Archiving the Semantics of Digital Engineering Artifacts in CIBER-U

William Regli, Michael Grauer, David Wilkie, Joseph Kopena, Martin Piecyk, Jordan Osecki
Drexel University
Philadelphia, PA 19104, USA

Abstract

This paper introduces the challenge of digital preservation in the area of engineering design and manufacturing and presents a methodology to apply knowledge representation and semantic techniques to develop **Digital Engineering Archives**. This work is part of an ongoing, multi-university, effort to create Cyber-Infrastructure-Based Engineering Repositories for Undergraduates (CIBER-U) to support engineering design education. The technical approach is to use knowledge representation techniques to create formal models of engineering data elements, workflows and processes. With these formal engineering knowledge and processes can be captured and preserved with some guarantee of long-term interpretability. The paper presents examples of how the techniques can be used to encode specific engineering *information packages* and workflows. These techniques are being integrated into a semantic Wiki that supports the CIBER-U engineering education activities across nine universities and involving over 3,500 students since 2006.

Introduction

Digital preservation is the mitigation of the deleterious effects of technology obsolescence, media degradation, and fading human memory (Gladney 2006). One of the fundamental challenges facing those developing Digital Preservation solutions is the development of digital representations that are self describing and assured to be interpretable over the long lifespans required by archival applications.

This paper utilizes semantic web technologies (Kopena & Regli 2003) to create of self-describing representations for archiving engineering data. The overall approach is to design a set of extensible ontologies that describe engineering file formats, their underlying data models, along with the software and hardware tools used to create and transform this data. Further, these form the basis for representations of process workflows and aggregations of engineering objects that capture relationships among the files and key data transformation processes. The ontologies can also be used to record the relationships of a file to different versions of itself as it evolves over time, and can record provenance metadata about a file such as the creating agent, time and location. This allows for digital archives that can record knowledge about their contents, rather than merely storing data.

There are specific contributions presented in this paper:

1. We develop a formal model for an extensible *Format Registry* (*format registry*) for engineering data elements and software. The *format registry* contains the fundamental ontology with which engineering data and processes can be captured and preserved with some guarantee of long-term readability.
2. We present a method of capturing and representing engineering workflows based on the *format registry* using the *Process Specification Language* (*PSL*), the ISO standard language for creating machine-readable representations of processes (Grüninger & Menzel 2003).

This offers a new approach create long-term digital archives. Prior work in digital preservation falls into two categories: *conversion*, where digital files are constantly updated and translated encodings; and *emulation*, where the original execution environment will be emulated on future platforms. The semantics-based approach is this paper aims to support design knowledge capture (Regli *et al.* 2001) into well-defined neutral forms, enabling future users the ability to extract meaningful information from archived digital objects.

The work presented in this paper is currently supporting the Cyber-Infrastructure-Based Engineering Repositories for Undergraduates (CIBER-U[1]) Collaboratory (Devendorf *et al.* 2009), an initiative to improve the ability of educators to teach engineering design by developing an extensible library of virtual product dissections. The structures we developed help support design knowledge capture, digital preservation planning and archive management in a semantic wiki that serves as the CIBER-U "collaboratory". Since its inception in 2006, this collaboratory has been used by over 3,500 students at nine universities.

Scientific and Project Background

The Challenge of Digital Engineering Archives The relationships among shape and form, structure and function, and behavior and semantics are among the most fundamental questions studied by science and engineering—and it is precisely these relationships that must be captured and

[1]`http://gicl.cs.drexel.edu/wiki/CIBER-U`

preserved in digital engineering archives. For many industries (aerospace, civil engineering & architecture, shipbuilding, geo-engineering), engineering design and manufacturing knowledge needs to be preserved over 50-to-75 year lifespans (Thilmany 2005). Traditional digital data management is highly dependent on the proprietary formats of commercial software systems, proving it hard to guarantee the utility of data over long periods. Hence, while nearly all modern engineering domains view the 3D computer-aided design (CAD) model as indispensable, the engineering part print (i.e., blueprint or 2-D drawing on paper, aperture cards, microfiche) remains as the principal method of design knowledge archival. From an archival standpoint, much of the knowledge generated during the modern engineering enterprise (i.e., by 3D CAD, simulation, etc.) is simply lost. Even if the CAD files are archived, the supporting infrastructure required to access and understand these designs will be obsolete and unusable. The rich digital knowledge in 3D CAD about features, design and production workflow, manufacturing processes and artifact behavior are simply not captured or lost in a myriad of data translations.

Other digital preservation domains (i.e., audio, video, still image, etc.), have developed format registries in order to better capture archival issues (Abrams & Seaman 2003; Library of Congress 2006). Such efforts are largely static, and focused on relatively few media types. In contrast, a comprehensive representation of an *single* engineering artifact might encompass thousands of files in hundreds of different formats.

