Book Section:
Mellor, P., Wheatley, P. and Sergeant, D. (2002) Migration on request, a practical technique for preservation. In: Goos, G., Hartmanis, J. and van Leeuwen, J., (eds.) Research and Advanced Technology for Digital Libraries : 6th European Conference, ECDL 2002 Rome, Italy, September 16–18, 2002 Proceedings. Lecture Notes in Computer Science, 2458 (2458/2). Springer , Berlin / Heidelberg , pp. 516-526. ISBN 978-3-540-44178-6

https://doi.org/10.1007/3-540-45747-X

Published paper
Mellor, P., Wheatley, P. and Sergeant, D. (2002) *Migration on request, a practical technique for preservation.* In: Goos, G., Hartmanis, J. and van Leeuwen, J. (eds) Research and Advanced Technology for Digital Libraries : 6th European Conference, ECDL 2002 Rome, Italy, September 16–18, 2002 Proceedings. Lecture Notes in Computer Science, 2458 (2458/2002). Springer , Berlin / Heidelberg, pp. 516-526. ISBN 978-3-540-44178-6

Migration on Request, a Practical Technique for Preservation

Phil Mellor[1], Paul Wheatley[1], and Derek Sergeant[1]

CAMiLEON Project*, Edward Boyle Library, The University of Leeds, Leeds LS2 9JT, UK
{P.R.Wheatley, D.M.Sergeant}@leeds.ac.uk

Abstract. Maintaining a digital object in a usable state over time is a crucial aspect of digital preservation. Existing methods of preserving have many drawbacks. This paper describes advanced techniques of data migration which can be used to support preservation more accurately and cost effectively.
To ensure that preserved works can be rendered on current computer systems over time, "traditional migration" has been used to convert data into current formats. As the new format becomes obsolete another conversion is performed, etcetera. Traditional migration has many inherent problems as errors during transformation propagate throughout future transformations.
CAMiLEON's software longevity principles can be applied to a migration strategy, offering improvements over traditional migration. This new approach is named "Migration on Request." Migration on Request shifts the burden of preservation onto a single tool, which is maintained over time. Always returning to the original format enables potential errors to be significantly reduced.

1 Introduction

In a digital library a new problem has surfaced, collections of digital objects become obsolete and unusable while technology rapidly evolves. Meanwhile, the field of digital preservation is only just beginning. Surely it is safest to wait for the advice to mature? However, in doing so many current digital objects that needed preserving will have already been lost (irretrievably).

Maintaining a digital object in a usable state over time is a crucial aspect of digital preservation. In order to preserve successfully, action must be taken to ensure that digital objects can be easily rendered on current computer platforms over time. Migration has been widely used to move obsolete data into current data formats. When a data format becomes obsolete, a migration tool is used to transform the digital object into a data format which can be rendered and used on a current computer platform. When this format becomes obsolete, another

* Research for this paper was supported in part by the National Science Foundation, Award #9905935, Digital Library Initiative - International, Emulation Options for Digital Preservation and by the Joint Information Systems Committee in the UK.

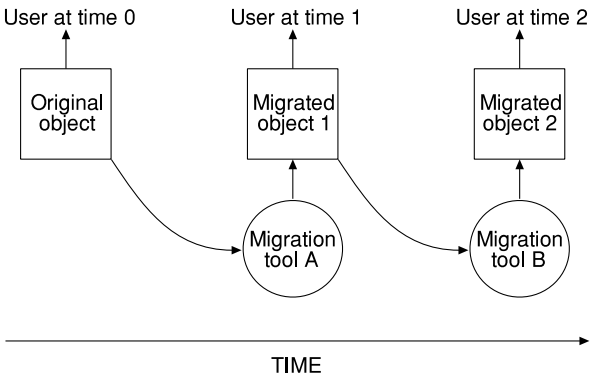

Fig. 1. A digital object preserved using traditional migration

transformation is performed, and so on. There are many drawbacks with this strategy of "traditional migration" (see figure 1). Any errors or omissions from a transformation will propagate throughout and hence be present in all future transformations (see figure 2). Existing methods of preserving digital data often fall short of accurately preserving and authentically rendering an original digital document. Continually producing new migration tools whenever a transformation is required and then applying them to possibly very large data holdings is costly. This paper describes some advanced techniques of data migration which can be used to support preservation work more accurately and cost effectively.

