# LONG-TERM STEWARDSHIP OF GLOBALLY-DISTRIBUTED REPRESENTATION INFORMATION

**David Holdsworth**
Information Systems Services
Leeds University
LS2 9JT UK
+44 113 343 5401
e-mail: ecldh@leeds.ac.uk

**Paul Wheatley**
Edward Boyle Library
Leeds University
LS2 9JT UK
+44 113 343 5830
e-mail: P.R.Wheatley@leeds.ac.uk

## Background

Leeds was a major participant in three projects looking at digital preservation, viz Cedars [1] (jointly with the Universities of Oxford and Cambridge), CAMiLEON [2] (jointly with the University of Michigan), and the Representation and Rendering Project [3]. With this background, work is beginning on setting up a digital curation centre [4]. for UK academia.

As a result of this work, we strongly favour a policy of retaining the original byte-stream (or possibly bit-stream, see below) as the master copy, and evolving representation information (including software tools) over time to guarantee continued access to the intellectual content of the preserved material. This paper attempts to justify that approach, and to argue for its technical feasibility and economic good sense.

Thus we need long-term stewardship of the byte-streams, and long-term stewardship of the representation information. We use the term *representation information* in the sense of the OAIS model [5]. The purpose of the representation information is to give future access to the intellectual content of preserved byte-streams. Without stewardship of the representation information we would not be exercising stewardship of the preserved data.

## Inevitability of Change in the context of long-term

Since computers were invented in the 1940s and 50s, there have many changes in the representation of data. The binary digit has survived as an abstraction, and in today's world the byte is a world-wide standard, although we sometimes have to call it an *octet*.

All we can be certain of for the long-term future is that there will be further change. However, even though the technology used for representing such bits and bytes has changed over time, the abstract concept lives on. Nonetheless, the uses to which those bits and bytes can be put have grown massively over the years.

Our work has always taken the view that "long-term" means many decades. As digital information technology is barely 60 years old, and we have already lost all of the software from the earliest machines, we need to mend our ways. We should plan that our digital information will still be safe and accessible in 100 years. It is then likely that developments over that time will render the material safe for millennia. In short, we are talking of a time span over which all of our existing hardware technology is likely to be obsolete, and also much of the software.

It is the representation information that makes the bridge between IT practices at the time of preservation, and IT practices at the time of access to the information.

## Abstraction is Vital

We can be confident that the concept of information will survive the passage of time, and even the concept of digital information. We need to bridge the longevity of the information concept to the certain mortality of the media on which the data lives. Our approach is to ensure that everything is represented as a sequence of bytes. We have confidence that the ability to store a sequence of bytes will survive for many decades, and probably several centuries. Current technology usually does this by calling this sequence a file, and storing it in a file system. There are many files in today's computer systems that had their origins in previous systems.

The challenge that remains is to maintain the ability to extract the information content of such byte-streams. The knowledge of the formats of such preserved data is itself information, and is amenable to being represented digitally, and is thus amenable to preservation by the same means as we use for the data itself.

By taking this focus on the storage of a stream of bytes, we divide the problem into two.

1. Providing media for storage, and copying byte-streams from older technology to newer technology.
2. Maintaining knowledge of the data formats, and retaining the ability to process these data formats in a cost-effective manner.

The OAIS representation net is the means by which the knowledge is retained. By treating all data as an abstract byte-stream at the lowest level, we have a common frame of reference in which we can record representation information, independent of any particular data storage technology, and any particular data storing institution. We have a framework in which representation information will be globally relevant.

### Keep the Original Data

We have no faith in long-lived media [6]. Our approach is always to keep the original data as an abstract byte-stream and to regard it as the master.

**Why?** Because it is the only way to be sure that nothing is lost. Format conversion can lose data through moving to a representation incapable of handling all the properties of the original. It can also lose data through simple software error in the conversion process that goes undetected until it is too late to read the previous data.

One of us (DH) has personal experience of both situations. One in which the data was damaged, and one in which potential damage was avoided by keeping the original and producing a format conversion tool.

**How?** We certainly cannot preserve the medium upon which the data is stored. In Cedars we developed the concept of an *underlying abstract form* which enabled us to convert any digital object into a byte-stream from which we could regenerate the *significant properties* of the original. Our approach is to preserve this byte-stream indefinitely, copying it unchanged as storage technology evolves.

