Citation for published version:
Ball, A, Patel, M & Ding, L 2008, 'Towards a Curation and Preservation Architecture for CAD Engineering Models' Paper presented at iPRES 2008: The Fifth International Conference on Preservation of Digital Objects, London, UK United Kingdom, 29/09/08 - 30/09/08, pp. 107-114.

Publication date:
2008

Towards a Curation and Preservation Architecture for CAD Engineering Models

Alexander Ball and **Manjula Patel**
UKOLN, University of Bath, Bath. BA2 7AY.

Lian Ding
IdMRC, Dept. of Mechanical Engineering, University of Bath, Bath. BA2 7AY.

Abstract

For many decades, computer-aided design (CAD) packages have played an important part in the design of product models within the engineering domain. Within the last ten years, however, the increasing complexity of CAD models and their tighter integration into the workflow of engineering enterprises has led to their becoming the definitive expression of a design. At the same time, a paradigm shift has been emerging whereby manufacturers and construction companies enter into contracts to take responsibility for the whole lifecycle of their products – in effect, to sell their product as a service rather than as an artefact. This makes necessary not only the preservation of the product's design, but also its continuing intelligibility, adaptability and reusability throughout the product's lifecycle. The CAD models themselves, though, are typically in closed formats tied to a particular version of an expensive proprietary application prone to rapid obsolescence. While product lifecycle management (PLM) systems deal with some of the issues arising from this, at present it is not possible to implement a comprehensive curation and preservation architecture for CAD models, let alone the other forms of engineering information.

In order to fill in some of the gaps in a possible architecture, we have developed two tools to aid in the curation and preservation of CAD models. The first is a preservation planning tool for CAD models: a Registry/Repository of Representation Information for Engineering (RRoRIfE). The tool uses Representation Information, as defined by the Open Archival Information System (OAIS) Reference Model, to advise on suitable strategies for migrating CAD models to archival or exchange formats. The second – Lightweight Models with Multilayered Annotations (LiMMA) – is an architecture for layering non-geometric information on top of a geometric model, regardless of the format used for the geometric model. We envision this architecture being used not only to create flexible, lightweight archival representations of model data, but also to facilitate better information flows between a design team and the rest of the extended enterprise.

Introduction

Within the engineering industry, Computer Aided Design (CAD) has grown steadily in importance since its introduction in the mid-1950s (Bozdoc 2004). Originally used to aid in the production of design drawings, CAD can now define a design more clearly than two dimensional drawings ever could, and within the past decade has started taking over as the definitive expression of a design. With the corresponding rise of Computer Aided Manufacturing (CAM) and Computer Aided Engineering (CAE) systems, not to mention Enterprise Resource Planning (ERP), Customer Relationship Management (CRM) and Supply Chain Management (SCM) systems, the potential for CAD models to be integrated with processes across a product's lifecycle is just starting to be realized.

The primary purpose of a CAD model is to represent the physical geometry of a design, typically in three dimensions. There are two different methods by which conventional CAD models represent the geometry of a product. Constructive Solid Geometry (CSG) constructs models as a combination of simple solid primitives, such as cuboids, cylinders, spheres, cones, etc. Boundary representations (B-rep), in contrast, represent shapes by defining their external boundaries: structured collections of faces, edges and vertices (McMahon and Browne 1996). Compared to CSG, B-rep is more flexible and has a much richer operation set, and so has been widely adopted in current commercial CAD systems. One of the ways in which B-rep models can be made highly expressive is through use of freeform surface modelling. This is where complex surface curvatures are represented using mathematical functions – such as Non-Uniform Rational B-Spline (NURBS) or Bezier surfaces – or approximations thereof.

CAD models can express more than just geometry, though. Most common CAD systems, whether using CSG and B-rep representations, can represent parts in terms of 'features', which encapsulate the engineering significance of the part as well as its geometry. Such features are often defined parametrically, allowing variations on the same basic part to be used throughout the model with little repetition of design data. Features are used not only for product design and definition, but also for reasoning about the product in a variety of applications such as manufacturing planning (Shah and Mäntylä 1995). While features are useful when coming to interpret a design, many are provided by vendors and/or embedded within CAD systems, making it hard to exchange the non-geometric information between systems. Additionally, features tend to be written from the designer's point of view, and may not fit the viewpoints of engineers in other parts of the extended enterprise.