CIBER-U Cyber-Infrastructure-Based Engineering Repositories for Undergraduates (CIBER-U) (Devendorf *et al.* 2009) is a ongoing collaboration among nine universities (Penn State, SUNY Buffalo, Drexel, Missouri S&T, Virgina Tech, Bucknell, Sweet Briar State, Norfolk State and Northwestern) to create a National Product Dissection Collaboratory to support engineering education. Product dissection has been used successfully in a variety of ways to actively engage students in learning engineering design (Sheppard 1992). Dissection can be used to increase awareness of the design process as well as teach competitive assessment and benchmarking (Marchese *et al.* 2003).

Despite the numerous advantages of using product dissection throughout the engineering curricula, product dissection has not yet become a national model for engineering design education. Products, tools, and their upkeep can be costly, workspace and storage space can be difficult to obtain, and even the best crafted dissection assignments can end in chaos. CIBER-U aims to create a living repository of product dissection activities, suitable for all levels of engineering undergraduate curricula. With CIBER-U, instructors use and contribute to an archival corpus of digital design repositories that include CAD models, simulation data (including kinematics, dynamics, physics, etc), video and other multimedia. The long-term goal is for this corpus to become a key element of engineering education nationally.

Example of CIBER-U Content. The principle contents of CIBER-U are "Case Studies," an example of which is shown in Figure 1 and includes:

1. a brief description of the product and how it works;
2. a list of all its parts (i.e., a Bill of Materials), including the quantify, material, manufacturing process, and photograph of each part;
3. step-by-step product disassembly instructions;
4. a set of files including 3D CAD models and 2D drawings of each part and an assembly model of the entire product;
5. descriptions of the functional, behavioral, and energy interactions of the project components and how they contribute to achieve the overall design objectives.

Some case studies also include animations (in CAD) of their disassembly and videotaped presentations of them being manually disassembled. These are particularly useful for first-time instructors who may not be familiar with a particular product, especially if they have never dissected it before.

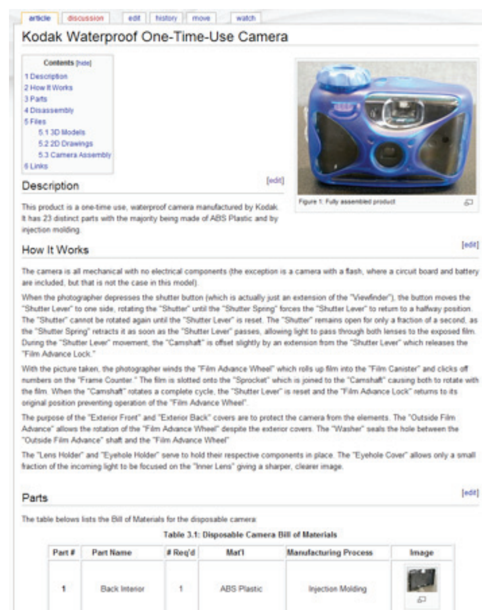


Figure 1: CIBER-U product dissection case study: Semantic MediaWiki for Kodak One-Time-Use Camera.

Typical CIBER-U Use Cases. The principle intended users of the CIBER-U Wiki are undergraduate engineering students studying engineering design. As part of a dissection-based design course, students are divided into lab groups and given a product to disassemble, evaluate and reassemble (i.e., cordless drills, disposable cameras, etc). Students had access to case studies from similar products that had been previously generated and entered into the CIBER-U Wiki. The CIBER-U Wiki provides students templates and user guides for dissection activities, enabling the display of many types of media and information in a manner that is easy for others to read and evaluate. Students were required to prepare a report in a MediaWiki format (e.g., Figure 1) that captures the product content (described above). The MediaWiki interface enabled collaborative editing, modification, and update—providing a forum for instructors and students to post their own comments and share their findings with others who are dissecting the same products. available, they can also be added to the website, creating a “living archive” of each product’s evolution.

CIBER-U Archiving Challenges. CIBER-U is an exemplar for the wider challenges in the creation of digital engineering archives. Consider:

- CIBER-U data is highly heterogeneous. Different CAD/CAE systems are used across institutions, resulting in many different file formats populating the case studies.
- Dissection studies involve a number of different activity workflows to interpret and document the product.
- For some case studies, product lifecycle data includes knowledge from different domains and software systems (i.e., physics-based models, kinematics, dynamics, etc).
- Data files for engineering models can be large (i.e., hundreds of files and gigabytes are required to represent even simple parts) and have complex internal structures (i.e., geometry, topology, features, joints, tolerances, math, physics, etc).
- Most importantly, for CIBER-U materials to be of ongoing usefulness, they must be interpretable by future engineering students and CAD/CAE systems.

Viewed in this way, CIBER-U is similar to problem in management of large-scale scientific data sets. However, the engineering domain offers significant complexity due to the size and internal structure of engineering and CAD file. This requires a practical approach to capturing engineering knowledge *at the time of creation* and transformation of this knowledge into format structures that can increase the usefulness of this data over time.

