Annotation of lightweight formats for long-term product representations

Lian Ding[1], Alex Ball[2], Jason Matthews[1], Chris McMahon[1], and Manjula Patel[2]

[1]Innovative Design & Manufacturing Research Centre, Department of Mechanical Engineering, University of Bath, Bath, U.K.
[2]UKOLN, University of Bath, Bath, U.K.

17th September 2008

Abstract

Companies operating in today's global economy are increasingly expected to manage the entire lifecycle of their products, and are finding advantage in a distributed, collaborative working style. Existing 3D CAD systems are not well adapted to this way of working, however. This paper highlights some limitations in the current applications, and presents a framework for overcoming them based on three strands of current research: lightweight representations, the annotation of CAD models, and representation information as defined by the Open Archival Information System Reference Model (ISO 14721:2003). In the proposed framework, a 'stand-off' method is used to layer information, in the form of annotations, on top of both CAD models and lightweight representations alike. These annotations can be circulated independently of the geometry, facilitating more flexible information flow across the whole product lifecycle. The approach is demonstrated with an industrial case study.

Keywords. Lightweight representation, markup method, Representation Information Registry/Repository, PLM, CAD model.

1 Introduction

With the advances in wireless communication and the Internet, the economy has become more global than ever. More and more engineering companies have realized design and manufacturing automation are not enough to succeed in this unprecedented, competitive market: the entire product lifecycle and the product's long-term sustained competitiveness must be considered. Meanwhile, globalization demands radically altered business processes. Collaboration between groups within and across different companies is inevitable, so as to take full advantage of the variations that exist between the different

ACKNOWLEDGEMENTS

The work reported in this paper has been supported by a number of grants for Engineering and Physical Sciences Research Council (EPSRC), involving a large number of industrial collaborators. In particular, current research is being undertaken as part of the EPSRC Innovative Manufacturing Research Centre at the University of Bath (reference GR/R67507/01). The authors gratefully express their thanks for the advice and support of all concerned.

This is an electronic version of an article published in the International Journal of Computer Integrated Manufacturing, Volume 22, Issue 11 (November 2009), pages 1037-1053. The International Journal of Computer Integrated Manufacturing is available online at: `http://www.informaworld.com/smpp/openurl?genre=article&issn=0951%2d192X&volume=22&issue=11&spage=1037`

regions of the world (e.g. sharing knowledge and innovations from different regions, cheap skilled labour in developing countries). Although various Product Lifecycle Management (PLM) systems have been developed (e.g. Agile, IBM/Dassault, MatrixOne, PTC, and UGS PLM), product information is still mainly stored in CAD models based on conventional product representations, including boundary representation (B-rep), freeform surface modelling, feature-based models or parametric-based models. CAD models aid designers during product development, but they cannot easily be used to support the whole product lifecycle without the following significant challenges being met.

Protection of intellectual property of a company. 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. However, the CAD models carry most of the important information and knowledge. In a collaborative situation, companies need to work with each other, but they 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.

Communication and sharing of product information. There are many commercial CAD software systems, each of which has its own proprietary format. For reasons of cost and administrative overhead, amongst others, it is not feasible for every user to install a copy of each CAD system to view or manipulate product models in their native representations, so companies must overcome software barriers by translating between formats. Even so, 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 et al. 2002).

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. Available from: `http://www.rti.org/pubs/STEP_Transport.pdf` (Accessed 01 February 2008).

A further barrier is the amount of resources required to edit and transfer full CAD models. For example, for a simple component such as a crankshaft, the file size may be 1.31 MiB in one leading CAD system and 5.4 MiB in another. Hundreds of such components may be included in a product (e.g. a ship), leading to file sizes unsuited to transmission between geographically distributed applications and users.

Long-term preservation. Since the first graphical system – the US Air Force's SAGE (Semi Automatic Ground Environment) air defence system – appeared in the mid-1950s (Bozdoc 2004), CAD technology has been rapidly developed. With enormous improvements in the performance of CAD systems, the readability of old proprietary CAD formats also becomes a more serious issue. Due to the ephemeral nature of CAD file formats and the applications that work with them, there is a real danger that the CAD models in which so much product

Bozdoc, M. 2004. The History of CAD. 2004-01-03. Available from: `http://mbinfo.mbdesign.net/CAD-History.htm` (Accessed 01 February 2008) data is encoded may prove impossible to read in the future (Kuny 1997, Ball and Ding 2007). The issue of long-term archives and information retrieval must be taken seriously.

Kuny, T. 1997. A digital Dark Ages?: Challenges in the preservation of electronic information. *Proceedings of the 63rd IFLA General Conference.* Copenhagen. The Hague: International Federation of Library Associations & Institutions. Available from: http://www.ifla.org/IV/ifla63/63kuny1.pdf (Accessed 01 February 2008).

Ball, A. and Ding, L. 2007. Proceedings of the Atlantic Workshop on Long Term Knowledge Retention 2007. kim40mee002ab10.pdf. KIM Project. Available from: http://www.ukoln.ac.uk/events/ltkr-2007/proceedings/ (Accessed 01 February 2008).

A number of avenues of research are currently being explored in response to these challenges, among them:

- lightweight representations that allow product information to be browsed, retrieved and manipulated by users at different stages of the product lifecycle and across geographically dispersed sites, without risking valuable intellectual property;
- the annotation of CAD models with extra information (metadata, application-specific information, etc.) allowing the same data to be shared by engineers with different viewpoints; and
- collections of representation information that assist in keeping product data readable, understandable and reusable long into the future.

This paper proposes a new approach to integrate the advantages of these three methods in order to address the issues of long-term preservation and the product lifecycle support. The subsequent content is organized as follows: sections 2 and 3 review the current status of lightweight representations and techniques for annotation and markup. Section 4 presents a new strategy of extending CAD models to support product lifecycle management and long-term preservation by combining the concept of lightweight representations with markup techniques and representation information repositories. Section 5 describes an implementation of the proposed approach and a case study from industry is given in section 6. Finally, section 7 provides conclusions and summarizes further work.

2 Lightweight representations

Although some researches on lightweight product visualisation are carried out by both of academic (e.g. Hwang et al. 2007, Hwang et al. 2008) and standard society (e.g. ISO/TC184/SC4N2243 2007), main players on lightweight representations are commercial software community. There are a number of different commercial lightweight representations in current use, each with properties and characteristics suited for some applications better than others. In this section we introduce a number of these formats, with particular regard to their capabilities with respect to: fidelity to the full model, metadata storage, data security, file size reduction, support for the format by software, and openness. The information is summarized in Table 1.

Hwang, J., Mun, D. and Han S., 2007. Neutral reference model for engineering change propagation in global top-down modelling approach. *International Journal of CAD/CAM,* 7, Paper Number 9. Available from: http://www.ijcc.org/sub_02_vo7.php (Accessed 09 September 2008).

Hwang, J., Mun, D., Kim, B. and Han, S., 2008. Securing enterprise intellectual property using a skeleton model in a collaborative product design environment. *Proceedings of 5th International Conference on Product Lifecycle Management (PLM08),* Seoul, Korea.

ISO/TC184/SC4N2243, 2007. SC4 Proposed Document (SC4N2243), Visualisation ad hoc group report, Industrial requirements for product data visualisation.

2.1 3D XML

3D XML (Versprille 2005, Dassault Systèmes 2007) is an XML-based format for describing a model's geometry, structure and visualization, and is optimized for interactivity and compactness. It can represent geometry using compact NURBS-like surface descriptions, XML polygon meshes and compact syntax polygon meshes, but does not

Versprille, K. 2005. Dassault Systèmes' strategic initiative: 3D XML for sharing product information. *Technology Trends in PLM.* Stamford, CT: Collaborative Product Development Associates. Available from: http://www.3ds.com/uploads/tx_user3dsplmxml/3DXML_for_sharing_product_information.pdf (Accessed 01 February 2008).

Dassault Systèmes, 2007. 3D XML User's Guide. 0th ed. Version 4.0.