The integration of CAD systems with other computerized systems in the manufacturing and in-service engineering phases is a significant part of Product Lifecycle Management (PLM), which aims to allow organizations to manage their products from conceptualization to disposal in the most efficient way possible. PLM is becoming increasingly important as organizations enter into more through-life contracts with their customers. Indeed, the extent to which customers are preferring to use a service model for acquiring products, particularly from the aerospace, defence and construction industries, has lead some authors to describe this in terms of a paradigm shift (Davies, Brady, and Tang 2003;

Oliva and Kallenberg 2003). The product-service paradigm places a number of requirements on CAD, not least that the product data be kept intelligible, adaptable and reusable throughout the product's lifecycle. When considering the lifespan of some of the products – of the order of thirty or more years – this is not an insignificant challenge, especially given the rate of change of CAD software.

The CAD software industry is intensely competitive, with market forces driving rapid functional and performance improvements. While this has obvious benefits, it also has negative consequences. The ways in which the improvements are implemeted cause conflicts not only with implementations on other CAD packages – and indeed with other types of systems – but also with those of earlier versions of the same CAD package. With little interoperability or backwards compatibility, and rapid turnover of software versions, CAD models can become unreadable within the timespan of three to ten years. That is not to say that CAD translation tools do not exist – they do – but due to the nature of the task they are not altogether reliable: in 2001 the manual correction of translated CAD data cost the aerospace, automotive, shipbuilding and tooling industries an estimated US$74.9m in the US alone (Gallaher, O'Connor, and Phelps 2002).

Even leaving aside the preservation issues, there are barriers to using CAD in a PLM context. Every participant in the collaborative enterprise throughout the whole product lifecycle is expected to share product information – the staff in various departments within a company, partners, contractors/subcontractors, service providers and even customers – and CAD models carry most of the important information and knowledge. On the one hand, the cost of CAD packages makes it infeasible for staff outside the design team(s) to have access to the models. On the other hand, companies are naturally unwilling to share full product models that include commercially sensitive information, especially with temporary partners, with whom collaborative protocols are not established and who may at other times be competitors.

Furthermore, current CAD models are 'resource-heavy', and restrict information transmission between geographically distributed applications and users. The file size of a relatively simple component (e.g. a crankshaft) could be over 1MiB in one leading CAD system. Hundreds of such components may be included in a product such as a car, leading to very large storage requirements for models and restricting the options for their communication.

In the remainder of this paper, we report the state of practice with regard to PLM systems. We then present our proposed additions to PLM architecture to better cater for the curation and preservation of product model data. Finally, we present in more detail the set of significant properties of product model data used by our proof-of-concept tools and give our conclusions.

Product Lifecycle Management

Engineering organizations of reasonable size are likely to use a PLM system for managing their data. PLM systems offer a number of different functions, for instance: file storage (typically with version control, access permission control, simple on-access format conversions), cross-file linkages (e.g. bills of materials generated directly from CAD models), cross-system linkages (typically with ERP, CRM, and SCM systems), portals for various activities across the lifecycle (e.g. simulation analysis, maintenance log management) and facilities for collaboration, both within lifecycle stages and between them. A number of PLM systems use lightweight formats – simple 3D formats that miss out much of the richness of a full CAD format – for communicating design information across the enterprise, and many claim to enforce compliance with various regulatory and certificatory requirements (Registration, Evaluation and Authorization of Chemicals; Six Sigma Quality; etc.).

While current PLM systems are certainly highly functional software environments, and do contain features pertinent to curation, they do not have any particular emphasis on preservation. None of the major PLM offerings (Dassault Systèmes, Siemens, SofTech, etc.) have integrated tools for preservation planning, monitoring when data storage media need to be refreshed, monitoring file format obsolescence, and so on. With some functions, such as wholesale migration from one CAD system and format to another, this is because the operation would be so complex, extensive and infrequent that it would need to be handled by a specialist team using specialist tools. With others, such as choosing appropriate lightweight formats for particular applications, it is because the PLM system architecture is only designed to support one option. Thus in order to fully support the curation and preservation of engineering documentation, additional tools are needed.

