A Framework for Understanding File Format Conversions

Peter Bajcsy, Rob Kooper, Luigi Marini, Kenton McHenry and Michal Ondrejcek
National Center for Supercomputing Applications (NCSA)
University of Illinois at Urbana-Champaign (UIUC)
Telephone Number: 217-265-5387, pbajcsy@ncsa.uiuc.edu

ABSTRACT

This paper addresses the workshop question: “Can data generated from the infancy of the digital age be ingestible by software today?” We have prototyped a set of e-services that serve as a framework for understanding content preservation, automation and computational requirements on preservation of electronic records. The framework consists of e-services for (a) finding file format conversion software, (b) executing file format conversions using available software, and (c) evaluating information loss across conversions. While the target audience for the technology is the US National Archives, these basic e-services are of interest to any manager of electronic records and to all citizens trying to keep their files current with the rapidly changing information technology. The novelty of the framework is in organizing the information about file format conversions, providing services about file format conversion paths, in prototyping a general architecture for reusing existing third-party software with import/export capabilities, and in evaluating information loss due to file format conversions. The impact of these e-services is in the widely accessible conversion software registry (CSR), conversion engine (Polyglot) and comparison engine (Versus) which can increase the productivity of the digital preservation community and other users of digital files.

Categories and Subject Descriptors

H.4 INFORMATION SYSTEMS: H.4.1. Office Automation, H.4.2. Types of Systems,

General Terms: Algorithms, Performance, Design

Keywords: File format conversions, Information loss evaluations

1. INTRODUCTION

In the current digital era, two basic questions are posed by the NIST organized workshop on roadmap development for Digital Preservation Interoperability Framework: “Can data generated from the infancy of digital age be ingestible by software today?” and “Will digital content created today be accessible and renderable throughout its lifecycle?”

These questions lead directly to the problem of file format conversions addressed in this paper. The file format conversion problem is motivated by a very large number of file formats in which digital content is stored, and by an increasing number of complex file formats containing multiple types of digital content (e.g., Adobe PDF, HDF) or having very elaborate specifications (e.g., STEP). However, there exists many software applications that support subsets of import and export operations to and from various file. It is well known that many file formats and software applications are ephemeral in the context of long term preservation. We have collected anecdotal evidence of this state for 3D file formats in [1] and found more than 140 different 3D file formats among 16 popular software packages.

In this work, our main objective is to design services using a computational cloud that would enable optimal and/or measurable data transformation from one data structure to another. The challenges of this task lie in (a) accessibility of services (file format conversions are inevitably one part of our daily life), (b) conversion software quality (software quality of file format conversions is unknown), (c) computational scalability (the volume of file format conversions and its corresponding computational resources), (d) increasing complexity of file formats (the complexity of file formats complicates our understanding of information loss due to file format conversions) and (e) the constraints imposed such as minimal information loss versus minimal financial costs.

The target audiences for such services include managers of electronic records, scientists conducting research with digital data, and citizens trying to keep their files current with the rapidly changing information technology. These communities of users have very heterogeneous requirements on file formats (content representation depends on file format) and the criteria defining information loss due to conversion. They also balance their budgets with the requirements on software quality (recovery and storage of content in a file format depends on the quality of software involved) and available hardware resources (software execution depends on access to storage media, operating system, and hardware platform).

Our work focuses on designing and prototyping the following services in order to assist the above communities:

(a) Find file format conversion software to convert from one file format to another file format

(b) Execute file format conversions with available third party software

(c) Evaluate information loss due to file format conversion over a set of files

Our approach to the technical part of the problem is designing (1) a conversion software registry (CSR) for aggregating the information about software conversion functionalities, (2) a scalable conversion software engine for executing file format conversions (called Polyglot), and (3) a scalable file comparison engine for measuring changes in content between the files before and after conversion. The overview of the services is illustrated in *Figure 1*.

The novelty of the work is in exploring the concept of ultimate conversion engines, conversion introduced information loss, comparison metrics, and scalability of conversions and comparisons of digital objects. The prototype services provide capabilities to improve efficiency with which archives would manage their holdings, and lead to better understanding of current and future preservation and reconstruction of electronic records via design and experimental evaluations of novel appraisal methodologies.