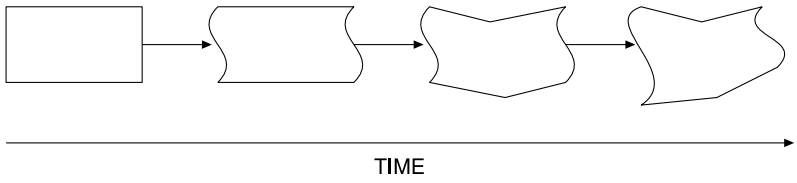

Fig. 2. Errors propagating through each conversion step of traditional migration

Traditional migration has been widely used to convert obsolete data into current data formats. In the short term this offers a way of keeping digital materials current, but it is not an effective long term strategy. This research performed by the CAMiLEON project[1] has investigated a way of applying migration in a more sensible and effective way. This new approach is named "Migration on Request." The CAMiLEON project has implemented a real Migration on Request tool to ensure that this theoretical work is truly practical. This paper explains this technique and presents findings from the experimental implementation. The Digitale Bewaring project is an excellent resource about migration [2].

Underpinning this approach is the notion of indefinite retention of an abstract byte-stream. This means preserving the original data object (not preserving

eight-inch floppy disks and giant reels of half-inch mag tape). These principles have been adopted from the Cedars project [3], whereby the term "migration" is only applied to operations which transform the data object [4].

2 Theoretical Basis to Migration on Request

Cedars suggested that, by preserving the original bytestream of a digital object, preservation work could be performed more effectively [5]. If a bytestream is preserved unchanged over time, a way of interpreting or rendering that original format will also be necessary. An obvious problem with this strategy is the short lifetime of any migration tool designed to perform this task. Previous research on the CAMiLEON project has developed techniques for software longevity [6]. While these techniques were originally developed to maintain software emulators over generations of platform, they equally apply to implementing migration tools. Combining these methods of software longevity with the principle of always maintaining the original bytestream, we see a new form of migration taking shape.

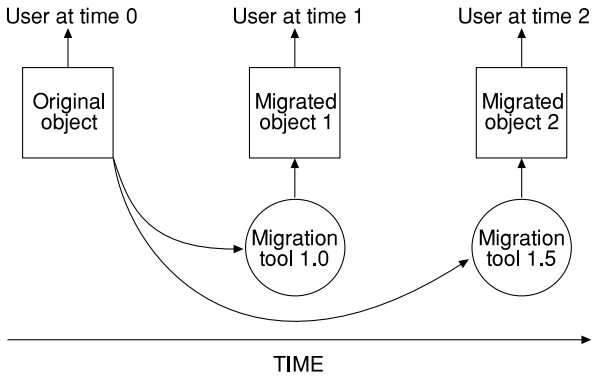

Fig. 3. Applying a Migration on Request tool over time

This foundation work points the way to a practical method of implementing a more useful migration strategy. Migration on Request shifts the burden of preservation from handling vast quantities of digital objects to a single tool for each class of data format in the archive. This tool renders all of the digital objects and is maintained over time (see figure 3) using the previously mentioned software longevity techniques. A digital object is simply archived in its original format. New output modules can be added to the tool to produce newer data formats as previously supported formats become obsolete. Always returning to the original digital format enables Migration on Request to significantly reduce the possibility of errors being introduced during the conversion process (see figure 4). There is always only one transformation step from the original to the current format. In any case the original is what is retained in the archive.

