An Approach to Accessing Product Data across System and Software Revisions

Alexander Ball[1], Lian Ding[2], and Manjula Patel[1]

[1]UKOLN, University of Bath, Bath, UK
[2]Department of Mechanical Engineering, University of Bath, Bath, UK

3rd October 2007

Abstract

Long-term users of engineering product data are hampered by 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 is used to identify suitable destination file formats for different purposes, and provides a resource to 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. In consequence, 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 scope for re-using the data remains severely limited, 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 of the advantages and disadvantages of STEP as an exchange and preservation format (section 3), and then identifies some candidate lightweight representations for inclusion 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 a demonstration of its use 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.

- Product life can be in the order of decades. This leads to two problems. Firstly, challenges of traceability and retrieval of information and knowledge during the product lifecycle need to be met such that files in older versions are accessed by the latest/future application tools. Secondly, during the product life, especially in a collaborative environment, changes occur many times across different situations and contexts. These changes further affect product content, metadata and configurations in files of various and varying format. In order to handle this evolution effectively throughout the whole product life, strategies are needed for updating, reporting and merging 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 [2–4]. 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) [5–8]. Although these parts have been widely applied in industry and the academic community — for example, AP203 delivers CAD data in a neutral format that is readable by most CAD systems — according to Pratt, Anderson and Ranger [9] the initial STEP parts have three potential 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) [10], providing basic mechanisms for procedural or construction history modelling [9]. 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) [11, 12]. Current commercial CAD systems adopt a hybrid representation including a procedural model, which describes the design history as a feature tree, and a set of parameters detailing how to instantiate a set of features (modelling functions) based on explicit geometry. Since 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) it 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) [13, 14]. AP239 aims to support the exchange of product information throughout the product lifecycle. It extends the capabilities of AP203 and AP214 to cover the entire lifecycle 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 [15, 16]. 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 challenge [17]. AP233 was proposed aiming to support the exchange and sharing of systems engineering data [18]. Although still under development, it focuses on analysis and test phases of product development.

3.2 Issues 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 issues that hinder the application of STEP in industry.

3.2.1 Application Protocol interoperability

In order for IT applications based on different Application Protocols to communicate meaningfully, they must be able to interpret the same data in the same way: the APs must be interoperable. Each AP is developed to exchange/store data for one application domain/industry, which results in two issues: APs in STEP are broad in scope; and the absence of any overall monitoring results in significant overlap between APs. To realise AP interoperability, the international standard ISO 18876 was published in 2003 [19]. Instead of using the initial STEP architecture, ISO 18876 adopts the newer modular architecture that breaks STEP into small, re-usable modules called Application Modules (AMs). AMs are reused by other AMs and ultimately in APs [20]. The advantages of the modular architecture have yet to be proven, however.

3.2.2 User-defined constraints

EXPRESS is the modelling language used to describe the data and information models used by the remaining parts of STEP [21]. One of the limitations of the language is that it provides no straightforward way to add additional constraints. This means that data written according to the information model of a given AP only declare the constraints envisaged when the AP was published, and do not make explicit any constraints that 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 to be created and transferred are large, especially in the absence of installed SDAIs [22];[1] 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. Product modelling has received much academic attention over the past forty years and several approaches have been proposed [24]: boundary representations (B-Rep) that represent shapes using limits such as connecting faces, edges and vertices [25]; 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 parameter-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 Lightweight representations for the lifecycle

Markup languages combine text with extra information expressed by markup (e.g. the text's structure or presentation) [26, 27]. 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[28].

[1] For example, using the *structural_design_schema*, an ASCII STEP file would need on average 2.2 kilobytes to represent the finite element data for a ten-node quadratic tetrahedron, and 0.25 kilobytes to represent finite element data for a three or four-node linear polygon [23].

4.1.1 **XML** [29]

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

4.1.2 **EXPRESS/XML** [30, 31]

Some efforts have been made to convey EXPRESS in an XML format. The result is EXPRESS/XML. In general, there are two ways to convey EXPRESS in 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 has been 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** [32]

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 [33].

