A Framework for Preservable Geometry-Centric Artifacts

William C. Regli Michael Grauer Joseph B. Kopena
Department of Computer Science
College of Engineering
Drexel University
Philadelphia, Pa, USA

ABSTRACT

Digital preservation is the mitigation of the deleterious effects of technology obsolescence, media degradation, and fading human memory. For engineering, design, manufacturing, and physics-based simulation data this requires formats that are semantically accessible for 30-to-50 year lifespans. One of the fundamental challenges is the development of digital geometry-centric engineering representations that are self describing and assured to be interpretable over the long lifespans required by archival applications.

This paper introduces the challenge of long-term preservation of digital geometric models. We describe a digital preservation case study for an engineering model which required, for just a single part, over 3.5 GB of data, including 39 file formats and over 750 distinct model and shape files.

Based on lessons learned in this case study, we present a framework for enhancing the preservation of geometry-centric engineering knowledge. This framework is currently being used on a number of projects in engineering education.

Categories and Subject Descriptors

I.3.5 [**COMPUTER GRAPHICS**]: Computational Geometry and Object Modeling, Modeling packages; H.3.7 [**Information Systems**]: INFORMATION STORAGE AND RETRIEVAL,Digital Libraries,Standards

1. INTRODUCTION

Digital Preservation is the mitigation of the deleterious effects of technology obsolescence, media degradation, and fading human memory. 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 describes some of the challenges related to the long-term retention of geometry-centric information in Digital Engineering Archives. Digital Engineering Archives are digital repositories for engineering design, manufacturing and life-cycle data. Engineering Archives require formats that can be accessible for 30-to-50 year lifespans. Further, the complexity of formats and multitude of metadata makes it particularly challenging to define structures that form integrative representations.

2009 SIAM/ACM Joint Conference on Geometric and Physical Modeling (SPM '09), October 4-9, 2009, San Francisco, CA.

Archiving digital engineering data presents additional problems beyond those of archiving in general, such as the mathematical complexity and the proprietary nature of many CAD file formats, as well as the need to capture organizational workflows. Unlike "traditional" digital media (i.e., videos, images, audio), engineering data is largely useless without the broader product model context. For example, the primary goal of an archive of an audio track is to support those that wish to analyze, study or simply listen to the audio. In contrast, the goal of the engineering archive is to support understanding of the product, its design and manufacture. Since so many of the processes and data formats involved in engineering design are complex and opaque, it becomes necessary to develop knowledge structures that can be used to archive a wide variety of individual data formats and capture the vital inter-relationships needed to understand the engineering records.

The needs of digital preservation for geometry-centric objects are diametrically opposed to traditional research advancements in geometric representation. The fundamental property of a preservable geometry object is to enable information exchange across temporally distinct software epochs. Ordinary data exchange focuses on translation between different software systems (i.e., data exchange) or across incremental version releases of the same software (i.e., data migration). Representations suitable for digital preservation need to inter-operate with "future" software systems (i.e., those yet to be implemented) and cannot assume that these systems share much, if any, similarity with existing tools. Depending on the domain, the temporal epoch could be measured in decades (i.e., for aerospace artifacts) or centuries (i.e., for architectural and civil structures)—thus ensuring that the very existence of corporations and software vendors cannot be assumed.

Our approach is to take representations for geometry-centric preservation and augmented them to include semantics beyond the geometry itself. For example, some geometry-centric representations are procedural or feature-based, resulting in the need to also understand how to interpret the execution of the process for generating the shape. Further, future interpretability of geometry-centric objects requires some capture and association of *context*. For purposes of this work, *context* describes the use, functions or process associated with a geometry-centric artifact that are essential to maintaining its future utility. For example, to understand an individual discrete part might also require some semantic model of its role in a larger assembly; its manufacturability may require having saved models that describe intermediate "in-process" shapes as well as jigs and fixtures, etc.

This paper presents a framework in which geometry-centric artifacts and their associated physics-based models, manufacturing models, simulations, etc can be captured and packaged with a reasonable guarantee of future interpretability. To motivate this work, we provide a real-world example using a digital preservation case study we performed on the Advanced Model-Based Engineering Realization (AMBER) part—an test artifact from an automated manufacturing demonstration by the Ministry of Defence of the United Kingdom. Describing just this single part required over 3.5 GB of data, including 39 file formats and over 750 distinct model and shape files. Our belief is that the case-study represented with AMBER is commonplace in engineering practice today and lessons learned from this dataset may be generalizable to other engineering preservation problems.

The paper is organized as follows. Section 2 introduces the general problem of digital preservation and presents background related to preservation of geometric models, physics-based models and engineering design information. Section 3 provides the case-study of the AMBER part and summarizes observations. Section 4 gives our framework for enabling long-term preservation of digital geometry objects and their related metadata. As part of this presentation, we provide a detailed application of the framework to the AMBER dataset. Sections 5 and 6 summarize our lessons learned and define goals for future work.

2. MOTIVATION AND BACKGROUND

General Background on Digital Archiving. An *archival information system* is 'an archive, consisting of an organization of people and systems, that has accepted the responsibility to preserve information and make it available for a Designated Community,' the latter being 'an identified group of potential Consumers who should be able to understand a particular set of information' [5]. Much of the work in digital archiving focuses on how to save digital objects so their content can be understood at some distant point in the future [14, 16, 14, 16, 15]. Basic assumptions include that no dialog will be possible with future users and the software and hardware platforms used will be totally different than those used currently.

Much of the work in digital archiving has come to be framed by ISO Standard 14721:2003, "Reference Model for an Open Archival Information System" (OAIS) [10]. The OAIS Reference Model began as an effort to meet the need for standardizing and constructing archives of datasets created in the space sciences (i.e., how to handle terabytes of data from Earth-orbiting satellites). OAIS defines archiving as "Long Term" preservation of information over cycles of technological obsolescence, rather than any specific time frame, i.e., *Long Term* is long enough to be concerned with the impacts of changing technologies, including support for new media and data formats, or with a changing user community. Hence, "Long Term" may mean "indefinitely". OAIS provides a common language with which to discuss archiving, in the form of an abstract breakdown of the information flows and functional components present in any archive. This section quickly overviews the OAIS model. In addition to the publicly available reference model, several other longer guides also exist [5].

Archives are either *bit-oriented* or *document-oriented*. Bit-oriented technologies aim to create format-agnostic, persistent, survivable storage and typically focus on hardware or infrastructure-centered solutions. Although a critical archiving component, this is not enough. Unless the syntax and semantics of those bits are also preserved, there can be little hope of interpreting the data in a meaningful fashion. This is particularly true in engineering design, which features intricately structured data with tight margins of error in interpretation. Document-oriented technologies focus on the domain content and semantics, aiming to ensure the preservation of the document content (i.e., the interpretation of the bits) over time. Typically some combination of the two approaches is what is pursued.

Given documents one wishes to preserve and a persistent bit storage mechanism, the two major approaches to digital preservation are *conversion*, where digital files are constantly updated and translated to new encodings that can be run on whatever computing platforms exist through time, and *emulation*, where the execution environment that the digital object originally was interpreted using will be emulated on future platforms [24]. Both of these approaches have the goal of attempting unambiguous communication with the future, and both attempt to maintain obsolete computing environments. The conversion approach is very much similar to data translation as implemented by the CAD industry today [32, 35, 31, 28].

For *emulation*, a complete record of the computational environment must be maintained, which is difficult, error-prone, and tends to waste effort preserving irrelevant information. Some emulation scenarios call for a "nesting doll"-type approach in which successive incremental generations of emulators each are responsible for accessing the previous generation. The *conversion* approach requires files be migrated as formats and computing environments change. This approach typically takes a file-centric view, often neglecting to maintain relationships among data elements. Hence, the conversion approach assumes that all relevant information can be circumscribed and converted—something that might prove difficult if one cannot formally define the relationships among all relevant metadata files.