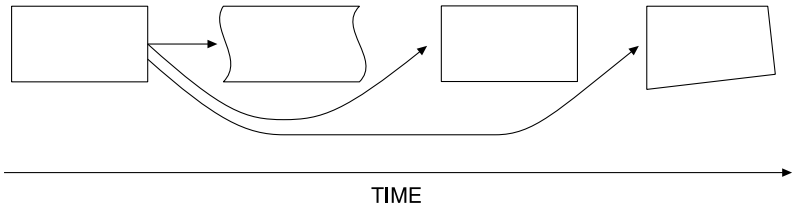

Fig. 4. Migration on Request may introduce minor errors, but these do not propagate

Migration on Request offers several key benefits over a traditional migration approach:

- The code which reads in and interprets a particular file format need only be implemented once.
- Using only one migration step increases migration accuracy.
- Issues of authenticity are greatly simplified as a digital object is preserved in its original form.
- The modular design of a migration tool makes the implementation of a "reversible migration" test much simpler and cheaper.
- The migration tool is only deployed "on request" and so offers massive savings where a large number of digital objects are preserved.

3 Migration on Request Tool Design

Figure 5 shows how a Migration on Request tool breaks down the elements required to migrate a digital object. This design is extensible, so as supported output formats become obsolete new output modules can be added without having to re-write existing input modules. This provides a major cost saving in comparison to a traditional migration approach.

The Consultative Committee for Space Data Systems OAIS Reference Model [7] reminds us that the only way of ensuring a migration step has been completed without error is by the proof of a reversible migration. If we can convert a migrated object back to its original form, and it matches the original object then no data has been lost. With a traditional migration approach the effort required to implement a reversible migration test effectively doubles the overall implementation required. On top of that, all this work must be repeated at each subsequent migration step! With Migration on Request, the modular framework allows us to make substantial savings in implementation time. The addition of an input and output module provides support for a new data format with the opportunity for a reversible migration test. When we add support for subsequent formats we make use of existing input modules. This represents a cost saving over the complete re-implementation at each step of a traditional migration process.

The ability to have a number of input as well as output modules means that one Migration on Request tool can support a number of different input formats.

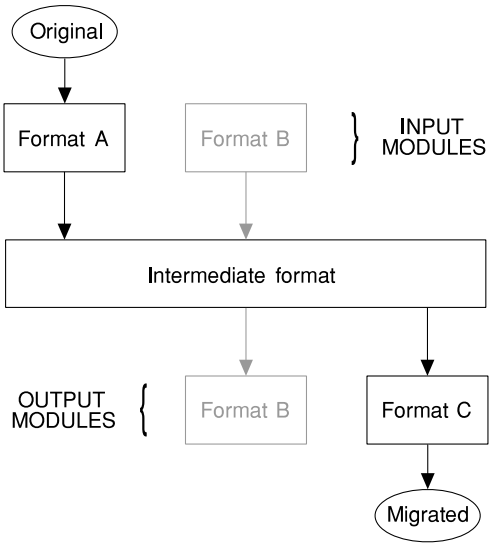

Fig. 5. The Migration on Request process

For example, an archive may contain textual data in a number of word processing formats, which are sufficiently similar for one Migration on Request tool to support them. Again, we see massive cost savings over traditional migration. A Migration on Request tool is maintained over time by adding new output modules as supported output formats become obsolete. These output formats can be used with all the supported input formats, stripping away all the wasteful implementation redundancy found in traditional migration.

The modularised design of a migration tool makes it easier to maintain and provides us with a functional record of the file format in which our preserved data is maintained.

4 Testing the Theory

The aim was to provide a practical test of a Migration on Request approach. A successfully working Migration on Request tool would provide strong evidence that this approach is useful, and also highlight any difficulties which became apparent during the implementation. It was important to make this test hard enough to tease out these implementation issues. Since vector graphic formats are sufficiently complex (unlike text or bitmap graphics), they were chosen as the focus of this test. This ensures that the Migration on Request strategy is tested thoroughly before progressing to other classes of digital objects.

