The International Journal of Digital Curation
Volume 7, Issue 1 | 2012

Assessing Migration Risk for Scientific Data Formats

Chris Frisz, Geoffrey Brown and Samuel Waggoner,
School of Informatics and Computing,
Indiana University

Abstract

The majority of information about science, culture, society, economy and the environment is born digital, yet the underlying technology is subject to rapid obsolescence. One solution to this obsolescence, format migration, is widely practiced and supported by many software packages, yet migration has well known risks. For example, newer formats – even where similar in function – do not generally support all of the features of their predecessors, and, where similar features exist, there may be significant differences of interpretation.

There appears to be a conflict between the wide use of migration and its known risks. In this paper we explore a simple hypothesis – that, where migration paths exist, the majority of data files can be safely migrated leaving only a few that must be handled more carefully – in the context of several scientific data formats that are or were widely used. Our approach is to gather information about potential migration mismatches and, using custom tools, evaluate a large collection of data files for the incidence of these risks. Our results support our initial hypothesis, though with some caveats. Further, we found that writing a tool to identify "risky" format features is considerably easier than writing a migration tool.

The *International Journal of Digital Curation* is an international journal committed to scholarly excellence and dedicated to the advancement of digital curation across a wide range of sectors. The IJDC is published by UKOLN at the University of Bath and is a publication of the Digital Curation Centre. ISSN: 1746-8256. URL: http://www.ijdc.net/

 doi:10.2218/ijdc.v7i1.212

Introduction

Over the past several years increasing pressure has been exerted by funding agencies upon research scientists to share the fundamental data generated by publicly funded research projects (Nature, 2009). One manifestation of this pressure is the new National Science Foundation policy on data sharing.[1] A fundamental issue with data sharing over the long term is the eventual obsolescence of all formats and the consequent need to migrate data to newer formats. The goal of the work described in this paper is to enable high quality, low risk, migration of scientific data from formats that have poor long term viability due to dependencies on legacy software and hardware to more viable formats. Specifically, we examine both the efficacy and development complexity for "risk assessment" tools, whose purpose is to evaluate a collection of files in obsolete formats to determine whether they can be safely migrated to newer formats.

Our work is based upon a simple hypothesis: where migration paths exist, most files can be safely translated leaving a (hopefully small) minority requiring closer inspection. For example, consider that Lotus 1-2-3, the dominant spreadsheet format in the 1980s and 1990s, is no longer supported in Microsoft Excel,[2] the current leader in spreadsheet software. Even when Excel supported migration for Lotus 1-2-3, significant differences, such as formula calculation and supported features, meant that some files could not be faithfully translated.[3] We found in early testing that the majority of Lotus 1-2-3 files utilize no formulas and hence can be translated to modern formats with no risk.

Although the imperative to share data has recently achieved a high degree of public awareness, data sharing is not new. In previous work we have examined the issues of preserving approximately 3,000 CD-ROMs distributed by the United States Government Printing Office over the past 20 years (Woods & Brown, 2009). These CD-ROMs contain many thousands of data files about the economy, the environment, society and physical sciences. In this paper, we discuss the development and analysis of risk assessment tools for key scientific data formats utilized in this collection.

The approach described in this paper was to inventory the file types utilized in this CD-ROM collection and then to select several examples of widely used formats. We then studied the available documentation and migration tools for these formats and developed risk assessment tools to evaluate the presence of known risk factors. The specific formats we chose were: Lotus 1-2-3, CDF and netCDF, and HDF. CDF and netCDF are related but incompatible data formats initially developed by NASA[4]. HDF is also widely used and is interesting because the transition from HDF4 to HDF5 introduced substantial incompatibilities.