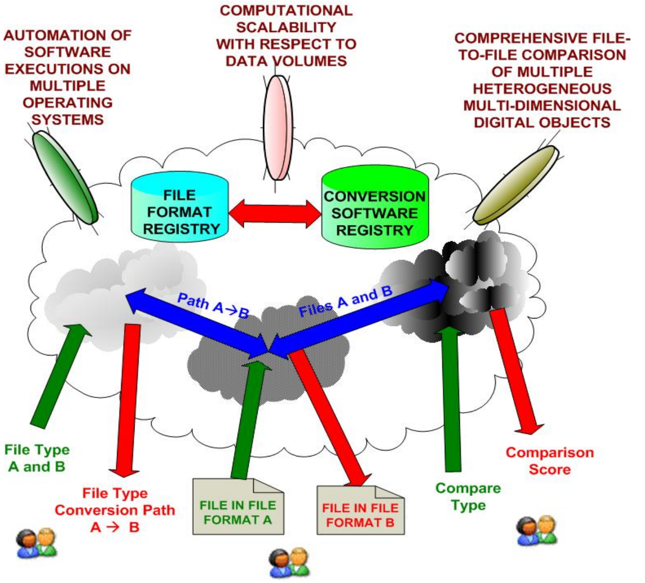

AUTOMATION OF SOFTWARE EXECUTIONS ON MULTIPLE OPERATING SYSTEMS
COMPUTATIONAL SCALABILITY WITH RESPECT TO DATA VOLUMES
COMPREHENSIVE FILE-TO-FILE COMPARISON OF MULTIPLE HETEROGENEOUS MULTI-DIMENSIONAL DIGITAL OBJECTS
FILE FORMAT REGISTRY
CONVERSION SOFTWARE REGISTRY
Path A→B
Files A and B
File Type A and B
File Type Conversion Path A → B
FILE IN FILE FORMAT A
FILE IN FILE FORMAT B
Compare Type
Comparison Score

Figure 1: An overview of the services and functionalities. The clouds represent basic operations of (a) finding conversion path (left cloud), (b) converting a file in format A to a file in format B, and (c) comparing files A and B and returning the comparison score. The green arrows refer to inputs for each operation and the red arrows show outputs. The blue arrows denote the interoperability of these services executed in computer clouds by sharing information about the Path A→B and the files to compare. The system can be viewed as a part of the preservation infrastructure or as a part of basic file management system.

2. CONVERSION SOFTWARE REGISTRY

2.1 Background

While there is a need to document all past, current and future file format conversion functionalities, there does not exist a common complex conversion software registry today according to our knowledge. There exist web-based solutions that given a file extension would return the information about the primary software capable of loading a file format defined by the extension. Two such databases of extensions are FILExt (http://www.filext.com) and Whatis?com (http://whatis.techtarget.com). However, none appear to provide the information about input and output file formats supported by applications (the information required for file format conversions). In addition to extensions, Wotsit.org (www.wotsit.org) provides not only extensions, name, and primary software description but also a sample file or link to a sample file.

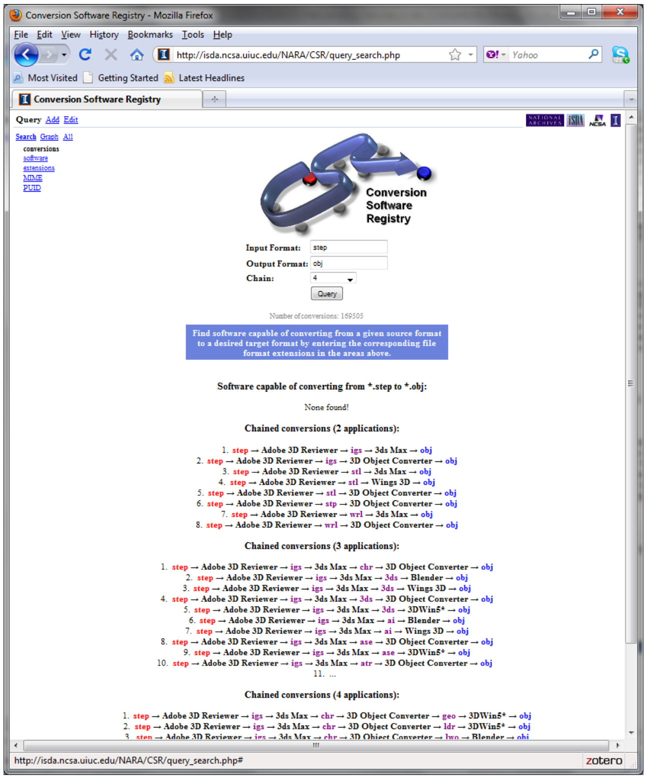
There have been efforts to catalogue software with various functionality of interest to a specific scientific community. For example, we are aware of the Geotechnical and Geoenvironmental Software Directory (GGSD) at http://www.ggsd.com/ggsd/index.cfm (1996-2006) and the Natural Language Software Registry (NLSR) at http://registry.dfki.de/. In the business world, software catalogues have been created for on-line shopping (e.g., Cnet at http://download.cnet.com/windows/) or for monitoring the use of software on intranets (e.g., the Bit9 Global Software Registry at http://www.bit9.com/products/gsr.php creating a white list of software). These software catalogues focus on general descriptions of software and do not provide information about file format conversion capabilities.