Three formats covering a cross section of existing vector formats were chosen for implementation: WMF [8], Draw [9] and SVG [10]. Windows Meta Files (WMF) were developed by Microsoft. Images are represented by a series of instructions that match the calls made by applications to the Windows Graphics

Device Interface. The Draw file was invented by Acorn Computers in the early 1990's and is commonly used for exchanging data between applications on the RISC OS platform. In the UK a lot of educational material still exists in this format. Scalable Vector Graphics (SVG) are a new XML based format developed by the W3O, and is mainly used for web site imagery. Figure 6 shows how an oversimplified vector diagram of a face is represented in these formats. Several vector graphics files, of varying complexities, were used in order to test the Migration on Request tool.

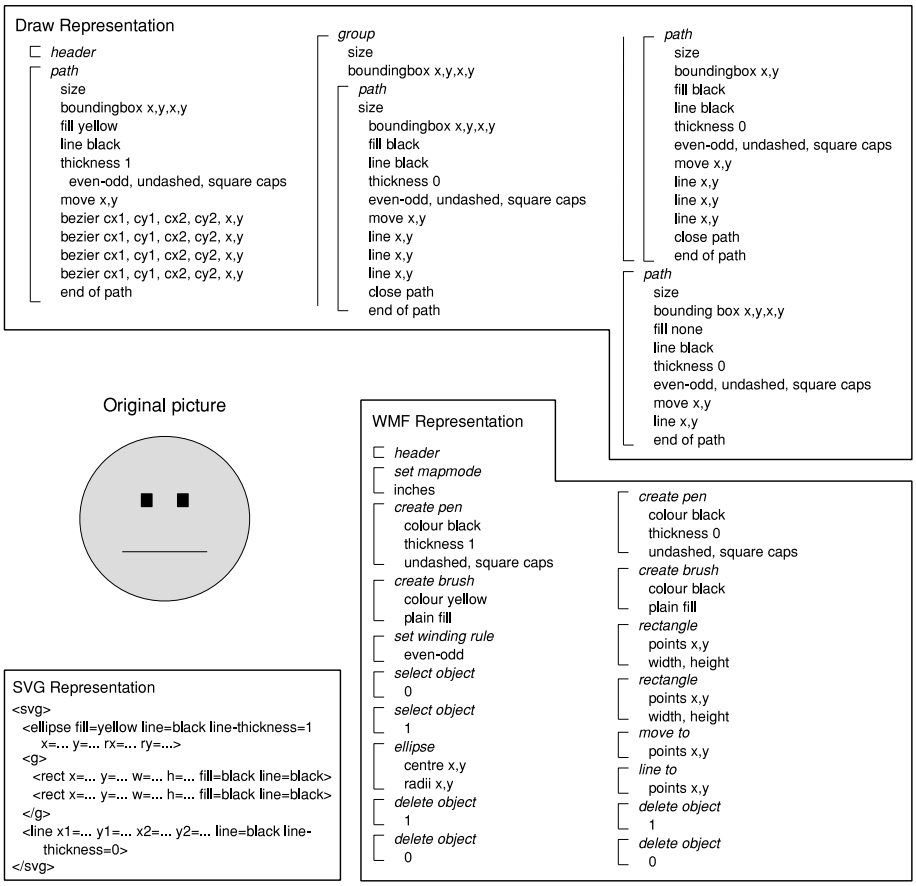

Fig. 6. WMF, Draw, and SVG representations for a simple face

The migration tool was constructed in a modular way. Separate functions were used to input each format and return an intermediate structure, a hierarchy of elements such as lines, ellipses and polygons. This structure could be passed to an output module, but it is likely to contain elements unsupported by the output format. To solve this the output module passes the structure on to a

series of functions that downgrade or convert any unsupported elements into ones that the format can handle. In order to minimise the amount of conversion routines that are needed, a chain of conversions can be applied; for example a curved path could be converted to a straight line path, and then to a series of individual lines; there is not need to create a special routine or clause to convert curved paths directly into individual lines.