Technical Objectives & Approach

The approach is to use open, semantic web, standards to represent vital engineering knowledge in a manner that is reasonably assured to be persistent and interpretable (and semantically consistent) over long periods of time. As it relates to CIBER-U, there are several key elements:

- We develop and extend ontology technologies to represent engineering knowledge in the form of *information packages.* These packages can refer to individual files or collections of associated engineering documents.
- Deploy a *format registry*, a knowledge-base with an extensible ontology of engineering file formats and related applications. This enables specification of relationships among packages and files, create associations between files and engineering activities, and define domain rules.
- Identify and capture *engineering process workflows* and *engineering context*. These workflows are similar to those in business processes re-engineering and scientific areas (Gil *et al.* 2007); context refers to knowledge beyond file format information that includes information about parts, assemblies, models, design decisions, simulation data, and other documentation.

The following sections introduce these concepts and provide detailed examples illustrating how they have been applied to complex engineering data elements archived in CIBER-U.

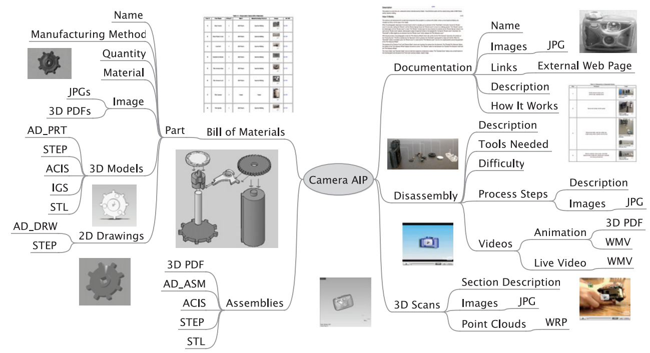


Figure 2: Camera *Information Package*.

Engineering Information Packages While there has been considerable research on ontologies for engineering design processes (e.g., engineering function (Kopena & Regli 2003)) none have addressed the taxonomy of engineering file formats and their interactions. Pattern recognition researchers have developed shape type ontologies, however these representations are sufficient to describe the varied geometries and formats of the population of engineering models found across the engineering lifecycle.

Rather than focus on such abstractions, the needs of CIBER-U require a focus on files as atomic elements in the preservation system. An *information package* (IP) is what the archive actually stores, containing four kinds of description information: *provenance*, the data history and rights; *context*, the relationships between the different data objects; *reference*, identifiers for the data; and *fixity*, keys and checksums used to check for integrity. The IPs will be stored over the life of the archive, and are subject to *preservation planning* migrations. These migrations include *refreshment*; where an IP is copied on the same medium; *replication*, where an IP is copied onto a different medium; *repackaging*, where an IP is copied and of the *packaging information* is changed; and *transformation*, where an IP is copied such that the content is changed with the goal of preserving the original IP content. In the engineering archives domain *transformation* would occur whenever a new format for a model (e.g., transforming a Pro/ENGINEER v16 file to Pro/ENGINEER v17 file) or a new standard (e.g., new ISO 10303 STEP APs) is developed.

Provenance, the history, set of events, users, and rights for a digital file, is another essential piece of digital archiving. CIBER-U uses a provenance ontology[2] developed based on the work in (Ram & Liu 2006), which encodes the semantics of provenance by providing the answers to such questions of “what”, “when”, “where”, “how”, “who”, “which”, and “why” for any digital file and transformations on that file.

[2] http://gicl.cs.drexel.edu/ontologies/provenance.owl

Underlying this provenance ontology is the OWL Time Ontology[3]. In the future, additional ontologies that will need to be integrated include those for engineering design (Darlington & Culley 2008), engineering function (Kopena & Regli 2003), and engineering requirements.

Example Information Package. Figure 2 illustrates the information package for the CIBER-U camera dissection shown in Figure 1. This can be viewed as a map of all of the information behind the artifact, including: the 3D solid model files (which can change over time); different resolutions of the model files for purposes of specific analyses; design and process rationale behind the creation of derivative models (i.e., surface mesh for input to a finite element analysis (FEA)) and their results.

An OWL ontology to captures the complexity of files and their relationships in the information package,[4] acting as an architecture for building up these engineering data information packages. The top level concepts in this ontology captures the `Design Product`'s IP: `Documentation`, `Models`, and `Workflows`. `Documentation` includes `Requirements`, `Specifications`, `Design Records`, `Videos`, `Images`, `Reports`, `Engineering Change Requests`, `User Manuals`, and `Bill of Materials`. The `Models` taxonomy describes various engineering models, such as `Geometric` (CAD 3D solid and 2D models), `Shape` (mesh and point cloud data), `Tolerance` (manufacturing tolerance specifications), `Kinematics` (models of how the object can move), `Dynamics` (physics based motion and interaction models), `Structure` (models of the physical relations of parts), `Function` (the purpose of each of the parts), `Behavior` (how each of the parts acts), and `Materials` (what each of the parts is made up of, and any relevant material properties). `Workflows` comprise `Design`, `Analysis`, `Assembly`, `Fabrication`, `Translation`, `Reverse Engineering`, `Simulation`, `Manufacturing`, and `Inspection`. All of the files that relate to the product go into the IP, and have an ontology instance describing the `Digital File`.