The question then remains as to how we continue to have *access to the intellectual content* (another Cedars phrase) of the data, and not merely a stream of bytes. Our answer to this is that we evolve the representation information over time so that it provides us with the means to transform our original into a form that can be processed with the tools current at the time of access. We believe that our work in the CAMiLEON project has shown this to be feasible in the case of a very difficult original digital object of great historical importance. Using emulation we successfully preserved the accessibility of the BBC's "Domesday" project, see below and [16].

The very essence involves identifying appropriate abstractions, and then using them as the focus of the rendering software. We achieve longevity by arranging that the rendering software is implemented so as remain operational over the decades. The application of our approach to emulation is covered in *Emulation, Preservation and Abstraction* [7]. We have also investigated the same technique of retention of the original binary data coupled with evolving software tools in the context of format migration [8].

### Format Conversion — when?

It is obvious that when data is to be accessed some time after its initial collection, the technology involved in this access will differ markedly from that in use when data collection took place. There is also the real possibility that other technologies have been and gone in the interim. Thus, format conversion is inevitable.

For data held in currently common formats, the amount of representation information needed is trivial. Meaningful access to the data normally happens at the click of a mouse.

A current computer platform will render a PDF file merely by being told that the format is PDF. Conversely, faced with an EBCDIC file of IBM SCRIPT mark-up, the same current platform might well render something with little resemblance to the original, whereas back in 1975, the file could be rendered as formatted text with minimal formality.

However, if we have representation information for IBM SCRIPT files that points us at appropriate software for rendering the file contents on current platforms, the historic data becomes accessible to today's users. Alternatively, we could have converted all the world's IBM SCRIPT files into Word-for-Windows, or $L^AT_EX$, or .... We could argue about the choice until all the current formats become obsolete, and we could well have chosen a format that itself quickly became obsolete. We could have been tempted to convert from EBCDIC to ASCII, but that could have lost information because EBCDIC has a few more characters than ASCII.

We recommend that the format of preserved data be converted only when access is required to the data, i.e. on creation of *the Dissemination Information Package* (DIP). For a popular item, it would obviously make sense to cache the DIP, but not to allow the reformatted DIP to replace the original as master. This means that the tracking of developments in storage technology involves only the copying of byte-streams. Moreover, when the format conversion has to be done, there will be improved computational technology with which to do it [9].

### Indirection is Vital

> There isn't a problem in computer science that cannot be solved by an extra level of indirection. *Anon*

The essence of our approach involves keeping the preserved data unchanged, and ensuring that we always have representation information that tells us how to access it, rather than repeatedly converting to a format in current use. We take the view that it is very difficult (impossible?) to provide representation information that will be adequate for ever. We propose that representation information evolves over time to reflect changes in IT practice. This clearly implies a structure in which each stored object contains a pointer to its representation information. This is easily said, but begs the question as to the nature of the pointer.

We need a pointer that will remain valid over the long-term (i.e. 100 years). We need to be wary of depending on institutions whose continued existence cannot be guaranteed.

Alongside this need for a pointer, we also have a need for a reference ID for each preserved object. This needs to be distinct from the location of the object, but there needs to be a service that translates a reference ID into a current location. This is the essence of the Cedars architecture [10].

Reference IDs could be managed locally within an archive store. Such IDs could then be made global, by naming each archive store, and prefixing each local name with that of the archive store.

There are various global naming schemes, ISBN, DNS, Java packages, URL, URI, URN, DOI, etc. It may even be necessary to introduce another one, just because there is no clear long-term survivor. What is certain is that there have to be authorities that give out reference IDs and take responsibility for translating these IDs into facilities for access to the referenced stored objects.

If we grasp the nettle of a global name space for reference IDs of stored objects and keep the representation information in the same name space, we have the prospect of sharing the evolving representation information on a world-wide basis. This will imply some discipline if dangling pointers are to be avoided.

**Enhance Representation Nets over time**

In the Cedars Project we produced a prototype schema for a representation net following the OAIS model, and populated it with some examples. After this experience, we had some new ideas on the schema of the representation net. We believe that it is inevitable that this area is allowed to develop further, and that operational archives are built so that evolution in this area is encouraged to take place. We must accept that there is likely to be revision in the OAIS model itself over the 100 year time-frame.