5 Data Formats and Their Interpretation

The intermediate format needs to encompass all the features of the input formats. There is usually more than one way to represent an element, but it is important that the method of representation does not lose any of the original information. It should not be a problem whether an ellipse is described with a centre point and the two radii, or with a bounding box, as the two methods are totally interchangeable (see figure 7). The intermediate format does not need to cater for both forms of representation.

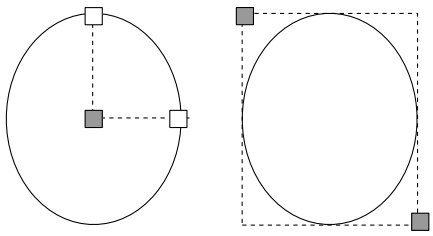

Fig. 7. Alternative descriptions of the same ellipse

However, the representation of some objects may be so dissimilar in different formats that one method in the intermediate format will not encompass them all without degradation. Therefore it is likely that there will be some 'duplication' in the intermediate format.

For example, SVG files have a rectangle element with a 'rounded edge' attribute. An ordinary rectangle can be produced by setting this attribute to zero. WMF files have both rounded and ordinary rectangle elements, despite also having the ability to create ordinary rectangles in the same manner as SVG. Draw files do not support rounded rectangles; instead a path of lines and curves would be needed.

The WMF structure is oriented in favour of the implementation of rendering rather than the content of the graphic itself. This means there are lots of elements specific to WMFs which are unlikely to be needed by file formats in the future and would get ignored or 'flattened out' by the output modules anyway. WMF files allow the presence of lots of redundant instructions. For example, a series of instructions to set the mapping mode to inches, millimetres, then back to

inches again with no objects being drawn in the meantime, would be pointless but possible. This is just a simple example but the possibilities are tremendously wasteful. The WMF is basically a simple programming language, and very poor, inefficient programs can be written with it.

Retaining any of this seemingly redundant information would only serve to clutter the intermediate format, thereby making it more complex, less intuitive, and increasing the risk of bugs or incompatibilities that are hard to track down. It seems reasonable, therefore, that such elements are not made part of the intermediate format and are dealt with and converted into more suitable representations by the WMF input module. The limitation of this approach would be that reversible migration of WMFs would become impossible.

5.1 Internal Number Representation

The Draw format measures values in OS units (1/180th of an inch), stored as fixed point 32-bit numbers with an 8 bit fractional part. WMFs offer a choice of units for a value, which is stored in 16 bits. Furthermore, an offset and scaling can be applied. SVG numbers are written as a string of ASCII characters, usually in base 10.

Care needs to be taken when storing such numbers in memory. Errors in precision can occur when numbers are stored in a floating point representation, particularly when storing exceptionally large or small numbers. However, the range of fixed point numbers is more constrained and can cause greater imprecision errors, usually through truncation of the fractional part.

There are many different units of measurement that could be used — inches, millimetres, pixels, and so on. These values need to be kept in their original units for as long as possible, since unnecessary conversion could lose accuracy.

In our Migration on Request tool a structure is defined to store a unit of measurement and a value (as either an integer or floating point number). Various functions can extract the value in different units. The design could go so far as to simulate various number representations itself, such as 16 and 32 bit integers, even ASCII strings. This would require a lot of maths routines to be implemented by hand, such as addition and multiplication, perhaps even square root functions.