Digital archeology places the burden on future consumers of information to extract any meaning from archived digital objects. This is only possible for simple file formats, and is an extremely expensive way to recapture information [34]. The digital archeology perspective does provide insight into how future users attempt to make sense of past data. If those future users can be seen as 'digging' the information out of the 'ground', then their job can be made much easier by 'placing' the archived data in the 'ground' in as structured a way as possible, so that semantic relationships between files are captured, and the context of a given file in a larger workflow is stored.

Digital preservation domains such as audio, video, still image, and GIS etc., have been developing format registries and "best practices" in order to better comprehend how to process archived files [37, 22, 2]. Such efforts are largely static, and focused on relatively few media types and relatively simple future use-cases for preserved data. Other domains of significant digital archiving efforts include document and science data preservation. In comparison to document archiving, e.g., [33, 30], the structure of both the data itself and the relationships between the data is much more complex in the engineering domain. General document and image formats also have a much larger user base, substantially improving the chances that data formats will be interpretable far into the future. Metadata extraction and effective search functionality is much easier to provide. Science data, e.g., astronomy observations or protein sequences, typically possesses a simpler and more homogeneous structure than engineering data, most often taking the form of voluminous databases of measurements or other records.

Challenges in Archiving Engineering Models. Design Repositories [36] and product lifecycle management (PLM) systems both aim to support design and operational activities throughout an engineered products' lifecycle. However, existing software technologies implicitly assume that stored data will always be accessible, irrespective of changes in the applications themselves, supporting platforms, or even the underlying hardware. Over the lifecycle timespans for current engineered products in the aerospace and civil infrastructure domains (i.e., decades, if not longer) such assumptions about data readability and software are not realistic. Without specific procedures and technology for preserving digital data through fundamental system changes, engineering design knowledge stands little chance of surviving.

While 3D computer-aided design (CAD) modeling has become an indispensable aspect of modern engineering, 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 in this process. Even considering data that are archived, over a long enough product lifetime, the data files and supporting infrastructure required to access CAD product designs will be obsolete and unusable [38].

Current industrial and legal policies for addressing this problem and archiving engineered designs remain largely based on paper drawings as the archival standard. Paper has the advantage that its long term preservation is well understood, drawings are unsuited to preserving design data for any number of reasons, including ambiguity, difficulty of reuse, and limited scalability of management and retrieval. Some archives have begun to incorporate bitmap encoded (e.g., TIFF), digital equivalents of paper drawings. Although perhaps easier to manage with existing library science technologies, paper archives are poorly suited for capturing a plethora of other critical born digital data, e.g., simulation results, manufacturing plans, etc.

STEP, the Standard Exchange of Product model data (ISO Standard 10303:2001), a standard open format for 2D and 3D geometric data developed to promote interoperability, only solves part of the problem. In addition to storing the product geometry in a non-proprietary format, the design history and context for a product must somehow be saved. There are numerous engineering data types that do not have any standard, and even for those data types with standards, the standards can change over time in subtle and complex ways.

One important archiving effort specific to engineering is the studies of the LOTAR Group, a consortium of aerospace manufacturers. Much of the group's work focuses on processes for preserving geometry [1], largely based on STEP. This is an important element, particularly as any successful engineering archive must ultimately be incorporated into business and design workflows.

There has been a great deal of research creating ontologies for engineering design processes, such as for engineering function [20] and engineering requirements [23], but none with the engineering file format as its central focus. Similarly, the Aim@Shape Digital Shape Workbench project has created an ontology of shape types and shape processing methods; however, the shape representation hierarchy there is not sufficient to describe geometries and formats of engineering models [3]. Thus there is a void in digital engineering archives work, as it lacks a model of the concepts of file formats, underlying model shapes and simulation data, engineering design processes, and the relationships between them. In order to archive engineering data with the goal of using and understanding the data in the future, it is clear that archives must contain unambiguous interpretations of the data and metadata. Due to the complexity of the data and standards, automated methods must be created to translate and maintain data interpretability. Both of these goals argue for formal representations with machine interpretable semantics.

Short History of Digital Representation of Geometry.

Developing digital representations of shape objects has been a central research challenge in geometric modeling, computer graphics and computer-aided design over the past 40+ years. The result of these efforts is a cornucopia of formats that relate to object geometry, product model data, features and other geometry-centric representations. Each of these representations has its own origin, merit and utility. These geometry-centric representations fall into several classes:

- Those that describe shape representations of discrete objects. These representations include mesh formats (i.e., .stl, .ply, .smf, vrml) as well as existing proprietary and international standards (i.e., STEP, DXF, .prt, etc).
- Those that describe shape metadata, such as features, tolerances, etc, usually associated with a single discrete object. In some cases, these descriptors are integrated into the shape representation. For example, proprietary formats from CAD vendors include these information elements embedded in them as annotations or other structures linked to the core geometry.
- Those that describe assemblies or aggregations of objects, such as a mechanical assembly or animation artifact. These representations must describe metadata associated with assembly or object aggregations, such as joints, assembly features, and their range of motion, parameters, etc.
- Those that describe external functional or behavioral properties associated with an object or aggregations. This includes information about kinematics and dynamics, parameters of motors and actuators, and their interconnections.

As these representations have advanced, emphasis has been on improving their mathematical robustness, expanding their representational power, or augmenting their capabilities to suit user needs or downstream application needs. These representations have vastly improved the capabilities of CAD systems, 3D geometric modeling environments and the underlying representations available for development of 3D computer graphics. The principle drivers for these developments are nearly always to enable near-term advancement and the development of new geometry-centric tools. In many cases these representational advancements are proprietary or protected intellectual property.

Preservation in a Geometric Modeling Context.

Geometry-centric models are at the core of the modern engineering enterprise [38], documenting trillions of dollars of intellectual property and human history across aerospace, civil, architectural and other industries. For long-term preservation, the natural question arises "Who's problem is this"? Is this a problem for the modeling and graphics community or one left to the digital librarians and curators?

This paper takes the position that challenges of digital preservation of geometry-centric engineering objects is a interdisciplinary challenge that requires the direct effort of geometric modeling, graphics and CAD community to develop representations, algorithms and techniques to ensure this valuable knowledge is sustainable over time. Early advances in graphics were driven by engineering and manufacturing needs, resulting in CAD systems and algorithms to automate the generation of NC codes to drive machine tools. Preservation is a critical current application need for the graphics and CAD communities. Just as bio-medical, gaming, animation, virtual reality and design domains have driven considerable advances in solid, physical and geometric modeling in the past