Also, we could see that to require a fully specified representation net before allowing ingest could act as a disincentive to preservation of digital objects whose value is not in doubt. In many cases, representation information existed as textual documentation. An operational archive needs to be capable of holding representation information in this purely textual form, although with an ambition to refine it later. Such information would not actually violate the OAIS model, but there is a danger of being over-prescriptive in implementing the model. For instance the NISO technical metadata standard for still images [11] has over 100 elements, at least half of which are compulsory.

For some formats the most useful representation information is in the form of viewing software. We need our representation nets to enable the discovery of such software (see below). Many current objects need only to be introduced to a typical desktop computer in order for them to be rendered. On the other hand, we experimented with obsolete digital objects (from 1970s and 1980s) in order to see some of the issues likely to arise when our grandchildren wish to gain access to today's material. We even tried to imagine how we would have gone about preserving for the long-term future using the technology of the 1970s. It was abundantly clear that ideas are very different now than they were 30 or 40 years ago. We must expect that today's ideas could well be superseded over the long-term.

In order to accommodate this, we must allow the content of objects in the representation net to be changed over time, in sharp contrast to the original preserved objects where we are recommending retention of original byte-streams. It is vital that the reference ID that is originally used for representation information is re-used for newer representation information which gets produced as a result of development of new tools and ideas. That way, old data gets to benefit from new techniques available for processing it. The representation information that is being replaced should of course be retained, but with a new ID, which should then be referenced by the replacement.

**Representation Nets should link to software**

Our representation nets in Cedars very deliberately contained software, or in some cases references to it. We have no regrets on this issue. Ideally we want software in source form in a programming language for which implementations are widely available, but it seems churlish to refuse to reference the Acrobat viewer as a way of rendering PDF files, just because we do not have the source, but see example 1 below.

A format conversion program that is known to work correctly on many different data objects is clearly a valuable resource for access to the stored data, and should be available via the representation network.

As regards the issue of longevity of such software, we argued earlier for the longevity of abstract concepts such as bits, bytes and byte-streams. Programming languages are also abstract concepts, and they too can live for a very long time. Current implementations of C or FORTRAN will run programs from long ago. Other languages which have been less widely used also have current implementations that function correctly.

The source text of a format conversion program which is written in a language for which no implementation is available is still a valuable specification of the format, and has the benefit of previously proven accuracy. We address the issue of evolving emulator programs in *C-ing Ahead for Digital Longevity* [12], which proposes using a subset of C as the programming language for writing portable emulators.

**Examples**

We illustrate the way in which we see representation information evolving over time, by reference to three examples drawn from rather different computational environments.

**Example 1: Acrobat files**

In today's IT world it is very common to use Adobe Acrobat® portable document format (PDF) for holding and transmitting electronic forms of what are thought of as printed documents. The only representation information needed by today's computer user is the URL for downloading the Acrobat® Reader™. The representation net for PDF files is basically this single node, detailing how to gain access to the software for rendering the data. In reality, it should be an array of nodes with elements for different platforms. All preserved PDF files would reference this one piece of representation information. The recent appearance of the GNU open-source Xpdf [13] would be reflected by adding it to this array.

**Example 2: IBM SCRIPT files**

One upon a time, the representation information for a preserved IBM SCRIPT file would point to the IBM SCRIPT program for the IBM/360 platform. Unfortunately we did not have the OAIS model in the 1970s, but if we had had an OAIS archive for storage of our VM/CMS data, this is the only representation information that would have been needed. (Actually the CMS file-type of SCRIPT performed the rôle of representation information, much as file extensions do today on a PC.)

As the 30+ years elapsed, our putative OAIS archive would have expanded the representation information for SCRIPT by information suitable for more current platforms — including the human readable documentation for a live-ware platform. There would probably also be reference to the Hercules project [14] which allows emulation of IBM/360/370 systems of yesteryear. This need to keep up-to-date was highlighted in the InterPARES project [15].

**Example 3: The BBC Domesday Project**

In 1986, to commemorate the 900th anniversary of the Domesday Book, the BBC ran a project to collect a picture of Britain in 1986, to do so using modern technology, and to preserve the information so as to withstand the ravages of time. This was done using a micro computer coupled to a Philips LaserVision player, with the data stored on two 12" video disks. Software was included with the package, some on ROM an some held on the disks, which then gave an interactive interface to this data. The disks themselves are robust enough to last a long time, but the device to read them is much more fragile, and has long since been superseded as a commercial product.