Table 1: Summary of the characteristics of selected lightweight representations.

Format	Feature	
3D XML	*Model Fidelity:*	Exact surfaces, polygon meshes
	Metadata support:	None
	Security features:	Data approximation
	File Size:	Reference-instance, instance modification, some compression
	Software support:	Dassault Systèmes products, Lotus Notes, Microsoft Word/PowerPoint, Internet Explorer, free viewer
	Openness:	Proprietary specification is cost-free to view
HSF	*Model Fidelity:*	NURBS surfaces, polygon meshes
	Metadata support:	Arbitrary user data, text objects
	Security features:	Data approximation
	File Size:	Data compression, streaming
	Software support:	Autodesk, Dassault Systèmes and PTC products
	Openness:	Proprietary specification is cost-free to view and implement
JT	*Model Fidelity:*	B-Rep, polygon meshes
	Metadata support:	Arbitrary user data, PMI
	Security features:	Data approximation
	File Size:	Reference-instance, data compression
	Software support:	UGS products, Microsoft Word/Excel/PowerPoint, free viewer
	Openness:	Proprietary specification is cost-free to view and implement, toolkit can be purchased
PLM XML	*Model Fidelity:*	NURBS surfaces, 2D and 3D vector graphics, feature modelling
	Metadata support:	Arbitrary user data, design or manufacturing notes, dimension information, surface finish information, mass and material information, text objects
	Security features:	Data approximation, access restriction
	File Size:	Reference-instance
	Software support:	UGS applications
	Openness:	Proprietary schemata are free to view, implement and extend; toolkit can be purchased
PRC	*Model Fidelity:*	B-rep, polygon meshes
	Metadata support:	Aribtrary user data, PMI, annotations
	Security features:	Data approximation
	File Size:	Reference-instance, data compression
	Software support:	TTF converters, Adobe PDF software
	Openness:	Proprietary, though may become ISO standard
U3D	*Model Fidelity:*	NURBS surfaces, triangle meshes
	Metadata support:	Aribtrary key/value data
	Security features:	Data approximation
	File Size:	Reference-instance, some compression
	Software support:	Adobe PDF software
	Openness:	ECMA standard, cost-free to view
X3D	*Model Fidelity:*	NURBS surfaces, polygon meshes, 2D and 3D vector graphics
	Metadata support:	Arbitrary key/value data
	Security features:	Data approximation
	File Size:	Reference-instance
	Software support:	Various open source and proprietary viewers and processors, e.g. Xj3D, Flux, BS Contact
	Openness:	ISO standard, cost-free to view, open source libraries
XGL/ZGL	*Model Fidelity:*	Triangle meshes
	Metadata support:	None
	Security features:	Data approximation
	File Size:	Reference-instance, whole-file compression
	Software support:	Autodesk, various minor CAD products
	Openness:	Specification no longer maintained

have any additional security features. File sizes are kept down by a reference-instance mechanism (allowing the same data to be re-used several times within a model), a modification mechanism (allowing an instance or reference object to build on the properties of another reference object) and raster graphic compression. Models may be expressed by a single file or split across several files. Native support for the format is largely restricted to Dassault Systèmes products, although free plugins are provided for Lotus Notes and Microsoft Word, PowerPoint and Internet Explorer, as well as a free standalone viewer. The format is owned and controlled by Dassault Systèmes; the specification for the format is available cost-free to those who register with the Dassault Systèmes website.

2.2 HOOPS Stream Format (HSF)

HOOPS Stream Format (Open HSF Intiative 2008) is a binary format for encoding both 2D and 3D geometry using tessellating polygons and (since version 7) NURBS surfaces; it also supports arbitrary user data, text and, by means of an OpenHSF extension to the format, model structures. It does not have any in-built security features other than data approximation. The format permits streams within files to be zlib-compressed, and as the name suggests it can be streamed. It is supported by a number of CAD vendors including Autodesk, Dassault Systèmes and PTC. The format is owned by Tech Soft 3D, but the specification is freely accessible on the Web and the licence to implement it is free.

Open HSF Initiative. The HOOPS 3D Product Suite. Available from: `http://www.openhsf.org/docs_hsf/index.html` (Accessed 01 February 2008).

2.3 JT Format

JT Format (UGS 2006) is a binary format for encoding product geometry using boundary representations and wireframes, and supports additional product manufacturing information and other metadata. It does not have any in-built security features other than approximating data using tessellating polygons. File sizes are kept down using a reference-instance mechanism, zlib compression of various data elements and datatype-specific compression using algorithms such as uniform data quantization, bitlength codec, Huffman codec, arithmetic codec, and Deering Normal codec. Models may be expressed by a single file or split across several files. Native support for the format is largely restricted to UGS products, although free plugins are available for Microsoft Word, Excel and PowerPoint, as well as a free standalone viewer. The format is owned by UGS, but the specification is freely accessible on the Web and blanket permission is given to implement it.

UGS, 2006. JT File Format Reference. Version 8.1. Available from: `http://www.jtopen.com/docs/JT_File_Format_Reference.pdf` (Accessed 01 February 2008).

2.4 PLM XML

PLM XML (UGS 2005) is a set of XML schemata for describing a model's geometry, structure, features, ownership, and visualization. It is designed to be interoperable between a number of different tools from across the lifecycle of a product. The native schemata for representing geometry can support 2D and 3D vector graphics, NURBS

UGS, 2005. Open product lifecycle data sharing Using XML. White Paper. url: `http://www.ugs.com/products/open/plmxml/docs/wp_plm_xml_14.pdf` (Accessed 01 February 2008).

surfaces and features, although non-native representations can also be used or referenced in a PLMXML document. It also allows for a single logical product model to have several different geometric representations, tailored to different purposes. Metadata of several different types – mass, material, texture, product manufacturing information, dimensions and tolerances, user markup, application-specific data – can be attached to logical parts of the model or specific geometric representations. File sizes can be reduced using a reference-instance mechanism and by splitting out various sections of data into separate files (so that data not needed for a particular purpose need not be transmitted). As well as approximating and subsetting data, PLM XML also supports mechanisms for restricting access to parts of the model data on the basis of person, organization or place. The format is used extensively by UGS products but is not widely supported otherwise. The format is owned and controlled by UGS; the XML schemata are freely accessible on the Web, but the software development kit must be purchased.

2.5 PRC

PRC (Adobe Systems 2007b) is a binary format that promises to encode the full range of CAD geometry, along with model trees, history trees and various forms of markup (Trade and Technologies France, 2006). Alternative geometries (e.g. exact and tessellated) can be provided for each part; markup can be associated with entire parts or tessellations but not with items of exact geometry. Arbitrary non-PRC data can be included at various points, notably at the end of entity code. Summary data sections enable files to be accessed without being fully parsed, but the format is not suitable for streaming. File sizes are reduced through a number of mechanisms: compact mathematical encoding of geometry, a reference-instance mechanism, and gzip encoding of data sections (header sections remain uncompressed). The precision of the geometry may be reduced to provide lossy compression. Proprietary converters are available for a wide range of CAD formats, and PRC is supported as a native 3D model format within the Portable Document Format (PDF) specification from version 1.7 (corresponding to Adobe Acrobat 8.1), which adds some conservative security measures on top of the otherwise unprotected format (Adobe Systems 2007a). The format was initially proprietary but is expected to form part of ISO 32000.

Adobe Systems, 2007b. PRC Format. Version 7094. Available from http://www.adobe.com/cfusion/entitlement/index.cfm?e=acrobat_prc_spec (Accessed 01 February 2008).

Trade and Technologies France. TTF to unveil PRC II compression technology: Release planned for January 2006. Press release. Available from: http://www.ttf-group.com/articles/PRCII.pdf (Accessed 01 February 2008).

Adobe Systems, 2007a. PDF Reference and Related Documentation. Adobe Acrobat SDK version 8.1. Available from: http://www.adobe.com/devnet/acrobat/pdfs/pdf_reference.pdf (Accessed 01 February 2008).