Over the past several years, the European Union has funded the Planets test bed project http://testbed.planets-project.eu/testbed/ that should provide a “controlled environment in which users can experience and compare different preservation tools and approaches through their Web browser.” Conceptually, the Planets project has similar goals as our proposed effort. However, the design of services is very different, the number of file format conversion software packages documented is relatively small[1] (about 18 as of 2010-03-25), the conversion service (also referred as the migration service) supports only one-hop conversion paths, and the extensibility and computational scalability have not been addressed One-hop conversion path refers to one file format conversion only.

2.2 Solution

Our approach to designing a conversion software registry (CSR) leverages from the past work on file format registries. We have followed the data models used in PRONOM and GDFR/UDFR to build a system available at https://isda.ncsa.uiuc.edu/NARA/CSR/. The current CSR system provides support for searching, editing and adding information about file format conversion software, as well as open access for queries and login-based access for modifications. The web interface to the prototype system is illustrated in *Figure 2*. While working with CSR one will observe the extreme difficulties in finding file format conversion software paths without the assistance of systems like CSR. The current version of CSR contains information about 72 software packages that lead to 169,505 possible one-hop conversion paths.

[1] https://testbed.planets-project.eu/testbed/public/browse_pathways.faces


Conversion Software Registry - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://isda.ncsa.uiuc.edu/NARA/CSR/query_search.php
Yahoo
Most Visited
Getting Started
Latest Headlines
Conversion Software Registry
Query Add Edit
Search Graph All
conversions
software
extensions
MIME
PUID
Conversion
Software
Registry
Query
http://isda.ncsa.uiuc.edu/NARA/CSR/query_search.php#
zotero

Input Format: step
Output Format: obj
Chain: 4

Number of conversions: 169505

Find software capable of converting from a given source format to a desired target format by entering the corresponding file format extensions in the areas above.

Software capable of converting from *.step to *.obj:

None found!

Chained conversions (2 applications):

1. step → Adobe 3D Reviewer → igs → 3ds Max → obj
2. step → Adobe 3D Reviewer → igs → 3D Object Converter → obj
3. step → Adobe 3D Reviewer → stl → 3ds Max → obj
4. step → Adobe 3D Reviewer → stl → Wings 3D → obj
5. step → Adobe 3D Reviewer → stl → 3D Object Converter → obj
6. step → Adobe 3D Reviewer → stp → 3D Object Converter → obj
7. step → Adobe 3D Reviewer → wrl → 3ds Max → obj
8. step → Adobe 3D Reviewer → wrl → 3D Object Converter → obj

Chained conversions (3 applications):

1. step → Adobe 3D Reviewer → igs → 3ds Max → chr → 3D Object Converter → obj
2. step → Adobe 3D Reviewer → igs → 3ds Max → 3ds → Blender → obj
3. step → Adobe 3D Reviewer → igs → 3ds Max → 3ds → Wings 3D → obj
4. step → Adobe 3D Reviewer → igs → 3ds Max → 3ds → 3D Object Converter → obj
5. step → Adobe 3D Reviewer → igs → 3ds Max → 3ds → 3DWin5* → obj
6. step → Adobe 3D Reviewer → igs → 3ds Max → ai → Blender → obj
7. step → Adobe 3D Reviewer → igs → 3ds Max → ai → Wings 3D → obj
8. step → Adobe 3D Reviewer → igs → 3ds Max → ase → 3D Object Converter → obj
9. step → Adobe 3D Reviewer → igs → 3ds Max → ase → 3DWin5* → obj
10. step → Adobe 3D Reviewer → igs → 3ds Max → atr → 3D Object Converter → obj
11. ...