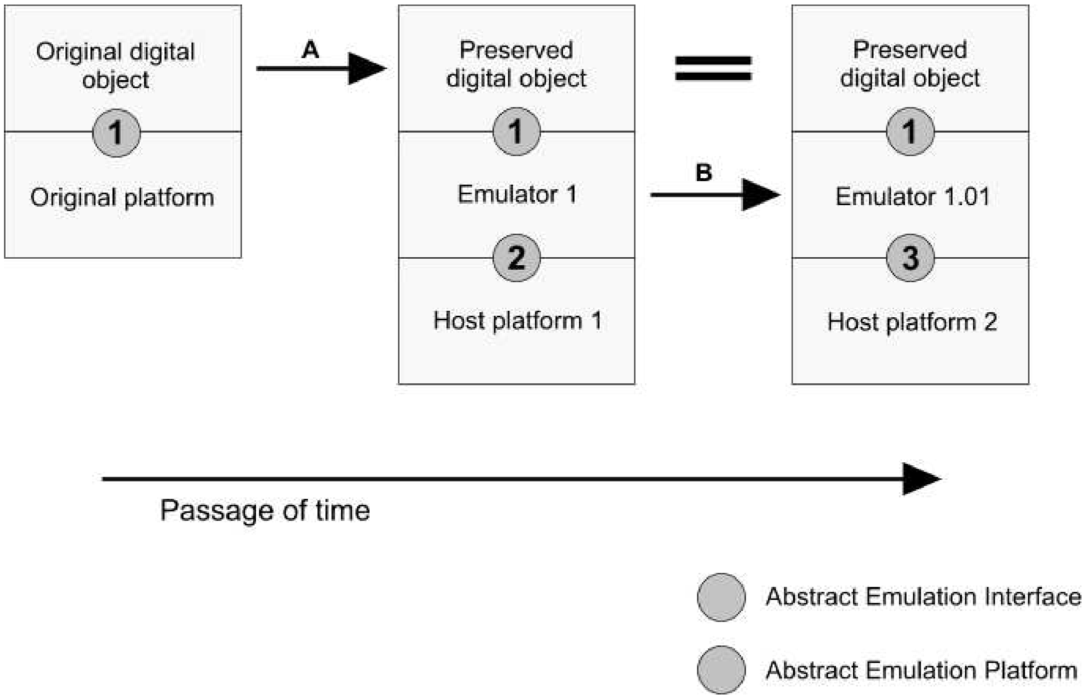
Here we have a clear example where the preservation decisions placed (mis-placed) faith in the media technology of the day, and more crucially in the survival of the information technology practices of the time.

The CAMiLEON project used this example as a test case to show the effectiveness of emulation as a preservation technique. A detailed treatment is to be found on the CAMiLEON web site [16].

We can look at this example with particular reference to its long-term viability, both with regard to the original efforts in 1986, and to the emulation work of 2002. We shall use it to illustrate our ideas about the appropriateness of emulation software as part of the representation information.

Firstly, a bit of background to the work.

We have taken our own advice and preserved the data from the original disks as abstract byte-streams. We can represent this step as the process marked A in the diagram (taken from reference [7]):



The technique was to show that we could use emulation to bridge from the Original platform to a different host platform, labelled Host platform 1 in the diagram. The ingest step (marked A in the diagram) involves identifying the significant properties of the original. The data consisted of the four disk surfaces, each with 3330 tracks, and some software in ROM held inside the BBC micro computer. Some tracks were video images and some held digital data which was often textual. We preserved the ROM contents straightforwardly as binary files, and made each track of the disk into a binary file of pixels for the video images, and a straightforward binary file for each of the digital data tracks. This we claim preserves the significant properties of the software and data necessary for it to run on the BBC computer with its attached video disk player. An example representation network describing the capture process was constructed as part of the Representation and Rendering Project [17]

To demonstrate the validity of this claim, we produced the emulator shown as Emulator 1 on the diagram. The original software relied on an order code and an API (applications program interface) labelled 1 in the diagram. In order to achieve successful preservation of this digital object, we need to reproduce this API with software that operates with a more modern API, labelled 2 in the diagram.

The emulation of the BBC micro-computer was obtained from an open-source emulation written by an enthusiast (Richard Gellman) and available on the net [18]. Although the achievements of enthusiasts are not always ideally structured for use in digital preservation work, they can often provide a useful starting point for further development. At the very least the source code can act as a handy reference point for new work.

