An Approach to Accessing Product Data across System and Software Revisions

Alexander Ball

UKOLN, University of Bath
Bath, UK

Lian Ding

Department of Mechanical Engineering
University of Bath
Bath, UK

1st November 2006

Abstract

Long-term users of engineering product data face a significant challenge due to the ephemeral nature of CAD file formats and the applications that work with them. STEP, the Standard for the Exchange of Product Model Data (ISO 10303), promises to help with meeting this challenge, but is not without problems of its own. We present a complementary solution based on the use of lightweight file formats to preserve specific aspects of the product data, in conjunction with a registry of relevant Representation Information as defined by the Open Archival Information System Reference Model (ISO 14721). This registry, currently under construction, may be used to identify suitable destination file formats for different purposes, and may aid in the recovery of information from these formats in the future.

1 Introduction

One of the major problems facing long-term users of engineering data is a lack of compatibility between software systems, specifically between competing systems and between different generations of the same system. This can partly be explained by market forces, and the consequent tactics used by vendors to encourage customer loyalty, and partly by genuine conceptual differences between systems. Whatever the reason, the consequence is that data created using a particular piece of software is in danger of becoming inaccessible to its

creators once that piece of software is retired or replaced as part of ongoing modernization.

Several possible solutions to this problem present themselves. One is investment in specialist migration tools that can convert files between competing file formats. Another is to develop a standard exchange format; this is the eventual aim of STEP [1]. The problem with both of these solutions is that the results of the conversions can be unpredictable in quality — or else rather costly — and may involve a certain degree of data loss. A third solution is to develop emulators or virtual machines that simulate the native application; however, without the ability to extract the data into the replacement application the data may not be re-usable, due to a lack of interoperability with newer systems or a lack of designer expertise in the old application.

The problem of access to data across software and system revisions is therefore serious, and it is unlikely that a wholly satisfactory solution will be forthcoming in the near future. Nevertheless, it is the contention of this paper that the meta-problem — how to determine the best possible course of action in a given set of circumstances — is soluble. Given a registry of information relating to the interpretation of data files, it would be possible to interrogate this registry to look up the characteristics of various file formats, and to assess the benefits and drawbacks of any available migration/emulation options.

The paper begins with a review the advantages and disadvantages of STEP as an exchange and preservation format (section 3), and then picks out some of the more significant lightweight representations that could play a part in an exchange or preservation strategy (section 4). The concept of representation information is introduced in section 5, along with an example of gathering representation information and an explanation of how it can be used as a decision-making resource via automated tools.

2 Challenges for current PLM practice

Product Lifecycle Management (PLM) aims to provide a shared platform for effectively capturing, representing, organizing, retrieving and reusing product-related lifecycle information across companies, and to support the integration of the existing software systems, including CAD/CAM/CAE and ERP/CRM/SCM. Currently, the main challenge for PLM is information sharing and exchange, for the following reasons:

- The scope of PLM includes not only data held in a highly structured form (e.g. geometric models, databases of one IT solution or another), but also information stored in less structured or formal ways (e.g. text documents), and even the tacit knowledge of employees (e.g. design rationale and lessons learned). It is the long term objective of PLM to represent and organize all this information and knowledge in digital format, and

to make it traceable and retrievable throughout the product lifetime, but currently there is still no effective solution.

- Several commercial application systems have been developed and rapidly enhanced to support different phases of the product lifecycle. For a PLM system to be successful, it must integrate these advanced application systems, and be able to incorporate newly developed applications into existing systems. To date, however, most PLM systems' integrative capabilities are limited to specific transactions. One of the reasons for this is the multiplicity of document and proprietary format types that need to be handled.

- The product life can be a long time, maybe in the order of decades. This leads to two problems. Firstly, how to ensure the traceability and retrieval of information and knowledge during the product lifecycle, such that files in older versions can be accessed by the latest/future application tools. Secondly, during the product life, especially in a collaborative environment, changes may occur many times across different situations and contexts. These changes may further affect product content, metadata and configurations in varying formats. Thus, the problem is how to handle this evolution effectively throughout the whole product life, i.e. how to update, report and merge these changes on different semantic levels.

- With the trend towards globalization, PLM needs to support a collaborative environment where information and knowledge is transmitted between geographically distributed applications and users. Obviously, conventional representations such as CAD models are not optimized for such environments.

From the above discussion, it can be seen that conventional representations/file formats are unable to meet the requirements of PLM; new representations — indeed the whole future direction of file formats — need to be explored.

3 Standard for the Exchange of Product Model Data

3.1 STEP as a preservation and access tool

STEP (STandard for the Exchange of Product Model Data) is an international standard addressing the representation and exchange of product data [1]. Over the last two decades, STEP has been expanded from the product design phase to incorporate later life-cycle phases, such as maintenance and repair, and extended to cover aerospace, automotive, electrical, electronic, and other industries. In its initial conception, STEP focused on information exchange in the

phases of product design and machining, producing Application Protocols (AP) such as AP203 (Configuration controlled 3D designs of mechanical parts and assemblies), AP204 (Mechanical design using boundary representation), AP214 (Core data for automotive mechanical design processes), and AP224 (Mechanical product definition for process plans using machining features) [2, 3, 4, 5]. Although these parts have been widely applied in industry and the academic community — for example, AP203 delivers CAD data in a neutral format, which is readable by most CAD systems — the initial STEP parts have three problems for CAD model exchange: 1) the original designer's intent (e.g. concerning features and constraints) may be lost or misunderstood; 2) the exchanged model is difficult to modify; 3) the construction history of the design is lost.

In order to overcome these problems, STEP has recently developed the integrated generic resource Part 55 (Procedural and hybrid representation) [6], providing basic mechanisms for procedural or construction history modelling [7]. Additionally, STEP has developed two integrated application resources: Part 108 (Parameterization and constraints for explicit geometric product models) and Part 111 (Elements for the procedural modelling of solid shapes) [8, 9]. Current commercial CAD systems adopt a hybrid representation including the procedural model, which describes the design history as a feature tree, and the parameters, which consist of a set of features (modelling functions) based on the explicit geometry. Thus, ISO 10303-111 is closer to current commercial CAD systems (e.g. the features in Part 111 are quite similar to the features defined in commercial CAD systems) and can be implemented more easily. The major problems for Part 111 are: 1) the number of versions and final stages are recorded, but the middle stages can not be stored, and therefore it is still difficult to describe the design process and rationale clearly; 2) features are application dependent — the features defined in Part 111 are based on geometric models and topology — so multiple viewpoints are a problem; and 3) Part 111 displays the modelling history by recording all constructional operations, so the memory requirements and processing speed of applications suffer if the component is complicated.