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) [34]

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) [35]

HSF is a proprietary 2D and 3D visualization format whose specifications have been made freely available through the OpenHSF Initiative. Its compression and streaming capabilities make it well-suited for collaboration over networks, 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 [36]. 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 [37]

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 [38]. 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 [37, 39] and in its compressed form, ZGL, may be used within a DWF file [36]. 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.

4.2.4 X3D [40–42]

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 in order to simplify processing. In addition, X3D utilizes an open profile/components-based architecture enabling customizing of implementations.

4.2.5 **3D XML** [43]

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, and can be embedded within office productivity software and a popular web browser [44].

4.2.6 **JT Format** [31, 45]

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. 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 is being promoted 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

As shown in sections 3 and 4, the problem of accessing and re-using product data over the product lifecycle can be eased, for some purposes at least, 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 [46]. 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 [47].

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[2] 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.' [46, §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. In the case of product model data, examples of Representation Information might include CAD software, design house rules, data dictionaries or more sophisticated product information specifications such as those developed by Lee et al. [48]. 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 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 [49].

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 [50]. 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 [51], 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),

[2] The adjective 'open' is used, somewhat awkwardly, to refer to the manner in which the OAIS Reference Model was developed, rather than to imply a lack of access restriction.

a batch file format identification tool [52]. Further developments will include a mechanism for resolving PRONOM Persistent Unique Identifiers so they can be used to retrieve representation information over the Internet [53], full object characterization (including validation and metadata extraction) in DROID, and tools for determining appropriate migration pathways [54].

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 [55]. 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 [56–58]. 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 *Registry/Repository of Representation Information* (RRoRI); while the other three registries are concerned solely with format-specific information and software, RRoRI will also include the instrument calibrations, data units and other information necessary to interpret e-Science datasets, for example [59]. In due course RRoRI will integrate into the preservation 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 [60, 61].

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 [62]. A similar but less automated architecture, under the name Preservation Web Services Architecture for New Media and Interactive Collections (PANIC) is being developed by the University of Queensland [63].

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 [57]. 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 (‘*deliv-*

ery'), 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 [64]; simple HTML + CSS can express most of the structure and formatting of a document;[3] 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 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 [68, 69], 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.

[3] XML conforming to the DocBook DTD [65] or Text Encoding Initiative DTD [66] would almost certainly be preferable to HTML + CSS for archival purposes, if not for display [67].

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 embed in PDF v1.6
  - Import from ACIS .sat to Right Hemisphere (RH) format using Acrobat 3D Toolkit, then import 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:[4]
  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:

[4] 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 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.

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 has its uses 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`, by which is meant the kinds of information the format can (or cannot) record. 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`, by which is meant the kinds of information that the conversion can (or cannot) translate. In both cases, the names and properties of the `features` are drawn from a controlled vocabulary to aid retrieval. The `features` named have been drawn from a number of different CAD format and lightweight format specifications, and informed by conversations with KIM Project industrial collaborators.

In file format documents, formats are specified as having one of three possible levels of `support` for a `feature`: '`full`', '`none`' or '`partial`'; where only `partial` support is available, the way in which the support falls short of `full` is recorded in a textual note. In software tool documents, a conversion is specified as having one of four different levels of support ('`preservation`') for a `feature`: '`good`', where all such information is handled well enough that the conversion could form a stage in a perfect round-trip conversion; '`none`', where all such information is discarded; '`poor`', where with any file of realistic complexity, the degradation or corruption of the information is at least as likely as it surviving intact; or '`fair`', which indicates a state between '`good`' and '`poor`'. Any use of the '`preservation`' value of '`fair`' is explained with a textual note. Also recorded is whether the conversion can degrade the information related to a `feature`. A value of '`configurable`' indicates that options are available to control how the information is degraded; for example, this is true of conversions that translate exact geometry into polygon meshes according to a minimum level of accuracy set by the user. A value of '`fixed`' indicates that, while the degradation can-

```
<?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="info:foobar/sw/2" version=
  "7.0.9">
  <conversion source="info:foobar/f/1" destination="info:foobar/f/4">
    <options>
      <option key="Collapse hierarchy to" value="N" />
    </options>
    <features>
      <feature degradation="configurable" preservation="none" property="
        NURBS surface geometry" />
      <feature preservation="fair" property="geometric dimensioning">
        <comment xml:lang="en-GB">
          Acrobat assumes metres as the model units (by default) regardless
          of which units are specified in the file.
        </comment>
      </feature>
      <feature preservation="good" property="assembly hierarchy">
        <comment xml:lang="en-GB">
          The assembly tree may subdivide parts into faces.
        </comment>
      </feature>
    </features>
  </conversion>
</converter>
```

