Proceedings of the ASME 2007 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference
IDETC/CIE 2007
September 4-7, 2007, Las Vegas, Nevada, USA

DETC2007-35652

TOWARDS A FORMAT REGISTRY FOR ENGINEERING DATA

Michael J Grauer
Iris K Howley
Joseph B Kopena
William C Regli
Geometric and Intelligent Computing Laboratory
Department of Computer Science
College of Engineering
Drexel University
Philadelphia, PA 19104

ABSTRACT

There has been a great deal of interest recently in the problem of long term archiving of digital data. This is especially so in engineering design, where the CAD software tools evolve rapidly but the manufactured products themselves have much longer lifetimes whose support requires archived design data in a usable form. The ISO Open Archival Information Systems (OAIS) Reference Model is a widely used standard for digital archiving, with an essential piece of this model being a file format registry. A file format registry is a system for housing information about file formats that allows for correct interpretation, rendering, storage, and translation of digital files. Currently there exists no file format registry specifically for CAD file formats.

This paper explains the purpose of a file format registry for CAD in the greater context of digital archiving, and then presents our approach to creating a CAD file format registry using the Resource Description Framework (RDF) language of the Semantic Web. By creating our file format registry in RDF, we allow archival systems to perform automated reasoning on the stored files. We hope that this paper will increase awareness of this element of engineering design repositories in the research community of this conference.

INTRODUCTION

With current engineering methods, it is quite easy to generate large volumes of digital data for product design, but much more problematic is storage and retrieval of that same data over a long period of time. In the same way as all other fields reliant on computer tools, engineering design firms must be able to use digital files created on previous systems and stored in older formats. Often in order to keep a file up-to-date, it must be imported into a new version of the software and converted to a new file format standard; however, this is usually supported by software vendors only for the most recent previous version [1]. So any data that needs to be usable in the future must have constant data hygiene practices performed on it, updating it as standards change and maintaining any relationships it has with other files.

In engineering design this problem is especially acute for CAD files. CAD software is the primary authoring tool for the geometry and topology data associated with a product (plane, train, auto, building, etc), both for 2D and 3D designs. It is central to product lifecycle management and is often integrated with manufacturing, analysis, simulation and other engineering and business functions. A product must have the relevant CAD files accessible to provide proper support throughout the product's lifetime, and longer in the case of beyond-lifecycle tasks such as disposal management, archaeology, or historical research.

Some engineered artifacts have extremely long lifetimes, such as the Boeing Corporation's B-52 Stratofortress, which was first deployed in 1955 and is expected to be in continued operation until 2040 [2, 3]. This aircraft will have to be supported throughout a lifetime that includes not only changes in CAD software packages, but through the evolution of CAD itself. This example helps provide the context for engineering archival of digital design data.

The problem of supporting product design and operation is well known to Product Lifecycle Management and Design Repository systems, which assume that the stored product data will be accessible throughout the lifetime of the products [4]. This perspective ignores the fact of constant change in software data formats, software applications, operating systems, computing hardware platforms, and even engineering designers' experience with a particular set of tools. Given a long enough product lifetime, the data files and supporting infrastructure required to access CAD product designs will be obsolete and unusable [5].

We have presented a previous paper exploring the impact of the problems of preservation of digital design knowledge in this community [6]. The ISO Open Archival Systems Reference Model (OAIS) was the framework given as the basis for our work in digital design repositories, the National Design Repository. A significant piece of the OAIS model is the creation of a file format registry as a reference for archived data files. A file format registry is an authoritative source of representation information for digital content streams, which can provide a reference for both syntactic and semantic properties of a particular file format [7]. Currently there does not exist any CAD file format registry, and the lack of such a registry is an impediment to the efforts of archiving digital design data.

The aim of this paper then is to explain the role of a file format registry in the OAIS model and hence in digital design repositories, and to present our current approach working towards the development of a file format registry for CAD data, which we have currently implemented using the RDF (Resource Description Framework) language of the Semantic Web. Toward that goal, the following section gives a brief overview of the OAIS model as the basis for our National Design Repository project, with particular emphasis on file format registries, along with a discussion of general efforts in the digital archiving community on file format registries. A short overview RDF is also included to help explain our implementation approach. We then present our approach for a CAD file format registry, and finally close with a few conclusions and a discussion of our future work.