2.6 Universal 3D (U3D)

Universal 3D (ECMA-363 2007) is a binary format for encoding product geometry using sets of tessellating triangles and (from the 4th edition) NURBS surfaces. A mesh update mechanism allows meshes to be rendered progressively, providing basic streaming support. Metadata, stored as key/value pairs, may be attached to any node in the model tree. It does not have any in-built security features other than approximating the geometry. File sizes are kept down using a reference-instance mechanism and a bit compression algorithm on numeric data

ECMA-363, 2007. Universal 3D File Format. 4th ed. Available from: http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-363%204th%20Edition.pdf (Accessed 01 February 2008).

fields. The format is most notably supported as a native 3D model format within the Portable Document Format (PDF) specification, with the 1st edition of U3D supported from version 1.6 (corresponding to Adobe Acrobat 7) and the 3rd edition supported from version 1.7 (corresponding to Adobe Acrobat 8.1). PDF also adds some conservative security mechanisms of its own (Adobe Systems 2004). It was developed by the 3D Industry Forum and is published and maintained as ECMA standard 363; the specification is freely available on the Web.

Adobe Systems, 2004. PDF Reference, Fifth Edition: Adobe Portable Document Format Version 1.6. Available from: http://www.adobe.com/devnet/pdf/pdfs/PDFReference16.pdf (Accessed 01 February 2008).

2.7 X3D

X3D (ISO/IEC 19775 2004, ISO/IEC 19776 2005, ISO/IEC 19777 2006) is an improved version of Virtual Reality Markup Language (VRML); it is an XML format optimized for animation and interaction. It can represent 2D and 3D vector graphics, 3D tessellating polygon meshes, and NURBS surfaces as well as identifying bones and joints for human animation. Any node in the model tree may have metadata attached, in a format specifying a value (string or number), a metadata schema and a key. It does not have any in-built security features other than data approximation. X3D has a reference-instance mechanism and a relatively compact XML syntax, with coordinates expressed as space/comma delimited lists within attributes, rather than through a hierarchy of tags; a binary syntax is available that compresses field values according to Fast InfoSet principles, using zlib compression, quantization of floating point number arrays, integer range reduction and conversion of absolute values to relative values. Open source libraries and viewers are available for processing and rendering X3D files. X3D was developed by the Web 3D Consortium, and is published and maintained as ISO standards 19775, 19776 and 19777; these standards are freely available on the Web.

ISO/IEC 19775, 2004. Information technology – Computer graphics and image processing – Extensible 3D (X3D). Available from: http://www.web3d.org/x3d/specifications/ISO-IEC-19775-X3DAbstractSpecification/ (Accessed 01 February 2008).

ISO/IEC 19776, 2005. Information technology – Computer graphics and image processing – Extensible 3D (X3D) encodings. Available from: http://www.web3d.org/x3d/specifications/ISO-IEC-19776-X3DEncodings-XML-ClassicVRML/ (Accessed 01 February 2008).

ISO/IEC 19777, 2006. Information technology – Computer graphics and image processing – Extensible 3D (X3D) language bindings. Available from: http://www.web3d.org/x3d/specifications/ISO-IEC-19777-X3DLanguageBindings/ (Accessed 01 February 2008).

2.8 XGL/ZGL

XGL (XGL Working Group 2006) is an XML-based encoding of the Open Graphics Library (OpenGL) application programming interface for rendering 2D and 3D computer graphics. When compressed it is known as ZGL. It uses tessellating triangles to encode geometry, and is optimized for display. It does not have any capabilities for storing metadata, nor does it have any in-built security features other than approximating the geometry. File sizes are kept small using a reference-instance mechanism and a relatively compact XML syntax, with vector coordinates expressed as comma delimited lists rather than through a hierarchy of tags. XGL is supported by Autodesk and a few smaller CAD vendors. It was developed by the XGL Working Group but no longer appears to be maintained; the specification of the format was once freely available on the Web, but now only appears in 'unofficial' locations.

XGL Working Group, 2006. XGL File Format Specification. Available from: http://web.archive.org/web/20060218/http://www.xglspec.org/ (Accessed 01 February 2008).

3 Annotation techniques

3.1 Definition of annotation and markup

Annotation may be defined as the act of adding explanatory notes (Compact Oxford English Dictionary of Current English 2005). These explanatory notes can have various purposes, such as commentary on, or viewpoint interpretation or extra description of, the existing entity. With the rapid development of digital technology, annotation is becoming an important tool for information communication, retrieval and management. Currently, annotation is being widely used in different digital items, including text documents, structured data (e.g. databases and tables), 2D images, 3D models and multimedia (e.g. video and audio) presentations.

Compact Oxford English Dictionary of Current English, 2005. 3rd ed. Oxford: Oxford University Press. ISBN: 978-0-19-861022-9.

The definition of markup differs between domains such as the publishing industry, document management, or the world of computer science. For the purposes of this paper, markup is regarded as a subtype of annotation and defined as a formally structured annotation for a purpose, normally to allow some kind of manipulation of the information entity. Under this definition, there are five basic types of markup (Coombs et al. 1987, Khare and Rifkin 1998):

Coombs, J. H., Renear, A. H. and DeRose, S. J., 1987. Markup systems and the future of scholarly text processing. *Communications of the Association for Computing Machinery,* 30(11), 933–947. ISSN: 0001-0782. Available from: `http://xml.coverpages.org/coombs.html` (Accessed 01 February 2008).

Khare, R. and Rifkin, A., 1998. The origin of (document) species. *Computer Networks and ISDN Systems,* 30(1-7), 389–397. Available from: `http://www.ifindkarma.com/attic/papers/www/origin-of-species.html` (Accessed 01 February 2008).

- *Punctuational markup.* Word, phrase, and sentence boundaries are identified by spaces, commas, full stops, and other punctuation characters inserted into the text.
- *Presentational markup.* The visual form of the document is specified directly.
- *Procedural markup.* Presentational instructions (or commands) for some particular processing system are embedded in the document.
- *Descriptive markup.* The structure of the document and the semantics of the various document elements are identified using tokens such as the tags found in SGML and XML.
- *Meta-markup.* A facility for controlling the interpretation of markup and for extending the vocabulary of descriptive markup languages (e.g. macros).

Of the above types of markup, descriptive markup has the most advantages; for example, it allows users to focus on the content of a document and its structure, it has greater portability, and it makes documents easier to maintain in the face of, for example, corporate stylistic changes. Indeed, the power of descriptive markup is the separation of how data is stored and how it is used. Different contexts require different software for the processing or rendering of data, and each of these different pieces of software has its own preferred data structures and formats. Ideally, though, the data should have a single authoritative form from which the others can be derived by transformation, and having a base form encoded using descriptive markup is programmatically the easiest way to achieve this. Thus, descriptive markup is probably the best solution for manuscript composition and distribution (Coombs et al. 1987).

3.2 Markup methods

Almost all markup strategies can be sorted into one of two categories, according the method used to apply the markup to the document:

- *'Inline' markup method.* The most common method for applying markup is to insert markup tokens directly into the text of the document. This is easy to accomplish using a text editor or a few lines of scripting language. However, the 'inline' markup method does actually change the document. More importantly, it is difficult to place multiple independent sets of markup in the same document as the syntax of the tokens used may easily interfere with one another (Tennison 2007).
- *'Stand-off' markup method.* In contrast, 'stand-off' markup is stored separately utilizing a system of references or pointers to indicate to which parts of the document the markup refers (TEI Standoff Markup Working Group 2003, Thompson and McKelvie 1997). The method is more suitable for applying multiple sets of markup to the same instance of a document, but implementing it is complex as it requires a persistent system of reference back to the document.