Since 1999, STEP has extended its scope from the product design phase to additional life-cycle phases, such as maintenance and repair, including AP239 (Product Life Cycle Support [PLCS]) and AP233 (Systems engineering data representation) [10, 11]. AP239 aims to support the exchange and sharing of product information throughout the product lifecycle. It extends the capabilities of AP203 and AP214 to cover the full product life and addresses the complete product support domain based on a single integrated information model. AP239 covers four key areas: support engineering, resource management, configuration management, and maintenance and feedback [12, 13]. AP239 is independent of specific processes so that it can be flexibly tailored according to different industry requirements. The implementation of AP239, however, is still a big challenge. AP233 was proposed aiming to support the exchange and sharing of systems engineering data [14, 15]. Although still under development, it

focuses on analysis and test phases of product development.

3.2 Problems with STEP

STEP has led to improvements in exchange and sharing of simple CAD information, product models, and complete product structures. Furthermore, STEP has improved communications within the extended enterprise (including suppliers, business partners, and customers) and helped to support global collaborations. However, due to the bulk of its documentation and its complexity, there are still some problems that hinder the application of STEP in industry.

3.2.1 Application Protocol interoperability

Application Protocol interoperability is the ability to exchange interpretable data between two (or more) IT applications which are based on different APs, i.e. to have a meaningful communication between IT applications. Each AP is developed to exchange/store data for one application domain/industry, which results in two problems: APs in STEP are too big; and no monitoring enforces AP teams to investigate the relationship of their own AP to any other AP, so that APs have too much overlap with each other. Some work has been done towards AP interoperability, including the development of a 'core model' for interoperability between pairs of APs, bi-directional mapping of 'semantic counterparts', the development of Application Modules, and the preparation of new editions of older APs, but the extra work required is considerable and increases the cost of the implementations, in terms of both time and money.

3.2.2 User-defined constraints

EXPRESS is good at describing the STEP information model, but constraints defined in EXPRESS are generated at the time of AP publication and not at the time of data creation. This is a significant limitation of EXPRESS, as many constraints are identified and captured during project planning and the early design stage.

3.2.3 Implementation process

It is not easy to implement STEP because: 1) STEP covers the whole product lifecycle so that the potential volume of software objects is extensive; 2) the development of STEP translators is time-consuming, and therefore costly; 3) the file sizes that will be transferred and created will be large, especially in the absence of installed SDAIs; 4) it may not be economically viable for older files to be converted to the STEP neutral file format; and 5) members in the extended enterprise lack STEP knowledge, which may slow down the implementation and result in higher training costs.

4 Lightweight representations

The initial objective of a product model is to represent the design and engineering aspects of a product during the configuration design, assembly design or detail design stages. During the last forty years, a large volume of research has focused on product models and several approaches have been proposed [16]: boundary representations (B-Rep) that represent shapes using limits such as connecting faces, edges and vertices [17]; surface models that represent a part by specifying some or all of the surfaces using functions or approximations such as non-uniform rational B-splines (NURBS) or Bezier surfaces; feature-based models or parametric-based models that capture the engineering significance of parts with a B-Rep or constructive solid geometry (CSG) using features instead of the underlying geometry. Currently, most CAD systems implement a hybrid-modelling strategy combining the best features of the various approaches; however, the models produced are usually proprietary to the specific product development system and unable to meet the requirements of PLM (see section 2).

The ideal representation for PLM would be one that: 1) is computer-interpretable and considers the requirements of the whole product lifecycle; 2) supports seamless information and knowledge exchange between functional components in PLM that are heterogeneous in programming language or representation model; and 3) supports geographically distributed applications and users by allowing information and knowledge to be transmitted over the Internet. The major efforts that have been made to develop such a representation focus on two aspects: lifecycle representations combined with markup languages, and lightweight 3D visualizations.

4.1 Lifecycle representations for the lifecycle

Markup languages combine text with extra information expressed by markup (e.g. the text's structure or presentation) [18]. Due to their many advantages, such as platform/application independence and machine-interpretability, markup languages are regarded as one of the important future computing approaches and have been widely applied to PLM.

4.1.1 **XML** [19, 20]

XML is a generic markup language that allows users to define their own tags (i.e. labels that mark portions of text as having special significance) based on the specific needs of a document. XML is extensible — schemata of tags can be plugged in as necessary — and represents a good compromise between being human-readable and computer-interpretable; thus it has been actively adopted. As a generic technology, it does not represent product information and knowledge natively, but instead provides a language in which representa-

tions, such as those incorporated into some existing product data exchange standards and schemata, can be written.

4.1.2 EXPRESS/XML [21]

Some efforts have been made to convey EXPRESS, the modelling language of STEP [22], in an XML format. The result is EXPRESS/XML. In general, there are two ways to implement the exchange between EXPRESS and XML: late binding (with XML markup independent of any particular EXPRESS schema) and early binding (with XML markup based on a particular EXPRESS schema). The late binding describes entities and attributes explicitly, and therefore it can define the data for any use. The early binding has a more economical structure, but is ill-suited to XML applications involving multiple EXPRESS information models. However, in comparison with late binding, early binding is less verbose and simpler to process, so it is still preferred in the EXPRESS user community. EXPRESS/XML develops a strategy for STEP in the context of XML and provides a simple way to describe product data for the Web. Thus, it is potentially very useful for collaborative partners using the Web to support product development.

4.1.3 PLM XML [23]

PLM XML is an XML-based PLM format created and supported by the US CAD/PLM company UGS. Through defining a set of XML schemata, PLM XML aims to integrate collaborative product lifecycle processes by offering a standardized protocol for data interoperability. The categories of information currently supported by PLM XML include: product structure, metadata, geometric representation data, feature descriptions, data ownership, visualization properties, application associability, and delta (difference) information. PLM XML can help with the integration and interoperability of application systems in PLM due to its simplicity, extensibility, support of multiple representations for shape definition, and incorporation of product, part, and process information. PLM XML has been used extensively in UGS applications; for example, Teamcenter products are able to communicate with other applications by externally generated PLM XML files, and UGS use PLM XML in their internal translator development [24].