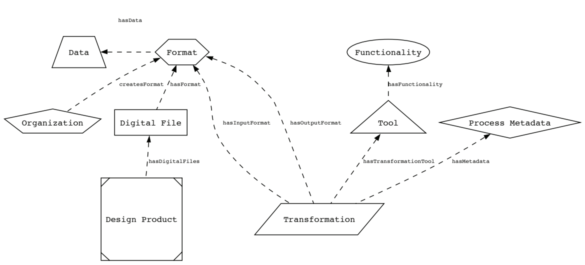


Figure 3: Top level concepts of the *format registry*.

An Extensible Engineering Format Registry The *format registry* is a knowledge-base with an extensible ontology of engineering file formats and applications. *engineering context* is knowledge beyond file formats that includes information about parts, assemblies, models, design decisions, simulation data, and other documentation. Engineering processes can be described with flows of data throughout an organization, hierarchically across levels of abstraction and horizontally through different functional teams and time as models are changed and updated.

The CIBER-U format registry captures key properties of the file formats and their interactions. In general, the registry is the source of common understanding between people developing engineering archives and the archiving applications themselves—thus enabling capture of workflow possibilities (i.e., how do I translate a `.step` file into a `.prt` file?) and related metadata. The registry captures core domain assumptions, i.e., as new workflow elements or design products/processes are identified, the ontology can be extended to capture this new information. Lastly, it provides methods for analyzing domain knowledge, e.g., workflows can be checked for validity and completeness.

The "competency questions" approach suggested by (Grüninger & Fox 1995) was used create the format registry's ontology. Examples of the questions used to create the ontology include:

- What formats hold CAD models?
- From what formats can a 3D mesh be derived?
- What software tools can translate from B-Rep to Mesh?
- What list of software tools and formats will translate between a pair of given formats? What information would be lost in this transformation?
- For a given model, what other models are related to it by any transformation?
- What are the analysis data types related to a particular model?
- What is the metadata on a given transformation between two models? (i.e., lossiness, shape distortions, etc).
- What are the attributes of a model? (i.e., does it capture discrete or continuous shape? is it feature based? etc).
- What formats contain material properties?
- Is there an example of metadata for a previous transformation using a model of a given starting type, a model of a given ending type, and a given software tool?
- What formats can be used to create a given analysis?
- What formats can be used to extract solid models?

Additional competency questions related to workflow capture, workflow design and verification, and workflow-assisted auto-archiving are:

- What are the inputs and outputs of tools?
- What metadata is necessary to specify for a given model, tool, and transformation?
- What files have been created in any of the formats listed in the workflow, or by any of the tools listed in the workflow, or capable of holding any of the model data specified in the workflow?

Format Registry Ontology Structure. The *format registry* is based on OWL[5] and its top level elements are `Format`, `Data`, `Functionality`, `Tool`, `Digital File`, `Process Metadata`, `Design Product`, `Transformation` and `Organization`. The `Design`

[3] http://www.w3.org/2006/time

[4] http://gicl.cs.drexel.edu/ontologies/engineering-information-packages.owl.

[5] http://gicl.cs.drexel.edu/FR/eng_fr.owl

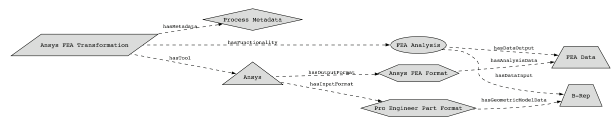
Product is overall artifact; the Digital File concept captures all files related to the artifact. The Tool concept represents Software Tools or a Hardware Tools used in the engineering workflow. These Tools each have Functionality, which is the set of functions (and preconditions) that the Tool is capable of performing. Organizations own and maintain Tools and Formats (Figure 3).

Formats and Data. Format is the most fundamental concept in the ontology, representing the format that encodes a digital file, and every instance of a Digital File must have a Format. Data is a classification for the types of data contained within formats (e.g., Geometric Model Data or Analysis Data). Metadata property elements (e.g., Name, Description, Extensions for Format build on other format registry projects from the digital preservation community (Abrams & Seaman 2003; Library of Congress 2006; **?**).