The emulation of the video disk player was done by our own project staff. This emulation software then becomes the major component of the representation information for this data. Its longevity depends crucially on the longevity of the interface labelled 2. Here we have used code that is written in C, and makes use of only a few Win32-specific API calls. In other words our interface labelled 2, is not the whole API of Host platform 1, but only the facilities that we have chosen to use. The move to another platform is made easier by choosing to use as few as possible of the proprietary features of Host platform 1. We may need to recode a few bits of the screen driving routines, but by and large we can expect to find on Host platform 2 an API (shown as 3) that has most of the features needed on the new platform. We expect that a slightly revised emulator called Emulator 1.01 will readily be generated (step B) to run on Host platform 2. Meanwhile, the preserved digital object will be completely unchanged, as indicated by the large equals sign.

**Example 3: The BBC Domesday Project — Evolution of Representation Information**

At the outset, the storage media consisted of two 12" video disks. The representation information (a booklet supplied with the disks) basically said buy the appropriate hardware including the two E-PROM chips holding software that is used in accessing the video disk player. In addition, the BBC microcomputer had a well documented API for applications programs. This API (or preferably the subset of this that happened to be used) provides the interface labelled 1 in the diagram.

Our preservation of the data from its original preservation medium created byte-streams that closely mirrored the actual physical data addressing. This maximised the validity of the existing representation information, *viz.* the documentation of the API mentioned above.

The emulator then implements this API, opening up the question of the API upon which it itself runs. Thus we add to the representation information the emulator, and the information concerning the API needed to run it. This is not yet stored in a real OAIS archive, but we do have the materials necessary to achieve this, and data from the disks is stored in our LEEDS archive[19].

Our care in producing an emulation system that is not tied too closely to the platform upon which it runs illustrates our desire to produce representation information that will indeed stand the test of time by being easily revised to accommodate newly emerging technologies. This revised emulator becomes an addition to the representation information, extending the easy availability of the original data to a new platform. InterPARES [15] identified clearly the desire of users to access the material on the technology of their own time.

So why emulate in this case? The interactive nature of the digital object is really a part of it. There is no readily available current product that reproduces that interaction, so we treat the interaction software as part of the data to be preserved. On the better examples of current desk-top hardware, it runs faster than the original.

### Share and Cross-Reference Representation Nets

We have argued earlier for the impossibility of producing an adequate standard for representation information which will retain its relevance over the decades. To attempt to do so would stifle research and development. We must therefore expect that different data storage organisations may develop different forms of Representation Information. Initiatives such as the PRONOM [20] file format database and the proposed Global File Format Registry will also produce valuable resources that should be linked from representation information.

It would seem that collaboration should be the watchword here.

The emerging solutions for IBM SCRIPT files in example 2 are likely to be applicable to any institution holding such data. With our proposed global namespace, they can all reference the same representation net, and benefit from advancing knowledge on the rendering of such files.

### Global Considerations

The implementation of preservation on a global basis means that there will be no overall command. Co-operation will have to be by agreement rather than by diktat. This situation has some aspects that resemble the problems of achieving true long-term preservation. We cannot predict the future accurately, nor can we control it to any great extent, so the ambition to operate on a global scale despite being unable to control activities everywhere in the world sits well with the need for future-proofing. The future is another country whose customs and practices we cannot know.

### Referential Integrity

We are proposing that no object that has a name in the digital store is ever deleted. It may be modified, but never deleted. Thus, anyone may use a reference to an object in the OAIS digital storage world confident that it will never become a dangling pointer.

However, the representation information in any OAIS archive will need to refer to information outside its control. (This is actually an inevitable consequence of Gödel's incompleteness theorem — reflected in Cedars by describing nodes holding such references as Gödel ends.) Many of these external references will relate to the current practice of the time.

A vital part of the management of such an archive will involve keeping an inventory of all such external references, and maintaining a process of review of the inventory in the search for things that are no longer generally understood or refer to information that is no longer available. The remedy in such cases is to update the referring nodes to reflect the new realities. Clearly it is in the interests of good management to try to keep such nodes to a minimum.

For example, a store would have a single node that describes the current version of Microsoft Word to which the representation information for any ingested Word file would refer. When this version becomes obsolete, this one node is updated with information on how to access data in the old format, or to convert to a newer format.

The two level naming proposed earlier helps greatly in implementation of such a policy.

## Digital Curation in the UK

The education funding authorities in Britain are currently in the process of setting up a digital curation centre [4]. This is seen as a centre for oversight and co-ordination of digital storage, and for R&D. The decision was announced shortly before Christmas. The centre will be based in Edinburgh, the home of the existing e-Science Centre [21], and EDINA [22].