4.2 Lightweight 3D visualizations

For PLM to support a collaborative environment, product representations of different degrees of complexity are needed so that systems can support users rapidly browsing, retrieving and manipulating a product model over the Internet. In recent years, some lightweight 3D visualizations for product models have been developed and applied in PLM.

4.2.1 Universal 3D (U3D) [25, 26]

U3D is a lightweight 3D graphics format intended to efficiently distribute 3D data on the Web and in applications. In order to reduce the U3D file size for quick Internet downloading and fast rendering on screen, most of the engineering data associated with the original model is eliminated. Additionally, the architecture of U3D is such that multiple nodes may use the same resource, further reducing the U3D file size. U3D provides a way to access and reuse 3D data in downstream applications, such as marketing, product documentation, sales, support, and customer service. U3D is supported natively within Portable Document Format (PDF) version 1.6 and higher.

4.2.2 HOOPS Stream Format (HSF) [27]

HSF is a proprietary 2D and 3D visualization format whose specifications have been made freely available through the OpenHSF Initiative. It is optimized for collaboration over networks, with significant compression and streaming capabilities, although it only handles tessellated geometry. The format can encode product information such as assembly structure, analysis data and object behaviours, as well as custom data, and intelligently associate it with the geometry. HSF is supported by a number of major CAD vendors, and a customized version of it is used in Design Web Format (DWF) under the name W3D Stream format [28]. A number of free tools are available for viewing HSF files and embedding them within documents produced by office productivity software.

4.2.3 XGL/ZGL [29]

XGL is a file format for visualizing 3D information. It is an XML-based encoding of OpenGL (Open Graphics Library), a cross-platform application programming interface (API) for the rendering of 2D and 3D computer graphics [30]. XGL supports several features to reduce file sizes; one of these is referencing, which allows different parts of a file to share the same data, and allows an XGL reader to recognize that two objects are the same. Another is compact XML syntax: a vector in XGL is expressed as a single comma-delimited list of numbers rather than as a series of child elements each representing a dimension. XGL is supported by a number of converters [29, 31] and in its compressed form, ZGL, may be used within a DWF file [28]. It can be applied for various visualization applications, such as CAD/CAM, Web sites, and gaming. In comparison with other XML-based 3D formats, however, the development of XGL has been slow, and due to its optimization for display, it lacks support for useful features such as non-uniform or off-axis scaling, non-triangular meshes and NURBS [32].

4.2.4 **X3D** [33, 34, 35]

X3D is a major upgrade from VRML (the Virtual Reality Modelling Language, a standard file format for representing 3D interactive vector graphics) and retains backwards compatibility with a huge base of available 3D content modelled using VRML. X3D incorporates numerous advanced 3D techniques including advanced rendering and multi-texturing, NURBS surfaces, GeoSpatial referencing, Humanoid Animation (H-Anim) and IEEE Distributed Interactive Simulation (DIS) networking. X3D is more lightweight than VRML and its nodes are represented in XML tags so as to take full advantage of the potential of XML on the Internet. In addition, X3D utilizes an open profile/components-based architecture enabling customizing of implementations.

4.2.5 **3D XML** [36]

3D XML is a lightweight and standard XML-based format that represents product graphics using NURBS-like freeform surfaces rather than tessellating polygons, and communicates product geometry, product structure, and graphical display properties using an XML schema. Due to its compact method of encoding surfaces, 3D XML can speed up product data transporting and improve the sharing of 3D product data. 3D XML can already be used to express real-time 3D applications with complex interactivity [37], and can be embedded within office productivity software and a popular web browser [38].

4.2.6 **JT Format** [39, 40]

JT is a 3D product visualization data format. It uses a combination of facet and B-Rep geometry along with Product and Manufacturing Information (PMI) and textual attributes [39]. JT supports a hierarchical product structure with assembly, sub-assembly, parts and instances; it is compressed and allows model data to be split across multiple files. JT format may be considered as a de facto standard and has been widely used in the automotive, aerospace and various manufacturing industries. A C++ library, JT Open Toolkit, has been developed; it is able to create, read and access JT formatted data. The Toolkit can be executed on various hardware and operating systems, such as Windows, Solaris, HP-UX, SGI Irix and AIX.

5 A combined approach

We have seen in sections 3 and 4 that, for some purposes at least, the problem of accessing and re-using product data over the product lifecycle can be eased by the use of open, lightweight formats which map easily onto other formats. By definition, though, lightweight formats cannot represent the full complexity of data possible in heavyweight formats. Thus, when considering the use

of lightweight formats, two questions present themselves: which lightweight formats can be generated from a given heavyweight format (within given parameters of cost and reliability), and which of these lightweight formats is best for a given purpose? These questions are relatively easy to answer, in principle, given a sufficient quantity and quality of *representation information*.

5.1 Representation information

Representation information is a term that originates in the Open Archival Information System (OAIS) Reference Model [41]. The OAIS Reference Model was developed by the Consultative Committee for Space Data Systems (CCSDS) as a first step towards generating formal standards for the reliable archiving of Space Science data. It is intended to provide a common point of reference when discussing archival information systems, rather than to recommend any particular implementation [42].

The Model is set in the context of Producers (who generate the information to be archived), Consumers (who retrieve the information) and Management (the wider organization hosting the OAIS). The term 'Open[1] Archival Information System' is defined as a repository or archive, 'consisting of an organization of people and systems, that has accepted the responsibility to preserve information and make it available for a Designated Community,' the latter being 'an identified group of potential Consumers who should be able to understand a particular set of information.' [41, §1.7.2]

In the terms of the Model, an *Information Object* is a piece of knowledge in an exchange format, manifested physically by a *Data Object* (a bitstream, a string of printed letters, etc.). A person extracts and understands information using knowledge from their *Knowledge Base* (e.g. the ability to read English); where this Knowledge Base is insufficient, some *Representation Information* that bridges the gap is required (e.g. a dictionary). On a practical level, the Model envisages a partnership between an OAIS and its Designated Community such that the Designated Community commits to maintaining a certain Knowledge Base among its members, and the OAIS commits to providing sufficient Representation Information for that Knowledge Base.