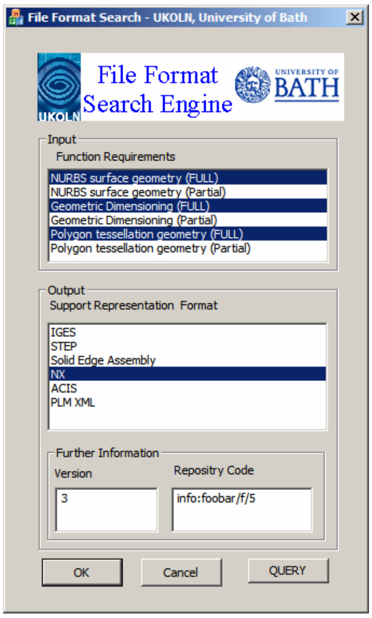
Figure 1 – Sample representation information document, describing the transformation from IGES to PDF v1.6. The identifiers (of form info:foobar/*type*/*number*) are dummy identifiers used within the standalone registry.

not be configured, the results can be reliably predicted; for example, this is true of a conversion that reduces rich text annotations to plain text annotations. If the conversion degrades information in an unpredictable fashion, the value 'unpredictable' is used. If the 'degradation' value is not set at all, then this implies the information is never degraded, but either preserved, corrupted or discarded, depending on the value of the preservation property.

A sample document, describing the transformation from IGES to PDF v1.6, is shown in Fig. 1.

As an initial proof of concept, a standalone registry has been constructed and populated with XML documents describing the results of the investigation in section 5.4. A prototype system for querying the registry has been implemented in C++. The main types of queries supported by the system are:

1. *Which file formats can support a given set of functional requirements?* Prior to running this query, the set of functional requirements is translated into a list of required and desirable file format properties. On running the 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 file format property under the property attribute of a feature


File Format Search - UKOLN, University of Bath
File Format Search Engine
UKOLN
UNIVERSITY OF BATH
Input
Output
Further Information
OK
Cancel
QUERY

Function Requirements

- NURBS surface geometry (FULL)
- NURBS surface geometry (Partial)
- Geometric Dimensioning (FULL)
- Geometric Dimensioning (Partial)
- Polygon tessellation geometry (FULL)
- Polygon tessellation geometry (Partial)