Proposed architecture

General framework

Within the digital library and digital preservation communities, several curation and preservation environments have already been developed.

PANIC (Preservation Web services Architecture for New media and Interactive Collections) is a semi-automated preservation environment developed by the University of Queensland (Hunter and Choudhury 2006). Its aim is to support three particular aspects of preservation: capture and management of preservation metadata, monitoring for format obsolescence, and interacting with preservation Web services. The architecture is modular, with separate local services for capturing and storing metadata, checking for obsolescence, discovering Web services, selecting Web services and invoking Web services. It relies on seperate, probably external, registries for file formats and preservation Web services.

CRiB (Conversion and Recommendation of digital oBject formats) is a similar environment developed by the University of Minho (Ferreira, Baptista, and Ramalho 2007). It includes local services for detecting the formats of ingested materials, checking for format obsolescence, determining suitable alternative formats for ingested materials, determining suitable migration pathways, recording details of available preservation services, invoking preservation services, and evaluating the success or otherwise of preservation actions to inform future decisions.

PLANETS (Preservation and Long-term Access through NETworked Services) is a European Union funded project looking at practical preservation strategies and tools (Farquhar and Hockx-Yu 2007). One of its deliverables is a modular preservation environment; among other things, the environment consists of: Plato, a preservation planning tool; a testbed for evaluating preservation approaches; a software emulation environment; a tool for designing automated preservation workflows; a set of modules for carrying out automated preservation workflows; a file format characterization registry; a preservation action registry; and a registry of preservation services.

It is clear that all three examples have much in common in terms of their architecture and the services they provide, and that these services are largely if not entirely absent from current PLM systems. That is not to imply that *all* of these services would be especially useful in the engineering context. For example, an obsolescence notifier would likely be of limited use as for large quantities of data within the organization, obsolescence comes about solely as a result of planned software upgrades rather than through environmental changes. Similarly, the migration of CAD models between major CAD formats is not a process to automate lightly, although other types of engineering documentation – reports, spreadsheets – may benefit from this sort of approach.

Another aspect that we feel deserves greater examination is the way PLM systems handle the communication of CAD data across the extended enterprise. Lightweight formats have particular advantages over full CAD formats, in that they are typically fairly simple, well documented and free from restrictive licences; this in turn means that it is relatively inexpensive to write software to support them, which means that such software is usually offered at little or no cost, and can be run across a number of platforms. All these things combined mean that they will likely remain readable for considerably longer than full CAD models. These advantages have not escaped CAD vendors, especially those who also produce PLM software, and a number have created their own. Because of this, there is a trend for PLM systems to support just one lightweight format for design review processes and the like. For example, Siemens Teamcenter uses JT, while Dassault Systèmes' PLM offerings use 3D XML. This is unfortunate, as different lightweight formats have different characteristics that make them particularly suited to specific use cases. Furthermore, feeding back information from later in the lifecycle is typically achieved through an entirely different set of functions, meaning that the benefits of tying, for example, in-service maintenance records directly to the original CAD models – in order to inform future design choices – are left unexploited in current PLM implementations.

The architecture we propose would add to PLM systems the following functions: a registry of format characteristics, a registry of format migration services, a registry of (evaluations of) preservation actions, and a preservation planning tool based on top of these three registries. We also propose that PLM systems should adopt a more flexible, modular and consistent approach to communicating design information throughout the extended enterprise, the better to aid the curation of engineering information.

To this end we have developed two proof-of-concept systems, demonstrating how some of these functions may be implemented. The first, LiMMA (Lightweight Models with Multilayered Annotations), is a system for representing CAD models using lightweight geometric models supplemented with layers of XML-encoded information. The second, RRoRIfE (Registry/Repository of Representation Information for Engineering), is a simple preservation planning tool that incorporates a registry of format characteristics and a registry of migration software.