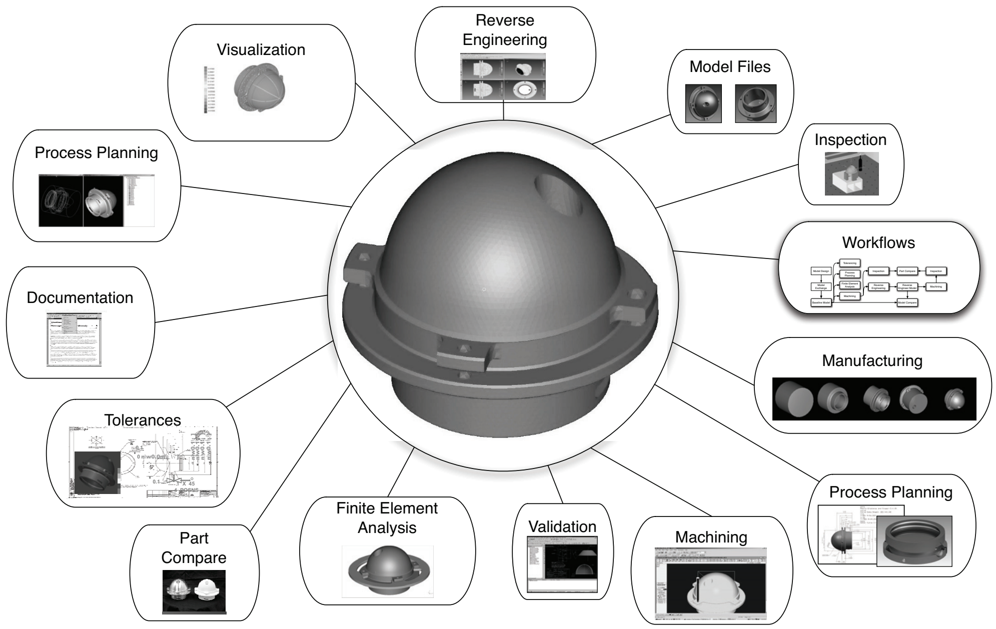


Figure 1: The AMBER part and its archival information space.

decade, the needs of geometry-centric preservation require both the domain knowledge of the curators as well as the mathematical and scientific techniques of the computer scientists, engineers and mathematicians working on these core algorithms and representations.

3. A CASE STUDY

The Advanced Model-Based Engineering Realization (AMBER) model, shown in Figure 1, is a single machined part designed for a manufacturing systems integration demonstration by the UK Ministry of Defence. In the UK, the part was the centerpiece of an end-to-end digital manufacturing exercise. This exercise was subsequently reproduced as part of an international collaboration with the US Department of Energy's Kansas City Plant (KCP). Our AMBER dataset is from this international collaboration project. In this collaboration, every step of all processes was driven by digital model files—thus all information for the project was "born digital". There are many complex models, transformations, and detailed workflows that were captured over the course of a "design", "simulate", "manufacture" and "verify" exercise. The effort involved dozens of scientists and engineers and produced, for just this single part, over 3.5GB of data, spanning nearly 1,000 files in 39 file formats, with over 750 model and shape files. Of these files, the vast majority were related to the task of inspection planning (64%) and approximately 3% were different versions of the same solid model that was the object of the design-analysis-manufacturing cycle. A depiction of this information space of the AMBER part is shown in Figure 1.

Capture of the AMBER Information Repository. The AMBER dataset was captured via post-event interaction with the technical staff on the project. Relevant data was gathered off of storage devices, shared data management systems as well as from individuals who created derivative data in the course of executing the exercise's workflows. In one case, a derivative model was created for simulation purposes as the simulation environment could not accept the baseline model format for input; in other situations, important manufacturing information relating to jigs, fixtures and setups was finalized on the machine shop floor and not associated with the product model or the manufacturing plan of record. A considerable amount of these elements were gathered as part of this exercise, however it is almost certain that hundreds of files relevant to achieving deeper understanding of the artifact and its manufacture were not captured in this process. For example, few stored previous versions of models, plans, etc let alone the or "dead ends" that might have been explored in the course of creating the artifact. These elements of the design history and rationale are not something most organizations systematically keep.

The current standard for engineering archives is the 2D drawing as shown in Figure 2. The advantages of this approach is that the archiving process is simpler and better understood. Since the 1990s engineering design work has become based largely on 3D solid models of products, and the conversion to 2D drawings loses information, including manifest and product data, manufacturing plans, and analysis and tolerance data. Note, however, that the blueprint only captures shape information and does not capture information about the manufacturing, inspection, simulation or other elements of the AMBER part's design-analysis-manufacturing lifecycle.

Organization of the AMBER Information Repository. Upon acquisition, the AMBER dataset consisted of several DVD ROMs of files. These files occasionally had organization: for example, all file related to inspection planning were in a set of subdirectories. However, there was considerable redundancy: there were dozens of copies of the "baseline" file, thus creating considerable ambiguity as to which was the "baseline" of record. workflows were implicit in the file structure and, occasionally, documented

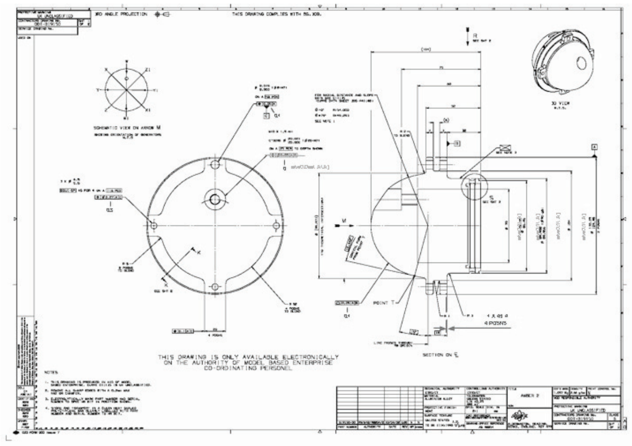


Figure 2: The current standard in engineering archives.

in detail in the associated text and other documentation [13, 26]. Hence, for several key files, the interpretation of the geometric information was not possible without access to the non-geometric metadata found in other files.

An analysis of the dataset identified patterns among the files (i.e., those that served as input to different workflow functions and those that were the outputs or results) as well as a set of files that, when taken together, provide a fuller representation of the geometric artifact.

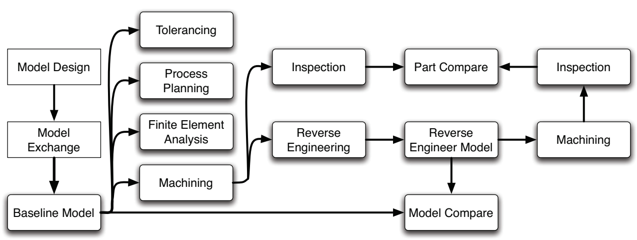


Figure 3: The top level workflow for the AMBER part.

Engineering Workflows for AMBER. Essential to understanding what geometry-centric information must be captured requires understanding the workflows associated with the AMBER, as shown in Figure 3. A workflow describes a business process related to the design and production of an artifact. The top level workflow for AMBER shows the relation between objects produced in the `Model Design` to those created as part of the fabrication process, requiring a `Model Exchange` phase, and the acceptance of one derivative model as the `Baseline Model` of record. The `Tolerancing` and `Process Planning` steps produced the manufacturing plans for the part; and structural analysis (`Finite Element Analysis`) activity verified the part's performance under operating conditions. The `Baseline Model` also drove the `Machining` step, where NC machine codes were created and the part was manufactured on a CNC lathe. The Inspection activity verifies the geometry based on acquired data from a high-precision point probe; the manufactured part was also verified via scanning into a point cloud model via `Reverse Engineering`. These point cloud models were then used to create a derivative model which was then used to create a new part via a second `Machining` step. The reverse engineered part underwent `Inspection` and was `Part Compared` against the `Inspection` data of the part machined from the `Baseline Model`. Finally, the `Reverse Engineer Model` was compared to the `Baseline Model` in the `Model Compare` step.

Observations about AMBER. The AMBER dataset is representative of the state of the practice. Preservation is typically an after-thought to the business processes, typically performed after-the-fact for legal or other compliance reasons and done by individuals who are outsiders to the process itself. What this means is that the organization, analysis and storage of archival engineered records is usually done by individuals who may not be intimately involved in the design and manufacturing process. Rationale at varying levels of an organization are captured in a raw form, perhaps suitable for future forensics or analysis. Data files will be organized by date, activity or based on the names of the individuals who may have touched them or from who's computers the files were copied. In contexts where a product lifecycle management or other groupware system is used for file management or version control, some portion of the records will be kept in a central store but need to be extracted and merged with data gathered from throughout the engineering activity.