3.3 Markup languages

A markup language is a formalized way of providing markup (Johnston 1998). The first structured markup language was a formatting language for document publishing, GML (Generalized Markup Language), developed by three IBM researchers in 1969 (Goldfarb 1990). GML evolved into SGML (Standard Generalized Markup Language) and became an international standard in 1986. Since HTML (HyperText Markup Language) – a subset of SGML – appeared in 1990, markup languages have developed quickly and there are dozens or maybe hundreds of markup languages in use today (Liu et al. 2008). The most common computer-interpretable markup languages are HTML and XML (eXtensible Markup Language). Both HTML and XML use tags to identify sections of text within a document. The difference is that the tags in HTML are defined along with the process for handling them, while XML allows the user to specify separately: valid tags, valid structure for data contained in elements, the valid type of data that each element may contain, and what manipulations should be applied to elements identified by the tags. Thus, XML has far more flexibility than HTML in how it is processed in different contexts, e.g. presenting a different subset of the information according to the purpose, or combining information from different documents.

Forms of annotation and markup have long been a part of engineering practice. Annotation is used to aid communication, both directly (e.g. engineers annotating a depiction of a design during a face-to-face discussion of the product) and indirectly (e.g. an engineer sending an annotated depiction of the product to a colleague). Markup – formally structured annotation – has also been used to augment engineering depictions of products with information that is difficult to represent unambiguously pictorially, such as information required for

Tennison, J., 2007. Creole: Validating overlapping markup. In: *XTech 2007: The Ubiquitous Web.* Paris. Alexandria, VA: IDEAlliance. Available from: http://2007.xtech.org/public/schedule/paper/46 (Accessed 01 February 2008).

TEI Standoff Markup Working Group, 2003. Stand-off markup. Working paper SO W 06. Text Encoding Initiative. Available from: http://www.tei-c.org/Activities/SO/sow06.xml?style=printable (Accessed 01 February 2008).

Thompson, H. S. and McKelvie, D., 1997. Hyperlink semantics for standoff markup of read-only documents: The next decade – pushing the envelope. *Proceedings of SGML Europe '97.* Barcelona. Graphic Communications Association. 227–229. Available from: http://www.ltg.ed.ac.uk/~ht/sgmleu97.html (Accessed 01 February 2008).

Johnston, P., 1998. What you have always wanted to know about SGML, HTML and XML but were afraid to ask: Why markup matters. Presentation to the Society of Archivists' Diploma in Archive Admininstration seminar, Edinburgh, 30 November 1998. Available from: http://www.gla.ac.uk/InfoStrat/socarcpj/ (Accessed 01 February 2008).

Goldfarb, C. F., 1990. *The SGML Handbook.* Ed., with a forew., by Y. Rubinsky. Oxford: Oxford University Press. ISBN: 978-0-19-853737-3.

Liu, S., McMahon, C. A. and Culley. S.J., 2008. A review of Structured Document Retrieval (SDR) technology to improve information access performance in engineering document management. *Computers in Industry,* 59(1), 3–16. DOI: 10.1016/j.compind.2007.08.001

manufacturing a product – tolerances, machining processes, surface finishes etc. Formally specified systems of markup symbols are normally recorded in some kind of standard, such as ASME Y14.41-2003. Work on the annotation and markup of CAD models also has been carried out by some researchers. For example, Elinson, Nau and Regli (1997) represent machining features using a graph whose nodes and edges correspond to features and their relationships; the nodes and edges are annotated with labels containing various parameters that may be useful for classification purposes. Hoffmann and Joan-Arinyo (1998) propose an architecture for a product master model, federating CAD systems with downstream application processes for different feature views; they especially address the need to make persistent associations of design information with net shape elements. Davies and McMahon (2006) explore the application of a markup approach to the attachment of information to CAD models and its subsequent organization and manipulation.

ASME Y14.41, 2003. Digital Product Definition Data Practices. ISBN: 978-0-7918-2810-6.

Elinson, A., Nau, D. S. and Regli, W. C., 1997. Feature-based similarity assessment of solid models. *Fourth Symposium on Solid Modeling and Applications.* New York, NY. Ed. By C. Hoffman & W. Bronsvoort. Atlanta, GA: ACM Press. 297–310.

Hoffmann, C. M. and Joan-Arinyo, R., 1998. CAD and the product master model. *Computer-Aided Design,* 30(11). 905–918. DOI: 10.1016/S0010-4485(98)00047-5.

Davies, D. and McMahon, C. A., 2006. Multiple viewpoint design modelling through semantic markup. *Proceedings of IDETC/CIE 2006, ASME 2006 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference.* Philadelphia, PA.

4 Product representations for a product lifecycle

Although lightweight representations have already shown benefits for collaborative product development, most of the representations developed so far can only be regarded as lightweight 3D visualizations and their applications are still limited. To address the problems of both lightweight representations and CAD models (discussed in section 1), a strategy of product representation for the product lifecycle is proposed, integrating lightweight representations, a markup method and a Representation Information Registry/Repository with a CAD model technique.

4.1 Proposed strategy

As shown in Figure 1, the proposed strategy allows:

- all users throughout all stages to mark up the CAD model according to their own experience and knowledge without directly changing the CAD model. As it does not allow CAD models to be accessible to all users, it avoids problems associated with managing multiple copies and versions of the same design and becomes acceptable with regard to security control;
- markup information to be recorded in a series of separate markup files written in XML format. Each markup file is based on a certain view for users or editors and linked to the CAD model through a specific element of the model. The elements of the CAD model cover different levels of detail from the assembly and parts to detailed parameters and tolerances. Such linkage mechanisms can avoid the CAD model becoming too heavy and help to speed the process when several users mark up the same CAD model at the same time;
- all users to retrieve information and knowledge about the CAD model by means of a lightweight representation corresponding

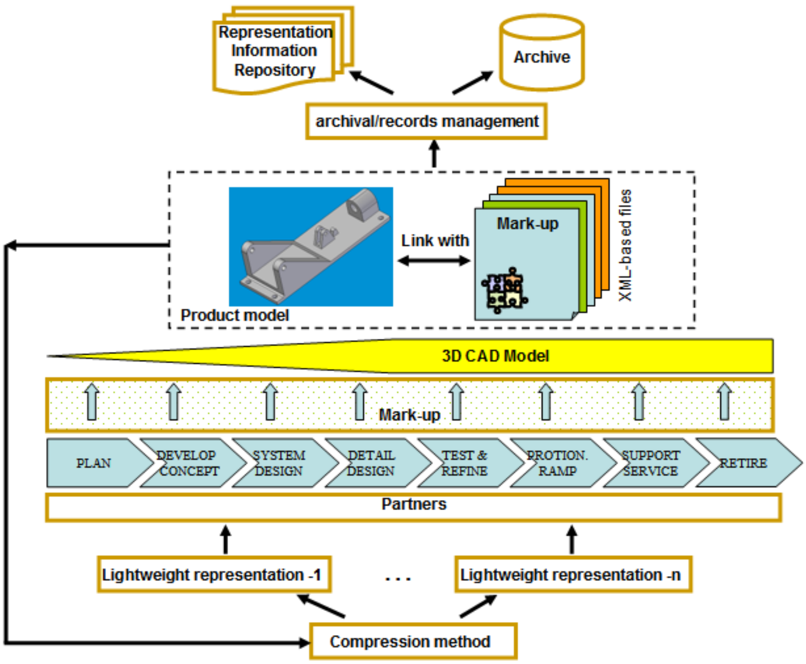


Figure 1: Framework of lightweight representations in product lifecycle

to the user's security level and viewpoint, accompanied by one or more markup files;

- data managers to keep track of which formats would be most suitable for particular viewpoints and purposes, and to support file recovery procedures. This is accomplished through the deposition, maintenance and subsequent retrieval of relevant representation information for the CAD model and lightweight representation formats, as well as the XML schemata for markup documents, within a Registry/Repository system.