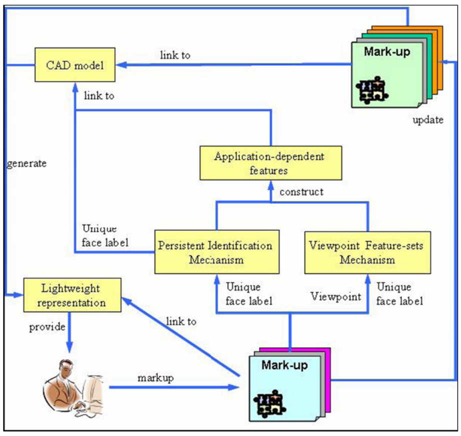

Figure 1: LiMMA -A Framework for the Annotation of CAD Models

LiMMA

LiMMA is not a single application or platform, but a series of individual tools based around a common XML schema and workflow. The premise behind it is that the same geometric model can exist in a number of different formats: full CAD formats, lightweight visualizations or exchange standards like STEP (ISO/TS 10303-203:2005) or IGES (US Product Data Association 1996). If extra information is added to the model in any one of these formats, and if that information is to be used to the widest possible extent, it ought to be visible in every other format, but this is problematic for at least three reasons: a) this would involve regenerating each version of the model every time information is added, b) different formats treat non-geometric information in different ways, and c) it would probably involve designing custom format translators. The solution in LiMMA is not to change the models at all but to store the information as annotations in a separate XML file and layer those annotations on top of the model using a system of persistent references (see Figure 1). Thus LiMMA consists of a series of plugins and viewers that allow one to interact with the annotation files whilst viewing the model, the system of persistent references used by the plugins and viewers, and the workflow of moving models and annotation files around the extended enterprise.

The multilayering of annotations in LiMMA is a way of offering additional flexibility and increasing the efficiency of the system. Not all the annotations will be of interest to everyone in the organization, and some may be confidential to a small group of engineers. By storing annotations in several different files according to access permission and interest groups, one can ensure that everyone receives all and only the annotations that they are allowed to see and that are of interest to them. The segregation of annotations into different files does not affect their usability as they are all layered on top of the model at once.

LiMMA has the potential to improve information flows throughout the product lifecycle. At the *design* stage desginers can embed and share design rationale, meanwhile geographically distant design teams may collaborate on the same design using lightweight formats that preserve the exact geometry of the model. Any additional design information not recorded by the lightweight format – such as materials and finishes – could be communicated using the annotation files. Similarly, an annotated lightweight/exchange version of the full model could be submitted to regulatory bodies for inspection, without either party having to invest in full CAD translations or multiple CAD package licences. Finally, the organization's customers could be provided with lightweight models (using approximate geometry in order to protect the organization's intellectual property) and function-related annotations. Similar models could be used as marketing materials to attract further customers.

By the *production* stage, the design has been finalized and lodged in the PLM system. A copy in a lightweight or exchange format, with accompanying annotations providing the additional design information and semantics to enable later re-editing, should also be kept in case the original model cannot reliably be opened when it is next needed. The CAD package in use by the design team and the Numerical Control (NC) software in use by the production engineers do not need to be so tightly integrated if the NC programmes can be generated from lightweight formats with exact geometry and maufacturing-related annotations. These could also be used by production engineers to feed back comments to the designers.

Once the product has reached its *in-service* phase, lightweight models with approximate geometry and annotations relating to disassembly and reassembly could be supplied to maintenance engineers, enabling them to have access to the design while inspecting the product. Inspection results could be marked up directly onto the model, allowing these results to be fed back to the designers as annotations. In this way, when the model is next opened for redesign or upgrade, any systematic in-service issues with the existing design can be spotted immediately and dealt with.

Finally, when the product has reached *end of life*, engineers could use a lightweight model with annotations relating to materials to determine which parts need to be disposed of in a controlled manner, and which can be recycled in some way; this type of information is also useful for input into future design and development.