---

[1] NSF: Dissemination and Sharing of Research Results: http://www.nsf.gov/bfa/dias/policy/dmp.jsp

[2] Deprecated features for Excel 2007: http://blogs.office.com/b/microsoft-excel/archive/2006/08/24/deprecated-features-for-excel-2007.aspx

[3] Tips for Importing Lotus 1-2-3 Files to Excel: http://support.microsoft.com/kb/q61941/ and the differences between Microsoft Excel and Lotus 1-2-3: http://office.microsoft.com/en-us/excel/HP051997741033.aspx

[4] CDF Frequently Asked Questions: http://cdf.gsfc.nasa.gov/html/FAQ.html

There exist several large, well funded projects that have made impressive claims relating to migration, yet careful analysis of publications and public source code repositories suggests that actual results are more modest. For example, the LOCKSS project has developed a tool that is widely used for bit preservation, and publications suggest that migration can be "built-in", yet the actual migration paths supported are limited to a small set of image formats (Rosenthal et al., 2005). The PLANETS project, a large European consortium co-funded by the European Union, has published extensively about migration. Their work has primarily consisted of limited testing of migration paths (Becker et al., 2009), identification of significant properties for some important formats (Dappert & Farquhar, 2009), development of an XML schema to encode significant properties (Becker et al., 2008a), and some extraction of structural properties from formats (Becker et al., 2008b). Risk assessment is a common preservation planning tool (Stanescu, 2005; Arms et al., 2002; University of Leeds, 2007; Pearson & Webb, 2008). However, in the case of formats it is generally applied at the macro level, i.e. determining that a format has poor long term viability (Hunter & Choudhury, 2005). Macro-level risk assessment, along with some investigation of file format structure, can be found in a number of recent projects (Chou, 2007; Walker & Thoma, 2004; Arms et al., 2002). Structured macro-level tools include the kopal Library for Retrieval and Ingest (developed in partnership with IBM) and the AONS and AONS 2 projects developed by the Australian Partnership for Sustainable Repositories to identify obsolescence risk in document collections (Curtis, 2006; Pearson, 2008). Related strategic factors in developing digital media archives are found in CLIR publications (Ide et al., 2002). The Preserv2 semantically enhanced file format registry relies on high-level profiles to (currently) provide risk assessment for only one format, PDF.[5]

The most directly related prior work on risk assessment (Lawrence et al., 2000) specifically examined Lotus 1-2-3 files as we do. Surprisingly, the single greatest risk identified was the use of floating-point numbers, which do not appear to pose a significant translation risk for most conversion tools. Our work differs significantly in scale, both in terms of the design objectives for our tools and the large data sets we analyze.

Data Formats

As discussed above, we selected scientific data formats for this study from the large collection of United States Government Documents published upon CD-ROMS that we have built over the past four years. In this paper, we examined 2747 CD-ROMs containing 9,657,954 files. The vast majority of files were html, images (gif, tif, jpg) and PDF files. From this collection we selected four file types for further analysis. Lotus 1-2-3 is an example of a proprietary format that was once dominant and has now dwindled to near extinction. CDF and netCDF are related open specification formats that have significantly diverged from their common roots. This divergence has introduced functional mismatch that makes conversion between formats somewhat problematic. HDF is an example of a format where there has been considerable change between versions – HDF5 was a large scale simplification of its predecessor, HDF4, which means there are features of the predecessor that are not supported by the successor. The two other common data formats in this collection were Excel and

[5] Preserv2 Semantically Enhanced File Format Registry: http://p2-registry.ecs.soton.ac.uk/

 doi:10.2218/ijdc.v7i1.212

dBase. We chose not to examine Excel, since it is the currently dominant spreadsheet format and current versions of Excel seem to provide good backwards compatibility. dBase, while proprietary, is a simple table format which is easily converted to other tabular formats.

Throughout this work we have made extensive use of available tools and code libraries. In the case of HDF, CDF, and netCDF there are extensive specifications and open source tools/libraries that are readily available. In the case of Lotus 1-2-3, our primary source of information was a published specification of an early version of the file format (Lotus Books, 1986). We did not have access to specifications for the the later wk4 format. For Lotus, another important source of information was the open source tool Gnumeric, which provides translation functions from Lotus to many other formats. In testing, wk1 files opened without errors in Gnumeric, though code review indicates that there are problems displaying wk4 files because of possibly incorrect formula operation codes.[6]