While the Model indicates that representation information should be logically encapsulated with the data object to which it refers, there is no requirement for this encapsulation to be literal; indeed, it anticipates circumstances where literal encapsulation would not be appropriate. For example, the representation information associated with a digital data object in respect of its format would be equally applicable to other data objects in the same format. Thus it would be more efficient for a repository to hold this representation information just once and refer to it from the metadata associated with data objects; in this way, the same piece of representation information can be part of

[1] The adjective 'open' is used, somewhat awkwardly, to refer to the manner in which the OAIS Reference Model was developed, rather than to describe OAISes.

several information packages. The phenomenon of pieces of representation information calling other pieces of representation information — either directly as in this case, or indirectly, as when a data object containing representation information requires representation information of its own — is referred to by the Model as a *Representation Information Network*.

5.2 Representation information registries

When representation information is kept in a separate store, for the purpose of long-term reference, such a store is known as a representation information registry. Registries are useful for deduplicating information and effort, not only within repositories but across repositories as well [43].

There are several projects underway to build representation information registries of various descriptions. In the UK, The National Archives (TNA) are developing the PRONOM format registry [44]. PRONOM is primarily intended to support TNA's own preservation work, but since February 2004 it has been available for others to consult through TNA's website. The registry contains information on over a hundred formats [45], and is used both to generate human-readable web pages and as the back end for automated tools such as DROID (Digital Record Object Identification), a batch file format identification tool [46]. Further developments will include a mechanism for resolving PRONOM Persistent Unique Identifiers so they can be used to retrieve representation information over the Internet [47], full object characterization (including validation and metadata extraction) in DROID, and tools for determining appropriate migration pathways [48].

The Library of Congress runs a simple format registry presented as part of a Web resource entitled *Sustainability of Digital Formats: Planning for Library of Congress Collections*, or *Digital Formats* for short [49]. The primary purpose of this registry is to provide the information necessary for determining the most appropriate format for depositing digital materials into the Library. Thus, the registry gives detailed information as to the sustainability, functionality and quality of various formats; on the other hand, the information is only available as a set of web pages and cannot be interrogated by an automated process.

Two other notable registries are in the process of being developed. The *Global Digital Format Registry* (GDFR), developed by Harvard University Library in partnership with OCLC, is intended to be a network of linked registries sharing representation information [50, 51, 52]. The GDFR will support both human and automated retrieval of information, and will allow the deposit of proprietary format documentation under escrow-type arrangements. The Digital Curation Centre (DCC) is constructing a *Representation Information Registry/Repository* (RI RegRep) [53]; as well as collecting format-specific information, the RI RegRep will collect rendering software and will also include instrument calibrations, data units and other information necessary to interpret e-Science datasets [54]. In due course the RI RegRep will integrate into the preser-

vation framework constructed by the CASPAR project, a European Union initiative aiming to ensure the preservation of cultural, artistic and scientific knowledge for access and retrieval into the future [55].

On the repository side, research is also being conducted into using representation information registries and associated tools to generate migration pathways automatically. The University of Minho is developing a prototype service-orientated architecture that uses a Format Detector and Format Evaluator to determine the current and optimum formats respectively for a given digital object, a Migration Advisor to generate optimum migration pathways, a Migration Broker to perform the migration and an Object Evaluator to perform quality assurance tests on the migrated objects and feed this information back to the Migration Advisor [56].

5.3 Using representation information registries

There are several practical uses to which a representation information registry can be put. Indeed, the GDFR project gathered approximately thirty use cases from different repositories detailing the ways they would expect to use a file format registry [51]. These use cases fell into six different categories: *identifying* or *validating* the format of a file; *looking up the characteristics* of a format, for example to identify automatic metadata extraction techniques; *assessing the risks* associated with a format, in particular whether the format is in danger of becoming obsolete; and *determining the optimum migration path*, either from the original format to a display format ('*delivery*'), or from the original format to a similarly functional format ('*transformation*'); the broad category of migration paths here does not exclude the possibility of using emulators.

Of these six categories of use case, the most pertinent from an engineering perspective are looking up the characteristics of individual file formats — for example, to determine the best format for communicating designs to service engineers — and determining migration paths. In both cases, one of the most important issues to consider is the likely data loss that would occur as a result of migrating the data to the new format. To take a simple example from common formats, there are a number of options for migrating complex word processor documents, besides converting to similar binary formats: Rich Text Format is ASCII-based and can express most aspects of a Microsoft Word document, notably excepting VBA scripting, Word user interface customization, document versions and some metadata fields [57]; simple HTML + CSS can express most of the structure and formatting of a document;[2] while plain text expresses only the textual content without any formatting. Depending on the circumstances, some data loss may be desirable and some may be unacceptable. With CAD/CAM/CAE files, there will be circumstances in which all the data needs to be retained (such as for archival purposes) and circumstances in

[2] XML conforming to the DocBook DTD [58] or Text Encoding Initiative DTD [59] would almost certainly be preferable to HTML + CSS for archival purposes, if not for display [60].

which full fidelity would compromise the organization's intellectual property (such as marketing material).

Thus the representation information we are concentrating on is that which would support these types of registry use cases. Specifically, we are considering file format references and specifications (where available), the kinds of information that can be stored by a format, any migration or emulation tools that can be used to transform files to or from it, the characteristics of such tools and the specifications of their APIs (if any).

While a representation information registry is not a solution in itself, it does have the potential to be a useful source of information driving the preservation planning functions of an engineering repository.

5.4 Collecting representation information

'Immortal information and through-life knowledge management (KIM): strategies and tools for the emerging product-service business paradigm' is an Engineering and Physical Sciences Research Council (EPSRC) Grand Challenge project involving eleven different UK universities and incorporating substantial industry collaboration. It is investigating a range of issues associated with the move towards a product-service paradigm in the engineering sector [61, 62], in particular the long-term curation of digital data, learning from production and use, and appropriate governance and management techniques. One of the elements of the research programme is to investigate the possibility of a representation information registry for engineering-specific file formats, and to explore the practical limits of its usefulness. We envision using a representation information registry, both as a decision-making tool for assessing migration/emulation options, and where possible as a reference tool for looking up the characteristics of various file formats. This would enable firms to take a flexible, needs-based approach to curating their data.