So far, LiMMA plugins have been written in C/C++ and NX Open for the CAD package NX and in JavaScript for the 3D PDF viewers Adobe Acrobat and Adobe Reader, while a standalone LiMMA X3D viewer has been written using Java. The annotations are currently linked to the models by means of unique identifiers attached to surfaces within the model, but a parallel system of reference using co-ordinate sets is also under development.

RRoRIfE

RRoRIfE is primarily a planning tool, enabling information managers to explore the options available for converting CAD models into other formats, whether for contemporaneous exchange or for long term archiving. It does this by means of stored information about the capabilities of various formats and processing software with respect to certain significant properties (see Figure 2); the precise details are given in the section on significant properties below.

As well as simply allowing one to browse through the information contained in the self-contained repository, RRoRIfE allows one to perform three different types of search on it. The first allows one to search for all the (known) formats that fit a chosen set of criteria with respect to significant properties. For each property, one can specify that it should be fully supported, fully or partially supported, or not supported at all; otherwise it is not considered in the search. The second type of search calculates the possible migration paths between two formats, in a given number of steps or fewer. The third type of search allows one to specify a starting format and, as in the first type of search, a set of criteria for the final destination format. RRoRIfE will then calculate a set of suitable migration pathways with the specified number of steps or fewer, and perform some simple ranking on them.

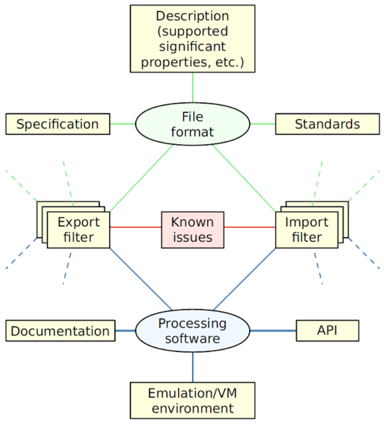

Figure 2: Capabilities of formats and conversion tools

We anticipate RRoRIfE being of use in at least the following scenarios: a) determining which lightweight formats would be suitable for which purposes when planning organizational LiMMA workflows and archival strategies; b) determining which tools or services to use to generate those lightweight formats from the full CAD models; and c) providing additional decision support when procuring a new CAD system to replace the exisiting one.

Further Work

There remain some outstanding issues with both LiMMA and RRoRIfE that need to be resolved in order to fully demonstrate their usefulness. One of the use cases for LiMMA is for annotated lightweight or exchange formats to be used as an archival backup in case the original CAD model ceases to be readable. In order to prove this concept, we plan to determine if a set of annotations can be generated automatically from the non-geometric information in a full CAD model, and to assess the feasibility of reconstructing a full CAD model from an annotated lightweight model.

With RRoRIfE, we intend to demonstrate how the Representation Information it stores may be synchronized with generic registries such as the Registry/Repository of Representation Information (RRoRI) developed by the UK's Digital Curation Centre and the European CASPAR Project (Giaretta 2007). There is also plenty of scope for expanding RRoRIfE to take account of more than just significant properties: openness of formats, price, availability and customizability of software, as well as evaluations of previous preservation actions.

Other areas of the proposed architecture we have not yet explored include the systematic evaluation of preservation actions, and the human and organizational issues associated with keeping the various registries up to date. Having argued against the need for an obsolescence notifier in the given context, it may yet be useful to have a tool that measures the potential impact, with respect to the readability of files, of a proposed software upgrade or system change.

Significant Properties

The utility of the LiMMA system is predicated on the understanding that different viewpoints and different stages of the product lifecycle have varying uses and requirements for CAD models; some features of a model may be vital for one engineer and irrelevant for another. In other words, the *significant properties* of a CAD model vary between viewpoints and lifecycle stages. Significant properties are those aspects of a digital object which must be preserved over time in order for it to remain accessible, usable and meaningful (Wilson 2007, 8).

In the general case, what may be considered significant about an object depends partly on the nature of an object – for a Mercator projection map, true bearings are significant while areas are not, whereas for a sinusoidal projection map, areas are significant but bearings are not – and partly on the purposes to which it is put – such as whether one is concerned about a graph's underlying data or its aesthetics. The latter dependency means that conceivably *any* property of an object may be siginificant to someone, so those entrusted with the preservation of the object have to prioritize the possible future uses of the object, and thereby the significant properties to preserve. In practice, for CAD models there are a limited number of 'business' uses (as opposed to academic uses) to which they could be put at present, although of course one cannot predict the future with any certainty.