An important aspect of this work is the identification of migration risks. The approach we are advocating can only mitigate risk and it is only as good as the information we can uncover about these risks. There are several obvious sources of risk information. The most valuable are published incompatibilities. As mentioned, Microsoft previously published a set of issues that arise when converting from Lotus to Excel. We designed our Lotus tool to identify these issues. In the case of CDF, netCDF, and HDF, the primary conversion risks are identified on the primary web sites providing format documentation. Surprisingly, while these risks are clearly identified, the conversion tools provided at these sites do not provide feedback when these risks are encountered. Instead, they silently convert the files using a "best-effort" approach. Unfortunately, this can have unintended consequences. For example, conversion from HDF4 to HDF5 is safe if the target files are treated as read-only, but may not be safe if they are subsequently modified. This danger arises because of the way shared objects are handled in the conversion process.

In the remainder of this section we review these formats, as well as the conversion issues we have identified.

Lotus 1-2-3

Lotus 1-2-3 was a dominant spreadsheet application through the 1980s and early 1990s, but was replaced by newer applications like Microsoft Excel. Due to its popularity, files from 1-2-3 could be opened in other spreadsheet applications, such as Excel, Gnumeric and OpenOffice. These tools can be used to migrate 1-2-3 files to other formats. While Gnumeric and OpenOffice still support Lotus 1-2-3, Excel dropped support after 2003. Complicating the situation for Lotus 1-2-3 is that the file format evolved over time. While the documentation for version 1 (wk1) files is good and consequently the support in Gnumeric appears fairly solid, later versions, such as wk4, are poorly documented and the corresponding support in Gnumeric is incomplete and full of errors, as noted in its own source code.

[6] Gnumeric source code repository: http://git.gnome.org/browse/gnumeric/

The primary source of risk information that we considered was a Microsoft support article mentioned previously, which provides information about key issues in migration from Lotus 1-2-3 to Microsoft Excel. Migration problems appear to be minimized by using Excel 7.0 and later. Some translation issues are due to fundamental differences in the programs. For example, charts and graphs can be part of the worksheet in Lotus 1-2-3, while conversion to Excel places these on separate chart pages. More fundamental are differences in calculations. For example, Excel and Lotus differ on the behavior of functions such as @MOD, @VLOOKUP, and @HLOOKUP. There are differences in operator precedence, such as exponentiation (^) and unary positive and negative (+, -). Some features are either not supported or not translated by default, including linked files and macros. Finally, key statistical functions are computed differently (McCullough, 2004).

While these issues were identified in the context of Excel, it is probably reasonable to conclude that Gnumeric has similar problems when the target conversion format is Excel. An important step that we have not yet taken is to closely evaluate the other conversion paths in Gnumeric, such as conversion to OpenOffice.

The presence of so many subtle differences suggests that migration from Lotus 1-2-3 to Excel cannot be performed reliably; however, the hypothesis motivating this work is that most Lotus 1-2-3 files do not manifest the risky features. To test our hypothesis we wrote a tool that parses Lotus 1-2-3 files using the format documentation published by Lotus (Lotus Books, 1986) and detects the categories of potential risks described above. It is far simpler to develop a tool to "walk" a file format, such as Lotus 1-2-3, than it is to translate to another format. Indeed, our program consists of about 500 lines of code. To understand why this is the case, consider the following abbreviated description of the wk1 file format.

The layout of Lotus 1-2-3 files uses a record-based format with a record for each of the cells and attributes of the file. Each record starts with a byte code denoting what element it describes and the length of the record, followed by the element's contents. Thus, parsing a wk1 file consists of reading and interpreting the byte codes in the file while preserving minimum context to find elements to analyze, namely formulas.

As mentioned, alternative migration paths for Lotus files include open source tools such as Gnumeric.[7] Testing and code review show that Gnumeric properly opens Lotus 1-2-3 version 1 files. Additionally, it opens version 3 and 4 files, but incorrectly displays the stored formula values. This is corrected on forcing recalculation.

While Lotus 1-2-3 is primarily of interest as an historical format, which was once widely used but is now largely abandoned, we note that even with its former dominance and a relatively long period of phased obsolescence, the state of migration tools from Lotus 1-2-3 can best be described as spotty and the feedback they provide in the face of inconsistent migration is non-existent. This is the case even when Microsoft at one time had a significant financial incentive to woo Lotus users to Excel. With the loss of this financial incentive, Microsoft support has evaporated. What then is the situation for data format migration where no financial incentive exists?