BACKGROUND

The ISO Standard 14721:2003 Reference Model for an Open Archival Information System (OAIS) has come to be the common framework used by much of the digital archiving community. The OAIS is a reference model rather than an actual implementation, and is primarily concerned with long term archival over durations that would indicate obsolescence of technology platforms [8]. This includes changing media and data formats, software and hardware platforms, and multiple generations of users. We described the OAIS model and our implementation of it in a engineering design context through the National Design Repository in a previous paper [6].

The three main participants as defined by the OAIS are the *Producers* of information, the *Archive* itself, and the *Consumers* of archived information. The four major components of the OAIS model for dealing with data are *Ingest*, which provides a means for *Producers* to add data, *Archival Storage*, which stores and maintains the archived data, *Data Management*, which provides searching and indexing into the archived data, and *Access*, which allows the *Consumer* to retrieve archived data. These components have as their inputs and outputs *Information Packages*, which combine the bitstream content of the data (*Data Object*), the syntactic and semantic information needed to interpret the data (*Representation Information*), and the information needed to identify, verify and certify the data (*Preservation Description Information*). These *Information Packages* include the information that a *Producer* would want to archive (*Submission Information Package [SIP]*), the information actually held by the archive (*Archive Information Package [AIP]*), and the information returned to the *Consumer* by the archive (*Dissemination Information Package [DIP]*). An additional administrative component specified by the OAIS model that is relevant to format registries is the *Preservation Planning* component, which can ensure the stability and readability of the archive data and can translate archived data into more sustainable formats. For a more detailed description of the OAIS model, refer to [9].

A file format registry is an authoritative source of representation information for digital content streams, which can provide a reference for both syntactic and semantic properties of a particular file format. According to the Global Digital Format Registry a format registry plays a major role in all aspects of the OAIS model.

Not only must the archived data be preserved over a long time scale, the format information must survive as well if proper preservation and interpretation is to occur. In the *Ingest* phase of archiving, format registries must be consulted for *SIP* validation and transformation from *SIP* to *AIP*. The *Access* phase will use format registry information for *AIP* to *DIP* transformation, which includes file metadata harvesting. The *Preservation Planning* stage of archiving uses format registries for monitoring formats for sustainability, and for finding appropriate formats and translators to preserve files currently stored in formats that are becoming obsolete.

OVERVIEW OF RDF

Resource Description Framework, or RDF, is a language of the Semantic Web [10]. The Semantic Web is an effort to distribute and encode knowledge throughout the Internet, and also allow computers to reason with that knowledge. Computers will not understand the actual knowledge encoded as humans do, but computers will be able to manipulate the encoded knowledge mechanically and make inferences about it.

RDF is a way for computers to describe information in a flexible manner, as it expresses relationships in a Subject-Predicate-Object triple which combine together into a labeled, directed graph structure. This structure allows for more expressive power than if information was encoded in hierarchies or relational tables. RDF files do not assume any centralized authority, so RDF files in various distributed locations can easily communicate with each other, and any system of RDF files can quickly and easily be included in another system, or extended into some larger scheme by including external files. The makeup of RDF also allows for different vocabularies to be mixed together, and for a canonical reference to be made so that different vocabularies can be sure that they are describing the same thing.

Together these attributes of RDF allow for encoding knowledge and performing computations on that encoded knowledge by taking different properties and relations on the same objects and combining them into more complex relationships. These attributes also allow for different RDF documents held by different organizations to be easily combined, which means that any other group that wanted to use our CAD file format registry could do so in the same manner as we do, or in any manner that they see fit. If some other group decides to write additional or competing RDF documents to describe CAD file formats, we can use those parts of their specification that we like, and ignore any other parts that we wish to. There are query languages such as SPARQL available for interrogating RDF documents, and these can be used to filter and examine the properties specified in a given RDF schema.

RDF can either be written in XML, or in a language of triples called N3. For our implementation of the CAD file format registry, we have used XML as it is a more common representation standard in this community.

RELATED WORK

There has been much work done by the digital archiving community towards the creation of file format registries. The most authoritative and relevant registries are the Global Digital Format Registry [7], The Library of Congress Digital Formats [11], and The PRONOM file format registry [12]. Of these, only PRONOM has any records related to CAD file formats, but CAD file formats are not the focus of that system. PRONOM provides detailed information about some CAD file formats, but these are only available as text fields on a web page.