4.2 Markup of CAD models

As reviewed in section 3, there are two methods for applying markup: the 'inline' markup method and the 'stand-off' markup method. 'Inline' markup has been widely used in digital items, such as text documents and 2D models. However, as the 'stand-off' method allows markup information to be stored in separate documents and linked back to the model using references or pointers, it is more likely to be suitable for CAD models. Firstly, it allows a 3D geometrical representation of a product to be progressively expanded to include more metadata without changing the representation method used for the geometry of the product. Secondly, with the 'stand-off' approach, the CAD model itself need not contain all the information needed for every user and purpose: context-specific information can be extracted out into a number of separate files (i.e. multi-layer markup) and passed around only as needed, allowing the CAD model to remain smaller in size. Thirdly, it allows the same markup to applied to different representations of the same model, granting the markup information some independence of the CAD format used. Fourthly, and significantly, it allows downstream processes (e.g. finite element analysis [FEA], manufacturing processes) to be independent of the CAD model by re-using markup; with transitional methods, like models associated with CAD, the process is only re-usable for invariant topologies.

In general, markup has particular utility for the following tasks:

- The insertion of the extra information that is needed for a certain point of view.
- The embedding of a commercial security level, restricting the access of certain partners or users to the data.
- The identification of the pieces of information in a CAD model that are necessary for a certain point of view.

The structure of the XML schema for recording markup for a product model is designed to be flat and to concentrate on recording metadata that will allow subsequent structuring and manipulation of the information:

- *Header section* (`document_header`). The `document_header` records the metadata for the markup file and the corresponding references to the CAD model. It includes: `model_file`, recording the path and file name of the CAD model that the markup refers to; `geometry_reference_list` enumerating the specific elements (i.e. faces, edges, bodies) that the markup links to by `geometric_element_type` and `persistent_id`; editor, indicating who has edited the markup document (name) and the group he/she belongs to (group); the dates of creation (`create_date`) and latest modification (`modify_date`); and `security_level`, which can be used to restrict the audience of the information.
- *Main body* (`markup_element`). The `markup_element` describes the detailed markup information. It consists of: `markup_type`, specifying whether the markup element is free-text, data, or a URI linking to resources like documents, other product models, databases, structured data or a combination of these; viewpoints, referring to what particular viewpoints the markup information belongs to, such as manufacturing process planning; and `markup_item`, recording the detailed markup information using a special structure. The structure for the `markup_item` is open for users to define as the number of different possible structures of product information throughout a product lifecycle is effectively infinite.

4.3 Representation information registry/repository

'Representation information' is a term that comes from the Open Archival Information System (OAIS) Reference Model (ISO 14721 2003), an international standard model for describing the activities of data repositories and other long-term stores of information. It is defined as the information required to turn a data object (commonly a stream of bits) into something meaningful, and therefore includes such things as format specifications, data dictionaries, ontologies and sets of hardware and software known to be relevant to a format. In short, it is the information needed to keep a data object perpetually understandable. While digital data objects typically need some

ISO 14721, 2003, Space data and information transfer systems – Open archival information system – Reference model.

representation information peculiar to themselves, they also share representation information with objects of a similar format or type. To avoid having to rediscover and store this information every time it is needed, it can be stored in a persistent registry and linked to whenever needed. The UK's Digital Curation Centre (DCC) and the European CASPAR Project are working on a Registry/Repository of Representation Information (RRoRI) for objects from across the spectrum of culture, art and science. The framework proposed in this paper would use RRoRI to store relevant representation information for CAD models and lightweight visualization formats, as well as XML schemata for markup documents. We are developing representation information for storage in RRoRI, and a tool for the local storage and manipulation of representation information downloaded from RRoRI.

In the short term, the principal usefulness of representation information is to enable informed decision-making on which formats would be most suitable for particular users, viewpoints and purposes, and which tools would be most reliable for which types of processing. Clearly, for these purposes the value of the information is in having a sizeable collection that can be cross-searched, rather than in individual pieces of information. Later in the lifecycle, though, the same information can be used to support file recovery procedures and other interpretive activities. In this case, the persistence of the relevant, individual pieces of representation information is critical, with preserved software tools taking on perhaps more significance than specifications and other more descriptive types of information, though descriptions of former practice and terminology will certainly be useful.

5 Implementation

According to the proposed approach, the markup tools must be provided not only for the users who own the original CAD model (e.g. designers), but also for the users who only have access to its derived lightweight representations (e.g. production engineers and service engineers).

Most CAD markup methods are proposed within a CAD environment, such as UGS's NX or Dassault Systèmes' Catia, or using 'viewer' software (Cimmetry 2007). This is useful for communication within a collaborative design team, for example inserting comments to other designers so that they can understand the reasons for a certain design choice, or adding a certain viewpoint of the model for later retrieval. Such markup methods, however, usually insert markup information as attributes associated with CAD models, meaning the markup cannot be re-used independently of the CAD model. There is also the issue of the readability of old proprietary CAD formats – there is a danger of investing large amounts of product data in CAD models that may prove impossible to read in the future (BBC 2007). As a consequence, an internal markup environment has been developed. The internal markup environment is built on a shared library written in UGS's NX3 Open C API, whose symbols get resolved at run time;

Cimmetry Systems Corporation, 2007. Autovue 19.2, Available from: `http://www.cimmetry.com/_products/autovue_software_suite.html` (Accessed 17 July, 2008).

British Broadcasting Corporation, 2007. Warning of data ticking time bomb. Available from: `http://news.bbc.co.uk/1/hi/technology/6265976.stm` (Accessed 01 February 2008.

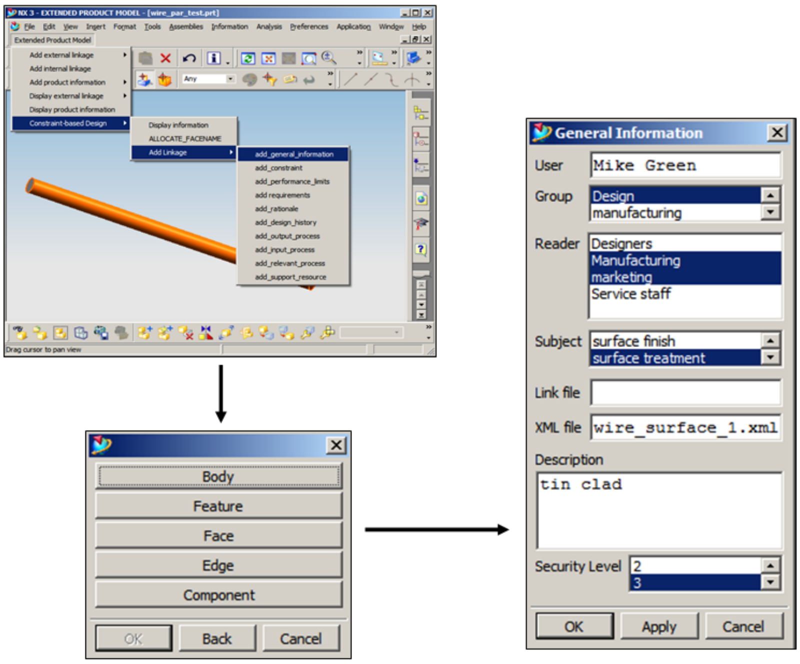


Figure 2: Interface of the internal markup environment of UGS's NX3

this occurs interactively within a NX session. Figure 2 shows how the internal markup environment allows users to mark up a CAD model within a CAD environment, including: choosing an entity (e.g. a body, a feature or a face) that the markup to be linked with, applying markup in a special structure according to different requirements and viewpoints, and recording the content of the markup in a separate XML document (with the schema discussed in section 4.2) linked by reference.

