**P21 e3 DIS Testing**

Minutes of June 5, 2104 Telecon

## Attendees

Martin Hardwick <hardwick@steptools.com>
Robert Lipman <robert.lipman@nist.gov>
Michael Benda <mjbenda@rockwellcollins.com>
Hedlind Mikael <mikael.hedlind@scania.com>
Dave Loffredo <loffredo@steptools.com>
Ed Paff  <ejp@transcendata.com>
Tom Thurman <thomas.r.thurman@imonmail.com>

## Apologies for Absence

Jochen Fritz <jfritz@steptools.com>

Nicolas Figay <nicolas.figay@eads.net>

## PMI Splitting

ITI has split the sp3\_boxy file produced by NIST into a geometry file and a pmi file. STEP Tools has viewed the geometry file and is working on a viewer for the separated pmi file. The two files and the original NIST data can be found at the following URL:

ftp://www.steptools.com/private/P21e3\_DIS\_testing/PMI/

STEP Tools is sponsoring the development of open source programs to split and merge P21 files using the Edition 3 specification. The programs are available at the following URL: <http://tinyurl.com/thundercode>.

## ZIP Assemblies

No visible progress has been made on this test case since the last telecom, but the PMI merge and split code is being updated to handle this case as well.

ftp://www.steptools.com/private/P21e3\_DIS\_testing/ZIP\_Assembly/

In the last telecom:

* We decided the sub-tree directory names should include the NUAO identifier(s) for their corresponding node in the assembly tree.
* We learned that the LZMA algorithm has been shown to make STEP files three times more compressed than the more commonly used deflate algorithm, but at the cost of an increase in the compression time.
* We learned that each compressed file is required to document the algorithm used for its compression in its header and that different components in the same ZIP can used different compression algorithms.
* We recommended that the choice of the best compression algorithm should be left to the end user and that the standard should be silent on the matter.

## Unit Definitions

The unit definitions have been merged into a single file called units.stp.

ftp://www.steptools.com/private/P21e3\_DIS\_testing/Units/

We continued our discussion on whether unit references should be encoded as a URL or a URN. The following two lines of code show equivalent URL’s and URN’s.

<http://standards.iso.org/iso/10303/tech/reference\_data/41/si\_base\_units.stp#METRE>;

<urn:iso:std:iso:10303:-41:tech:unit:metre>;

Explanation of URN:

urn Indicates this URI is a URN, instead of the more common URL (http)

iso   URN namespace  (other examples are oid, usbn)

std ISO standard

iso originating organization (other examples are iec, iso-ies, iso-cie)

10303 STEP standard

-41 part of multipart standard (hyphens required)

Tech associated or embedded resource defined by committee that created the standard

<the rest> unspecified -- controlled by committee.

In the above example, the URN has small advantage over the URL but perhaps the following URN will be acceptable because a file does not have to exist to define a URN.

<urn:iso:metre>;

It is also conceivable that the following URN might be acceptable even though there are currently small technical issues stopping the deployment of unit definitions for a Fahrenheit measure in a STEP file.

<urn:iso:Fahrenheit>;

We further discussed the requirements if all the unit definitions are to be defined in STEP files. The definition of literal constants such as PI, and EXPRESS constants such as negri\_pi, have both led to extensive additions to the edition 3 format for which other strong use cases are currently lacking.

On the other hand the problem might be that we have not been sufficiently ambitious. It was pointed out that if constants can be defined then we are close to allowing expressions to be defined. The use of the PLIB schema for expressions has never been very satisfactory because the expressions are too hard to read when encoded into the STEP files. However, if Part 21 has an expression evaluation capability (probably in the anchors) then potential STEP applications such as parametrics and construction history might become more tractable. One interesting possibility might be to use Modelica as the expression language.

## Digital Signatures

STEP Tools gave a demonstration of how to make a digital signature using a private key and a signing certificate and how to verify that a file has not been edited since it was signed using a public key.

ftp://www.steptools.com/private/P21e3\_DIS\_testing/Digital\_signatures/

We discussed white space and the issues that might arise if a file is read into Notepad/Wordpad and converted from line-feeds to carriage returns or vice versa. These characters are explicitly excluded from the Part 21 character set so they will not be included when computing and verifying the hash value.

We discussed supporting multiple signatures. The easiest procedure is for each signature to be applied to all of the characters that precede that signature. Thus each new signee is also verifying the signature of the previous signees.

For this to be consistent it would be best if the signatures came after the END-ISO-10303-21 keyword as shown in the example on the ftp site.

* The next conference call will be held on Friday June 27 at 4PM Paris, 3PM London, 10AM New York and 7AM Seattle.

## Action Items

1. Complete the first ZIP assembly example.
2. Consider the best approach for unit definitions: URN’s or URLs’?
3. Consider if Edition 3 should offer any support to the definition of parametrics and if so how?
4. Extend the digital signatures example to include the creation of signing certificates.
5. Demonstrate signing at one site (ITI) and verification at another site (STEP Tools).

As recorded by Martin Hardwick <hardwick@steptools.com>