Of particular relevance to the problems as presented in this paper, the central element to all of these activities is the geometry-centric representations of the artifact. These representations are common across all engineering activities for AMBER. The AMBER model gets translated, remodeled, re-factored and imaged across all of the tasks in the repository. Hence, understanding the geometry model and its role in connecting all of these activities is the central challenge in the digital preservation of engineering artifacts.

4. A FRAMEWORK FOR 100-YEAR GEOMETRY-CENTRIC OBJECTS

The following were guiding principles for developing requirements for preservable geometry-centric objects.

- The cost of physical storage is negligible. Previous work on geometry-centric representations occasionally focuses on compactness, compression (for storage or transmission) and memory footprint. For purposes of preservation, the size of the data object for storage is not a significant issue.
- Redundancy is a good thing. Rather than argue the merits and challenges of developing unambiguous representations, redundancy can be exploited to reduce ambiguity and improve data preservation. Having two, three or more ways to represent an mathematical concept or topological structure provides additional methods of coding data for preservation so that future generations can disambiguate the intention of the designer and software developers.
- *Pluralitas non est ponenda sine necessitate*.[1] A layered approach to representation, one that starts with common assumptions and the simplest representations, will be the easiest to preserve. For example, a STL file is quite rudimentary and largely self-describing for those with a basic geometry knowledge and the ability to read and parse a text file.
- The future use-cases for the preserved data are open ended. In previous studies [6, 25], results from surveys and industrial users revealed that future uses are often unexpected and the greatest need is to ensure a comprehensive and accurate set of data for a wide variety of these use-cases exist rather than a set of data tuned to any one possible application.

[1]Entities should not be multiplied unnecessarily, William of Ockham (c1285 -1349)

- Preservation of context is vital to future data interpretability. In this way, digital preservation of geometry-centric engineering objects shares many issues with design rationale capture and retrieval [33] and work in scientific workflow management [13]. Most data files do not exist in a vacuum and the context representation is vital to the interpretation of the meaning in the data.
- No one geometric or CAD representation method is universal, nor is there one best method. In the history of CAD, solid and geometric modeling, representations are all developed with some application or specific purpose or application in mind. As a corollary, there is no one subset of representations that encompass all potential future use-cases, hence it is expected that various aspects of design semantics, shape, feature or other metadata will be lost with time. A practical approach, therefore, aims to be satisficing, a "best effort", and not theoretically ideal.
- Generative knowledge, where available, is preferable over pure geometry and topology representations. For example, in Architecture, Engineering and Construction (AEC) domains, the generative descriptions in the Building Information Model (BIM) [12] contain considerably more semantic and design intent than does the geometry description of a building alone. Similarly, for CAD/CAM models, the CSG tree, feature-based models, feature construction history, and parametric constraints reveal (in general) more about the design than the mathematical objects that form the geometric or topological model [28, 39]. Bluntly, a feature construction history contains considerably more information than a Brep and if one had to choose between the two, it is better to capture the feature construction history as the Brep can be uniquely derived from the features (but not vice versa).
- No single model is isolated, process and workflow knowledge are vital to maintain in order to preserve interpretability. For example, a single model of an artifact may reveal its final design geometry but not the manufacturing models, the "in process" shapes, required to machine and fixture the model. Hence, preservation of a single artifact may require links to dozens of other models as well as workflow and process descriptions that describe the relationships among and between these models.

4.1 Technical Basis

The basis for a possible solution is the concept of *forward error correction* (FEC) from information theory[]. In a networking and telecommunications context, forward error correction adds redundancy to data, in the form of redundancy checks or other error correcting codes, in order allow the receiver to properly decode the message after is transmitted over a lossy channel with no need for a back-channel for communications back to the transmitter. FEC can provide a framework for guiding the design of representations for exchange of geometry-centric engineering data over time, as one cannot assume that the receiver has access to the transmitter of the data. In this context, the transmitter includes not only the native software applications that may have created the data but also the human agents and other software and systems that provide the technical and social context for interpreting the semantics of this data.

In this context, one is not concerned with the typical communications issues, such as maximizing bandwidth; rather the objective is to encode a representation or set of representations that maximize the the receiver's likelihood of being able to interpret the data and understand its semantics. Such an encoding is not at the bit-level, but lies instead at the level of the application semantics and potential use-cases for the data by the receiver.

Geometry-Centric Representation(s). Digital geometric representations are all approximations of an idealized mathematical object. Some approximations are, naturally, more accurate; others may sacrifice accuracy for processing costs, memory or ease of use. Depending on the method by which the object is defined, system parameters regarding the mathematical tolerances[2] are essential for defining a consistent interpretation of a shape or an assembly model. Here, issues such as round-off error, floating point inaccuracies, and the representational precision of the underlying mathematical objects all contribute to the subjectivity of the interpretation of models. In addition to the well-known ambiguities in these interpretations, there are morphologies that are introduced by the need to maintain consistent topological interpretation for a given object. For example, if an object is represented as a set of feature-based operations, feature editing operations can have multiple, mathematically consistent—but perhaps non-intuitive— interpretations. Usually system implementors select a single interpretation in order to produce consistent geometry, however this may not be explicitly represented in the procedural model. A similar problem occurs with handing the merger of surfaces and edges resulting from blending operations.

While we would like to believe that our mathematical sophistication and computer-aided tools have created an idealized universe in which we can interact with virtualized physical models, the situation shares similarities with the pre-digital era of drawings and paper. Engineering drawings, such as the one shown in Figure 2, were intended for the conveyance of ideas among human beings. Drawings were used to communicate design intent, prescribe manufacturing operations, illustrate how an artifact or building was to be assembled, or show how a device might be used. The role for digital representations of 3D shape and engineer data is largely the same; what has changed is that we have a computer-mediated form of communications in which software is required for the humans to fully interpret and visualize the information they are communicating over.

Given these issues and limitations, the proposed method is to encode geometry-centric model information as a triple:

$$\mathscr{M} = \langle D, S, N \rangle$$

where

- D is a discrete mathematical representation of the geometry-centric artifact. For most CAD/CAM objects, solid models, etc, D will be an approximate representations such as a mesh, polygon or point-based graphics file. Common extensions for the current generation of these file formats are WRL, STL, PLY, etc. These representations have the advantage of being very simple and nearly self-describing. For example, the grammar with which to parse the STL format is easily reverse engineered by simply looking at an example STL file.
- S is an appropriate standard model representation format. For most 3D models, CAD models, etc, the current representation of choice would be ISO 10303 STEP or, perhaps, IGES [18]. These representations have several advantages over discrete representations, including of higher mathematical precision and, in some case, methods for capturing constraints, features and model construction history.
- N is a copy of the model in its native representation. As an representation of object of record, the model N is the closest

[2] As opposed to manufacturing tolerances.

approximation and it is with respect to N that metadata elements are associated. Models D and S are derivative of N, as are the inputs and outputs of the other processes (described below).

While in some domains it is not possible to have each of D, S and N (i.e., for some graphics/modeling applications the discrete model is also the native model), the triple can be viewed as forward error correction encoding scheme designed to enhance, but not necessarily guarantee, future interpretability of the geometry information.