Chained conversions (4 applications):

1. step → Adobe 3D Reviewer → igs → 3ds Max → chr → 3D Object Converter → geo → 3DWin5* → obj
2. step → Adobe 3D Reviewer → igs → 3ds Max → chr → 3D Object Converter → ldr → 3DWin5* → obj
3. step → Adobe 3D Reviewer → igs → 3ds Max → chr → 3D Object Converter → lwo → Blender → obj

*Figure 2: Conversion software registry with search results for converting step file format to *.obj file format.*

3. FILE FORMAT CONVERSION

3.1 Background

We are aware of file format conversion engines that are constrained to one data type and in-house software base, as listed in our previous report [1]. For example, FileFormat.info (http://www.fileformat.info) includes file format conversion tools for images only based on Java Advanced Imaging libraries (javax.imageio.* and javax.media.jai.*). There exist a few file format conversion services that support only certain conversion types (e.g., http://www.ps2pdf.com – 1 conversion type, http://media-convert.com - about 20 multi-media formats; http://www.zamzar.com – selected conversions of document, image, music, video and couple of CAD formats). The main drawback of the existing conversion systems is that they are not extensible (limited by the availability of specific libraries).

In order to design an extensible file format conversion system based on utilizing third party software several problems have to be addressed. First, the problem of automated execution of the software, most GUI based, without having access to an application programming interface (API). We are aware of AutoHotKey[2] scripting as a viable option for the Windows operating system (OS) and we have based our current Polyglot implementation on it. Second, the problem of distributed computational resources has been approached in the past by the Grid community (TeraGrid[3] and Globus Toolkit[4] for building computational grids) and the design of workflow middleware that would manage the execution, such as Cyberintegrator [2], Kepler [3] DAGMan, CCA[5] or Taverna[6] [4]. Due to the heterogeneity of computational hardware, this problem also requires considerations about options for parallel processing, for instance, the use of (1) a message-passing interface (MPI is designed for the coordination of a program running as multiple processes in a distributed memory environment by using passing control messages.), (2) open multi-processing (OpenMP is intended for shared memory machines. It uses a multithreading approach where the master threads fork any number of slave threads.), (3) the map reduce parallel programming paradigm for commodity clusters (which allows programmers write simple Map and Reduce functions, which are then automatically parallelized without requiring the programmers to code the details and communications of parallel processes) and (4) novel architectures (FPGAs, GPUs, multiple CPUs). Unfortunately, none of the existing grid solutions are an option when utilizing 3rd party binaries compiled for specific hardware on one machine. Workflow solutions could potentially orchestrate calling computational resources based on a conversion sequences, however most do not robustly deal with solely GUI based software and we must also consider tasks specific needs, such as clustering the conversion execution sequence into segments that do not require data movement, and then managing and monitoring entire conversion executions.

3.2 Solution

Our approach to designing file format conversion services is based on an idea of 'Imposed Software Reuse'. We define imposed software reuse as the wrapping of 3rd party software, utilizing whatever published application programming interface (API), command line interface or graphics user interface (GUI) the software vendors make available to access the embedded functionalities. This approach is selected because of the simple fact that fully supporting the many available formats is such an enormous undertaking which is made all the more difficult when you consider that many formats are closed/proprietary and thus difficult to extract data from, and vendor file formats sometimes store application specific features. Although there are many commercial solutions that provide subsets of conversion capabilities based on available file loading libraries (e.g. ImageMagick, ps2pdf, Zamzar, Google multi-media converters for YouTube posting, etc…), none appear to utilize the imposed software reuse philosophy that would allow the reuse any software for conversion purposes.

The key aspects of our solution are in (a) automated execution of third party software using workflows of scripts (AutoHotKey, AppleScript, and others), (b) distributed execution over a set of software/hardware resources with distributed software licenses, hardware access restrictions, and temporally varying

[2] http://www.autohotkey.com

[3] https://www.teragrid.org

[4] http://www.globus.org/toolkit

[5] http://www.cca-forum.org

[6] http://taverna.sourceforge.net

computational loads, and (c) computationally scalable execution of large volumes of file format conversions with additional hardware resources (shared folder approach, layered approach to improve robustness and scalability). All three aspects are shown in the Polyglot overview in Figure 3 and described in a detail in [11]. The web interface to Polyglot is illustrated in Figure 4. The service is available at http://teeve3.ncsa.uiuc.edu/polyglot/convert.php.