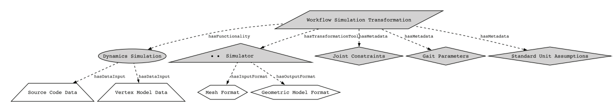
There are three subclasses of Format: Analysis Format, a format of some analysis transformation, Model Format, which describes any engineering model, and Documentation Format, which describes documentation such as requirements, specifications, screenshots, videos, etc. Documentation Formats are extensively researched in existing format registry projects, the under-documented formats for engineering design are Model Format and Analysis Format. Analysis Formats contain Analysis Data, which is data that has been created by applying an analysis using a scientific tool to a given model. An example of this would be taking a Pro Engineer Part file, bringing it into Ansys (a Software Tool), performing Finite Element Analysis (a Functionality), and outputting a Ansys FEA Format file, which would contain Analysis Data.

Model Format is a subclass of Format that has some underlying Model Data. Model Data subclasses represent common engineering models such as Behavior Data, Function Data, Joints and Constraints, Material Properties, Source Code, and Topology. Any of these model types is rich enough to support a separate ontology, and should a suitable one be developed or discovered, its concepts could be integrated here. Our work specifically focuses on shape and geometry, as a result we chose to develop the Geometric Model Data subclass.

Challenges Specific to CAD/CAE Data. The complexity and diversity of Geometric Modeling Data poses unique challenges. In the course the engineering process, many derivative model representations must be created for the various workflows central to design, manufacturing, and lifecycle activities. Geometric Model Data is data that specifies shape, including Vertex Model Data, Curve Model Data, Wireframe Models, Surface Model Data, and Solid Model Data. Solid Model (a Functionality in the *format registry* ontology), is a branch of computer graphics and modeling that deals with water-tight representations of 3D objects, with 3D boundaries and Topology. A B-Rep, a subclass of Solid Model Data, is a boundary representation of a solid model. The vast majority of objects produced by modern CAD software are 3D models or solid models. The complex internal representations are typically proprietary and encode design history, rationale, parametric constraints, etc. Some of the derivative models from 3D solid models include those in Mesh Format, which use Surface Model Data to create a polyhedral approximations of a 3D object (usually used in graphics for rendering). For example, Pro Engineer Part and Pro Engineer Assembly are both CAD Formats and have B-Rep as their Geometric Model Format data, whereas Stereolithography is a Mesh Format holding triangular polygonal surface representations of objects, and has Surface Model Data as its Geometric Model Data data.



(a) Finite Element Analysis Transformation.



(b) Workflow Simulation Transformation.

Figure 4: Example engineering archive process workflows.

Transformations and Process Metadata. Two other ontology classes describe transforming Model Data either into another Model Data, and performing an analysis and creating Analysis Data. An illustrative example is presented in Figure 4(a), in which Finite Element Analysis (a Functionality) is performed on a part stored in a Pro Engineer Part Format. A Transformation specifies a Tool used and the allowed input and output formats to the transformation, as well as the Functionality performed (i.e., FEA Analysis). Finally, the Transformation must specify its Process Metadata, which records specific arguments to the transformation. Through these classes it is possible to store all Model Data and Analysis Data instances associated with any Model Data, along with storing enough information about the Transformation to recreate it. This allows for the ability to store in an engineering archive the sequence of changes and enrichments a model undergoes throughout the design process and beyond, so that analyses and model translations can later be investigated and understood.

Engineering Context and Workflows

PSL provides a core ontology of activities and can represent the workflow as an unambiguous and reusable plan that can be input to inference engines(Kim, Spraragen, & Gil 2004). The flow of data in engineering organizations is a complex combination of efforts by individuals spanning disciplines and roles that needs to be modeled formally in order to be accurately captured(Barkmeyer *et al.* 1995). The transformations and branchings of the data are often captured only by those local to the changes taking place. For example, an engineer may need to alter the data representation of a part in order to import it into an analysis tool, but this transformation is likely to be outside the scope of the managements' vision, and the relationship between the transformed data and the original design may be unclear. Additionally, the flow of data will cross disciplinary lines and acquire different assumptions, which are unlikely to be recorded. Hence, capturing the workflow of the engineer is a vital step in archiving. Archiving workflows also helps to ensure reproducibility of analyses and to provide a platform and technology independent view of the system processes (Gil *et al.* 2007). Without an explicit account of the semantics of the workflow's transformations, it will never be clear for what reason a modification was made. The modification could have arisen from the constraints of an analysis package; it could have arisen for manufacturing considerations such as machine tool wear; or it could have arisen from later cost-estimates by managers or from aesthetic considerations by the marketing department.