The representation information being collected for the prototype registry consists of two main types: format reference documentation and specifications, and information on migration pathways. Since information of the second kind is only infrequently available in the detail required, an initial investigation was carried out to generate and codify some for the registry.

For this initial investigation, the source file chosen was a model of a layshaft produced in UGS Solid Edge v16 (SE). This was migrated using SE and Adobe Acrobat 3D v7.0.9 in the following ways:

- Export from SE v16 to PLM XML using precise geometry
- Export from SE v16 to PLM XML using triangular facets
- Export from SE v16 to ACIS .sat
  - Import from ACIS .sat to U3D using Acrobat 3D Toolkit then embedded in PDF v1.6

  - Import from ACIS .sat to Right Hemisphere (RH) format using Acrobat 3D Toolkit, then imported from RH to U3D embedded in PDF v1.6

- Export from SE v16 to NX3
  - Import from NX3 to U3D embedded in PDF v1.6
- Export from SE v16 to IGES
  - Import from IGES to U3D embedded in PDF v1.6
- Export from SE v16 to STEP
  - Import from STEP to U3D embedded in PDF v1.6
- Export from SE v19 to JT with the following option sets:[3]
  1. including precise geometry but not translating visible constructions, and splitting the JT document into assembly and part documents
  2. not including precise geometry and not translating visible constructions, and splitting the JT document into assembly and part documents
  3. including precise geometry and translating visible constructions, and splitting the JT document into assembly, sub-assembly and part documents
- Export from SE v19 to Catia v4

The migrated files were viewed — using Adobe Reader v7 for PDF, NX3 for NX3 and JT formats, and Solid Edge v19 for the other formats — and checked to ascertain whether:

- the parts visually resembled those of the original model;
- the parts were assembled correctly within the model;
- the logical hierarchy of parts and sub-assemblies was preserved;
- the dimensions of the parts were accurately preserved;
- the file size was reduced.

[3]The options common to all three migrations were: exporting both part and assembly documents; not only updating modified parts; using a simplified body when available for the top-level assembly, sub-assemblies and parts; removing unsafe characters from the JT document name; translating visible parts only; not translating inter-part copies as constructions; translating Solid Edge document properties; deleting unused JT part documents; not saving PMI data; and using metres as the base unit of measure.

Model	Assembly correct	Model tree correct	Length /mm	File size /kB
Solid Edge [original]	Yes	Yes	1178.18	3 319
PLM XML (precise)	Yes	Yes	1178.18	619
PLM XML (triangles)	Yes	Yes	1178.18	734
ACIS .sat	Yes	No	1178.18	957
PDF (ACIS .sat)	Yes	No	1178.1753[a]	1 743
PDF (ACIS .sat via RH)	Yes	No	1178.1753[a]	939
NX3	Yes	Yes	1178.18	2 096
PDF (NX3)	Yes	Yes	1178.1750[a]	911
IGES	Yes	Yes	1178.18	5 069
PDF (IGES)	Yes	Yes	46.3848[a]	637
STEP	Yes	Yes	1178.18	981
PDF (STEP)	Yes	Yes	1178.1750[a]	816
JT (option set 1)	Yes	No	1178.15	390
JT (option set 2)	Yes	No	1178.18	184
JT (option set 3, attempt 1)	No	No	1178	313
JT (option set 3, attempt 2)	Yes	No	1178	391
Catia v4	Yes	No	1178.175	10 050

[a] Model units; unless the correct units are specified at conversion time, these are interpreted as metres. See text for explanation.

Table 1: Comparison of different migration routes for a Solid Edge CAD model

The results of the investigation are summarized in table 1. There were no discrepancies found in the geometry of the migrated parts, in any of the tests. The parts were assembled correctly in each case, except for the third JT migration in which one of the parts was lost (although, on repeating the migration, the fault did not occur). The logical significance of the sub-assemblies in the model was lost on migration to ACIS .sat format and JT format. While the IGES file preserved the logical significance of the sub-assemblies, it also treated each part as a sub-assembly of faces.

Variations in the measuring precision of the different viewers makes direct comparison of the length of the layshaft in each model difficult, but to the nearest hundredth of a millimetre, there was no variation in length among the majority of the models. The apparent discrepancy in the case of the IGES file imported into U3D/PDF was due to a characteristic of the importer in Acrobat, which interprets all model units as metres unless the correct units are supplied as an option of the conversion process. The IGES file was the only model to have base units of inches; the others used millimetres. There was some difficulty measuring the JT files in NX3, but the dimensions appeared to be correct to tenths of a millimetre. One other interesting result was that the PDFs produced from the ACIS .sat file contained models 0.3 µm longer than those in the PDFs produced from the NX3 and STEP files.

On the matter of file sizes, it should be borne in mind that the relative effectiveness of the different compression methods used cannot be judged on the basis of two tools migrating a single, simple model, especially when the methods of approximating the geometry with facets cannot be precisely aligned. Having said that, a few broad trends can be seen. The two migrations that caused an increase in file size were to IGES and Catia. The latter should not be surprising, given that Catia is another heavyweight CAD format, although the scale of the increase is noteworthy. Among the remaining migrations, two other points stand out: firstly, that when importing models into PDF via the Acrobat toolkit, the resulting file was significantly better compressed when Right Hemisphere format was used as the intermediate format; and secondly, the JT format files appeared to be significantly better compressed than the other lightweight formats.