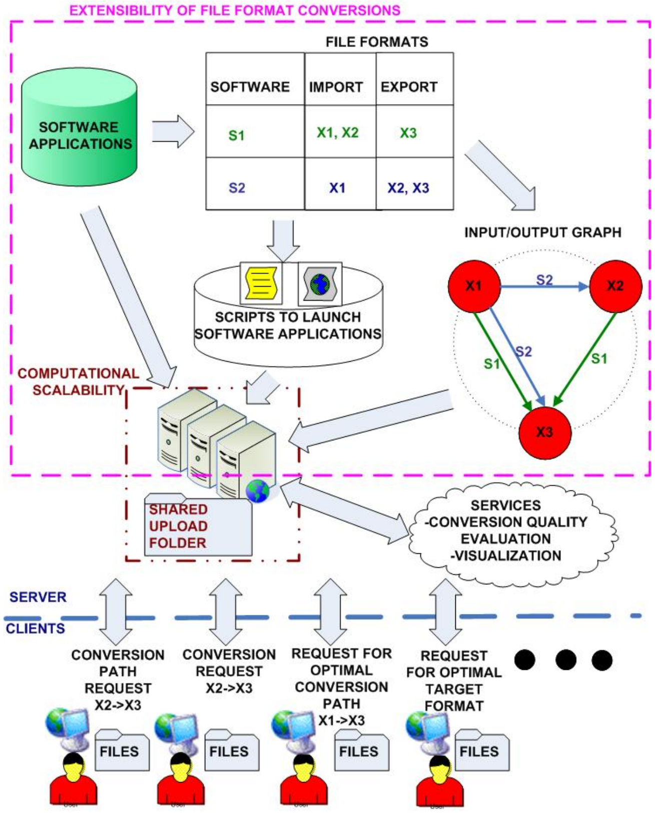
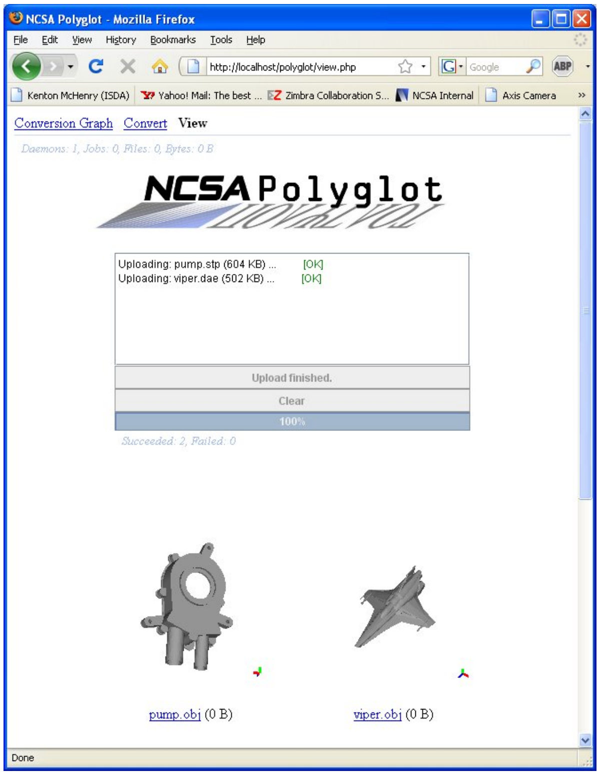

EXTENSIBILITY OF FILE FORMAT CONVERSIONS
FILE FORMATS
SOFTWARE
IMPORT
EXPORT
SOFTWARE APPLICATIONS
S1
X1, X2
X3
S2
X1
X2, X3
INPUT/OUTPUT GRAPH
SCRIPTS TO LAUNCH SOFTWARE APPLICATIONS
X1
S2
X2
S2
S1
S1
X3
COMPUTATIONAL SCALABILITY
SHARED UPLOAD FOLDER
SERVICES
-CONVERSION QUALITY EVALUATION
-VISUALIZATION
SERVER
CLIENTS
CONVERSION PATH REQUEST X2->X3
CONVERSION REQUEST X2->X3
REQUEST FOR OPTIMAL CONVERSION PATH X1->X3
REQUEST FOR OPTIMAL TARGET FORMAT
FILES
FILES
FILES
FILES

Figure 3: An overview of the Polyglot conversion system. Third party applications are documented and scripted so as to be used as modules within the overall system. A web interface provides one method of user access for file conversion and visualization.