Support Representation Format

- IGES
- STEP
- Solid Edge Assembly
- NX
- ACIS
- PLM XML

Version: 3

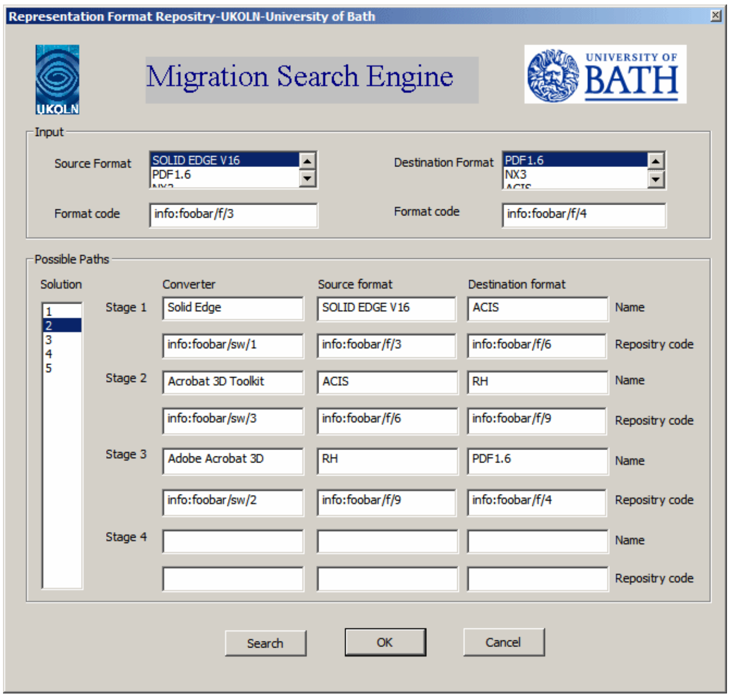
Repository Code: info:foobar/f/5

Figure 2 – The query interface for searching for file formats matching a given set of functional requirements.

element, alongside a `support` attribute with a value of either ‘`good`’ (for both required and desirable properties) or ‘`partial`’ (for desirable properties). In each case where all the requested properties are so described, the search script returns the details of file format being described. Fig. 2 shows this type of query being performed on the standalone registry by the prototype system. In the example, full (‘`good`’) support is requested for NURBS surface geometry, geometric dimensioning and polygon tessellation geometry. The system returns five suitable formats (IGES file, STEP physical file, Solid Edge assembly file, NX, ACIS, PLM XML) along with version information and the internal IDs of the formats.

2. *What are the properties of a given file format?* For this query, the search engine 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) cy-



Input

Source Format: SOLID EDGE V16, PDF1.6, NX3
Format code: info:foobar/f/3

Destination Format: PDF1.6, NX3, ACIS
Format code: info:foobar/f/4

Possible Paths

Solution: 1, 2, 3, 4, 5

	Converter	Source format	Destination format	
Stage 1	Solid Edge	SOLID EDGE V16	ACIS	Name
	info:foobar/sw/1	info:foobar/f/3	info:foobar/f/6	Repository code
Stage 2	Acrobat 3D Toolkit	ACIS	RH	Name
	info:foobar/sw/3	info:foobar/f/6	info:foobar/f/9	Repository code
Stage 3	Adobe Acrobat 3D	RH	PDF1.6	Name
	info:foobar/sw/2	info:foobar/f/9	info:foobar/f/4	Repository code
Stage 4				Name
				Repository code

Figure 3 – The query interface for searching for migration paths between two formats, allowing a maximum of four migration stages.

cle. 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. Fig. 3 shows this type of query being performed on the standalone registry by the prototype system. After selecting the maximum number of migration stages (in this case, four) the source format (Solid Edge v16) and destination format (PDF v1.6), the system returns five possible routes. On selecting a route, the details of the migration path are displayed; the second of the five routes is displayed in the figure.

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

With these queries, and a sufficient quantity and quality of representation information, one is able to shortlist the most suitable dissemination or archival formats for data in a given format, provided the significant characteristics of the data for the given purpose are identified. The workflow necessary to create the shortlist is given in Fig. 4; as an example, with the limited set of representation information just presented, the following enquiry was performed.

1. *Determine the starting format for data:* Solid Edge v16.

2. *Significant characteristics for in-service feedback on the design:* dimensions, approximate geometry. Additionally the destination format must support annotations and compression.

3. *Which formats support these characteristics?* Registry returns: IGES (v5.3), STEP, PDF (v1.6), JT (8.1).

4. For each format (for example, PDF):

    (a) *What are the properties of the format?* PDF supports approximate geometry but not exact geometry; it supports geometric dimensioning; it supports the addition of annotations, etc.

    (b) *What migration paths exist from Solid Edge v16 to the format?* There are three paths with two stages: Solid Edge exporting to STEP, Acrobat importing from STEP; Solid Edge exporting to IGES, Acrobat importing from IGES; and Solid Edge exporting to NX, Acrobat importing from NX. There are also two paths with three stages: Solid Edge exporting to ACIS, Acrobat 3D Toolkit converting from ACIS to U3D, Acrobat importing from U3D; and Solid Edge exporting to ACIS, Acrobat 3D Toolkit converting from ACIS to RH, Acrobat importing from RH.