5.5 Re-using representation information

The information just presented may be useful as part of a larger collection of representation information, but it is hard to re-use in this format. To this end two XML schemata have been devised to encode the information in a machine readable format: one for file formats and one for software tools. In essence, the file format documents identify a `format` and list its `features`. The software tool documents identify a `converter` which performs one or more `conversions`, each of which is identified by the `source` format, the `destination` format and the conversion `options`, and which is described as having certain `features`. In

```
<?xml version="1.0" encoding="UTF-8"?>
<converter
  xmlns="http://www.ukoln.ac.uk/projects/grand-challenge/
    conv-issues.rnc"
  toolname="Adobe Acrobat 3D"
  toolid="urn:uuid:76887daf-9ee7-59b2-90bd-8617435f2cf7"
  version="7.0.9">
  <conversion>
    <source>urn:uuid:1c223183-239d-540a-35a5-
      6948a1ea4e97</source>
    <destination>urn:uuid:2df086ec-f5d4-a362-f776-
      af24c39bf041</destination>
    <options>
      <option key="Collapse hierarchy to" value="N" />
    </options>
    <features>
      <feature property="NURBS surface geometry"
        preservation="none" degradation="configurable" />
      <feature property="geometric dimensioning"
        preservation="fair">
        <comment xml:lang="en-GB">Acrobat assumes metres as the
          model units (by default) regardless of which units are
          specified in the file. The correct units must be
          specified in the conversion options.</comment>
      </feature>
      <feature property="assembly hierarchy" preservation="good">
        <comment xml:lang="en-GB">The assembly tree may subdivide
          parts into faces.</comment>
      </feature>
    </features>
</converter>
```

Figure 1: Sample representation information document. The identifiers (of form `urn:uuid:`*number*) refer to registry objects within the DCC RI RegRep.

both cases, the names and properties of the `features` are drawn from a controlled vocabulary to aid retrieval. A sample document, describing the transformation from IGES to PDF v1.6, is shown in Fig. 1. This document has been entered into the DCC RI RegRep as a piece of representation information describing processing software, and given its own unique identifier.

Representation information encoded in this form allows the decision-maker to answer the following queries using a representation information registry.

1. *Which file formats can support a given set of functional requirements?* For this query, a search script parses each piece of representation information in the registry written in the XML schema for describing engineering file formats. The script looks for each requested functional requirement under the `property` attribute of a `feature` element, alongside a `support`

attribute with a value of either '`good`' or '`partial`'. In each case where all the requested functional requirements are so described, the search script returns the details of file format being described.

2. *What are the properties of a given file format?* For this query, the search engine simply looks up the representation information document associated with the file format, parses the contents and returns all the information encoded in the `feature` and `comment` elements.

3. *What migration paths exist between two given formats?* For this query, a recursive method is required; a maximum number of levels of recursion must be set, representing the maximum number of migration stages allowed. The first step in the recursive cycle is a search script that parses each piece of representation information in the registry written in the XML schema for describing engineering processing software. Any `converter` with the correct (current) source format listed among its `conversions` is returned into an nth-order result set. Any `converter` in that set with a `conversion` having both the correct destination format and the correct (current) source format is returned into a set of direct paths for that pair of formats, then into the nth order set of possible paths, prepended by any path saved from the $(n-1)$th-order (parent) cycle. If n is equal to the maximum recursion level, the cycle then terminates. Otherwise, for each remaining `converter`, the supported `destination` formats are enumerated. For each destination format, a check is performed for saved sets of paths between that format and the correct destination format; where such a set exists, each path in the set is added to $(n+1)$th-order set of possible paths, prepended by any path saved from the $(n-1)$th-order cycle, plus the current `converter` and the current intermediate format; otherwise, the current `converter` and current format are appended to the path saved from the $(n-1)$th cycle (or saved anew in the case of the first cycle) and an $(n+1)$th-order (child) cycle is performed with the current format as the source format. Following completion of the $(n+1)$th-order cycle, the current `converter` and current format are removed from the end of the saved path, and the nth-order cycle continues until all `converters` in the cycle's result set have been processed, at which point the cycle terminates.

4. *What are the characteristics of a given migration?* In this case, the search engine simply looks up the representation information document associated with the `conversion`, and returns all the information encoded in the `feature` and `comment` elements.

A tool for conducting these queries is being developed as part of the KIM Project, and is due to be delivered by summer 2008.

6 Discussion

The use of a representation information registry as a tool for making decisions about the right formats for archival and exchange purposes has several advantages. The primary advantage concerns the re-use of information. Whenever organizations acquire new or replacement pieces of software, common sense dictates that they pay at least some regard to how that software deals with legacy data and documents, and how it interoperates with the other software in place. The results of any research they perform into data exchange and interoperability would most likely be summarized at a fairly high level and recorded in text documents, with much of the detail omitted. Under the scheme outlined in this paper, the detail of the results would be encoded in machine-readable format and placed in a registry, allowing the information to be mined and reused in future, and permitting a greater return on the investment in creating the information. The possibility of automating queries is another notable advantage over a purely text-based approach.

The value of the registry, of course, depends heavily on the depth and breadth of information it contains, and this raises a few issues beyond the purely technical. On the issue of depth, the granularity of information encoded in the XML representation information documents is an aspect that requires careful tuning; this is currently being refined in consultation with the KIM Project's industrial collaborators. On the issue of breadth, registries would undoubtedly benefit from supplementing their data with data from other registries, but the case for facilitating this by sharing data with other registries (whether directly or through a neutral, third-party registry such as the DCC RI RegRep) would need to be explicitly considered when designing a real-world implementation.

7 Conclusion

This paper has outlined some of the major problems facing engineering firms due to software and platform revisions. Several sets of solutions have been identified, and while each approach has its advantages, none are entirely free of drawbacks. Even STEP, the most comprehensive exchange format, has serious shortcomings, not least the lack of interoperability between Application Protocols, the inability of EXPRESS to convey user-defined constraints and the difficulties associated with implementing STEP.

The proposed solution is to construct a representation information registry that can be used to determine the least unsatisfactory migration path between two known formats, and the least unsatisfactory destination format for a migration given a set of desirable characteristics. The University of Bath is constructing such a registry as part of the EPSRC Grand Challenge Project 'Immortal Information and Through Life Knowledge Management', in consultation with

a set of industrial collaborators, and the results will be incorporated into the Digital Curation Centre's Representation Information Registry/Repository. A demonstrator will be produced to test the implementation.

8 Acknowledgements

This work is supported by the UK Engineering and Physical Sciences Research Council (EPSRC) and the Economic and Social Research Council (ESRC) under Grant Numbers EP/C534220/1 and RES-331-27-0006.

References

[1] ISO 10303. 'Industrial automation systems and integration – Product data representation and exchange.' Multipart standard.

[2] ISO/TS 10303-203:2005. 'Industrial automation systems and integration – Product data representation and exchange – Part 203: Application protocol: Configuration controlled 3D design of mechanical parts and assemblies (modular version).'