Metadata Representation. As noted by [31], the semantics of the geometry-centric representations can only really be determined at run time, within the operating context of the software systems (CAD, CAE, FEA, etc) that act on these representations. In the context of geometry-centric preservation, however, certain aspects of these semantics need to be made explicit. Our approach advocates adopting metadata representations techniques from other preservation domains, in which there are many emerging technologies and techniques for creating and preserving metadata representations, including:

- The Metadata Encoding and Transmission Standard (METS)[3] is an XML format that can reference files or contain embedded files.
- The XML Formatted Data Unit (XFDU)[4] contains the metadata information in a manifest, and may contain the referenced files or they may be separated.
- ISO/IEC 21000-2:2005, better known as the MPEG-21 DIDL, is focused on allowing multimedia resources to be executed across differing devices and networks, and must be a single document which combines metadata with referenced files.
- IMS Content Packaging[5] is intended to describe the contents of physical container files and logical file packages. The manifest document describes the contents of the package, and can recursively reference included manifests.
- Ontological Representations have found successful application in a number of engineering design contexts, including the representation of metadata associated with material properties [4], engineering designs [11], function [20], requirements [23], and manufacturing processes [21].

These methods present varying degrees of flexibility for noting relationships between the packaged contents. The relationships considered focus on the logical relationship of files in the packages as dependencies or parts of assemblies. What these current methods lack is a way of describing the semantic relationships between the files, and placing the files in the context of a semantically described process, such as a workflow.

Our approach uses W3C and Semantic Web standards to create ontologies in OWL [27] that describe the metadata associated with geometry-centric objects. These representations can capture model relationships, define taxonomies of functional attributes, joints, simulation parameters and other (usually proprietary or ad hoc) information required to interpret geometry-centric data in the context in which it was originally created. These representations have several attributes that are beneficial to preservation requirements, including that they are text-based, open standards that are relatively self-describing.

[3] `http://www.loc.gov/standards/mets`

[4] `http://sindbad.gsfc.nasa.gov/xfdu`

[5] `http://www.imsglobal.org`

Process and Workflow Representation. Scientific workflow representations that handle large, distributed, scientific datasets have begun to emerge in the course of a variety of large-scale collaborative projects. For example, the Southern California Earthquake Center (SCEC) and USC ISI's Pegasus[6] provide formal mechanisms for capturing executable workflows. Similar representational ideas based on work on automated planning technologies as well, including the Business Process Modeling Language[7], the Shared Planning and Activity Representation[8] and the Core Plan Representation (CPR)[9]. In the design and manufacturing domain there have been several studies of workflow and process modeling [7] These techniques commonly use plan-oriented representations in which activities are modeled as actions with various pre-and post-conditions.

The proposed approach is to employ the *Process specification Language* (PSL) for a formal representation of the workflow. PSL is ISO TC 184 Standard 18629 language for creating machine-readable representations of processes [17]. 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. Hence, it is possible to create applications that validate workflows to ensure completeness and correctness, as well as interactively suggesting valid options to workflow designers to correct errors [19].

4.2 Implementing the Approach

Our recommended implementation of this approach is top-down, starting with identification of the processes and workflows. Workflows drive the metadata requirements and, ultimately, the selection of appropriate geometry-centric representations for creating an *archive information package* (AIP) [10]. An *archive information package* contains four kinds of preservation description information: *provenance* [30], a formal representation of the model's ownership history and any rights associated with it; *context*, as defined by the process, workflow and metadata descriptions, capturing the set of relationships between the different data objects required for future interpretability of the model; *reference*, any identifiers for the data, such as ISBNs, URIs or pointers to descriptors of the formats and models involved; and *fixity*, which are keys, checksums or other domain-specific techniques used to check for model integrity.

The AIPs are stored over the life of the archive, and are subject to *preservation planning* migrations. These migrations include *refreshment*; where an AIP is copied on the same medium; *replication*, where an AIP is copied onto a different medium; *repackaging*, where an AIP is copied and some of the packaging information can be changed; and *transformation*, where an AIP is copied or created such that the content is changed with the goal of preserving the original AIP content. For example, the *transformation* activity would occur whenever a new format for a model is developed (e.g., transforming a file in the Pro/ENGINEER version 16 format to Pro/ENGINEER version 17 format), or a format describing a new standard is developed (e.g., in the case where a successor to ISO STEP is created). With each of these transformations, the original content in the original format should be maintained to preserve the original context, along with the transformed content in the new format.

4.3 Example: The AMBER Inspection Plan

To demonstrate the application of this approach we describe how to build an archive information package for the `Inspection` sub-

[6] `http://www.isi.edu/ikcap/cat`

[7] `http://www.bpmi.org`

[8] `http://www.aiai.ed.ac.uk/project/spar`

[9] `http://projects.teknowledge.com/CPR2`

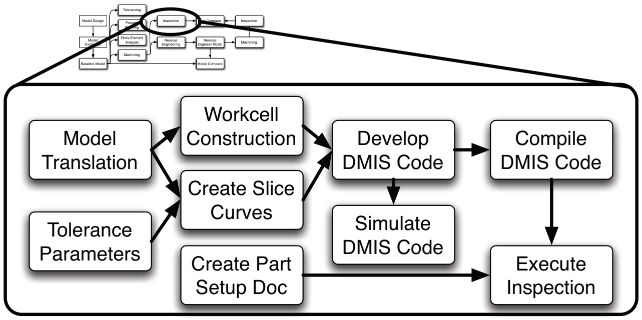


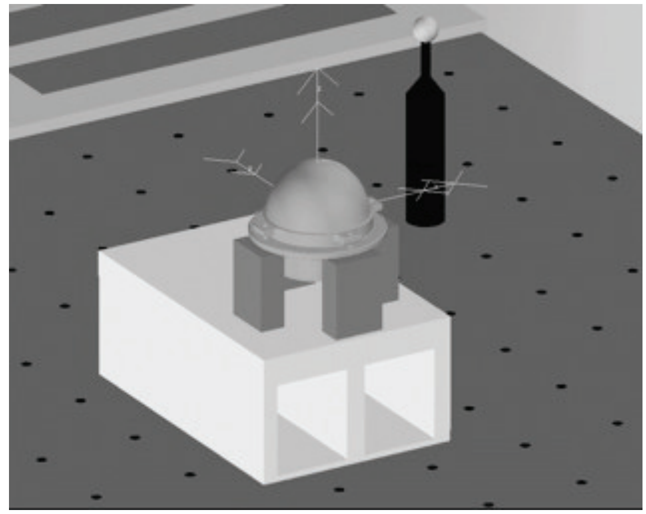
Figure 4: A detailed view of the `Inspection` workflow step.

section of the AMBER part's workflow. At this point in the actual workflow, the part has been machined and will be inspected to determine how closely the physical part resembles the CAD model of the part. First, we create a workflow for the `Inspection` stage, depicted in Figure 5(b), that comprises instances of classes from a `Workflow` ontology describing the metadata associated with this activity. These workflow elements are ultimately represented with PSL and describe inputs and outputs of each stage, along with succeeding and preceding workflow stages, and each of these is linked to the `Inspection` stage as a sub-workflow element.