For the purposes of constructing RRoRIfE, whose purpose is to compare different methods of expression and the processes of translating between them, we had to take a view on significant properties that was one step removed from the definition given above. We considered significant properties to be those aspects of a digital object which any new expression of that object must exhibit in order to fulfil its intended function while being faithful to the original; the notion of faithfulness is intended to encapsulate the given definition's notion of preservation over time with respect to access, utility and meaning.

In the previous section on the proposed architecture, we outlined a number of use cases for CAD models. From these, several types of requirements can be identified:

- Some use cases require exact geometry, others approximate geometry.
- Some use cases require the modelling history;
- Some use cases require geometry-related metadata (tolerances, finishes, etc.);
- Some use cases require transmission of the model over a the Internet;
- LiMMA relies on persistent identification of (subsets of) geometry.

In the following subsections we present our working list of significant properties for CAD models, based on these requirements. The properties are structured in a hierarchy in order to take advantage of logical dependencies between them; for example, if a format is capable of analytically expressing an ellipse, it can certainly express a circle analytically. This allows for greater brevity when recording the expressiveness of different formats.

Geometry

There are two factors to consider when judging whether geometry expressed in one format may be expressed exactly in another format. The first is whether the entities used in the first expression have an equivalent in the second format, and the second is whether the conversion from the original entity to its equivalent can be done programmatically. The first of these can be determined relatively easily by comparing the basic entities supported by each format. Thus the first set of significant properties concerns geometric entities (Table 1).

These entities were compiled with reference to a previous study of the significant properties of vector graphics, and a number of different format specifications and software manuals (Coyne et al. 2007; ISO/TS 10303-203:2005 ; ISO/IEC 19775:2004 ; Shene 2007; Shene 1997; US Product Data Association 1996).

Geometric construction techniques

In order to build geometric entities into full CAD models, one or more construction techniques have to be used. The methods of construction available within a file format have a significant impact on its expressiveness, thus the second set of significant properties relates to these (Table 2).

One of the main distinguishing features of a format is whether it only allows parts to be made up of Boolean operations on solid objects (Constructive Solid Geometry), or whether individual surfaces can be used as well or instead (Boundary representation). There are further distinctions in the use of parametrically defined parts and construction history modelling. Finally, some formats have facilities for including several different versions of the same part; commonly this is used to speed up rendering – so viewers can render small or distant parts using low-fidelity meshes – but may be used to provide alternative organizational viewpoints on the same data.

Geometry-related metadata

The third set of significant properties is concerned with information about particular parts of the geometry, apart from shape information (Table 3).

In addition to the actual geometry, manufacturing and quality control processes require at the least geometric dimensioning and tolerancing information (giving the size of the various components and acceptable limits for errors), as well as information on the materials from which to make the components and the required finishes. Certain re-editing applications also require the preservation of the semantics associated with model 'features' (predefined geometry with established engineering meaning).

If a format provides a way of adding arbitrary metadata to a node in the assembly (a subassembly, part or perhaps surface), this can provide a way for additional geometry-related information to be embedded within the model. Even if the currently available software is unable to make use of this information, additional tools or plugins may be developed to interpret it.

Compression and identification

One of the factors that determine whether a format is likely to be suitable for transmission over the Internet, which may be necessary with geographically dispersed design teams, is whether a format tends to produce smaller file sizes. It was not considered within the scope of this project to devise a