[3] ISO 10303-204:2002. 'Industrial automation systems and integration – Product data representation and exchange – Part 204: Application protocol: Mechanical design using boundary representation.'

[4] ISO 10303-214:2003. 'Industrial automation systems and integration – Product data representation and exchange – Part 214: Application protocol: Core data for automotive mechanical design processes.'

[5] ISO 10303-224:2001. 'Industrial automation systems and integration – Product data representation and exchange – Part 224: Application protocol: Mechanical product definition for process planning using machining features.'

[6] ISO 10303-55:2005. 'Industrial automation systems and integration – Product data representation and exchange – Part 55: Integrated generic resource: Procedural and hybrid representation.'

[7] Michael J. Pratt, Bill D. Anderson, and Tony Ranger. 'Towards the standardized exchange of parameterized feature-based CAD models.' *Computer-Aided Design* 37(12): 1251–65, 2005. URL `http://dx.doi.org/10.1016/j.cad.2004.12.005`.

[8] ISO 10303-108:2005. 'Industrial automation systems and integration – Product data representation and exchange – Part 108: Integrated application resource: Parameterization and constraints for explicit geometric product models.'

[9] ISO/DIS 10303-111. 'Industrial automation systems and integration – Product data representation and exchange – Part 111: Integrated application resource: Elements for the procedural modelling of solid shapes.' Under development.

[10] ISO 10303-239:2005. 'Industrial automation systems and integration – Product data representation and exchange – Part 239: Application protocol: Product life cycle support.'

[11] ISO/WD 10303-233. 'Industrial automation systems and integration – Product data representation and exchange – Part 233: Application protocol: Systems engineering data representation.' Under development.

[12] PLCS Inc. 'Product life cycle support: Frequently asked questions.', 2002. URL http://www.ams.mod.uk/content/docs/ils/ils_web/plcs/plcs_faq.htm.

[13] John Dunford. *Validation report for ISO/DIS 10303-239, STEP Part 239, Application protocol: Product life cycle support.* WG3 N1451, ISO TC184/SC4, 2004. URL http://www.tc184-sc4.org/SC4_Open/SC4_and_Working_Groups/WG3/N-DOCS/Files/wg3n1451_PLCS_804_AP239%20DIS%20Validation%20Reportv1.2.doc.

[14] Roland Eckert. 'The STEP systems engineering project AP233.' Presentation, 2005.

[15] SEDRES-2. 'Systems Engineering Data Representation and Exchange Standardisation-2.' Brochure, 2001. Information Society Technologies Project IST-1999-11953, URL http://www.sia-av.it/newsite/brochure/SEDRES-2.pdf.

[16] Chris McMahon and Jimmie Browne. *CADCAM: Principles, Practice and Manufacturing Management.* Harlow: Addison-Wesley, 2nd ed., 1998. ISBN 0-201-1781-9.

[17] Ian C. Braid. 'Designing with volumes.' Ph.D. thesis, Cambridge University, 1974.

[18] Wikipedia. 'Markup language.', 2006. 15 Jun., URL http://en.wikipedia.org/wiki/Markup_language.

[19] W3C. 'Extensible markup language (XML) 1.1.', 2004. URL http://www.w3.org/TR/xml11/.

[20] W3Schools. 'XML tutorial.', 2006. URL http://www.w3schools.com/xml/.

[21] ISO/TS 10303-28:2003. 'Industrial automation systems and integration – Product data representation and exchange – Part 28: Implementation methods: XML representations of EXPRESS schemas and data.'

[22] ISO 10303-11:2004. 'Industrial automation systems and integration – Product data representation and exchange – Part 11: Description methods: The EXPRESS language reference manual.'

[23] UGS. 'Open product lifecycle data sharing using XML.' White Paper, 2005. URL http://www.ugs.com/products/open/plmxml/docs/wp_plm_xml_14.pdf.

[24] ———. 'Real-time engineering collaboration: Using web-based communities to collaborate throughout the product lifecycle.' White Paper, 2006. URL http://www.ugs.com/products/teamcenter/docs/wp_community_collaboration.pdf.

[25] ECMA-363. 'Universal 3D file format.', 2007. 4th edition, URL http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-363%204th%20Edition.pdf.

[26] 'U3D.' *Computer Desktop Encyclopedia*, Computer Language Company. 2005. URL http://www.answers.com/topic/u3d.

[27] Open HSF Initiative. 'The HOOPS 3D product suite.' URL http://www.openhsf.org/docs_hsf/index.html.

[28] Autodesk. 'DWF 6 specification.', 2003. Part of the Autodesk DWF Toolkit 7.3.

[29] XGL Working Group. 'XGL file format specification.', 2006. URL http://web.archive.org/web/20060218/http://www.xglspec.org/.

[30] Wikipedia. 'OpenGL.', 2007. 2 Jul., URL http://en.wikipedia.org/wiki/OpenGL.

[31] Ron LaFon. '3D without boundaries: New tools to publish and share designs.' *Cadalyst* 22(12): 18–29, 2005. ISSN 0820-5450. URL http://www.nxtbook.com/nxtbooks/questex/cadalyst1205/index.php?startpage=18.

[32] Okino Computer Graphics. 'XGL/ZGL exporter.', 2007. URL http://www.okino.com/conv/exp_xgl.htm.

[33] ISO/IEC 19775:2004. 'Information technology — Computer graphics and image processing — Extensible 3D (X3D).' URL http://www.web3d.org/x3d/specifications/ISO-IEC-19775-X3DAbstractSpecification/.

[34] ISO/IEC 19776:2005. 'Information technology — Computer graphics and image processing — Extensible 3D (X3D) encodings.' URL http://www.web3d.org/x3d/specifications/ISO-IEC-19776-X3DEncodings-XML-ClassicVRML/.

[35] ISO/IEC 19777:2006. 'Information technology — Computer graphics and image processing — Extensible 3D (X3D) language bindings.' URL http://www.web3d.org/x3d/specifications/ISO-IEC-19777-X3DLanguageBindings/.

[36] Ken Versprille. *Dassault Systèmes' Strategic Initiative: 3D XML for Sharing Product Information*. Technology Trends in PLM, Collaborative Product Development Associates, 2005. URL http://www.3ds.com/uploads/tx_user3dsplmxml/3DXML_for_sharing_product_information.pdf.