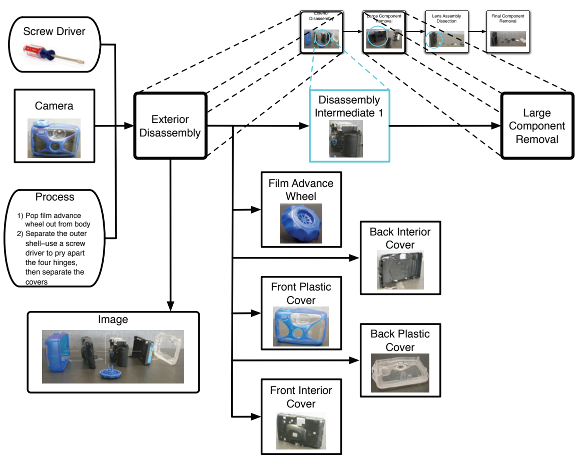


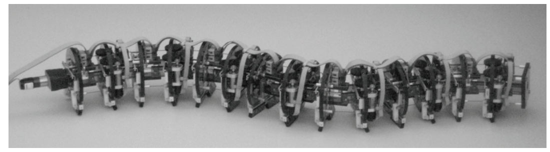
Figure 5: Product disassembly workflow for the camera.

Applying the Semantic Models

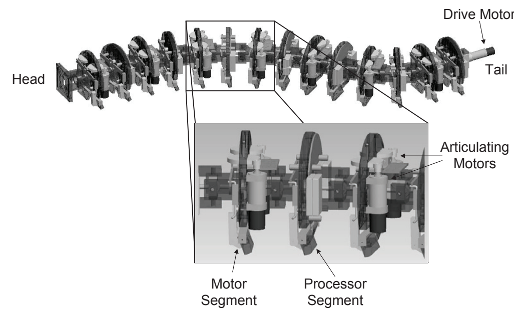
Three examples are provided to illustrate how the formal semantic techniques are being used to support the digital preservation and long-term archival of engineering data.

Example. CIBER-U Camera Disassembly Workflow. Figure 5 shows a portion of the `Exterior Disassembly` workflow for the waterproof camera. This workflow is underlies the lesson plans for the CIBER-U student product dissection assignments. While the example is specific to the camera, general workflow models are behind all dissection activities and can be used to automate the packaging, organization and annotation of information captured in the course of CIBER-U assignments.

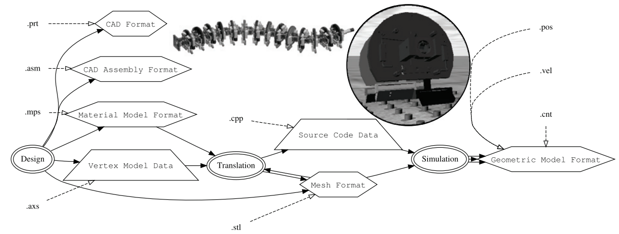
Example. CIBER-U Bio-Robotic Snake. As a more detailed case study, consider the capture the semantics



(a) The "live" snake robot.



(b) A 3D CAD model of the snake robot.



(c) Design and simulation workflow.

Figure 6: CIBER-U Case Study: A Bio-Inspired Robot.

of a complex design and analysis task for simulation of the biologically-inspired (snake-like) robot shown in Figure 6(a). The simulation/analysis workflow is shown in Figure 6(c) as *workflow function* (*WF*) nodes and *ontology class* (*OC*) nodes. Each *WF* can take *OC*s as inputs and can produce *OC*s as output. The *OC*s join the *WF*s to form the workflow. `Design` activity produces the CAD solid models of the robot (Figure 6(b)), which includes individual `CAD Format` files, parts assembled together into `CAD Assembly Format` files, and then sub-assemblies grouped together into larger assemblies. For this particular object, file format instances include `Pro Engineer Part` (.prt) and `Pro Engineer Assembly` (.asm). The CAD data is used to create a physics-based simulation of the robot using `Material Model Format` (`Material Properties` format, .mps), `Vertex Model Data` (.axs), and `Mesh Format`. For the simulation software tool used in this example, a `Stereolithography` format (.stl in Figure 6(b)) is derived from the `Surface Model Data` of the `B-Rep` defining the solid model via a `Translation` task. This *WF* represents an engineer relating the different `Mesh Format` *OC*s, which are surface representations of the robot's structure, with geometric axes used for positioning defined on individual parts of the model from the `Vertex Model Data` *OC*, which represents geometric axes, and with the

Material Model Format *OC*, which contains densities and coefficients of friction. One output of this *WF* is the *OC* Source Code Data (C++ specifically). Additionally, the Translation *WF* creates a simplified tessellation of the Mesh Format. The last *WF* step, Simulation, involves an combining Source Code Data with other libraries (i.e., Open Dynamics Engine, source code, a gait, and other elements. The result is a physics-based simulation that animates a model of the robot walking over a specified terrain (pictured at the top in Figure 6(c)) along with three Geometric Model Format *OC* outputs, which are the custom defined formats Position (.pos), Velocity (.vel), and Contact Point (.cnt).

