schema. It contains all the PDM-relevant information (e.g. part numbers, versions) along with the model structure and associated information (e.g. validation properties). The leaf nodes of the assembly tree then reference the appropriate geometry externally.

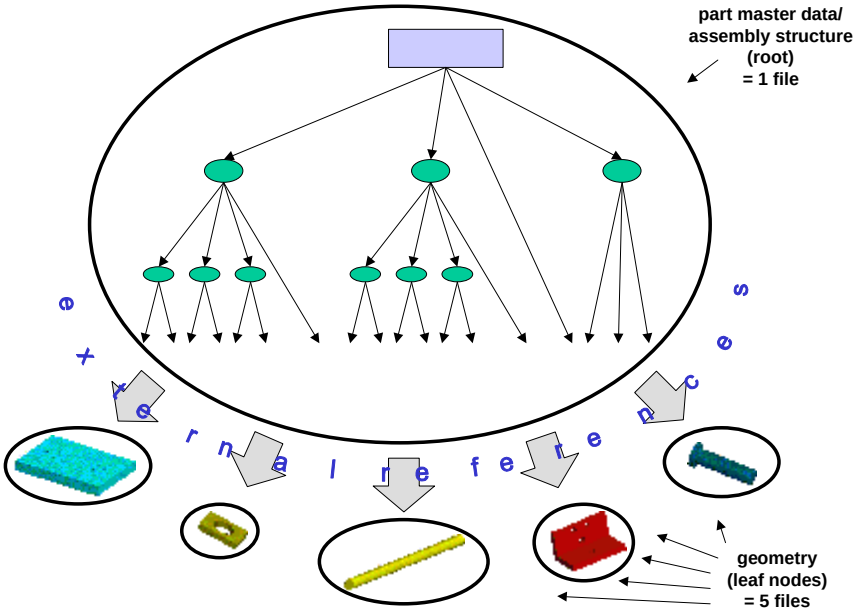

Figure 1: Relationship of generated files

The geometry part files shall be conformant to either AP203 or AP214 and contain only geometry definitions. In general, these files can also be native models, but in the context of CAx testing, STEP files should be used.

1.2 Nested External References

It has turned out that when dealing with large assemblies, it is often very unhandy to exchange the complete structure in a single file. Especially when only a single part or small subassembly of the assembly has been changed, transferring the complete structure means a lot of unnecessary data exchange.

There, the mechanism for “nested” external references has been introduced, i.e. the “master” file itself is split into several parts, where each file represented one node in the assembly structure.