Table 1: 2D and 3D Geometric entities

Entity	**Special case of**
Point	–
Polyline	–
Line	Polyline
Conic arc	–
Elliptical arc	Conic arc
Circular Arc	Elliptical arc
Open composite curve	–
Closed composite curve	–
Ellipse	–
Circle	Ellipse
Polygon	–
Triangle	Polygon
Rectangle	Polygon
Square	Rectangle
NURBS curve (open or closed)	–
Rational Bézier curve	NURBS curve
Non-rational Bézier curve	Rational Bézier curve
Cubic Bézier curve	Non-rational Bézier curve
Quadratic Bézier curve	Cubic Bézier curve
Point cloud	–
Helix	–
Plane	–
Ellipsoid	–
Sphere	Ellipsoid
Cylinder	–
Cone	–
Cuboid	–
Cube	Cuboid
Torus	–
Mesh of surface segments	–
Mesh of tessellating triangles	Mesh of surface segments
Lofted surface	–
Ruled surface	Lofted surface
Translation surface	–
Normal swept surface	–
Polylinear swept surface	Normal swept surface
Extrusion surface	Polylinear swept surface
Swung surface	Normal Swept surface
Rotation surface	Swung surface
NURBS surface	–
Rational Bézier surface	NURBS surface
Non-rational Bézier surface	Rational Bézier surface

Table 2: Geometric construction techniques

Entity	**Special case of**
Constructive Solid Geometry	–
Boundary representation	–
Trimmed surfaces (surfaces trimmed by boundary curves/surfaces)	–
Parameterized re-use of instances	–
Simple re-use of instances	Parameterized re-use of instances
Construction history modelling	–
Multiple alternative representations	–
Levels of detail	Multiple alternative representations

Table 3: Geometry-related metadata

Entity	**Special case of**
Feature semantics	–
Material metadata	–
Geometric dimensioning and tolerancing	–
Dimensions	Geometric dimensioning and tolerancing
Assembly node metadata	–
Assembly hierarchy	–

Table 4: Compression and Identification

Entity	**Special case of**
Field-wise compression	–
Stream-wise compression	–
Whole-file compression	–
Streaming	–
Identification of subassemblies	–
Identification of parts	–
Identification of surfaces	–
Identification of edges	–
Identification of vertices	–

reliable and fair metric for determining this quantitatively, so in lieu of this, our significant properties include various ways in which file sizes may be reduced. One method was mentioned above – re-use of a single part several times within a model – and the remainder are given here (Table 4). Another factor to be considered is whether the format allows streaming: allowing the file to be opened before it has been entirely transferred.

Finally, there is the matter of identification of the parts of a model. We are particularly interested in this from the perspective of using LiMMA, but there are other technologies which would benefit from being able to refer to identifiers within models.

Implementation in RRoRIfE

RRoRIfE uses two different XML schemata to store Representation Information, one for file formats and one for conversion processes; each schemata is based on the above ontology of significant properties.

The first schema relates to file formats and describes whether or not the format supports a particular property. As well as 'full' support and 'none', an intermediate value of 'partial' support is allowed, to indicate that support is limited in some way; for example, NURBS surfaces may be allowed, but only with 256 or fewer control points. In cases of partial support, explanatory text must be provided.

The second XML schema relates to conversion processes, grouped by software product. For each format conversion – and each optional variation of that conversion – the software is able to perform, the schema allows one to record how well the conversion preserves each property. Four levels of preservation are allowed. 'None' indicates that the property has never knowingly survived the conversion intact (most frequently because the destination format does not support the property). 'Good' indicates that the conversion has so far preserved examples of the property sufficiently well that it would be possible to reconstruct the original expression of the property from the new expression. 'Poor' is used when

Start format: [Please select]

Weight: 1 ● Required ○ Desired ○ Irrelevant ○ Forbidden

Max. no. of stages: 4

Please make your selection and then press 'Search'.

Figure 3: User Interface to RRoRIfe

tests have found it at least as likely for the property to be corrupted or lost as it is to survive. Lastly, ‘fair’ is used in all other cases, alongside an explanatory note.

Where preservation is less than ‘good’, it is possible to record whether the property survives in a degraded form, and if so, whether this degradation always happens in a fixed way, a configurable way or an unpredictable way. For example, when moving from a format that supports NURBS to one that only supports tessellating triangles, there may be a fixed algorithm for approximating surfaces, or one may be able to specify how detailed the approximation is.