[7] Welcome to Gnumeric: http://www.gnome.org/projects/gnumeric/

 doi:10.2218/ijdc.v7i1.212

CDF and netCDF

In this section we consider two important and closely related scientific data formats – CDF and netCDF. CDF (common data format) is a library and toolkit developed by NASA for managing array data. Key format features described on the CDF homepage[8] include:

- Self-describing data format for storage and manipulation of scalar and multi-dimensional data in a discipline-independent fashion.
- Scientific data management package (known as the "CDF Library") which allows programmers and developers to manage and manipulate scalar, vector, and multi-dimensional arrays.

There are translation tools available between CDF and a number of other data formats, including netCDF, FITS, ASCII and HDF4. These tools can be downloaded but are also accessible through the Data Translation Web Service.[9]

At a high level, netCDF is similar to CDF. It is a self-describing data format and has libraries for sharing array-oriented data, and it was initially derived from NASA CDF. However, it has diverged and is no longer compatible with CDF. More recently it has begun to support the HDF data formats described below. There is more than one netCDF file format, including the "classic," 64-bit offset, and, more recently, the netCDF-4 which is HDF5 based. Furthermore, the netCDF libraries support interoperability with a variety of other data formats[10].

From the CDF FAQ[11] review of NASA's CDF-to-netCDF conversion code, and CDF and netCDF API source code[12] we found the following differences between CDF and netCDF:

1. CDF supports a multi-file format for storing variables in separate files with a metadata file, while netCDF does not.
2. CDF supports native encoding for data representation for speed optimizations while netCDF only supports platform-independent network encoding.
3. CDF includes a native epoch data type for storing high-resolution time information, while netCDF does not.
4. NetCDF uses named dimensions, while CDF does not.
5. CDF supports up to ten dimensions per variable and netCDF supports 32.

Multi-file format is an organizational choice and does not affect data. Nothing needs to be done in converting multi-file CDFs to netCDF. Similarly, the conversion can directly change from native to network encoding without data loss. Though

[8] The Common Data Format (CDF): http://cdf.gsfc.nasa.gov/

[9] Data Translation Web Service: http://cdf.gsfc.nasa.gov/html/dtws3.html

[10] NetCDF (Network Common Data Form) homepage: http://www.unidata.ucar.edu/software/netcdf/

[11] CDF Frequently Asked Questions: http://cdf.gsfc.nasa.gov/html/FAQ.html

[12] CDF Software Download: http://cdf.gsfc.nasa.gov/html/sw_and_docs.html NetCDF downloads: http://www.unidata.ucar.edu/downloads/netcdf/index.jsp

netCDF does not have an epoch data type, the CDF API provides functions to convert an epoch variable to an arbitrary date string and vice versa. CDF has no way to represent dimension names from netCDF, so converted data sets may need additional metadata to retain this information. Since CDF does not support more than ten dimensions per variable, conversion from netCDF data sets could be a problem. The DTWS source code does not address this, but as shown in our results this rarely arises.

HDF

The Hierarchical Data Format (HDF), another common scientific data format, exhibits migration risks because of the changes between HDF4 and HDF5. HDF is a self-describing format that includes interfaces for numerical, multi-dimensional and image data. It represents data objects hierarchically by relating them through Vgroups. Each version of the format through to version 4 was completely backwards-compatible. With the release of HDF5, the data model was significantly simplified. However, this broke backwards compatibility. In general, there is a well-defined mapping from objects in previous versions of HDF to HDF5 objects. However, default conversion methods to HDF5 do not always maintain the relationships of the original data set.

The HDF Group provides two methods for converting HDF4 data sets to HDF5. One is a stand-alone application which blindly performs the default conversion according to the mapping of HDF4 data objects to those in HDF5. The other method is an API used to manually perform conversions object-by-object. According to the documentation, the latter method is intended for making changes to the data set during translation, such as changing object names and merging Vgroups.