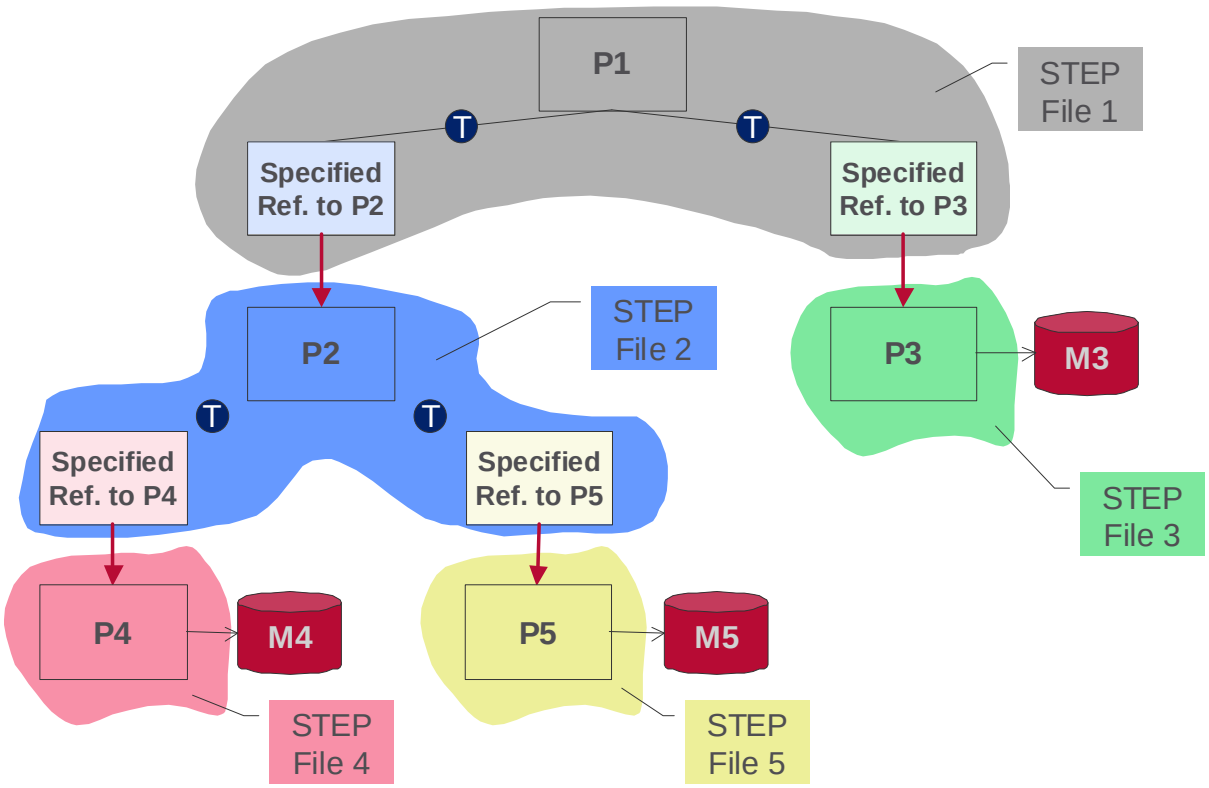

Figure 2: Reference mechanism with nested external references. Leaf nodes (P3, P4, P5) are geometry files.

The complete assembly structure is broken down level by level, and subassemblies are referenced the same way as the single geometry parts.
A way to distinguish whether the referenced file is a single part (geometry) or a sub-assembly (structure), is by using the Document Format Properties, see section 2.3. Therefore, all structure files belonging to a split assembly structure shall include the document format property.
A first test case for nested external references is defined in the Round15J Test Suite document.

1.3 Possible scenarios

There are basically two usage scenarios for the background of this document: CAx to CAx and CAx to PDM exchange.

In CAx to CAx exchange, the set of STEP files is both written and read by – possibly different – CAx systems. There may be various reasons to apply this functionality in this context.

In CAx to PDM exchange, the set of STEP files is written by a CAx system, while a PDM system is on the receiving end. With this functionality, the PDM system only needs to read the 'master' file to obtain the full information needed. With regard to joint testing activities between the two forums, a model may be generated from a PDM system, then imported and exported again by a CAx system and finally fed back to the originating PDM system in order to see which information could be maintained and which was lost.

1.4 Scope of this document

This document gives an overview on how to apply the external reference mechanism described in the PDM Schema Usage Guide (http://www.pdm-if.org/pdm_schema/) and is designed to give the CAx vendors a guideline for their implementations. It therefore gives a list of relevant sections in the PDM Schema Usage Guide.

2 Master File Structure

The structure within the master file which identifies the external references, i.e. links the assembly nodes to their associated documents, has been harmonized between the CAx- and PDM-IF. It is shown in the following diagram:

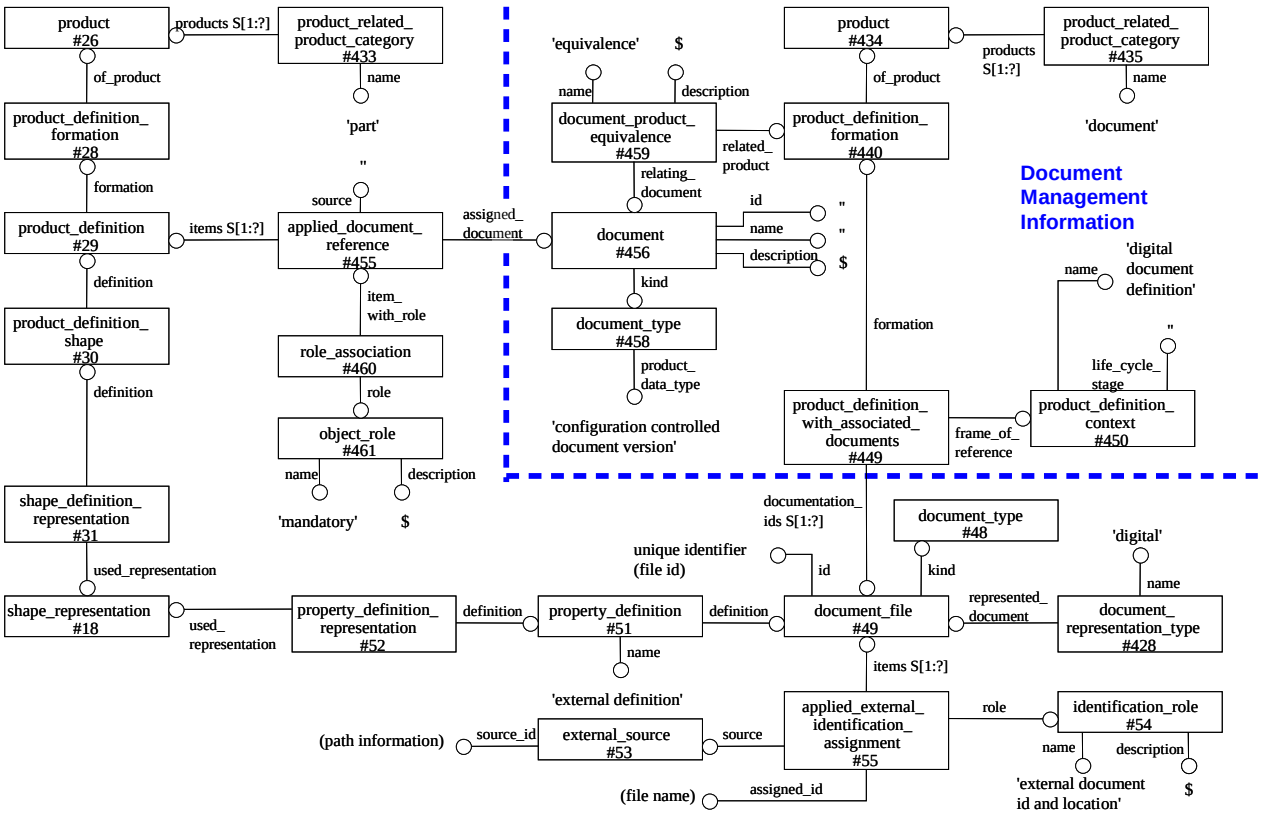

Figure 3: Instance diagram for external references

2.1 Placement of the file name

The following agreement was made on where to put the file name for an external reference file in the master file, and concerns the entities #49, #53, #54, and #55 in the diagram above.

2.1.1 On import

- The post-processor should look for the file name first in `applied_external_identification_assignment(AEIA).assigned_id (#55)`.
- If the file name is there, it should be used.
- If the `AEIA` is there, but its `assigned_id` attribute is empty, look in `external_source.source_id (#53)`. This is to ensure downward compatibility with older files.

- If the AEIA is not there, take whatever is found in document_file.file_id (#49) and go on from there.