4. FILE COMPARISON

4.1 Background

A file format conversion can lead to a change of information organization due to format specifications, change of information value due to data representation, or a change in information accuracy due to the software involved. Furthermore, file formal conversion is applied to all types of digital objects contained in a file, for example, text, images, vector graphics, 3D objects, and animations (2D video) in the case of document formats. Thus, content-based file comparison has to encompass a wide spectrum of digital object types and be invariant to some content changes that preserve the information while being sensitive to other content changes that drop relevant information during conversions. The problem of text based file comparison leads to the study of natural language processing and a large number of publications and prototyped methods [5]. Similarly, the problem of X based file comparison where X is image, vector graphics, 2D video, and 3D objects, has been addressed in the abundance of past work on content based image retrieval ([6] and [7]), video retrieval (e.g., YouTube), or 3D shape retrieval [8]. According to all references consulted, the primary challenge in content based retrieval systems is in overcoming the so-called *semantic gap*, which is the gap between low-level features of, for instance, an image and its abstract meaning to a human viewer. This gap has not been adequately bridged by current systems [9]. From Smeulders et al., it is apparent that much early work on content based image retrieval focused on finding appropriate similarity metrics between image features but there has not been a single comparison method and metric that would outperform all other methods and metrics. Further, there has not been an exploratory framework that would deliver a comprehensive comparison of file content with multiple types of digital objects and would scale with an increasing volume of file comparisons.



Daemons: 1, Jobs: 0, Files: 0, Bytes: 0 B

```
Uploading: pump.stp (604 KB) ...     [OK]
Uploading: viper.dae (502 KB) ...    [OK]
```

Succeeded: 2, Failed: 0

pump.obj (0 B)   viper.obj (0 B)

Figure 4: The web interface to Polyglot's universal converter and viewer. Users drag files to the top area. In the View mode, there is no choice for the output format as it is automatically set to the type supported for viewing by the web interface (.obj for 3D). When the files are converted they are displayed in the area below where the user can then rotate and zoom in on the objects.*

4.2 Solution

Our approach to designing file comparison services is based on our previous prototypes focusing on comparing complex file formats such as Adobe PDF (called Doc2Learn [12]), 3D file formats (called ModelBrowser), and on the experience of other projects, such as the Planets test bed project http://testbed.planets-project.eu/testbed/.

Figure 5 shows the overview of file comparisons in Doc2Learn. The comparison process starts with extraction of individual digital objects from complex files. In this case, the extraction includes text, images and vector graphics. The comparison method is based on comparing frequencies of occurrence of words, image pixels and vector graphics primitives. Figure 6 shows the user interface to browsing frequency of occurrence features in Doc2Learn. The comparison computations are implemented using the Map and Reduce paradigm which provides computational scalability on cluster computers.

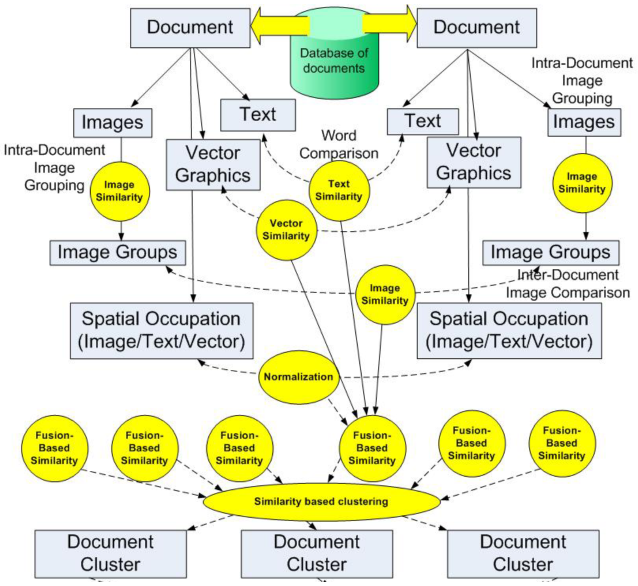


Figure 5: Overview of Doc2Learn for comparing complex files in Adobe PDF formats. A pair-wise comparison consists of comparing individual digital elements first, and then assigning a final similarity score based on weighted combination of individual similarity scores.