An external markup environment allows users to apply markup outside of the CAD environment, with the content of the markup being recorded solely in an XML document linked by reference both to the representation being used and back to the CAD model. Work has been carried out based on Adobe Acrobat. Since version 1.7 (corresponding to Acrobat version 8.1), Portable Document Format (PDF) has supported the display of two different formats of 3D model: U3D (editions 1 and 3) and PRC. Both Adobe Acrobat and the free Adobe Reader support a JavaScript API which allows one to extend the functionality of the 3D interface. In the API, an object with a 3D representation within a scene hierarchy is accessible as a node; Acrobat 8.1 introduced a new property of the scene object, `selectedNode`, which enables one to query which node in the tree currently has focus. Therefore, this could be used to generate an XML file containing details of the currently selected node alongside the comment type and text. The markup interface in Adobe Acrobat shown in Figure 3 has been developed; the markup content may be freely imported from and exported to an external XML document. The markup document can then be read and linked back to a specific entity (i.e. the entity with same label as the entity in the PDF model that the markup document is linked with) in the original CAD model through a transfer interface (as shown in Figure 4). The transfer interface is a standalone

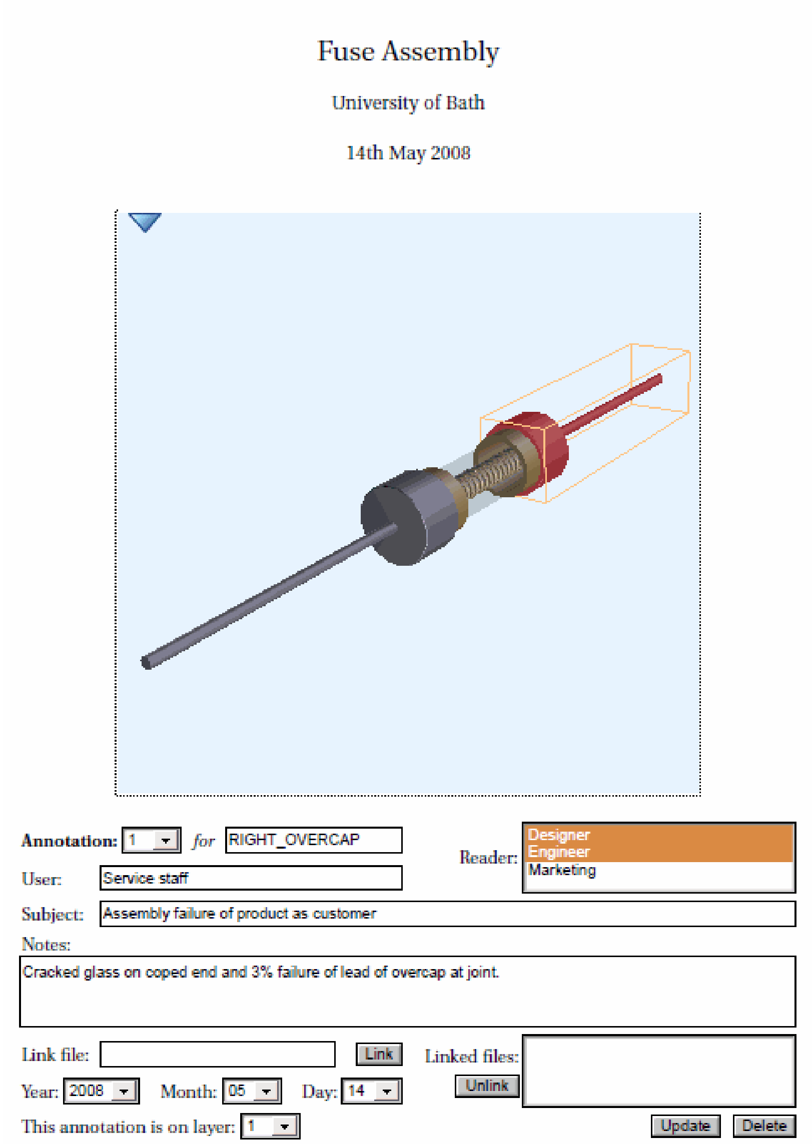


Figure 3: Markup interface in PDF

executable program developed and executed within Microsoft Visual Studio.NET. External UGS NX Open API functions are embedded so that the transfer interface can be linked using the same shared libraries as NX, but its symbols are resolved at link time rather than run time. With the transfer interface, the feedback information coming from the users, who only have access to the lightweight representation (e.g. U3D or PRC), can be retrieved through the CAD model directly. It is particularly important for designers to share the experiences and knowledge of the users at later stages of a product's lifecycle, on such

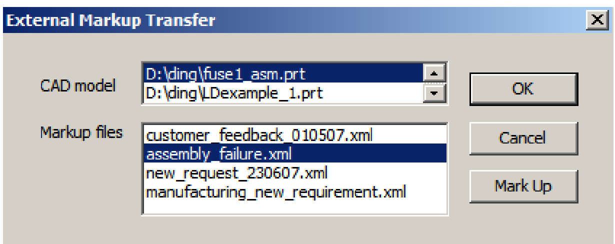


Figure 4: An interface of external markup transfer

matters as product performance, flexibility and failures (e.g. Matthews et al. 2006).

Matthews, J., Singh, B., Mullineux, G. and Medland, T., 2006. Constraint-based approach to investigate the process flexibility of food processing equipment. *Computers & Industrial Engineering*, 51(4), 809-820.

A basic representation information registry has been implemented in Java, with the representation information itself stored in the form of a set of XML documents. The documents are based around an ontology of significant characteristics; the documents relating to file formats detail the extent to which the format supports each characteristic, while the documents relating to software indicate how well each characteristic is preserved in each of the format transformations of which the software is capable. The registry interface allows one to search for formats and software based on their support for a given set of characteristics, figure 5 shows the interface for a combined search, retrieving both suitable destination formats and suitable migration pathways to those formats.

Figure 5: Representation information repository

6 Case study

As with most high volume manufacturing industries, in order to stay financially competitive fuse manufacturers are forced to operate in lower cost manufacturing areas; commonly US manufacturers use South American assembly companies and European manufacturers use companies in south east Asia and the Indian sub-continent.

This case study concerns a US-owned European company transferring its production to mainland China. Prior to the transfer, the manufacturer developed, assembled and packed its own fuse links. Only the raw components such as end caps, bodies, fuse elements, etc. were externally manufactured. Now, the assembly, testing and packaging of the fuse links can take place in companies on opposite sides of the globe. The manufacturer was keen to protect its intellectual property, namely the operational characteristics of the fuse link, and so only released partial information to each player in the manufacturer and user chain. This is shown in table 2.

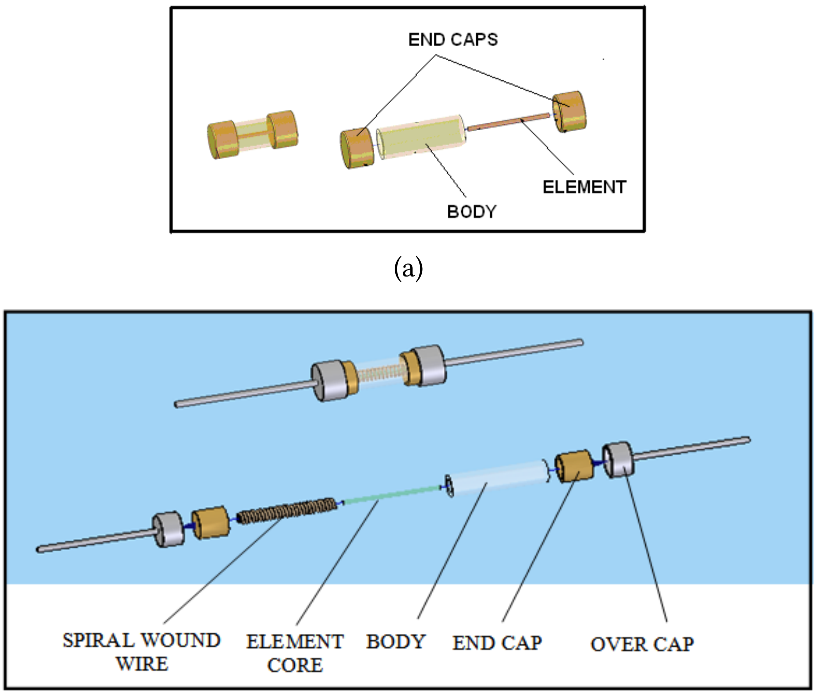
Figure 6 shows two examples of a fuse link. Typically, a fuse link is constructed from eight main parts: two end caps, two over

Table 2: Relevant information.