The centre will not be a repository for the data itself.

It will provide consultancy and advice services, and a directory of standard file formats.

There will be a significant research activity, and a particular focus on digital integration, the enabling of research combining data from different sources.

Academia is addressing its own problems, but what about the rest of the world of digital information, e.g. engineering data? How confident are we that the CAD data for nuclear power stations has an appropriate lifetime, or even half-life?

## Summary

We argue strongly for retention of the original in the form of a byte-stream derived as simply as possible from the original data, and for the use of representation information to enable continued access to the intellectual content.

We take the view that for much material it is impossible to have perfect representation information at the time of ingest, but that we must preserve the data and develop its representation information over time.

Ideas on the nature of representation information will evolve over time. We must have systems capable of taking on board changing schemas of representation information.

A two-level naming system, separating reference ID from location (and translating between them) should be the practice for implementing pointers in an OAIS archive, as a prerequisite for our proposed policy of evolving representation information over time, and sharing it on a global scale.

## A Footnote on Bits versus Bytes

The OAIS model uses the bit as the lowest level. However, the byte is the ubiquitous unit of data storage. In today's systems one cannot see how the bits are packed into bytes. When a file is copied from one medium to another we know that whether we read the original or the copy, we shall see the same sequence of bytes, but we know nothing of the ordering of bits within the byte, and these may be different on the two media types. On some media (e.g. 9-track tape) the bits are stored side-by-side.

Pragmatically, we regard the byte as the indivisible unit of storage. If the OAIS model requires us to use bits, then we shall have a single definition of the assembly of bits into a byte. This would enable us unambiguously to refer to the millionth bit in a file, but not constrain us to hold it immediately before the million-and-oneth bit.

## References:

[1] Cedars project http://www.leeds.ac.uk/cedars/
[2] CAMiLEON project http://www.si.umich.edu/CAMILEON/
[3] Representation and Rendering Project http://www.leeds.ac.uk/reprend/
[4] UK National Digital Curation Centre http://www.jisc.ac.uk/index.cfm?name=funding_digcentre
[5] Reference Model for an Open Archival Information System (OAIS) ISO 14721:2002: http://www.ccsds.org/documents/pdf/CCSDS-650.0-B-1.pdf
[6] The Medium is NOT the message. *Fifth NASA Goddard Conference on Mass Storage Systems and Technologies* NASA publication 3340, September 1996. http://esdis-it.gsfc.nasa.gov/MSST/conf1996/A6_07Holdsworth.html
[7] Emulation, Preservation and Abstraction, RLG DigiNews vol5 no4, 2001. http://www.rlg.org/preserv/diginews/diginews5-4.html#feature2
[8] Research and Advances Technology for Digital Technology : 6th European Conference, ECDL 2002 http://www.springerlink.com/openurl.asp?genre=article&issn=0302-9743&volume=2458&spage=516
[9] *Migration - A CAMiLEON discussion paper —Paul Wheatley, Ariadne, Issue 29(September 2001)* http://www.ariadne.ac.uk/issue29/camileon/
[10] Cedars architecture. http://www.leeds.ac.uk/cedars/archive/architecture.html
[11] NISO technical metadata standard for still images. http://www.niso.org/committees/committee_au.html
[12] *C-ing Ahead for Digital Longevity* http://www.leeds.ac.uk/CAMiLEON/dh/cingahd.html
[13] Xpdf Acrobat® renderer http://www.foolabs.com/xpdf/about.html
[14] Hercules IBM Emulator http://www.schaefernet.de/hercules/index.html
[15] InterPARES project http://www.interpares.org/book/index.cfm, see also [23
[16] Domesday. http://www.si.umich.edu/CAMILEON/domesday/domesday.html

[17] Representation and Rendering Project case study
http://www.leeds.ac.uk/reprend/repnet/casestudy.html
[18] Richard Gellman and David Gilbert, BBC Emulator
http://www.mikebuk.dsl.pipex.com/beebem/
[19] LEEDS archive http://www.leeds.ac.uk/iss/systems/archive/
[20] PRONOM http://www.records.pro.gov.uk/pronom/
[21] EDINA http://www.edina.ac.uk/
[22] UK National e-Science Centre http://www.nesc.ac.uk/
[23] InterPARES2 http://www.interpares.org/ip2.htm