6 Reversible Migration

The reversible migration test compares a migrated version of the digital object to the original digital object. This is done by migrating the migrated version back into the original format, see figure 8. A Migration on Request tool can be used to perform all of the migrations required for the reversible migration test. CAMiLEON has performed a reversible migration test successfully with the Draw format, and a reversible migration was achieved (bar some minor information such as user interface preferences). It would even be possible to reversibly migrate WMF files, if its features were integrated with the intermediate format. Such features would significantly increase the complexity of the intermediate

format, making the tool harder to maintain. A choice between complexity and reversibility must be made. It seems unlikely that the rendering structure found in WMF will also appear in future vector graphic formats, so it seems logical not to support this style here. ASCII formats such as SVG raise an interesting quandary. The amount of white space (new lines, spaces etc) is irrelevant to the information stored in the file, but is often used to indent nested items or to separate different sections to enable human readability. For true reversible migration (using a "diff" tool), this white space would also have to be preserved in the internal format. Alternatively a parsing tool could be used to remove this non-essential information before applying the "diff" test. Migration on request certainly makes the 'holy grail' of reversible migration easier to reach than through conventional migration techniques.

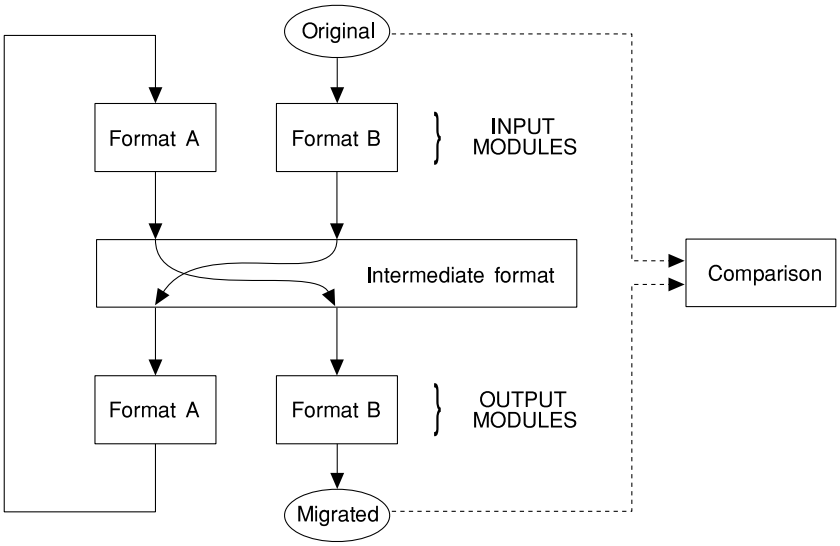

Fig. 8. The reversibility test

7 Reliance on Original Evidence

When developing the tool, it was essential to view the output from the migration tool in a graphics application (rather than a file editor), and compare this with a rendering of the original file. A simple visual check was used to confirm whether elements had migrated properly. Although not particularly accurate, this method allowed obvious errors to be identified quickly.

Imagine a collection of abstract paintings by Mondrian existed only in a poorly documented, (now) unused vector format. If the colours were mixed up or some shapes distorted during migration, without a true comparison to the

original, these errors may go unnoticed. Technically it would be possible to examine each file manually to determine what it would look like, but for large or complex files this would be an arduous task.

Once the migration tool can be shown to be working correctly, the need for such evidence is less necessary, but still relevant. Later modifications to the tool would require all the modules to be tested again for any discrepancies in the migration. Such modification might be, for example, adding new features to the intermediate format to support a new input module. Evidence of an original rendering of test files would again be useful here, perhaps via emulation.

Using the original applications need not be the only way to acquire this evidence; screenshots, written documents, etc, are other useful sources. These resources may also have to be preserved as time passes. Implementation of a Migration on Request tool should be done when the format is still in a usable form and such evidence can be gathered. This echoes Holdsworth and Wheatleys observations on the timeliness of emulation for preservation [6].

8 The Evolution of File Formats

In order to prepare any Migration on Request tools for the future, study into trends of data formats is needed. Vector graphics used to be the preserve of design and publishing, but it is a reasonable assumption that in a few years their most widespread deployment will be on the World Wide Web. Designers of migration tools should concentrate on preserving features that are most likely to be used in future developments of their format genres.