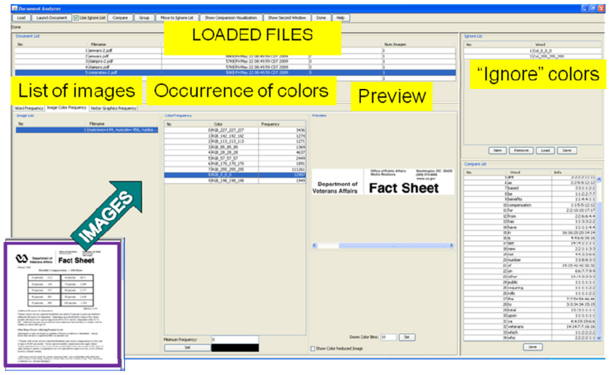


Figure 6. User interface to Doc2Learn. This view shows all information about images contained in a PDF file.

Figure 7 illustrates a comparison of the 3D model of heart in three different file formats using the ModelBrowser application within our prototype library of 3D utilities. This example shows how information loss due to file format conversion could be analyzed. In both, Doc2Learn and ModelBrowser, the users are not able to choose criteria defining information loss but can benefit from visual browsing of data files and from analyzing the resulting similarity/dissimilarity scores. We have been designing a next generation system called Versus which is extensible in terms of comparison methods and would allow users to choose a comparison method. As shown in Figure 8, the comparison method is decomposed into a choice of data representation, features/signature characterizing a digital object, and a similarity measure (a triplet defining information loss measurement). Using this decomposition, a user could map the application specific definition of information loss to the triplet and perform analyses of file format conversions in accordance with institutional perspectives on information loss.

With the support for content based file-to-file comparison, one can utilize the similarity/dissimilarity scores for evaluating average information loss per file format conversion executed by software. The information loss measurements are stored in the CSR database and can be used for optimal selection of complex file format conversion paths in order to minimize the information loss and license costs.

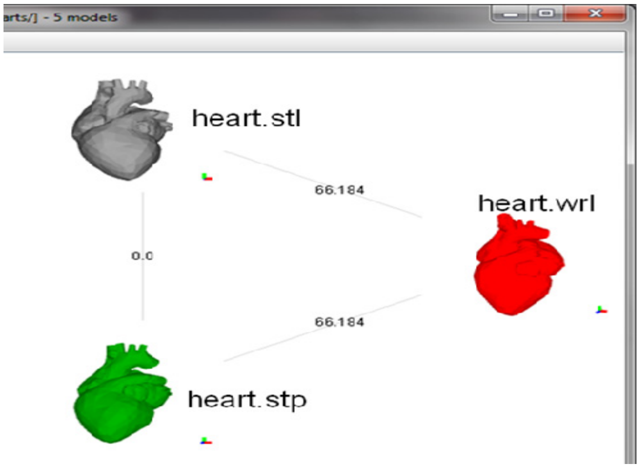
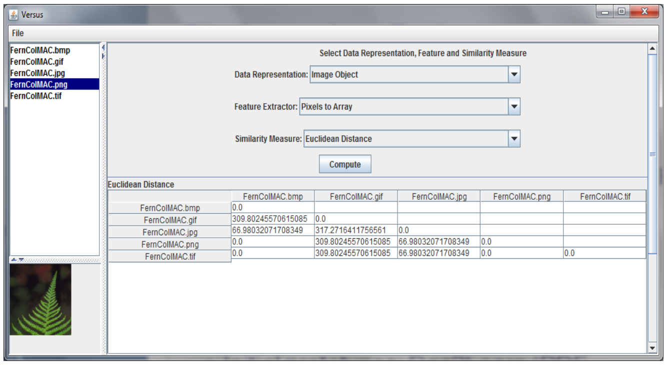


Figure 7: Visualization of pair-wise comparison using ModelBrowser tool. The original file heart.wrl was converted using Adobe 3D Reviewer to STP and STL file formats and all three files were compared. The comparison is based on the Light Fields method [10] which compares silhouettes from various viewing angles around the objects. The results show that there is information loss introduced and the converted files are the same with respect to the Light Fields method.

5. CONCLUSION

We have presented three types of services for digital preservation communities. These services provide a way to better understand preservation and reconstruction of electronic records in terms of file format conversions. The benefits of such systems lie in answering questions such as: what is the infrastructure needed for documenting existing file format conversion software, what is the framework for imposing code reuse on closed 3rd party software, how to measure information loss due to file format conversions, what is the computational cost of file format conversions, file comparisons, and information loss evaluations, and how to achieve computational scalability of file format conversions?