    (c) For each path (for example, via IGES):

        i. *What are the known issues with the path?* The model units have to be set explicitly on import into Acrobat. The degradation of the geometry can be set with parameters.

        ii. *Does this path risk the significant characteristics?* The dimensions are at risk if conventions are not applied.

    (d) *Collate and rank remaining paths according to known issues:* Solid Edge exporting to STEP; Solid Edge exporting to IGES; two-stage Solid Edge to PDF paths; Solid Edge exporting to JT; three-stage Solid Edge to PDF paths.

5. *Shortlist best migration paths according to format characteristics and known issues with path:* Solid Edge exporting to STEP; Solid Edge exporting to IGES; two-stage Solid Edge to PDF paths (the other paths rejected due to known issues with preservation of the model tree).

6. *Decide on chosen path according to format characteristics, known issues with path, and costs:* Solid Edge exporting to STEP, Acrobat importing from STEP (intermediate format well supported; PDF reading and annotation software available at low cost).

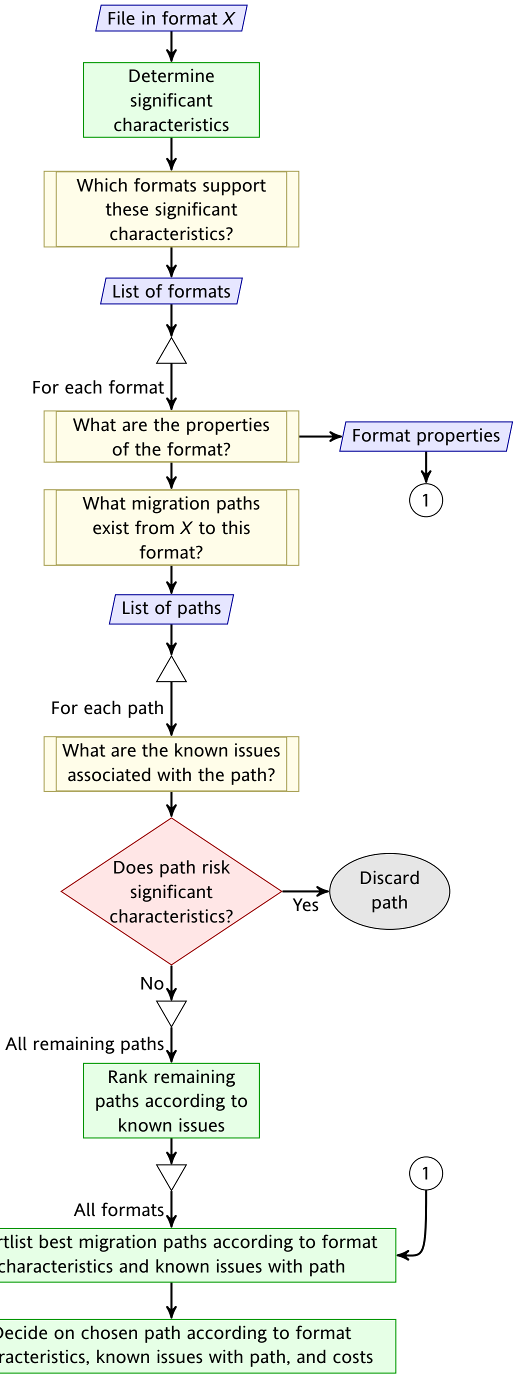

File in format *X*
Determine significant characteristics
Which formats support these significant characteristics?
List of formats
For each format
What are the properties of the format?
Format properties
1
What migration paths exist from *X* to this format?
List of paths
For each path
What are the known issues associated with the path?
Does path risk significant characteristics?
Yes
Discard path
No
All remaining paths
Rank remaining paths according to known issues
1
All formats
Shortlist best migration paths according to format characteristics and known issues with path
Decide on chosen path according to format characteristics, known issues with path, and costs

Figure 4 – Flowchart representing a strategy for choosing an appropriate dissemination or archival format for product data. The four subprocesses are queries to a representation information registry.