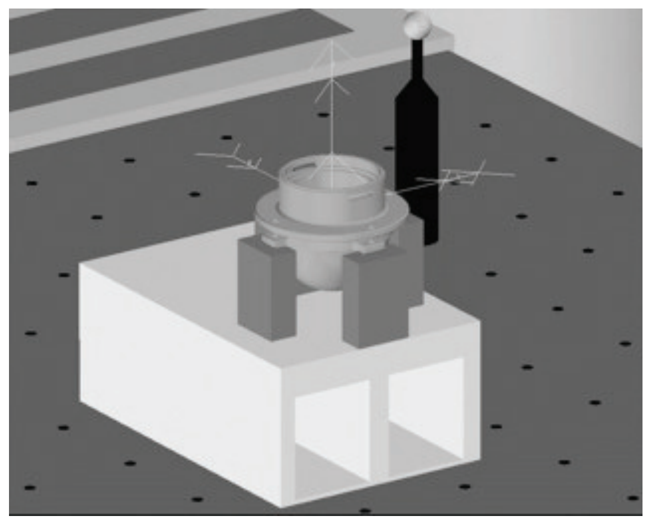
Any digital file that is included in the archive will have a `Digital File` ontology instance created and linked with it, which will be linked to the `Design Product` instance, the AMBER part in this case, and will additionally have a link to the appropriate `Format` instance and an instance of `Provenance` holding the proper values for that Digital File (e.g., the creation time, creating agent, and reason). These `Digital File` instances can be linked to as many other ontology instances as necessary to situate them in their proper context, for example various versions of the same file over time, or any `Workflow` stages that the file is involved with. The `Inspection` stage has documents relevant to the entire sub-workflow, including the Microsoft Word documents "process summary report.doc" and "inspection process plan.doc", which will have the appropriate Documentation ontology element instances linked to them. Each of the sub-workflow elements of the `Inspection` stage will now be described in turn.

Embedded in the inspection plan are several sub-processes that all require capture and preservation in order to interpret the results in the archive.

- `Model Translation:` The AMBER part was inspected by a coordinate measuring machine (CMM), which is a machine that uses a touch probe following a predetermined sequence of actions to collect data on the physical dimensions of the part. Before this sequence of measurement actions is executed, the inspection is simulated in software. The CMM simulation software, CimStation v5.01, can only import Unigraphics v17 CAD models, but the AMBER part is specified in a Unigraphics v19 format—requiring a translated step via Unigraphics v19 software to an intermediate STEP format and then again to Unigraphics v17 software to a Unigraphics v17 format. This sequence needs to have its own sub-workflow created, with ontology entries for each of the digital files, formats for these files, software used for the translations, and parameters of each of the translations. Each of the CAD files will also be described by an appropriate element of the `Model` ontology, and each piece of software will have a corresponding `Software Tool` ontology instance.
- `Tolerance Parameters:` An additional element to the CMM sequence are the tolerances and bilateral dimensions of the part. These must be extracted from a 2D drawing file Model of the part, which is extracted from the Unigraphics v19 CAD file using the Unigraphics v19 software package.
- `Workcell Construction:` The sequences of the CMM process are separated into two workcells which describe two different inspections, one inspection with the part in an up orientation, which can inspect the top of the dome, and a second inspection with the part in a down orientation, which can inspect the underside of the dome. These workcells are shown in Figure 5.
- `Create Slice Curves:` The dome of the AMBER part was designed using a rotated spline curve, but as the Cim-Station software does not have an angular increment option, a set of points along the curve slice must be manually created and entered into the inspection plan. These curve points, depicted in Figure 6, are saved as a comma separated values (CSV) file, which would also be archived.

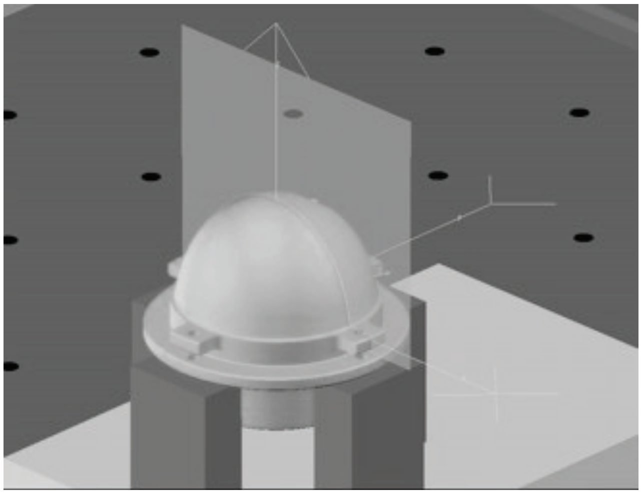


(a) Workcell for up orientation inspection.

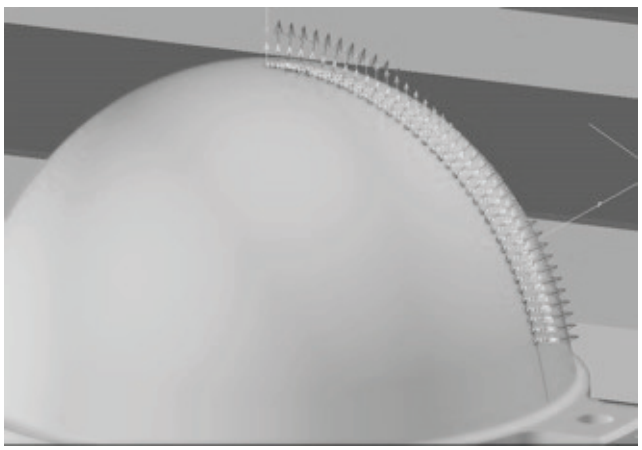


(b) Workcell for down orientation inspection.

Figure 5: The workcells constructed for CimStation inspection simulation.



(a) Plane of the slice curve.



(b) The datapoints, one every 2 degrees.

Figure 6: Development of the slice curve points for inspection.

- `Develop DMIS Code:` The slice curves and workcells in the previous workflow elements are combined in this stage as shown in Figure 7 to create Dimensional Measuring Interface Standard (DMIS)[10] code, which is the standard for CMM machinery. There are two separate DMIS source code files to be archived, one for the "up" oriented inspection and one for the "down" oriented inspection.
- `Simulate DMIS Code:` The DMIS code developed in the previous step is then simulated and tested on the CimStation software platform. An animation of the inspection simulation process is created for each orientation, and these animations must be added to the archive.
- `Compile DMIS Code:` The DMIS source code that is now tested is then compiled for the actual inspection machine, in this case a Sheffield Measurement Apollo RS70 CMM. This inspection machine must have a `Hardware Tool` entry supplied to the archive, and the compiler for this source code also needs to be archived, as is the object code output by the compiler for each orientation inspection.
- `Create Part Setup:` An MS Word document,

[10] `http://www.dmisstandards.org/`

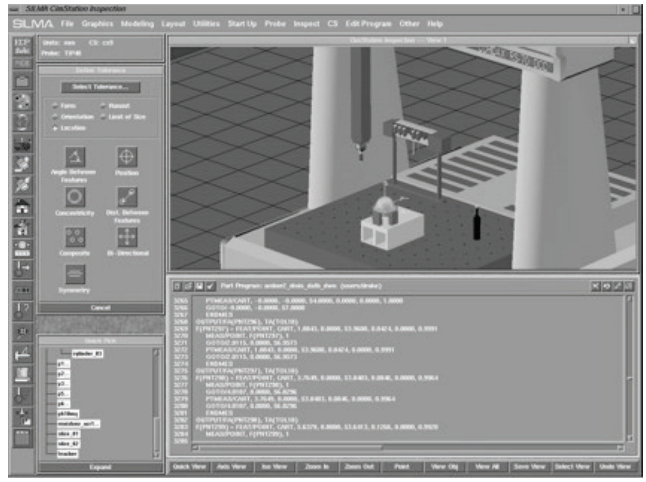


Figure 7: DMIS code development process.

`CMM_SetupSheet_Amber2.doc`, which contains inspection instructions for the users is created at this stage, and this `Digital File` should be included in the archive.
- `Execute Inspection:` The final stage of the `Inspection` workflow is to actually execute the physical inspection itself, which is shown in Figure 8. The inspection process creates `Metris` format datapoint files for each orientation, and then summary reports. The archive should include a `Digital File` entry for each of these files, along with a `Metris` format instance. The datapoints are then visualized and analyzed into an image, also shown in Figure 8, which would also be added to the archive.