An advantage that our approach of creating a CAD format registry in RDF provides over the method of text keyword storage is that the RDF encoded knowledge about the CAD file formats allows for automated reasoning and inferences about archived CAD files. Imagine a situation where a design engineer submits a CAD file to a design archive. The archive system could look up the file format type in the format registry, then using automated reasoning processes, could convert the file to a more sustainable format through an automatic translator to a correct format type, could create a 2D model from a 3D model, create a rendered image of the model, and add all of these files to the archive of the entered CAD model. If a new file format is entered into the archive, the archive could alert administrators to the fact that a new file type has been entered and new translator types would be needed. If information on the new file format is entered into the format registry, similarities to existing format types could aid in the suggestion of proper translation and rendering tools.

Our work on the National Design Repository addresses many of the functions of the OAIS model, but still lacks some critical components. Some of these concerns can be addressed through the development of a file format registry for CAD formats. Shortcomings of the National Design Repository that can be addressed through the creation of an appropriate CAD file format registry are an automated process to watch for formats becoming obsolete and an authoritative reference to supply formal semantic and syntactic interpretation of data file formats.

TECHNICAL APPROACH

There are two important spheres of information about CAD files that must be captured in a CAD file format registry. The first is basic identification, sustainability, versioning, and other administrative information. The second is classification information such as the types of geometry representations and application domains of a file. By creating a well-detailed **Format Description Schema** that contains the basic roots of a classification hierarchy, we can create a format registry schema that may be custom tailored to suit different organization's needs.

Our proposed structure for this format registry schema is displayed in Figure 1. The Format Description Schema, holding the basic classes, properties and other CAD metadata, contains links to a **Taxonomy of Application Types**, which can be customized based upon a project's needs. The language defined by the Format Description Schema and Taxonomy of Application Types are combined together to discuss instance data (i.e. particular CAD file formats).

The administrative data about a file format held in the Format Description Schema is necessary for tracking the constantly changing specifications of that format. This information is particularly useful as the industry standard CAD format changes between industries and from one year to the next, so it becomes vital for a manufacturer to be able to access the correct attributes of a particular file format. If older CAD files are to be archived and retrieved, it is important to store metadata necessary for maintaining these files. This includes concepts such as the format

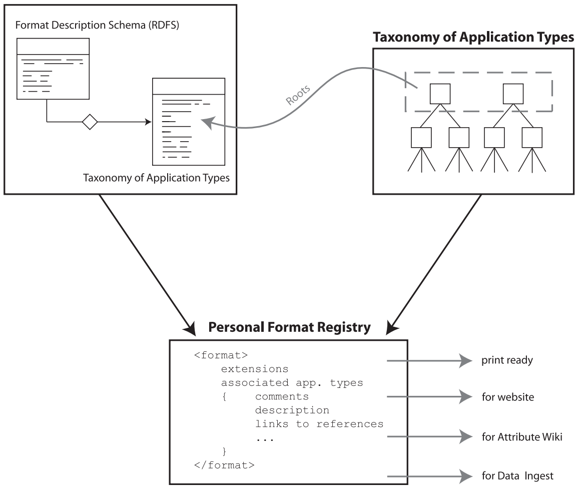

Figure 1. Our proposal for creating a partially customizable File Format Registry.

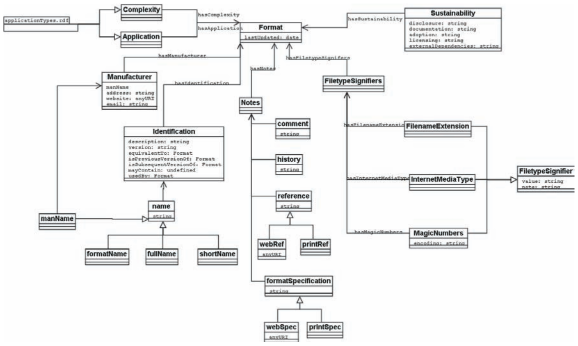

Figure 2. The Format Description Schema defines a file format's manufacturer, methods of identification, filetype signifiers, and sustainability capabilities.

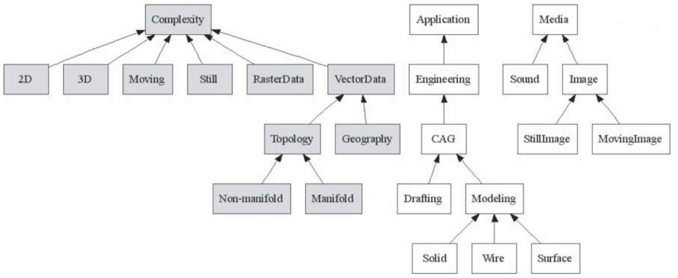

Figure 3. The structure of the Taxonomy of Application Types. Root concepts shared with the Format Description Schema are located at the top of the hierarchy.

name, extensions, manufacturer, documentation, and licensing. These properties are all declared as concepts within the Format Description Schema, which serves to define common information necessary for all CAD file formats.