[37] Virtools. '3D XML Virtools™ plugin.' URL http://www.virtools.com/solutions/products/virtools_3dxml_plugin.asp.

[38] Dassault Systèmes. 'Dassault Systèmes delivers 3D XML specifications and player.', 2005. Press release, URL http://www.3ds.com/news-events/press-room/release/899/1/.

[39] Wikipedia. 'JT (visualization format).', 2006. 18 Mar., URL http://en.wikipedia.org/wiki/JT_(visualization_format).

[40] UGS. 'JT files.', 2005. URL http://www.jtopen.com/technology/jt_files_overview.html.

[41] CCSDS. *Reference Model for an Open Archival Information System (OAIS)*. Blue Book CCSDS 650.0-B-1, Consultative Committee for Space Data Systems, 2002. Also published as ISO 14721:2003, URL http://public.ccsds.org/publications/archive/650x0b1.pdf.

[42] Brian F. Lavoie. *The Open Archival Information System Reference Model: Introductory Guide*. DPC Technology Watch Series Report 04-01, Digital Preservation Coalition, 2004. URL http://www.dpconline.org/docs/lavoie_OAIS.pdf.

[43] Margaret Hedstrom and Seamus Ross, eds. *Invest to Save: Report and Recommendations of the NSF-DELOS Working Group on Digital Archiving and Preservation*. Network of Excellence for Digital Libraries (DELOS), 2003. URL http://delos-noe.iei.pi.cnr.it/activities/internationalforum/Joint-WGs/digitalarchiving/Digitalarchiving.pdf.

[44] Adrian Brown. *PRONOM 4 User Requirements*. Kew: The National Archives, 2004. URL http://www.nationalarchives.gov.uk/aboutapps/fileformat/pdf/pronom_4_user_reqs.pdf.

[45] ———. 'Automating preservation: New developments in the pronom service.' *RLG DigiNews* 9(2), 2005. ISSN 1093-5371. URL http://www.rlg.org/en/page.php?Page_ID=20571&Article_ID=1717.

[46] ———. *Automatic Format Identification Using PRONOM and DROID*. Digital Preservation Technical Paper 1, The National Archives, Kew, 2006. URL http://www.nationalarchives.gov.uk/aboutapps/fileformat/pdf/automatic_format_identification.pdf.

[47] ———. *The PRONOM PUID Scheme: A scheme of persistent unique identifiers for representation information*. Digital Preservation Technical Paper 2, The National Archives, Kew, 2005. URL http://www.nationalarchives.gov.uk/aboutapps/pronom/pdf/pronom_unique_identifier_scheme.pdf.

[48] Jeffrey Darlington. 'PRONOM: A practical online compendium of file formats.' *RLG DigiNews* 7(5), 2003. ISSN 1093-5371. URL http://www.rlg.org/preserv/diginews/v7_n5_feature2.html.

[49] Caroline R. Arms and Carl Fleischhauer. 'Digital formats: Factors for sustainability, functionality, and quality.' *IS&T Archiving Conference*. Washington, DC: Society for Imaging Science and Technology, 2005. 26–29 Apr., URL http://memory.loc.gov/ammem/techdocs/digform/Formats_IST05_paper.pdf.

[50] Stephen L. Abrams. 'Establishing a Global Digital Format Registry.' *Library Trends* 54(1): 125–43, 2005. ISSN 0024-2594.

[51] Stephen Abrams and Dale Flecker. 'A proposal for a Global Digital Format Registry.', 2005. URL http://hul.harvard.edu/gdfr/documents/Proposal-2005-09-29.doc.

[52] GDFR. 'Global Digital Format Registry data model v4.', 2004. URL http://hul.harvard.edu/gdfr/documents/DataModel-v4-2004-01-12.doc.

[53] David Giaretta. 'Representation information in the DCC Registry/Repository.' Digital Curation Centre, 2005. URL http://dev.dcc.rl.ac.uk/twiki/bin/view/Main/DCCRegRepV04.

[54] David Giaretta, et al. 'Supporting e-Research using representation information.' *Proceedings of the UK e-Science All Hands Meeting*. Nottingham: EPSRC, 2005. ISBN 1-904425-53-4. 19–22 Sep., URL http://www.allhands.org.uk/2005/proceedings/papers/447.pdf.

[55] David Giaretta. 'CASPAR and a European infrastructure for digital preservation.' *ERCIM News* (66): 47–9, 2006. URL http://www.ercim.org/publication/Ercim_News/enw66/giaretta.html.

[56] Miguel Ferreira, Ana Alice Baptista, and José Carlos Ramalho. 'A foundation for automatic digital preservation.' *Ariadne* (48), 2006. ISSN 1361-3200. URL http://www.ariadne.ac.uk/issue48/ferreira-et-al/.

[57] Microsoft Technical Support. *Microsoft Office Word 2003 Rich Text Format (RTF) Specification*. White paper, Microsoft Corporation, 2004. URL http://www.microsoft.com/downloads/details.aspx?familyid=ac57de32-17f0-4b46-9e4e-467ef9bc5540&displaylang=en.

[58] Norman Walsh and Leonard Muellner. *DocBook: The Definitive Guide*. Sabastopol, CA: O'Reilly, 1999. ISBN 1-56592-580-7. URL http://www.docbook.org/tdg/en/html/docbook.html.

[59] C. Michael Sperberg-McQueen and Lou Burnard, eds. *TEI P4: Guidelines for Electronic Text Encoding and Interchange*. Oxford: Text Encoding Initiative Consortium, 2002.

[60] Ian Barnes. *Preservation of Word Processing Documents*. Working paper, Australian Partnership for Sustainable Repositories, 2006. URL http://www.apsr.edu.au/publications/preservation_of_word_processing_documents.html.

[61] Andrew Davies, Tim Brady, and Puay Tang. *Delivering Integrated Solutions*. Brighton: SPRU/CENTRIM, 2003. ISBN 0-903622-98-X.

[62] Rogelio Oliva and Robert Kallenberg. 'Managing the transition from products to services.' *International Journal of Service Industry Management* 14(2): 160–72, 2003. ISSN 0956-4233.

All links were correct on 1st July 2007.

This article was submitted to Advanced Engineering Informatics on 25th July 2007.