The next stage of development for the registry will be to integrate the functionality into RRoRI. The process of expanding the set of representation information held by the registry is ongoing.

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, they must give consideration to how that software deals with legacy data and documents, and how it interoperates with the other software in place, in order that existing IP and other assets are not compromised. The results of any research performed into data exchange and interoperability are typically 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 re-used 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.

In terms of the service-oriented architecture for automatic migration being worked on by the University of Minho [62], the tools presented here correspond to the Migration Advisor and Format Evaluator services. In terms of format evaluation, the registry presented in the current paper provides information that is much more specific to the needs of industry than the prototype Format Evaluator can provide; what the registry cannot do in its current form is deal with more general information such as documentation and links to standards bodies, although such information is handled already by RRoRI. In terms of migration advice, the registry presented in the current paper is again based on information more specifically relevant to industrial needs than the prototype Migration Advisor. What the registry lacks in comparison with the Migration Advisor is the ability automatically to factor in statistics captured from previous migrations.

The value of the registry 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. While finer-grained information enables more precise decision-making and greater flexibility in the face of changing information needs, there is an increased danger of the information varying with respect to different files, requiring much additional research to delimit the applicability of any given piece of information. This aspect 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 RRoRI) needs to be explicitly considered when setting up real-world interoperability between registries.

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 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 most appropriate of the available migration paths between two known formats, and the most appropriate 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 Registry/Repository of Representation Information (RRoRI). A demonstrator for querying from the registry based on the proposed method has been implemented; the next stage will focus on integrating the functionality into RRoRI and expanding the set of representation information in the registry.

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] W. D. Li & Z. M. Qiu (2006). 'State-of-the-Art Technologies and Methodologies for Collaborative Product Development Systems'. *International Journal of Production Research* 44:13. 2525–2559. ISSN: 0020-7543. DOI: 10.1080/00207540500422080.

[3] J. Y. H. Fuh & W. D. Li (2005). 'Advances in Collaborative CAD: the State-of-the-Art'. *Computer-Aided Design* 37:5. 571–581. ISSN: 0010-4485. DOI: 10.1016/j.cad.2004.08.005.

[4] Sherman Y.T. Langa, John Dickinson & Ralph O. Buchal (2002). 'Cognitive Factors in Distributed Design'. *Computers in Industry* 48:1. 89–98. DOI: 10.1016/S0166-3615(02)00012-X.

[5] 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).*

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

[7] 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.*

[8] 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.*

[9] Michael J. Pratt, Bill D. Anderson & Tony Ranger (2005). ‘Towards the standardized exchange of parameterized feature-based CAD models’. *Computer-Aided Design* 37:12. 1251–1265.

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

[11] 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.*

[12] 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.

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

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

[15] Michael J. Pratt (2005). ‘ISO 10303, the STEP Standard for Product Data Exchange, and its PLM Capabilities’. *International Journal of Product Lifecycle Management* 1:1. 86–94. DOI: 10.1504/IJPLM.2005.007347.

[16] John Jeremy Dunford (2004). *Validation report for ISO/DIS 10303-239, STEP Part 239, Application protocol: Product life cycle support.* WG3 N1451. ISO TC184/SC4. 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.

[17] Rohit Sharma & James Gao (2006). ‘STEP PLCS for Design and In-service Product Data Management’. In: *Advances in Design.* Edited by Hoda A. ElMaraghy & Waguih H. ElMaraghy. Springer Series in Advanced Manufacturing. Springer : London. 293–301. ISBN: 978-1-84628-004-7. DOI: 10.1007/1-84628-210-1_24.

[18] Roland Eckert, Wolfgang Mansel & Günther Specht (2005). ‘CASE Tools in Systems Engineering’. *Systems Engineering* 8:1. 41–50. ISSN: 1098-1241. DOI: 10.1002/sys.20018.