The *WF*s can later be clarified with greater detail, for example the Workflow Simulation Transformation of Figure 4(b) can further specify that the Simulation step uses a custom Robot Simulator which has an input of Mesh Format and an output of Geometric Model Format. The Process Metadata of the step can be specified, with Joint Constraints detailing the relationships and alignment between parts, Gait Parameters recording the force and torque inputs to the robot's joints, and Standard Unit Assumptions storing the system of units expected by the Simulation.

Example: Using the Formal Workflow Models. The formal workflow models can be used to define rules for model translation and format migration. For example, a conversion of the Translation task in PSL is performed by a function to define the Activity of Translation, parameterized by the inputs and outputs of the task:

$$\forall v, w, x, y, z \cdot \texttt{translation-activity}(v, w, x, y, z) \supset [\texttt{material-model-format}(v) \wedge \texttt{mesh-format}(w) \wedge \texttt{vertex-model-data}(x) \wedge \texttt{source-code-data}(y) \wedge \texttt{mesh-format}(z)]$$

Then, another function defines the Occurrence of Translation, parameterized by its inputs and outputs:

$$\forall o \exists v, w, x, y, z \cdot \texttt{occurrence\_of}(o, \texttt{translation}(v, w, x, y, z)) \supset [\texttt{input}(v, o) \wedge \texttt{input}(w, o) \wedge \texttt{input}(x, o) \wedge \texttt{output}(y, o) \wedge \texttt{output}(z, o)]$$

Next, the overall workflow is rendered (note only a single I/O at eachstep is given for brevity):

$$\forall o \cdot \texttt{occurrence\_of}(o, \texttt{design-and-simulate}) \supset \exists p, q, r, x, y, z \cdot [\texttt{occurrence\_of}(p, \texttt{design}(x)) \wedge \texttt{occurrence\_of}(q, \texttt{translation}(x, y)) \wedge \texttt{occurrence\_of}(r, \texttt{simulation}(y, z)) \wedge \texttt{subactivity\_occurrence}(p, o) \wedge \texttt{subactivity\_occurrence}(q, o) \wedge \texttt{subactivity\_occurrence}(r, o) \wedge \texttt{output}(x, p) \wedge \texttt{input}(x, q) \wedge \texttt{output}(y, q) \wedge \texttt{input}(y, r) \wedge \texttt{output}(z, r) \wedge \texttt{next\_suboccurrence}(p, q, o) \wedge \texttt{next\_suboccurrence}(q, r, o)]$$

These formal models can be used to automate the ingest of data into persistent archives and the ongoing maintenance of the files and their relationships over time.

Deployment, Assessment & Lessons Learned

To date, 3,434 students have participated in one or more CIBER-U activities (Devendorf *et al.* 2009), ranging from high school students working as part of summer research experience programs, undergrads as well as graduate students. These activities developed product dissection exercises to be shared across all of the partner institutions and created rubrics and evaluation methods to assess these activities. CIBER-U datasets and exercises have been being actively incorporated into curriculum materials at the involved institutions since 2006 and the team is engaged in longitudinal studies to assess the effectiveness of these materials.

The implementation of the engineering semantics extensions to the CIBER-U MediaWiki was performed by a team of team of 3 graduate and 3 undergraduate student researchers over a two year period. Most of the research effort went into modeling the digital engineering archive domain, creating workflow models and various ontologies for capturing engineering form, function, and file format relationships. Tools such as Protégé were used for modeling ontologies, workflows were captured visually as mindmaps and translated into formal representations as needed. These ontologies and rules are currently operated within and behind Semantic MediaWiki. The ontologies are used to "markup" template pages and capture formal relationships among pages on the Wiki, i.e., file formats are cross referenced using the format registry ontologies. The current system is maintained by the teaching assistants using the CIBER-U wiki in the classroom; extensions to the ontology and workflow content are developed on an as-needed basis by research assistants. Data migration tasks have not yet proven to be an issue.

For the student or faculty CIBER-U user, the semantic representations are largely invisible. The format registry, workflows and rule sets are operational behind the Semantic Media Wiki interface. The information package representations, workflows, context models etc all facilitate the backend processing of data elements uploaded into the CIBER-U Semantic Wiki. For example, the format registry drives the automated processing of model translations into neutral file formats and the indexing of these 3D models for queries. Context information provides associations among file types based on date and workflow relationships (i.e., models can be automatically connected with associated simulation or manufacturing data).

There are two ways to view assessment. First, how well does Semantic MediaWiki technology support the student education and learning objectives for the CIBER-U Project? While typical CIBER-U users are not aware of the use of semantics, the team has initiated a comprehensive evaluation of Wiki technology for undergraduate engineering education (Devendorf *et al.* 2009). One of the strong conclusions of our early evaluation is that typical engineering undergraduates want cyber-infrastructure technologies to be easy to use and they are quite adverse to having to become "techies" in order to accomplish engineering-centric work and educational goals. From this viewpoint, the semantic structures reduce the need for students and practitioners to understand all of the vagaries and nuances in the multitude of engineering file formats and software relationships; they also improve the likelihood of data being directly usable by students and educators without considerable manipulation with software systems that they may not be familiar with or that may be unavailable.