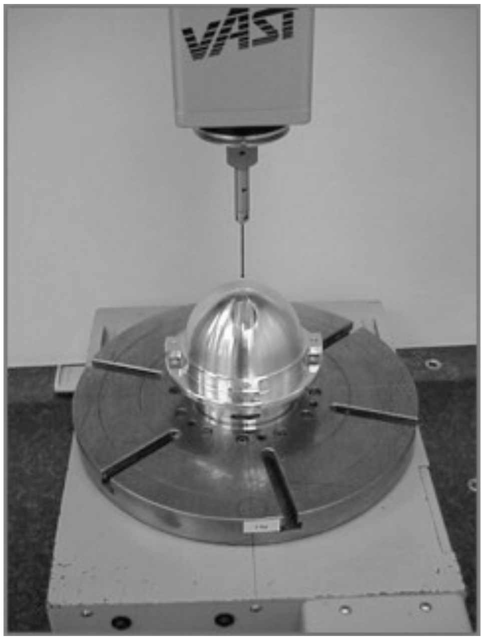
4.4 Resulting Archive Information Package

The previous example focused solely on the package required for capturing the inspection workflow for the AMBER part. With respect to the dataset explored in the AMBER case-study, a reasonable Archive Information Package for the full model would include of:

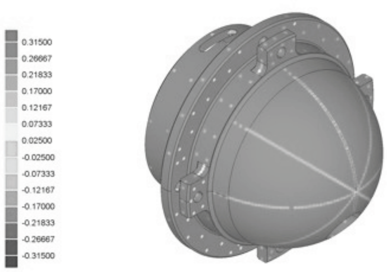
- Workflows for Inspection, Analysis and Simulation, Manufacturing and Process Planning.
- 12 3D models, encompassing native CAD as well as different representations (mesh, STEP) for each model used at the origin of each of the workflows.
- Many ancillary documents, including files for simulation set ups, describing the process plans, the DMIS code for the inspection, etc.
- Several derivative 3D models, including those that represent the "in process" manufacturing shape (i.e., the material removal sequences in the process plan), adjacent parts in the simulation assembly, etc.

Some observations need to be noted. There is currently no formal or precise way to specify the necessary and sufficient conditions on a suitable file inventory for an Archive Information Package. The actual list implemented for an AIP will naturally depend on the end purposes of the archive. In the case of the AMBER part, the assumed "use case" was to ensure the object could be fabricated in the future. For this purpose, one can argue that the above list serves this purpose considerably better than a single proprietary CAD file.

5. DISCUSSION



(a) A photo of the inspection process.



(b) Visualization of the data points from the inspection.

Figure 8: The inspection process and results.

Long term preservation of design data introduces many new challenges to existing work on geometric representation. Some of these are procedural concerns to address in practice, such as monitoring for media degradation and technological obsolescence. Others require application and continued development of current research. As an opener, renewed focus must be placed on capturing a complete dataset to describe a model. Geometry, analysis meshes, simulation code, requirements, functional models, etc must all be archived as any data successfully archived may rapidly become meaningless outside its original context. Long-term archiving also increases the pool of geometry-centric data which must be captured. For example, in addition to any generated data, the software tools used to create that data, and any operating context or parameters, must also be preserved. Archiving of software raises its own concerns at least as complex as those of the design of CAD systems themselves.

One consideration unique to archiving is that in the absence of perfect solutions to capture all aspects of design, it may be better to "simply grab everything." This may require substantial "digital archeology" efforts to utilize such unstructured, unspecified data. However, it is certainly better than preserving no data and such efforts may be worth it in the future for significant artifacts. This also addresses the consideration that it is hard to predict precisely how data will be used over the long term. Such a wide range of data would need to be captured and packaged in some manageable form. Standards such as ISO 10303 STEP will clearly play a large role in this. However, there are many types of engineering data for which no standards exist. As is illustrated by the AMBER case study, standards are needed that can capture the complete web of model interactions and workflows.

Archive management for engineering objects can be viewed as an extension of existing data migration and product lifecycle management processes. Periodic maintenance of the preserved engineering objects is necessary, including migration of file formats into newer software versions and verification of the integrity of the archived models to ensure stability and data integrity over time.

More problematic, however, are the extremely long lifetimes of engineered artifacts that introduce challenges beyond simple software integration. Generational changes in the process of science and engineering, or even the use of language, may make preserved data difficult or impossible to interpret. For example, the Boeing Corporation's B-52 Stratofortress was first deployed in 1955 and planned to be in operation until 2040. This aircraft's lifetime extends not just over changes in CAD software packages, but across the development of CAD technology and the entire computing and software industry itself. In order to maintain the future interpretability of digital models for current generation of airframes one will have to record substantial information about engineering context, scientific and engineering language, the state-of-practice and the business processes assumed during the creation of the artifact. This problem is further compounded for civil infrastructure projects where the expected lifecycle (i.e., for a building, bridge, dam or archaeological site) might be reasonably measured in centuries.

There are also problems related to the general nature of computational representations of geometry. Generative geometry/topology knowledge poses unique representation and preservation challenges [8, 9]. For real-world artifacts (i.e., a 3D mechanical part, building layout, animated entity) the generative knowledge is captured in the geometry/topology authoring software as a list of editing operations (i.e., feature tree, constraints, etc). Not only is the specific structure of this tree not unique, as there are many ways to arrive at the same goal; but it is also highly dependent on the skills of the human operator working with the software system. As a result, not all features or edits are equally meaningful and the capture or filtering of raw "design histories" to determine which are most meaningful poses a significant challenge in preservation and design rationale capture. In this case, the needs of preservation go beyond that of feature-based data exchange, in which merely re-mapping of the features is adequate for other to edit the object–but not enough for enabling understanding of the design's semantics.

Another open issue for long-term preservation is that of consistency maintenance and management over time and across different underlying geometry/topology representations [29]. This issue can be looked at in two ways, one is as a matter of provenance (i.e., the labels on the data), the other is more model-specific (i.e., the data itself). Representation of provenance can be used to track the pedigree and history of data objects in a preservation system as well as track associations among related file-centric objects. In the context of CAD and engineering objects, this means having the complete chain, back to the original file, of transformations and translations of this object into derivative objects. Tracking provenance would not require claims about the underlying consistency of these representations themselves, as this would necessitate an agreed upon methodology for directly comparing the underlying representations. In the context of CAD and solid models, this would mean some kind of formal definition for "consistency" and then algorithmic methods for checking "consistency" among the data files themselves. For example, checking "consistency" between a native Pro/E PRT file and that of a STEP AP 203 file produced by Pro/E's translator would require deciding of "consistency" meant assessing geometry/topology, feature history, shape, or other model properties. Currently, while there has been considerable work in recent years in shape similarity assessment, clearly "consistency" of shape alone is not sufficient to ensure that underlying data models are equivalent. Recent work in modeler-independent representations may provide useful tools for addressing this problem, as would be formal methods for describing and recording equivalences among underlying representations, i.e., between analytical geometry, meshes, features, etc.

6. CONCLUSIONS

This paper introduced the problem of long-term archival of geometry-centric engineering data. In all domains in which Computer-Aided Design has become an essential tool, geometry-centric artifacts are the common element in most, if not all, engineering activities. Long-term preservation of engineering knowledge requires we understand the role of these geometry-centric representations in uniting different engineering activities and how these relationships can be used to build frameworks for capturing essential metadata and process knowledge. The position of the authors of this paper is that the central challenges required the detailed attention and research effort of those that study geometry, physics-based and other model representations and their associated algorithms. The representational challenges presented by the need to steward geometry-centric data over decades and centuries can be key drivers of innovation, much in the same way the needs of CAD, visualization and manufacturing were the drivers of advances in the early days of graphics.