[19] ISO/TS 18876. *Industrial automation systems and integration — Integration of industrial data for exchange, access and sharing.* Multipart standard.

[20] Allison Barnard Feeney (2002). ‘The STEP Modular Architecture’. *Journal of Computing and Information Science in Engineering* 2:2. 132–135. DOI: 10.1115/1.1511520.

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

[22] Abhijit V. Sawant & John W. Nazemetz (1998). *Impediments of STEP.* 009. Oklahoma State University : Stillwater, OK.

[23] Vailin Choi & Mike Folk (2007). ‘Investigations into using HDF5 as an Alternative to STEP for Finite Element Modeling Data’. In: *Proceedings of the 9th NASA-ESA Workshop on Product Data Exchange.* National Aeronautics & Space Administration : Santa Barbara, CA. URL: http://step.nasa.gov/pde2007/Investigations-into-Using-HDF5-as-an-Alternative-to-STEP-for-Finite-Element-Modeling-Data.pdf.

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

[25] Ian C. Braid (1974). *Designing with volumes.* PhD thesis. Cambridge University.

[26] C. F. Goldfarb (1981). 'A Generalized Approach to Document Markup'. In: *Proceedings of the ACM SIGPLAN SIGOA Symposium on Text Manipulation.* Portland, Oregon, United States. ACM Press : New York. 68–73. ISBN: 0-89791-050-8. DOI: 10.1145/800209.806456.

[27] James H. Coombs, Allen H. Renear & Steven J. DeRose (1987). 'Markup Systems and the Future of Scholarly Text Processing'. *Communications of the ACM* 30:11. 933–947. DOI: 10.1145/32206.32209.

[28] John B. Bedunah (1999). 'XML: the Future of the Web'. *Crossroads* 6:2. 5–10. DOI: 10.1145/333104.333109. URL: http://doi.acm.org/10.1145/333104.333109.

[29] *Extensible Markup Language (XML) 1.1 (Second Edition)* (2006). W3C Recommendation. World Wide Web Consortium. URL: http://www.w3.org/TR/xml11/.

[30] 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.*

[31] Russell S. Peak et al. (2005). 'STEP, XML, and UML: Complementary Technologies'. *Journal of Computing and Information Science in Engineering* 4. 379. DOI: 10.1115/1.1818683.

[32] UGS (2005). *Open Product Lifecycle Data Sharing Using XML.* White Paper. URL: http://www.ugs.com/en_us/Images/wp_plm_xml_14_tcm53-11521.pdf.

[33] UGS (2006). *Real-Time Engineering Collaboration: Using Web-Based Communities to Collaborate Throughout the Product Lifecycle.* White Paper. URL: http://www.ugs.com/en_us/Images/wp_community_collaboration_tcm53-3191.pdf.

[34] ECMA-363 (2007). *Universal 3D File Format.* 4th edition. URL: http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-363%204th%20Edition.pdf.

[35] Open HSF Initiative. *The HOOPS 3D Product Suite.* Specification documentation. URL: http://www.openhsf.org/docs_hsf/index.html.

[36] Autodesk (2003). *DWF 6 Specification.* Part of the Autodesk DWF Toolkit 7.3.

[37] XGL Working Group (2006). *XGL File Format Specification.* February 2006. URL: http://web.archive.org/web/20060218/http://www.xglspec.org/.

[38] Dave Shreiner (ed.) (2006). *OpenGL Reference Manual : the Official Reference Document to OpenGL, Version 2.0.* Addison-Wesley Professional : Harlow. ISBN: 0-321-33571-6.

[39] Ron LaFon (2005). '3D without Boundaries: New Tools to Publish and Share Designs'. *Cadalyst* 22:12 (December). 18–29. ISSN: 0820-5450. URL: http://www.nxtbook.com/nxtbooks/questex/cadalyst1205/index.php?startpage=18.

[40] 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/.

[41] 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/.

[42] 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/.

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

[44] Dassault Systèmes (2005). *Dassault Systèmes Delivers 3D XML Specifications and Player.* Press release. June 2005. URL: http://www.3ds.com/news-events/press-room/release/899/1/.