Figure 2 portrays the relationships between the various concepts deemed necessary for defining and maintaining CAD file formats. These basic concepts include ways to identify the format of a file, each file's filetype signifiers (such as file extensions), sustainability, as well as the format's manufacturer. Each of these broad concepts have more specific details defined within the schema.

Classification of the purpose of a CAD file type is another important piece of information, which can be specified using our Taxonomy of Application Types. Our current taxonomy, as portrayed in Figure 3, shows a proposed hierarchy for this portion of the Format Registry. As stated previously, this document can be easily adjusted according to a project's needs. The main root concepts, however, cannot be changed and are declared in the Format Description Schema. We further define these main root concepts below:

- **Complexity** defines what information can be retrieved from a particular file format.
- The **Application** concept defines typical domains in which the file format is applied.
- The **Media** concept allows the inclusion of audio and visual files as well as CAD files.

A file format could potentially be a child of one or more of any of these concepts. For instance, file formats are generally of Complexity/2D or Complexity/3D as well as either Complexity/Moving or Complexity/Still.

This particular portion of the Format Registry is constantly evolving, as well as easily changeable as we determine improved ways to model the varying complexity of CAD file formats, among others. It could easily be expanded to include new application types such as text files.

The **Personal Format Registry** combines concepts from the Format Description Schema with concepts from the Taxonomy of Application Types to create instance data for a particular organization. It is in the Personal Format Registry that the metadata about the specific file formats are created. As an example, we will demonstrate selected parts of our Personal Format Registry *GICLFormatRegistry.rdf*, where we describe the STEP CAD file format. This example assumes that all class definitions have already been made within the Format Description Schema and the Taxonomy of Application Types. At this point, we create instance data for the STEP CAD file format. Rather than show the instance data in its entirety, we will depict an abridged set, as showing the entire definition is overly verbose and repetitive.

Figure 4 shows selections from our GICLFormatRegistry.rdf (implemented in XML), where we declare and define the STEP file format instance, starting with the Format instance member itself. The Format definition includes the date this RDF declaration was last updated, as well as pointing to other instance data

```
<format:Format rdf:ID="STEP">
  <format:lastUpdated rdf:datatype="&xsd;date">2006-09-19
  </format:lastUpdated>
  <format:hasApplication rdf:resource="#STEPapp"/>
  <format:hasComplexity rdf:resource="#STEPcomp"/>
  <format:hasFiletypeSignifiers rdf:resource="#STEPfiletypesignifiers"/>
</format:Format>

<app:Modeling rdf:ID="STEPapp">
  <rdf:type rdf:resource="&app;#Drafting"/>
</app:Modeling>

<app:VectorData rdf:ID="STEPcomp">
  <rdf:type rdf:resource="&app;#_3D"/>
  <rdf:type rdf:resource="&app;#_2D"/>
</app:VectorData>

<format:FiletypeSignifiers rdf:ID="STEPfiletypesignifiers">
   <format:hasFilenameExtension rdf:resource="#STEPext1"/>
   <format:hasFilenameExtension rdf:resource="#STEPext2"/>
 </format:FiletypeSignifiers>

<format:FilenameExtension rdf:ID="STEPext1">
   <format:value rdf:datatype="&xsd;string">step</format:value>
</format:FilenameExtension>

<format:FilenameExtension rdf:ID="STEPext2">
   <format:value rdf:datatype="&xsd;string">stp</format:value>
</format:FilenameExtension>
```

Figure 4. An abridged example of the Step file format instance in the GICLFormatRegistry.rdf .

that defines additional information about the file format. Having properties that point to other classes like this is part of the definitions in our Format Description Schema. This structure makes the otherwise unwieldy RDF code a little more organized and readable. If this were a complete example this format declaration would also refer to identification, manufacturer, and sustainability instances.

Next we define the STEP format's typical application, in this particular example, we will say that STEP's application is both Modeling and Drafting. Due to the structure of the Taxonomy of Application Types, this definition also implies that STEP is used for both computer aided geometry and engineering applications. We then define STEP's complexity to be both 3D and 2D vector data. This shows that is possible to use terms placed deeper within the Taxonomy of Application Types, rather than just higher-level complexity concepts.