The prototype services are freely available to digital preservation community and serve as a framework for decisions related to (a) selection of an 'optimal' file format to be preserved (the target file format), (b) evaluation of file format conversion software, (c) selection of minimum cost for a chosen file format conversion path (quality cost, license cost, hardware cost). It is our hope that in the future these prototype services will become a foundation of the infrastructure needed to manage and preserve the increasing amount of digital information.



FernColMAC.bmp
FernColMAC.gif
FernColMAC.jpg
FernColMAC.png
FernColMAC.tif

Select Data Representation, Feature and Similarity Measure

Data Representation: Image Object

Feature Extractor: Pixels to Array

Similarity Measure: Euclidean Distance

Euclidean Distance

	FernColMAC.bmp	FernColMAC.gif	FernColMAC.jpg	FernColMAC.png	FernColMAC.tif
FernColMAC.bmp	0.0				
FernColMAC.gif	309.80245570615085	0.0			
FernColMAC.jpg	66.98032071708349	317.2716411756561	0.0		
FernColMAC.png	0.0	309.80245570615085	66.98032071708349	0.0	
FernColMAC.tif	0.0	309.80245570615085	66.98032071708349	0.0	0.0

Figure 8: An example of the front end to the generic file comparison application programming interface (API) called Versus where a user can choose a comparison method. In this example, a user chose to evaluate four files by defining information loss as a sum of Euclidean distances over all pixels represented by Image Object. The images were generated by MS Paint software from the original TIF file. The numerical results show the information loss when converting to GIF and JPEG and no-loss when converting to PNG and BMP.

6. ACKNOWLEDGMENTS

This research was partially supported by a National Archive and Records Administration (NARA) supplement to NSF PACI cooperative agreement CA #SCI-9619019 and NCSA Industrial Partners. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the National Archive and Records Administration, or the U.S. government.

7. REFERENCES

[1] K. McHenry, and P. Bajcsy, An Overview of 3D Data Content, File Formats and Viewers., NCSA, Urbana-Champaign, IL, 2008.

[2] R. Kooper, et al., "CyberIntegrator: A Highly Interactive Scientific Process Management Environment to Support Earth Observatories," Proc. Geoinformatics conference, 2007.

[3] B. Ludascher, I. Altintas, S. Bowers, J. Cummings, T. Critchlow, E. Deelman, D. D. Roure,J. Freire, C. Goble, M. Jones, S. Klasky, T. McPhillips, N. Podhorszki, C. Silva, I. Taylor, andM. Vouk. Scientific Process Automation and Workflow Management. In A. Shoshani andD. Rotem, editors, Scientific Data Management: Challenges, Existing Technology, and Deployment, Computational Science Series, chapter 13. Chapman & Hall/CRC, 2009.

[4] T. Oinn, et al., "Taverna: Lessons in creating a workflow environment for the life sciences," Concurrency and Computation: Practice and Experience, Grid Workflow Special Issue, vol. 18, no. 10, 2005, pp. 1067-1100.

[5] Christopher D. Manning and Hinrich Schütze, Foundations of Statistical Natural Language Processing, MIT Press. 1999.

[6] Smeulders et al. "Content-Based Image Retrieval at the End of the Early Years", IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000.

[7] Lew et al., "Content-Based Multimedia Information Retrieval: State of the Art and Challenges", ACM Transactions on Multimedia Computing, Communications and Applications, 2006.

[8] Lydia Lei, "Three Dimensional Shape Retrieval using Scale Invariant Feature Transform and Spatial Restrictions," Technical Report NISTIR 7625, National Institute of Standards and Technology, Gaithersburg, MD, August 2009.

[9] Yu-Jin Zhang, "Toward High-Level Visual Information Retrieval", appearing in the book Semantic-Based Visual Information Retrieval, 2007.

[10] D. Chen, X. Tian, Y. Shen, and M. Ouhyoung. On visual similarity based 3d model retrieval. Eurographics Computer Graphics Forum, 2003.

[11] K. McHenry, R. Kooper and P. Bajcsy, "Towards a Universal, Quantifiable, and Scalable File Format Converter," the 5th International IEEE eScience conference, Oxford, UK from Dec 9-11. 2009 (oral presentation), http://www.escience2009.org/

[12] R. Kooper and P. Bajcsy, "Comprehensive Appraisals of Contemporary Documents," Microsoft Research eScience workshop, Pittsburg, PA, October 15-17, 2009 (oral presentation), http://research.microsoft.com/en-us/events/escience2009/.