2.1.2 On export

- Older (AP214 DIS), or AP203 1st Edition, processors, which do not support the AEIA, should put the file name into document_file.file_id.
- Newer, AP214 or PDM Schema, processors should create the AEIA and its appending entities, and put the file name into AEIA.assigned_id. The external_source.source-_id (path information) should be left empty.
- In most cases, the unique identifier in document_file.file_id is a replicate of the file name, not at last to ensure compatibility with older processors. (There are, however, scenarios where this is not necessarily the case, e.g. if the uniqueness of the identifier has been ensured company-wide, and there are several versions of the respective file.)
- The three entities below the document_file (AEIA + ...) are there to exchange the semantics, i.e. that the information found is indeed the name of the file as found on the computer or storage media, and not any arbitrary identifier. In the light of pure CAD systems providing external references, this information is not usually populated, but it should be included to support other scenarios where additional path or source information might become of vital importance.
- To sum this up, the recommendation to the CAx-IF participants on External Reference export is:
  - Put the file name into AEIA.assigned_id ,
  - And replicate file name into document_file.file_id, if a separate (unique) file ID is not available.
  - Use external_source.source_id for, e.g., path, directory, URL, hyperlink or equivalent information.
  - Leave external_source.source_id empty in the case of an asynchronous data exchange.

2.2 Important Notes:

- The following attributes shall have the same values in the master file, where an external shape is referenced, and in the referenced file containing the respective shape:
  - shape_representation.name
  - shape_repredentation.id
  - product.id
  - product.name

  where the consistency of the shape_representation attributes is of most importance.
- The last attribute in document_file (characterized_object.description) is required to be $ (not instantiated).
- Systems not supporting document management at all should skip the entities within the blue box in Figure 3 and link the applied_document_reference.assigned_document directly to the document_file (in the example above, #455 would point to #49 instead of #456).
- There is a known issue regarding the attribute values of #54 identification_role. The AP214 mapping requires the name to be 'external document id and location' and the description attribute to be not instantiated ($). These values are not in the list of recommended (allowed) values given in the PDM Schema Usage Guide. Since changing the values in AP214 would require modifications to the mapping, for the time being, the AP214 values will be added to the list in the PDM-UG.

```
#18= SHAPE_REPRESENTATION('',(#17),#8);
#26= PRODUCT('10001','L-BRACKET','NOT SPECIFIED',(#25));
#28= PRODUCT_DEFINITION_FORMATION('','/ANY',#26);
#29= PRODUCT_DEFINITION('design','',#28,#24);
#30= PRODUCT_DEFINITION_SHAPE('','SHAPE FOR L-BRACKET',#29);
#31= SHAPE_DEFINITION_REPRESENTATION(#30,#18);
[...]

#48= DOCUMENT_TYPE('');
#49= DOCUMENT_FILE('l-bracket_prt.stp','','',#48,'',$);
#51= PROPERTY_DEFINITION('external definition','',#49);
#52= PROPERTY_DEFINITION_REPRESENTATION(#51,#18);
#53= EXTERNAL_SOURCE(IDENTIFIER(''));
#54= IDENTIFICATION_ROLE('external document id and location',$);
#55= APPLIED_EXTERNAL_IDENTIFICATION_ASSIGNMENT
     ('l-bracket_prt.stp',#54,#53,(#49));
[...]

#428= DOCUMENT_REPRESENTATION_TYPE('digital',#49);
#433= PRODUCT_RELATED_PRODUCT_CATEGORY('part',$,(#26,#72,#117,#159,
     #213,#245,#300,#374,#402));
#434= PRODUCT('20001','L-Bracket_Doc','',(#436));
#435= PRODUCT_RELATED_PRODUCT_CATEGORY('document',$,(#434,#441,#443,
     #446,#448));
#440= PRODUCT_DEFINITION_FORMATION('1','',#434);
#449= PRODUCT_DEFINITION_WITH_ASSOCIATED_DOCUMENTS('1','',#440,#450,
     (#49));
#450= PRODUCT_DEFINITION_CONTEXT
     ('digital document definition',#437,'');
#455= APPLIED_DOCUMENT_REFERENCE(#456,'',(#29));
#456= DOCUMENT('','',$,#458);
#458= DOCUMENT_TYPE('configuration controlled document version');
#459= DOCUMENT_PRODUCT_EQUIVALENCE('equivalence',$,#456,#440);
#460= ROLE_ASSOCIATION(#461,#455);
#461= OBJECT_ROLE('mandatory',$);
```

Figure 4: Physical File excerpt for Figure 2

A complete physical file according to this structure is available for reference testing.