According to the HDF Group's HDF4 to HDF5 Programmer's notes,[13] the following issues can arise when converting from HDF4 to HDF5:

1. Separate HDF4 Vgroups containing different data objects with the same name when the Vgroups are to be merged into a single group in HDF5.
2. HDF4 data objects which are shared between multiple Vgroups, and the Vgroups are to be converted into separate groups in HDF5.
3. HDF4 data objects that are not named.

The first difference could arise from records sharing the same attribute across elements of a data set. If the affected Vgroups are merged during conversion, one of the data objects will get a default name. The standard conversion process will not merge Vgroups, so this is not a concern for the default conversion application. The second difference arises when multiple Vgroups alias the same object. Standard conversion will create a copy of the object for each Vgroup. For read-only data sets this is not an issue since data in each copy is accurate. If converted, data sets are modified. This is an issue because changes in one copy of the object will not occur in the other copies. Data object names are required in HDF5 but not in HDF4, leading to the third difference. The conversion program assigns a default name based on the object's type and internal reference number in this case. While this is not technically

[13] H4 to H5 Programmer's Notes: http://www.hdfgroup.org/h4toh5/H4H5ProgrammersNotes.pdf

 doi:10.2218/ijdc.v7i1.212

an error, it is not clear that future users of the data will find this naming scheme particularly useful.

Tools

To test our hypothesis that most file conversions are "safe" we wrote risk analysis tools for each of the four file formats described previously. The most challenging of these efforts was the tool to examine Lotus 1-2-3 files. This was written entirely from scratch, based upon the published format information as well as an examination of format support provided in Gnumeric. The program itself is fairly small – approximately 500 lines of C. Most of the design effort (approximately 80%) was used to parse formulas for the major migration risks. The program performs a single pass over Lotus 1-2-3 wk1 files and hence is quite fast to execute on a large collection of files. Indeed, it is much faster to identify the risks in a Lotus file than to convert the file.

In the development of tools to analyze CDF and netCDF we were greatly aided by the availability of well-documented library APIs. Using these APIs only a simple skeleton program is needed to traverse files to identify potential risks. For CDF and netCDF, our file analysis programs consisted of 300 and 150 lines of C code, respectively. These used the CDF version 3.3.0 API obtained from NASA and the netCDF version 4.1.3 API from Unidata.

The HDF tool was the largest at 900 lines of code because of the large number of separate interfaces for different data objects. It utilized the HDF4 version 4.2.6 API. Each tool could analyze any single file in our data set in under one second and could analyze large data sets (2600 to 61,000 files) within a few minutes.

Results

We evaluated 2,747 CD-ROMs of published government data. Of these, 110 contained files in one of the four formats considered in this study. 36 contained a total of 14,878 Lotus 1-2-3 files. 14,022 of the 1-2-3 files were wk1 files which we analyzed. The remaining were 281 wk3, 568 wk4, and 7 123 files. 68 CD-ROMs contained 61,247 CDF files, four contained 3,162 netCDF file, and two contained 2,605 HDF files.

Evaluation of Lotus 1-2-3 files (see Table 1) found 2,266 (15.2%) files that contained one or more formulas, meaning that nearly 85% of the files were simple tabular data with very low migration risk. The data set included only 147 (0.99%) files containing formulas with operations having potential migration risks for Excel.

Our data set contained 61,247 CDF files, all of which were in version 2. Of these, only 14,574 (23.8%) had no potential risk considerations for converting to netCDF. A large fraction, 46,669 (76.2%) used the epoch data type to provide basic timestamp information. Since netCDF has no such data type, the standard conversion translates this data type to strings containing date/time information. This represents a mismatch in file functionality that cannot be avoided; however, it is not clear that this conversion represents a significant risk, since the CDF API itself provides methods to

convert between epoch data and strings. Furthermore, Unix systems provide library calls to convert bidirectionally between date/time records and strings.

We found only four CDF data sets that used the multi-file format, indicating that it is a rarely-used feature. While it is important to note this in conversion, it is not a high-risk conversion factor. We found no CDF files that used native encoding, indicating that even though native encoding may optimize performance it is not often used in practice. The native/network encoding issue is relatively low-risk because the conversion can be performed in a totally error-free manner; however, it is very important that such conversions be checked to ensure they are performed correctly.