The hierarchy of the ontology has been programmed into RRoRIfE, so that it knows that if a format supports NURBS surfaces, for example, it also supports non-rational Bézier surfaces. It does not make these inferences, though, if the Representation Information file in question already contains a statement about the ‘child’ property. Figure 3 shows the GUI to RRoRIfe.

Conclusions

In this paper we have argued that CAD packages and PLM systems do not currently provide the functionality required for the preservation of engineering materials, nor for taking full advantage of potential information flows within organizations responsible for the full lifecycles of their products. We therefore propose the addition of several new components to the PLM system architecture: a system of lightweight models and layers of annotation, to facilitate easier and more far-reaching information flows; a registry of file format characteristics, to help determine the suitability of the formats for specific purposes; a registry of format migration software and services, and a registry of (evaluations of) preservation actions, to aid in planning migration strategies; and a preservation planning tool based on top of these three registries. In order to test the feasibility of these architectural additions, we have developed two proof-of-concept systems. LiMMA demonstrates how annotations stored in dedicated XML files may be layered on top of CAD models in a variety of formats, using application plugins and custom viewers via a persistent reference mechanism. These annotations may be passed around an organization independently of each other and used with any translation of the referent model. In addition, as they are simpler and better documented than full CAD formats, lightweight formats are better suited to long term preservation, and some of the information lost in translation may be preserved instead as annotations. RRoRIfE demonstrates how information about the support that file formats and processing software have for the significant properties of CAD models can be used to support preservation planning decisions. There are, of course, still a number of issues to resolve with the proposed architecture, not least the human and organizational aspects of maintaining such a system, but we believe that it is promising, worth studying and developing further.

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. The Digital Curation Centre is funded by the UK Joint Information Systems Committee (JISC).

References

Bozdoc, M. 2004. The history of CAD. Available from: `http://mbinfo.mbdesign.net/CAD-History.htm` (2008-02-01).

Coyne, M.; Duce, D.; Hopgood, B.; Mallen, G.; and Stapleton, M. 2007. The significant properties of vector images. Technical report, Joint Information Systems Committee.

Davies, A.; Brady, T.; and Tang, P. 2003. *Delivering Integrated Solutions*. Brighton: SPRU/CENTRIM.

Farquhar, A., and Hockx-Yu, H. 2007. Planets: Integrated services for digital preservation. *International Journal of Digital Curation* 2(2):88–99.

Ferreira, M.; Baptista, A. A.; and Ramalho, J. C. 2007. An intelligent decision support system for digital preservation. *International Journal on Digital Libraries* 6(4):295–304.

Gallaher, M. P.; O’Connor, A. C.; and Phelps, T. 2002. Economic impact assessment of the international Standard for the Exchange of Product model data (STEP) in transportation equipment industries. RTI Project 07007.016, National Institute of Standards & Technology, Gaithersburg, MD.

Giaretta, D. 2007. The CASPAR Approach to Digital Preservation. *International Journal of Digital Curation* 2(1):112-121.

Hunter, J., and Choudhury, S. 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.

ISO/IEC 19775:2004. Information technology – Computer graphics and image processing – Extensible 3D (X3D).

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).

McMahon, C., and Browne, J. 1996. *CADCAM: From Principles to Practice*. Wokingham: Addison-Wesley.

Oliva, R., and Kallenberg, R. 2003. Managing the transition from products to services. *International Journal of Service Industry Management* 14(2):160–172.

Shah, J. J., and Mäntylä, M. 1995. *Parametric and Feature-Based CAD/CAM*. Chichester: Wiley.

Shene, C.-K. 1997. A user guide to the surface subsystem of DesignMentor. Available from: `http://www.cs.mtu.edu/~shene/COURSES/cs3621/LAB/surface/surface.html` (2008-08-07).

Shene, C.-K. 2007. Introduction to computing with geometry notes. Available from: `http://www.cs.mtu.edu/~shene/COURSES/cs3621/NOTES/` (2008-08-07).

US Product Data Association. 1996. Initial Graphics Exchange Specification (IGES) v5.3. Withdrawn standard ANS US PRO/IPO-100-1996, ANSI.

Wilson, A. 2007. InSPECT significant properties report. Technical report, Arts and Humanities Data Service/National Archives.