2.3 Document Format Properties

The Document Format is an additional property which completes the structure displayed in Figure 3, and is attached to the `document_file` entity. It provides information about the data format of this referenced file.

The format of the referenced file is stored in the `descriptive_representation_item.description` (see Figure 5). Recommended values are:

- `'STEP AP203'` if the referenced file is a geometry file conforming to the first edition of AP203
- `'STEP AP203E2'` if the referenced geometry file conforms to the second edition of AP203
- `'STEP AP214'` in case the referenced file is a geometry file in AP214 IS format
- `'STEP AP214 CC06'` if the referenced file contains a subassembly, and conforms to AP214 CC06.

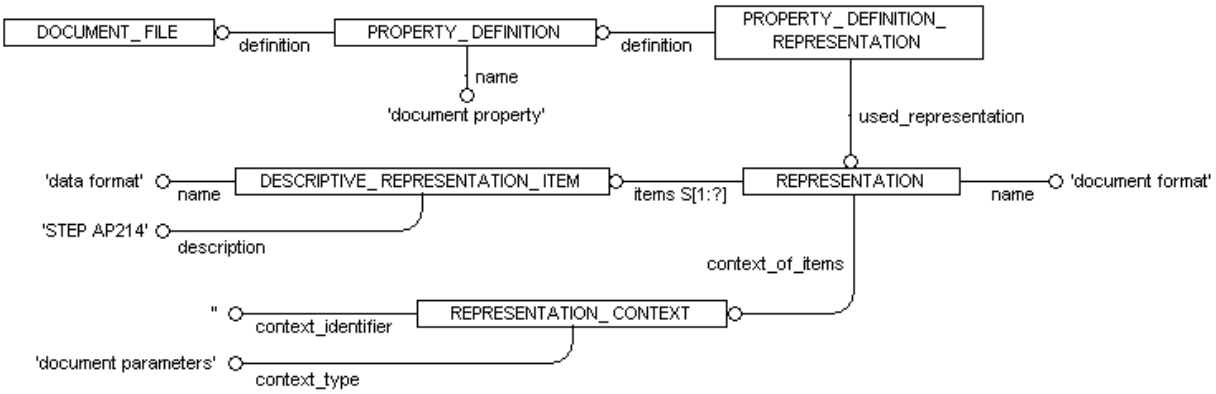

Figure 5: Structure Diagram for Document Format Properties

2.4 File Example

There is a set of example files available from the CAx-IF web site under "Joint Testing Information", dated 11-25-2002. It is based on the 'AS1' test case. The master file is according to the structure given above (not including document format properties or nested external references), and all geometry files are based on AP214-IS.

2.4.1 Master file containing assembly structure and part master data

See file: ext_ref_master.stp in the ZIP file.

2.4.2 Leaf node geometry files

See files: nut_prt.stp, plate_prt.stp, rod_prt.stp, bolt_prt.stp, and l_bracket_prt.stp in the ZIP file.

3 References to the PDM Schema Usage Guide

This section gives an overview on relevant sections in the PDM Schema Usage Guide, which is available from http://www.pdm-if.org/pdm_schema/. The current version is V4.3 for the PDM Schema V1.2 (January 2002).

- Section 3.2: External Part Shape
  - Section 3.2.1: Geometric Shape Property
  - Section 3.2.3: Relating Externally Defined Part Shape to an External File
- Section 3.3: External Geometric Model Structure
  - Section 3.3.1: Relating Part Shape
- Section 4.1: Explicit Assembly Bill Of Material
- Section 4.4: Relating Part Shape Properties to Product Structure
  - Section 4.4.1: Explicit Representation of Complete Assembly Geometry
  - Section 4.4.2: Implicit Relationships between Assembly Components
- Section 7.1: External File Identification
- Section 9.6: Document source property
- Section 10.2: External File Reference