	Information								
	Geometry	Topology	Dimensions	Tolerance	Materials	Surface finish	Characteristics	Test criteria	Package volume
Development	◇	◇	◇	◇	◇	◇	◇	◇	◇
Part manufacture	◇		◇	◇	◇	◇			
Assembler		◇	◇	◇				◇	
Packager (+ Tester)			◇					◇	◇
Marketing			◇		◇		◇		◇
Customer			◇		◇		◇		
Service/Operation			◇	◇	◇		◇		

caps, a spiral wound wire, its fibreglass core, filling of glass spheres and a glass body. Each design has been developed to satisfy multiple users, each of whom has a different pack requirement, and some of whom also require 100% independent testing before they will accept the product.

- The part product manufacturer is only given the information it needs in order to produce the part; it is not given the full specifications for the end product.
- The assembling company is supplied with sufficient information to assemble the product to the customer's needs. The assembler is required to inspect and test the product in-process. To satisfy this, the company is issued with specific dimensions and tolerances along with required cold resistance values and over-current blowing times. Post manufacture, the assembling



(b)

Figure 6: Fuse link

company will bulk pack the product into boxes or reels for shipping to OEMs or to an independent testing company prior to shipment to the OEM.

- Most independent fuse-link test companies will also put the product into its primary and secondary packaging prior to shipment. The test companies are issued with testing criteria similar to those of the assembler. There may also be application specific tests which are not normally carried out in the process of production but which form part of the customer's requirements.
- The companies' marketing teams require an overview of the product's dimensions and operational characteristics. This will enable them to target markets and answer customer enquiries.
- The customer will be issued with the full operating characteristics of the product along with the material information about the product. If the product is supplied to a secondary customer, i.e. one for which the product was not originally designed, then some of the operational characteristics may be withheld.
- The manufacturer's customers and operational services are required to 'field' questions on any product which may have failed inexplicably or induced effects further down in the operational

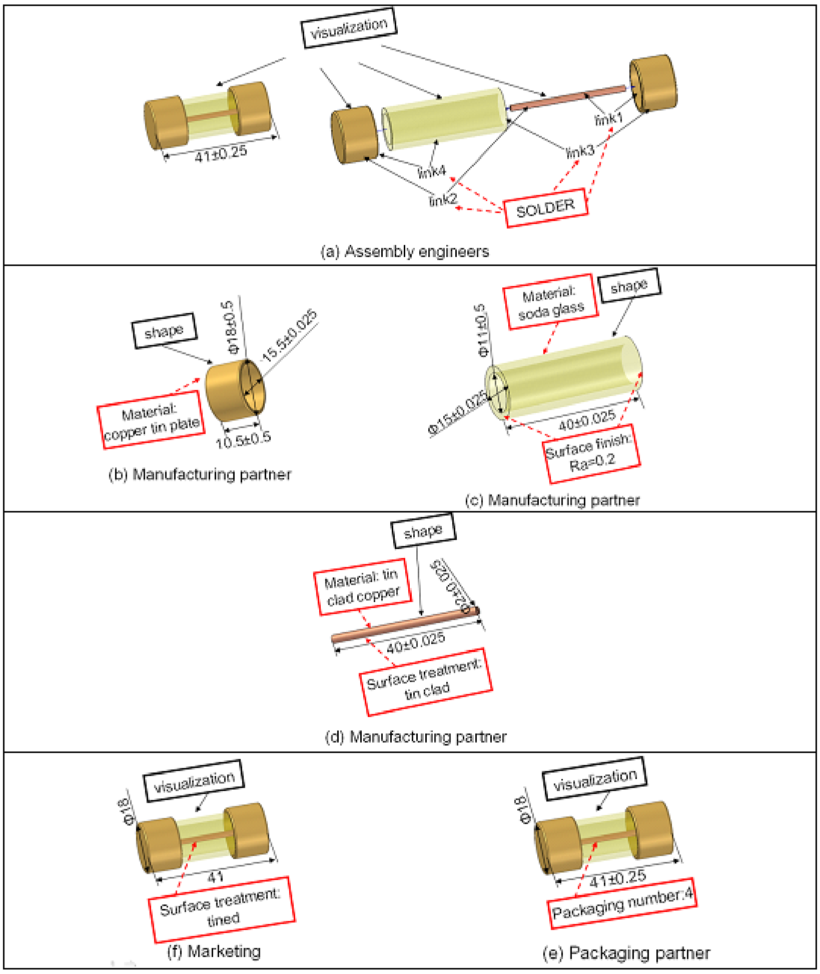


Figure 7: Markups for the fuse link

```xml
<?xml version="1.0" encoding="utf-8"?>
<markup_document xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="D:\Ding\markupschema.xsd">
  <document_header>
    <model_file>
      <file_name>element_core.prt</file_name>
      <filepath>D:/ding/</filepath>
    </model_file>
    <geometry_reference_list>
      <geometric_element>
        <persistent_id>body_001</persistent_id>
        <geometric_element_type>body</geometric_element_type>
      </geometric_element>
    </geometry_reference_list>
    <editor_list>
      <editor>
        <name>Mike Green</name>
        <group>Design</group>
      </editor>
      <document_security_level>3</document_security_level>
      <Date>
        <create_date>03/05/08</create_date>
      </Date>
    </editor_list>
  </document_header>
  <markup_element>
    <markup_type>free-text</markup_type>
    <viewpoints>
      <viewpoint>manufacturing</viewpoint>
      <viewpoint>marketing</viewpoint>
    </viewpoints>
    <markup_item>
      <surface_treatment>tin clad</surface_treatment>
    </markup_item>
  </markup_element>
</markup_document>
```


(a)


```xml
<?xml version="1.0" encoding="utf-8" ?>
<markup_document xmlns:xsi="
http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="D:\Ding\markupschema.xsd">
  <document_header>
    <model_file>
      <file_name>fuse_assembly.prt</file_name>
      <filepath>D:/ding/</filepath>
    </model_file>
    <geometry_reference_list>
      <geometric_element>
        <persistent_id>assembly_body</persistent_id>
        <geometric_element_type>body</geometric_element_type>
      </geometric_element>
    </geometry_reference_list>
    <editor_list>
      <editor>
        <name>Jane Lee</name>
        <group>Business</group>
      </editor>
    </editor_list>
    <document_security_level>4</document_security_level>
    <Date>
      <create_date>08/05/08</create_date>
    </Date>
  </document_header>
  <markup_element>
    <markup_type>text</markup_type>
    <viewpoints>
      <viewpoint>packaging</viewpoint>
    </viewpoints>
    <markup_item>
      <general_information>
        Packaging number is 4
      </general_information>
    </markup_item>
  </markup_element>
</markup_document>
```


(b)


```xml
<?xml version="1.0" encoding="utf-8" ?>
<markup_document xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="D:\Ding\markupschema.xsd">
  <document_header>
    <model_file>
      <file_name>tube.prt</file_name>
      <filepath>D:/ding/</filepath>
    </model_file>
    <geometry_reference_list>
      <geometric_element>
        <persistent_id>face_001</persistent_id>
        <geometric_element_type>face</geometric_element_type>
      </geometric_element>
    </geometry_reference_list>
    <editor_list>
      <editor>
        <name>Jason Matthews</name>
        <group>Design</group>
      </editor>
    </editor_list>
    <document_security_level>2</document_security_level>
    <Date>
      <create_date>06/05/08</create_date>
    </Date>
  </document_header>
  <markup_element>
    <markup_type>text</markup_type>
    <viewpoints>
      <viewpoint>manufacturing</viewpoint>
    </viewpoints>
    <markup_item>
      <surface_finish>
        <variable>Ra</variable>
        <value>0.2</value>
      </surface_finish>
    </markup_item>
  </markup_element>
</markup_document>
```


(c)


```xml
<?xml version="1.0" encoding="utf-8" ?>
<markup_document xmlns:xsi="
http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="D:\Ding\markupschema.xsd">
  <document_header>
    <model_file>
      <file_name>tube.prt</file_name>
      <filepath>D:/ding/</filepath>
    </model_file>
    <geometry_reference_list>
      <geometric_element>
        <persistent_id>body_001</persistent_id>
        <geometric_element_type>body</geometric_element_type>
      </geometric_element>
    </geometry_reference_list>
    <editor_list>
      <editor>
        <name>Jason Matthews</name>
        <group>Design</group>
      </editor>
    </editor_list>
    <document_security_level>2</document_security_level>
    <Date>
      <create_date>06/05/08</create_date>
    </Date>
  </document_header>
  <markup_element>
    <markup_type>text</markup_type>
    <viewpoints>
      <viewpoint>manufacturing</viewpoint>
    </viewpoints>
    <markup_item>
      <material>coda glass</material>
    </markup_item>
  </markup_element>
</markup_document>
```


(d)

Figure 8: Examples of XML-based markup documents for the fuse product

circuit. In order to answer these questions, dimensional and operational characteristics for the product are required.