Finally we define one kind of administrative information for the STEP format, which is filetype signifiers. The most basic of filetype signifiers is the filename extension. In this example we state that STEP has two filename extensions, *.stp* and *.step*.

After defining STEP as a CAD file format in the GICLformatRegistry.rdf, we can use this information in our research group's other efforts towards creating an archiving system for CAD files. When a CAD file is added to our archive, we take information from the GICLformatRegistry.rdf to tag and identify the CAD file format. By adding this metadata from our format registry to the archive of the CAD file, we can better document, track, and classify the types of files we have received.

Our plans for extending this work include expanding on the set of classifications and building a method for tracking the relations between file formats as they change over versions. We would like to open up our system to the public, possibly through a moderated Semantic Wiki, which would allow us to gather expertise from the community and increase our coverage of CAD file types. Through exposing this system to the greater community we hope to gain valuable feedback which could let us know additional directions for research. Additionally, we can expand our system to encompass the recommendations of developers of format registries in other digital archiving domains.

CONCLUSIONS

This paper has aimed to explain the role of a file format registry in the context of engineering design archives and to present our current approach to building a file format registry for CAD. We hope to make the community more aware of the challenges in digital engineering archives and how the role of a file format registry fits in with this. We have based our work in digital archiving on the OAIS model, in which the notion of the format registry is needed by many different components. As there is currently no format registry dedicated to CAD formats, we feel this is an important need for the engineering design archiving community.

Our approach to creating a CAD file format registry using RDF was presented as a first step towards this goal. By using RDF, the format information about the various CAD file formats is well structured and enables automated reasoning. The advantages of this are having a system that can automatically perform translations between file formats when one format is nearing obsolescence, or could perform filtering and searching based on format attributes rather than simply over text keywords.

It is hoped that as we deploy our system to the wider community and integrate our CAD file format registry with our other efforts in the digital engineering archiving domain (archiving CAD files in the National Design Repository, 3D and temporal searches, file type translation, semantic relationships of files) this work will stimulate further research and development in this area.

ACKNOWLEDGMENT

This work was supported in part by the Library of Congress National Digital Information Infrastructure Preservation Program (NDIIPP) via National Science Foundation (NSF) DIGARCH Program grant CISE/IIS-0456001. Additional support was provided by the National Science Foundation Cyber-Infrastructure Teams (CI-TEAMs) Program grants CISE/SCI-0537370 and CISE/SCI-0537125.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or the other supporting government and corporate organizations.

REFERENCES

[1] Fallows, J., September 2006. "File not found". *The Atlantic*.

[2] US Air Force, 2001. Long-range strike aircraft white paper. On the WWW, at `http://www.af.mil/library/posture/bmap01.pdf`, November. PDF file.

[3] Defense Science Board, 2004. B-52H re-engining. On the WWW, at `http://www.acq.osd.mil/dsb/reports/2004-06-b52h_re-engining.pdf`, June. PDF file.

[4] Szykman, S., Bochenek, C., Racz, J., Senfaute, J., and Sriram, R., 2000. "Design repositories: Engineering design's new knowledge base". *IEEE Intelligent Systems,* **15**(3), May/June, pp. 48–55.

[5] Thilmany, J., 2005. "Ephemeral warehouse". *Mechanical Engineering*, September.

[6] Kopena, J., Shaffer, J., and Regli, W., 2006. "CAD Archives Based On OAIS". *Proceedings of IDETC/CIE 2006 ASME 2006 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference September 10-13, 2006, Philadelphia, USA*.

[7] Abrams, S., and Seaman, D., 2003. "Towards a global digital format registry". *World Library and Information Congress: 69th IFLA General Conference and Council 1-9 August 2003, Berlin*.

[8] Consultative Committee for Space Data Systems, 2002. Reference model for an open archival information system (OAIS). Adopted as ISO Standard 14721:2003. Tech. rep., NASA, January.

[9] Bull, A., 2006. Briefing paper: The OAIS reference model. Tech. rep., UKOLN, University of Bath, February.

[10] Tauberer, J., 2006. What is RDF. On the WWW, July. `http://www.xml.com/pub/a/2001/01/24/rdf.html`.

[11] Library of Congress, 2006. Sustainability of Digital Formats: Planning for Library of Congress Collections. On the WWW, November. `http://www.digitalpreservation.gov/formats/fdd/descriptions.shtml`.

[12] The National Archives, 2007. Pronom. On the WWW, February. `http://www.nationalarchives.gov.uk/pronom/`.