Our data set also contained 3,162 netCDF files, all of which used the "classic" file format. As expected, all of them used named dimensions which would be omitted in translating to CDF. None of the files in the set included variables with more than CDF's maximum of ten dimensions. While this is a potentially high-risk migration issue, it does not seem to occur in practice.

We found 2,213 HDF files in our data set. Of these 352 (15.9%) were HDF3 and 1,861 (84.1%) were in HDF4 format. 1,891 (85.4%) contained multiple objects with same name belonging to the same Vgroups. All but two of those files (1,889) also contained data objects shared between multiple Vgroups. The objects with duplicate names do not pose any migration risks for default conversion, but the majority of the HDF files have migration risks if the converted data are to be modified because of the copied aliased objects. Finally, 324 (14.6%) of the HDF files had no conversion considerations and could be converted and modified without any risk.

Risk	Occurrences	Formulas	Files
Operation order (exp., neg.)	0	0	0
@MOD	704	639	74
@HLOOKUP	829	828	117
@VLOOKUP	257	193	49
Total files		14,022	
Total formulas		392,736	
Files with formulas		2266	
Formulas with conversion issues		1660	
Files with conversion issues		147	

Table 1. Results for Lotus 1-2-3 WK1 files.

Conclusions

While data migration is commonly practiced and many data formats are supported by one or more conversion tools, we found that most formats exhibit potential migration risk. The primary sources of these risks are mismatches between source and target formats and differences in interpretation. For example, CDF and netCDF have

 doi:10.2218/ijdc.v7i1.212

diverged sufficiently that each contains features that are not supported by the other, and HDF has undergone a major simplification that led to features being abandoned. Nevertheless, it appears that most files can be safely converted, although potential risks should be noted in the conversion process.

Our initial hypothesis in starting this work was that, where incompatibilities exist, most files would not be subject to migration risk and hence do not require special attention. For example, we found that fewer than 1% of Lotus 1-2-3 files used operations with the potential for conflicting behavior when converted to Excel. A robust translation process would then need to examine this relatively small fraction of files to determine if the risks are actually manifest.

The situation for CDF and netCDF is not quite as clear. While the use of epoch (temporal) data was prevalent in our test files, this is not necessarily a risk (assuming the conversion tools behave correctly). The use of named dimensions is common, but the information provided could be saved in a separate metadata file. There is no indication that the available conversion tools provide a mechanism to do this. We found that HDF files that use aliased data objects would be copied on conversion to HDF5. While this conversion leads to no loss of data accuracy, subsequent modification to the HDF5 files might lead to errors because changes to one copy of an object might not be made to another copy.

In writing our analysis tools we found a fundamental difference between analyzing proprietary formats like 1-2-3 and open ones like CDF, netCDF and HDF. For 1-2-3 we had to write code to directly parse files based on information from the Lotus Developer's Guide (Lotus Books, 1986), code from open source projects like Gnumeric, and reverse engineering test files. In contrast, all three open formats we examined had a curating organization which provided APIs for interacting with files abstractly and documentation for the structure and use of the format. We developed the tools for open formats more quickly and reliably than for 1-2-3.

Surprisingly, we found no evidence that the available conversion tools note the presence of conversion risks. It seems that the common approach is to silently translate in a best-effort manner. As expected, writing associated risk assessment tools was not particularly difficult. None of our programs required more than 1,000 lines of C code. This work supports our hypothesis that most scientific data files do not exhibit high-risk migration issues and that writing programs to categorize formats by their migration risks is simple compared to writing conversion software. Thus, most files can be converted with existing best effort migration tools, leaving a small number of files to convert manually. Thus, we believe that any large-scale data migration effort should include the creation of risk assessment tools for the particular migration paths being utilized.

Acknowledgements

The authors would like to gratefully acknowledge the support of the Data to Insight Center, a partnership of the School of Informatics and Computing, Digital Libraries and Pervasive Technology Institute at Indiana University. This research funded in part by a grant provided by the Lilly Endowment, Inc.