The basic approach advanced in this paper are (1) to insure future interpretability, one can encode models with multiple redundant representations, providing a form of forward error correction that can enable future users the ability to interpret the shape data; (2) that open Semantic Web standards can be used to represent vital metadata associated with the shape descriptions or that enables interpretation of the shape descriptions; and (3) that organizational process and workflow knowledge is essential to understand the transformation of data that occurs in the course of the development of a product, model or other artifact.

These challenges are certainly not going to disappear. As geometry-centric representations and our tools and algorithms become more sophisticated, increasingly import aspects of our physical world will be captured and modeled with them. The preservation and long-term stewardship problem will persist, increasing in complexity with each software and mathematical improvement. . . but also increasing in importance as the digital media eclipses print and other physical media as the object "of record."

Acknowledgments. 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 NSF Cyber-Infrastructure Teams (CI-TEAMs) Program grants CISE/SCI-0537370 and CISE/SCI-0537125. 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. The authors would like to thank the members of the technical team at the United States Department of Energy Kansas City Plant (KCP) and the UK Ministry of Defence for providing us with the AMBER dataset; as well as the technical reviewers for their useful comments.

7. REFERENCES

[1] Project LOTAR: Long term archiving and retrieval of product data within the areospace industry (lotar): Technical aspects of an approach for application. Technical report, ProSTEP Association, August 2002.

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

[3] Aim At Shape. http://shapes.aim-at-shape.net/index.php. On the WWW, 2007.

[4] T. Ashino and M. Fujita. Definition of a web ontology for design-oriented material selection. *Data Science Journal*, 5:52–63, 2006.

[5] A. Ball. Briefing paper: The OAIS reference model. Technical report, UKOLN, University of Bath, February 2006.

[6] A. Ball and L. Ding. Proceedings of the atlantic workshop on long term knowledge retention. Technical report, The University of Bath, 2007.

[7] E. J. Barkmeyer, T. H. Hopp, M. J. Pratt, and G. R. Rinaudot, editors. *Systems Integration for Manufacturing Applications: Background Study*. National Institute of Standards and Technology, Gaithersburg, MD 20899, September 1995 1995. NISTIR 5662.

[8] J. Chen, M. Freytag, and V. Shapiro. Shape sensitivity of constructive representations. In *SPM '07: Proceedings of the 2007 ACM symposium on Solid and physical modeling*, pages 85–95, New York, NY, USA, 2007. ACM.

[9] J. Chen, M. Freytag, and V. Shapiro. Shape sensitivity of constructively represented geometric models. *Comput. Aided Geom. Des.*, 25(7):470–488, 2008.

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

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

[12] C. Eastman, P. Teicholz, R. Sacks, and K. Liston. *BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors*. Wiley, Englewood Cliffs, NJ, 2008. ISBN 978-0470185285.

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

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

[15] H. M. Gladney. Trustworthy 100-year digital objects: Evidence after every witness is dead. *ACM Trans. Inf. Syst.*,

22(3):406–436, 2004.

[16] H. M. Gladney and R. A. Lorie. Trustworthy 100-year digital objects: durable encoding for when it's too late to ask. *ACM Trans. Inf. Syst.*, 23(3):299–324, 2005.

[17] M. Grüninger and C. Menzel. The process specification language (psl) theory and applications. *AI Mag.*, 24(3):63–74, 2003.

[18] T. C. Kelly, R. Wolf, P. Kennicott, and R. N. Nagel. A technical briefing on the initial graphics exchange specifications (IGES). Technical report, National Bureau of Standards, N.B.S. Report NBSIR 8-2297., July 1981.

[19] J. Kim, M. Spraragen, and Y. Gil. An intelligent assistant for interactive workflow composition. In *IUI '04: Proceedings of the 9th international conference on Intelligent user interface*, pages 125–131, New York, NY, USA, 2004. ACM Press.

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

[21] S. Lemaignan, A. Siadat, J.-Y. Dantan, and A. Semenenko. Mason: A proposal for an ontology of manufacturing domain. pages 195–200, June 2006.

[22] Library of Congress. Sustainability of Digital Formats: Planning for Library of Congress Collections. On the WWW, November 2006.

[23] M. F. Lin, J. and T. Bilgic. A Requirement Ontology for Engineering Design. *Concurrent Engineering*, 4(3):279–291, 1996.

[24] R. Lorie. A methodology and system for preserving digital data. *Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries*, pages 312–319, 2002.

[25] J. Lubell, S. Rachuri, E. Subrahmanian, and W. Regli. Long term knowledge retention workshop. Technical report, National Institute of Standards and Technology, NISTIR 7386, 2006.

[26] R. McClatchey, Z. Kovacs, F. Estrella, J.-M. Le Goff, G. Chevenier, N. Baker, S. Lieunard, S. Murray, T. Le Flour, and A. Bazan. The integration of product data and workflow management systems in a large scale engineering database application. pages 296–302, Jul 1998.

[27] N. Noy and D. McGuinness. Ontology Development 101: A Guide to Creating Your First Ontology. Technical report KSL-01-05, Stanford Knowledge Systems Laboratory, March 2001.

[28] M. J. Pratt and J. Kim. Experience in the exchange of procedural shape models using iso 10303 (step). In *SPM '06: Proceedings of the 2006 ACM symposium on Solid and physical modeling*, pages 229–238, New York, NY, USA, 2006. ACM.

[29] S. Raghothama and V. Shapiro. Consistent updates in dual representation systems. In *SMA '99: Proceedings of the fifth ACM symposium on Solid modeling and applications*, pages 65–75, New York, NY, USA, 1999. ACM.

[30] S. Ram and J. Liu. Understanding the Semantics of Data Provenance to Support Active Conceptual Modeling. Springer, 2006.

[31] A. Rappoport. An architecture for universal cad data exchange. In *SM '03: Proceedings of the eighth ACM symposium on Solid modeling and applications*, pages 266–269, New York, NY, USA, 2003. ACM.

[32] A. Rappoport, S. Spitz, and M. Etzion. One-dimensional selections for feature-based data exchange. In *SPM '05: Proceedings of the 2005 ACM symposium on Solid and physical modeling*, pages 125–134, New York, NY, USA, 2005. ACM.

[33] W. C. Regli, X. Hu, M. Atwood, and W. Sun. A survey of design rationale systems: Approaches, representation, capture and retrieval. *Engineering with Computers*, 16:209–235, 2001.

[34] S. Ross and A. Gow. Digital Archaeology: Rescuing Neglected and Damaged Data Resources. *Library Information and Technology Centre, London, available at: www. ukoln. ac. uk/services/elib/papers/supporting/# blric*, 1999.

[35] S. Spitz and A. Rappoport. Integrated feature-based and geometric cad data exchange. In *SM '04: Proceedings of the ninth ACM symposium on Solid modeling and applications*, pages 183–190, Aire-la-Ville, Switzerland, Switzerland, 2004. Eurographics Association.

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

[37] The National Archives. Pronom. http://www.nationalarchives.gov.uk/pronom/, February 2007.

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

[39] H. A. van der Meiden and W. F. Bronsvoort. Tracking topological changes in feature models. In *SPM '07: Proceedings of the 2007 ACM symposium on Solid and physical modeling*, pages 341–346, New York, NY, USA, 2007. ACM.