Figure 7 presents an example from the fuse manufacturing industry. It shows the markup needed for different points of view, such as manufacturing engineers, the assembly partner, the packaging partner and the marketing staff. With the internal markup environment (shown in Figure 2), this extra information can be marked up on the CAD model and stored in separate XML-based documents. Figure 8 shows some examples of the XML-based markup documents generated; Figure 8 (a), shows a piece of markup recording surface treatment information, attached to the fuse body for the viewpoints of manufacturing engineers and marketing staff; Figure 8 (b) shows the information stored for packaging partner; Figures 8 (c) and (d) show the extra information provided on the BODY component (i.e. material and surface finish) for manufacturing engineers. In addition, Figure 3 also gives an example of how service staff can feed back information when they only have the lightweight model in PDF.

Both internal and external markup information can be retrieved later by all users with a certain security level as all of them are linked to certain elements of the CAD model. For this case study, Figure 9 shows the interface allowing designers to retrieve the feedback information inserted by the service staff through the interface shown in Figure 3, and to open corresponding support documents automatically when clicking the 'link' button.

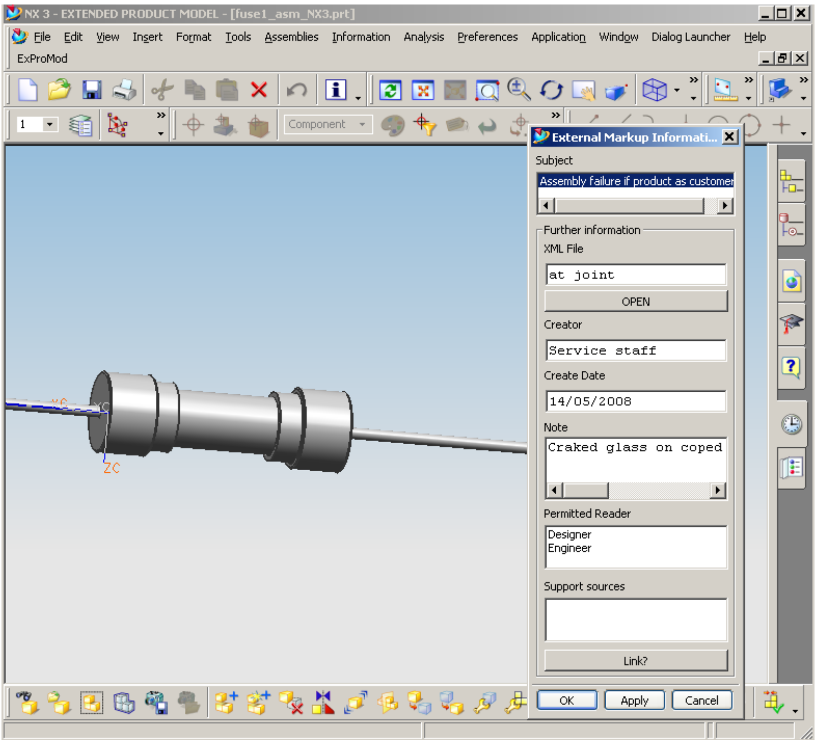


Figure 9: Interface of information retrieval in a CAD environment

7 Conclusions and future work

Currently, companies face the unprecedented challenges of the global market, collaborative environments and the entire product lifecycle. There are new demands on product representations, including platform/application independence, support for the product lifecycle, rapidly sharing information between geographically distributed applications and users, and protection of commercial security. To meet these requirements, some lightweight representations have been developed and applied in different industries. From the survey, it can be seen that current lightweight representations have been successful in some areas, such as the reduction of file sizes, cross-platform support, and enhancement of progressive streaming. However, current lightweight presentations adopt approximate or simplified geometric representations and domain-specific compression methods, and therefore their applications are limited to visualization. Meanwhile, though XML is widely used, there is still a risk of losing access to the information in current lightweight representations.

Aiming to address the problems of current product representations, a strategy of integrating lightweight representations with a markup method for the whole product lifecycle is presented in this paper. More engineering and non-engineering information, commercial security information, and viewpoint-specific information, are attached to the CAD model by a series of separate markup files. Based on these markup files, a system of levels of security and the original CAD model, various lightweight representations for different viewpoints during the product lifecycle can be generated. Similarly, an archival function can store the product information and its representation information into a private archive and a representation information registry (or network of private and public registries) respectively. As all CAD models are application dependent and highly complex, lightweight representations are a good solution for archival purposes. Meanwhile, the markup files linked to the CAD model can also provide the evidence needed by the archival function to decide which information is worth preserving for the long term. An implementation has been developed to demonstrate the practicality of the proposed strategy, including the internal markup environment in NX3, the external markup environment in Adobe Acrobat through its JavaScript API, the transfer interface, the retrieval interface, and representation information repository.

Further work will focus on the following tasks:

- Persistent identification of geometry. A number of different authors have tackled the issue of the persistent identification of geometry, also called the 'persistent naming problem', such as the work of Marcheix and Pierra (Marcheix and Pierra 2002), and Mun and Han (Mun and Han 2005). The initial focus of the problem was how to identify the surviving entities of a pre-edit B-rep in the post-edit B-rep. Here, the persistent identification problem concerns how to consistently identify the same geometric element (e.g. face or edge) present in the original CAD model and the derived lightweight representation. Most lightweight representations currently use facet representations to reduce the file size, but do so at the cost of losing the geometry identifiers within the CAD models. Thus, an intensive investigation on persistent identification on geometry among CAD models and lightweight representations and work on exploring the fundamental solution of persistent geometric identification are needed.

Marcheix D. and Pierra G., 2002. A survey of the persistent naming problem. *Proceedings of the Seventh ACM Symposium on Solid Modeling and Applications.* Saarbrücken, Germany. New York: ACM. 13–22. ISBN: 978-1-58113-506-0. DOI: 10.1145/566282.566288

Mun, D. and Han, S., 2005. Identification of topological entities and naming mapping for parametric CAD model exchanges. *International Journal of CAD/CAM,* 5(1). Available from: `http://www.ijcc.org/online2(pdf)/on_5_6.htm` (Accessed 01 February 2008).

- Improvement of the markup mechanism. The major disadvantage of 'stand-off' markup method is that implementing it is complex, as it requires a persistent system of reference back to the document. Thus, a more robust, persistent reference mechanism for stand-off markup method should be explored, especially in the circumstances of model change and format change.

- Partial geometry compression methods. Current geometry compression methods simplify the whole assembly/part model, but are unable to optimize according to users' requirements, such as compressing part of a CAD model but retaining other parts of the model uncompressed.

- Enforcement of version management for stand-off markup files. For products such as aircraft, ships, and machines, the lifecycle could last 20 to 30 years or in some cases even longer. The product data, including the stand-off markup documents, could be updated continuously, complicating the archiving, retrieval and access control of different versions of the markup documents from globally distributed remote locations. Effective solutions need to be explored, perhaps involving version control systems such as Subversion, or more comprehensive OAIS-style repositories.