References

Nature. (2009). Data's shameful neglect. *Nature 461*. Retrieved from http://www.nature.com/nature/journal/v461/n7261/full/461145a.html

Arms, W.Y., Hillmann, D., Lagoze, C., Krafft, D., Marisa, R., Saylor, J., Terizzi, C., Van de Sompen, H., Gill, T., Miller, P., et. al. (2002). A spectrum of interoperatbility: The site for science prototype for the NSDL. *D-Lib Magazine 8*(1). Retrieved from http://www.dlib.org/dlib/january02/arms/01arms.html

Becker, C., Kulovits, H., Kraxner, M., Gottardi, R., Rauber, A. & Welte, R. (2009). Adding quality-awareness to evaluate migration web-services and remote emulation for digital preservation. Paper presented at ECDL 2009 Proceedings of the 13th European conference on Research and advanced technology for digital libraries, Corfu, Greece.

Becker, C., Rauber, A., Heydegger, V., Schnasses, J., & Thaller, M. (2008a). A generic ML language for characterizing objects to support digital preservation. Paper presented at the ACM Symposium on Applied Computing, New York, United States.

Becker, C., Rauber, A., Heydegger, V., Schnasses, J., & Thaller, M. (2008b). Systematic Characterization in Digital Preservation: The eXtensible Characterisation Languages. *Journal of Universal Computer Science 14*(18).

Chou, C. (2007). Format identification, validation, characterization, and transformation in DAITSS. Paper presented at IS&T Archiving 2007, Arlington.

Curtis, J. (2006). Aons system documentation revision 1692006-09-29. Australian Partnership for Sustainable Repositories.

Dappert, A. & Farquhar A. (2009). Significance is in the eye of the stakeholder. Paper presented at ECDL 2009 Proceedings of the 13th European conference on research and advanced technology for digital libraries, Corfu, Greece.

Hunter, J. & Choudhury, S. (2005). Semi-automated preservation and archival of scientific data using semantic grid services. Paper presented at the Semantic Infrastructure for Grid Computing Applications workshop at the International Symposium on Cluster Computing and the Grid (CCGrid 2005). Retrieved from http://metadata.net/panic/Papers/SIGAW2005_paper.pdf

Ide, M., MacCarn, D., Shepard, T., Weisse L. (2002). Understanding the preservation challenge of digital television. *Building a National Strategy for Preservation: Issues in Digital Media Archiving*. Council on Library and Information Resources and the Library of Congress, Washington, D.C.

Lawrence, G., Kehoe, W.,Rieger, O., Walters, W., Kenney, A. (2000). *Risk management of digital information: A file format investigation.* Technical Report

 doi:10.2218/ijdc.v7i1.212

LKRWK-2000. Council on Library and Information Resources, Washington, D.C.

Lotus Books. (1986). *Lotus file formats for 1-2-3 Symphony & Jazz.* Reading, Massachusetts: Addison-Wesley.

McCullough, B. (2004). *Fixing statistical errors in spreadsheet software: The cases of Gnumeric and Excel.* Report for the Computational Statistics and Data Analysis Statistical Software Newsletter. Retrieved from http://www.csdassn.org/software_reports/gnumeric.pdf

Pearson, D. (2008). AONS II: Continuing the trend towards preservation software 'nirvana.' *New Technology of Library and Informations Service: iPRES2007 Special Issue.*

Pearson, D. & Webb, C. (2008). Defining file format obsolescence: A risky journey. *International Journal of Digital Curation 3*(1).

Rosenthal, D., Lipkis, T., Robertson, T. & Morabito, S. (2005). Transparent format migration of preserved web content. *D-Lib Magazine 11*(1).

Stanescu, A. (2005). Assessing the durability of formats in a digital preservation environment. *OCLC Systems & Services: International Digital Library Perspectives 21*(1).

University of Leeds. (2007). *Survey and assessment of sources of information on file formats and software documentation.* Report of the Representation and Rendering Project, University of Leeds.

Walker, F. & Thoma, G. (2004). A web-based paradigm for file migration. Paper presented at the IS&T Archiving Conference, San Antonio, Texas, USA.

Woods, K. & Brown, G. (2009). Creating virtual CD-ROM collections. *International Journal of Digital Curation 4(2).*