The development of open standards is interesting. If the trend is to define and follow standards, then choosing an internal format similar to one of these standards would be a sensible way forward. SVG seems a good basis for the intermediate format in a vector migration tool. Unfortunately it seems unlikely that relying on standards will be sufficient to ensure preservation. The need to maintain a commercial advantage over competitors has meant in the past that standards are extended or not adhered to, HTML being a case in point. We have to accept that standards can change over time and will at some point become obsolete. Fortunately a Migration on Request strategy can benefit from the stability and longevity of open standards but is not tied to them. It must also be remembered that a format is not necessarily a good design just because it is a standardised format [11].

9 Evaluation

The practical implementation of a Migration on Request tool was a valuable test of our theoretical strategy. The modular migration tool successfully imports, converts and exports a number of vector graphic formats. The experiences encountered in developing a tool of this kind were useful in raising problem areas and providing the chance to develop solutions to tackle these difficulties. In particular, implementing a reversible migration test was not as easy in practice as

was originally thought but this was not a specific problem with Migration on Request. Ordering of data elements, non-critical information and multiple methods of representing the same data are problems likely to be encountered with most data formats, using any migration strategy.

The implementation work showed that the initial development of a Migration on Request tool is not overly laborious or costly. Over a short to medium term period Migration on Request should offer major cost savings in comparison to a traditional migration strategy, even where standard/open formats are utilised.

10 Conclusions

The CAMiLEON project has developed a Migration on Request tool which shows that a preservation strategy of this kind can work in a practical environment. Migration on Request provides a more accurate and cost effective strategy for preserving digital objects than traditional migration. Because Migration on Request relies on the preservation of the original bytestream of a digital object it can effectively work alongside an emulation strategy. If open source emulation and Migration on Request tools become available, a digital repository can effectively offer different ways of rendering its digital materials at a very low cost. The time is right to move forward from the "thinking" to the "doing" and provide the preservation community with well designed, but cost effective tools for the preservation of digital materials.

References

1. The CAMiLEON Project http://www.si.umich.edu/CAMILEON/
2. Testbed Digitale Bewaring: Migration : Context and Current Status (2001) http://www.digitaleduurzaamheid.nl/bibliotheek/Migration.pdf
3. Cedars Guide To : Digital Preservation Strategies (2002) http://www.leeds.ac.uk/cedars/guideto/dpstrategies/
4. Wheatley, P: Migration - a CAMiLEON discussion paper Ariadne **29** (2001) http://www.ariadne.ac.uk/issue29/camileon/
5. Holdsworth, D and Sergeant, D M: A blueprint for Representation Information in the OAIS Model (1999) http://www.personal.leeds.ac.uk/~ecldh/cedars/ieee00.html
6. Holdsworth, D and Wheatley, P: Emulation, Preservation and Abstraction. RLG Dignews **5,4** http://www.rlg.org/preserv/diginews/diginews5-4.html#feature2
7. Consultative Committee for Space Data Systems: Reference model for an Open Archival Information System (OAIS) (2001) http://www.ccsds.org/documents/pdf/CCSDS-650.0-R-2.pdf
8. GFF Format Summary: Microsoft Windows Metafile O'Reilly's Encyclopedia of Graphics File Formats http://www.oreilly.com/centers/gff/formats/micmeta/download.htm
9. (RISC OS) Programmers Reference Manual: Acorn Computers Technical Publications **5** (1994)
10. World Wide Web Consortium: Scalable Vector Graphics (SVG) 1.0 Specification September (2001) http://www.w3.org/TR/SVG/
11. Hedstrom, M and Lee, C: Digital Objects: Definitions, Applications, Implications. Proc 3rd DLM Forum, Barcelona, May (2002) (forthcoming)