[45] UGS (2006). *JT File Format Reference: Version 8.1.* URL: http://www.jtopen.com/docs/JT_File_Format_Reference.pdf.

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

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

[48] Ghang Lee et al. (2006). 'Grammatical Rules for Specifying Information for Automated Product Data Mining'. *Advanced Engineering Informatics* 20:2. 155–170. ISSN: 1474-0346. DOI: 10.1016/j.aei.2005.08.003.

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

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

[51] Adrian Brown (2005). 'Automating Preservation: New Developments in the PRONOM Service'. *RLG DigiNews* 9:2 (April). 25 para. ISSN: 1093-5371. URL: http://digitalarchive.oclc.org/da/ViewObjectMain.jsp?fileid=0000070519:000006289143.

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

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

[54] Jeffrey Darlington (2003). 'PRONOM: A Practical Online Compendium of File Formats'. *RLG DigiNews* 7:5 (October). 20 para. ISSN: 1093-5371. URL: http://digitalarchive.oclc.org/da/ViewObjectMain.jsp?fileid=0000070519:000006289273.

[55] Caroline R. Arms & Carl Fleischhauer (2005). 'Digital Formats: Factors for Sustainability, Functionality, and Quality'. In: *IS&T Archiving Conference.* 26–29 04. Society for Imaging Science & Technology : Washington, DC. URL: http://memory.loc.gov/ammem/techdocs/digform/Formats_IST05_paper.pdf.

[56] Stephen L. Abrams (2005). 'Establishing a Global Digital Format Registry'. *Library Trends* 54:1. 125–143. ISSN: 0024-2594. URL: http://muse.jhu.edu/journals/library_trends/v054/54.1abrams.pdf.

[57] Stephen L. Abrams & Dale Flecker (2005). *A Proposal for a Global Digital Format Registry*. URL: http://hul.harvard.edu/gdfr/documents/Proposal-2005-09-29.doc.

[58] GDFR (2004). *Global Digital Format Registry Data Model v4*. URL: http://hul.harvard.edu/gdfr/documents/DataModel-v4-2004-01-12.doc.

[59] David Giaretta et al. (2005). 'Representation Information for Interoperability Now and with the Future'. In: *Local to Global Data Interoperability: Challenges and Technologies, 2005*. 20–24 06. IEEE : Sardinia. ISBN: 0-7803-9228-0. DOI: 10.1109/LGDI.2005.1612462.

[60] David Giaretta (2006). 'CASPAR and a European Infrastructure for Digital Preservation'. *ERCIM News* 66 (July). 47–49. URL: http://www.ercim.org/publication/Ercim_News/enw66/giaretta.html.

[61] David Giaretta (2007). 'The CASPAR Approach to Digital Preservation'. *International Journal of Digital Curation* 2:1. 112–121. ISSN: 1746-8256. URL: http://www.ijdc.net/ijdc/article/view/29/32.

[62] Miguel Ferreira, Ana Alice Baptista & José Carlos Ramalho (2006). 'A Foundation for Automatic Digital Preservation'. *Ariadne* 48. 64 para. ISSN: 1361-3200. URL: http://www.ariadne.ac.uk/issue48/ferreira-et-al/.

[63] Jane Hunter & Sharmin Choudhury (2006). 'PANIC: an Integrated Approach to the Preservation of Composite Digital Objects using Semantic Web Services'. *International Journal on Digital Libraries* 6:2. 174–183. ISSN: 1432-5012. DOI: 10.1007/s00799-005-0134-z.

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

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

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

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

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

[69] Rogelio Oliva & Robert Kallenberg (2003). 'Managing the Transition from Products to Services'. *International Journal of Service Industry Management* 14:2. 160–172. ISSN: 0956-4233. DOI: 10.1108/09564230310474138.

All links were correct on 1st October 2007.

This paper was submitted to Advanced Engineering Informatics *on 3rd October 2007, accepted on 7th October 2007, and made available online on 11th December 2007.*