The second evaluation element creates a very difficult problem: how does one assess if the semantic technologies foster preservation planning, archive management and the exchange of knowledge through time? Does one need to evaluate the effectiveness of a semantic wiki (as opposed to just a wiki) for these purposes? The position of this team is that formal representation and capture of semantics can (and will) improve long-term digital preservation over 30-to-50 year periods. This kind of assessment cannot be done directly, but will be considered longitudinally as part of the system and information maintenance process.

Conclusions

This paper presented a methodology to support long-term preservation of "born-digital" engineering artifacts. Digital preservation for engineering enterprises poses unique challenges due to the complexity and diversity of the datatypes involved in CAD/CAM, simulation and other aspects of the product realization process. The approach presented in this paper develops shared knowledge about engineering formats and workflows. The *format registry* captures how these diverse and complex formats interact, enabling long-term interpretability and monitoring their evolution over time. The *engineering workflows* formally capture the context required to interpret engineering data over time, as well as understand vital relationships among datatypes.

Currently, these techniques are actively deployed as part of the CIBER-U National Engineering Dissection Collaboratory. While the techniques exist behind the scenes of the Cyber-Infrastructure used by CIBER-U, they are actively being used by hundreds of engineering design students annually. It is expected that the use of Semantic Wiki technologies to support this effort will improve the ability of educators to teach engineering design and enable the creation of an extensible library of virtual product dissections.

Beyond CIBER-U, the digital engineering data format registry is being contributed to support the ongoing efforts of several private and government organizations in tracking engineering data elements and their interactions in order to create persistent, 100-year digital engineering archives.

Acknowledgments. This work was supported by the Library of Congress' National Digital Information Infrastructure and Preservation Program (NDIIPP) through the National Science Foundation (NSF) DIGARCH Program Grant IIS-0456001. Additional support was provided by the NSF's Office of Cyber-Infrastructure Grant OCI-0636273. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF or the other supporting government and corporate organizations.

References

Abrams, S., and Seaman, D. 2003. Towards a global digital format registry. *World Library and Information Congress: 69th IFLA General Conference and Council*.

Barkmeyer, E. J.; Hopp, T. H.; Pratt, M. J.; and Rinaudot, G. R., eds. 1995. *Systems Integration for Manufacturing Applications: Background Study*. NIST-IR 5662.

Darlington, M. J., and Culley, S. J. 2008. Investigating ontology development for engineering design support. *Adv. Eng. Inform.* 22(1):112–134.

Devendorf, M.; Lewis, K.; Simpson, T. W.; Stone, R. B.; and Regli, W. C. 2009. Evaluating digital product repositories to enhance product dissection activities in the classroom. *ASME Transactions on Computer and Information Science in Engineering*.

Gil, Y.; Deelman, E.; Ellisman, M.; Fahringer, T.; Fox, G.; Gannon, D.; Goble, C.; Livny, M.; Moreau, L.; and Myers, J. 2007. Examining the challenges of scientific workflows. *Computer* 40(12):24–32.

Gladney, H. 2006. Principles for digital preservation. *Communications of the ACM* 49(2):111–116.

Grüninger, M., and Fox, M. S. Methodology for the Design and Evaluation of Ontologies. In *IJCAI Workshop on Basic Ontological Issues in Knowledge Sharing*.

Grüninger, M., and Menzel, C. 2003. The process specification language. *AI Mag.* 24(3):63–74.

Kim, J.; Spraragen, M.; and Gil, Y. 2004. An intelligent assistant for interactive workflow composition. In *International Conference on Intelligent User Interface*, 125–131. New York, NY, USA: ACM Press.

Kopena, J., and Regli, W. 2003. Functional Modeling of Engineering Designs for the Semantic Web. *IEEE Data Engineering Bulletin* 26(4):55–62.

Library of Congress. 2006. Sustainability of Digital Formats: Planning for Library of Congress Collections.

Marchese, A.; Ramachandran, R.; Hesketh, R.; Schmalzel, J.; and Newell, H. 2003. The Competitive Assessment Laboratory. *IEEE Trans. on Education* 46(1):197–205.

Ram, S., and Liu, J. 2006. Understanding the Semantics of Data Provenance to Support Active Conceptual Modeling. *Active Conceptual Modeling of Learning Workshop*.

Regli, W. C.; Hu, X.; Atwood, M.; and Sun, W. 2001. A survey of design rationale systems. *Engineering with Computers* 16:209–235.

Sheppard, S. 1992. Dissection as a Learning Tool. *IEEE Frontiers in Education Conference*.

Thilmany, J. 2005. Ephemeral Warehouse. *Mechanical Engineering* 127(9):1–10.