TOWARD A WORKABLE EMULATION-BASED PRESERVATION STRATEGY: RATIONALE AND TECHNICAL METADATA

DAVID ANDERSON, JANET DELVE, and DAN PINCHBECK

Humanities Computing Group, School of Creative Technologies, University of
Portsmouth, Portsmouth, UK

The KEEP project is the first of its kind to seriously research the kind of emulation based on a virtual machine as put forward by Lorie (2002). In addition to creating such a virtual machine, a number of other supporting tools and techniques are also being developed as part of this EC FP7 project. One of these is an emulation metadata data model with a dual purpose: first, for use as the basis of a database that forms part of the Emulation Framework that will run on the KEEP Virtual Machine (KVM); and, second, for use as the core of an emulation metadata standard, envisaged to be taken up by the wider community. This paper is thus very much geared toward a practical discussion of emulation. However, before the digital preservation community will consider emulation as a viable option compared to migration; it is imperative that the polarized positions exemplified by Rothenberg and Bearman are carefully analyzed, deconstructed, and, where necessary, set aside. In this way, some options that have previously been dismissed out of hand can be allowed to resurface, and their relative merits be reconsidered. The second part of the article comprises a detailed investigation of the technical environment necessary to emulate a given digital object. The technical environment data, thus obtained, is then used to create the core of the emulation metadata model. The article concludes with a consideration of video games' metadata, as games represent the most complex digital objects planned to be emulated as part of the KEEP project.

Introduction

Digital “objects” are, in many ways, a curious product of the 20th century quite different from any of the objects that preceded

The KEEP project is co-financed by the European Commission with contract No. IST-231954. The views expressed in this article do not represent the opinion of the European Community, and the European Community is not responsible for any use that might be made of them.

Address correspondence to David Anderson, Humanities Computing Group, University of Portsmouth, School of Creative Technologies, Portsmouth, Hampshire PO5 4BP, UK. E-mail: david.anderson@port.ac.uk

them. Their immaterial nature gives rise to diverse digital preservation issues, requiring substantial changes to be made to preservation practice and demanding significant alterations to the way we think about the nature of objecthood in the digital context.

We depend on documents to carry messages through space and time. In many cases, this reliability is achieved through fixity: letterforms inked on paper can survive for long periods of time. But with newer media, such as video, this reliability is achieved not by fixity but by repeatability. The moving images on a video screen are by their very nature transient. I will never be able to see those very images again. But I can play the tape repeatedly, each time seeing a performance that, for all practical purposes, is "the same as" the one I saw the first time (Levy, 2000).

Digital objects are not capable directly of human creation or subsequent access but require one or more intermediate layers of facilitating technology. In part, this comprises further digital objects; software such as a BIOS, an OS, or a word processing package, and, in part, it is mechanical, a computer. Even a text file (ASCII format) has a series of relationships with other digital and physical objects from which it is difficult to isolate it completely. This complexity and necessity for technological mediation exists not only at the time when a digital object is created but is present on each occasion when it is edited, viewed, preserved, or otherwise interacted with. Furthermore, the situation is far from static as each interaction with a digital object may bring it into contact with new digital objects (a different editor for example) or new physical technologies.

Given the underlying complexity of the issues involved, it is perhaps surprising that within the digital preservation community only two approaches to preservation are frequently discussed: migration and emulation. The degree to which opinions appear to be polarized between these two strategies (Bearman, 1999; Rothenberg, 1999; Stawowczyk Long and Pearson, 2009), neither of which can claim to be a complete solution, is perhaps indicative of the extent to which the fundamental issues of long term digital preservation have not yet been addressed (Stawowczyk Long, 2009). The arguments and "evidence" offered on both sides of the debate are often far from convincing.

One of the things about which there is some measure of agreement is that computer museums have little role to play in

digital preservation. The truth of this observation seems almost to be taken for granted and little effort is expended on providing detailed justification for what seems to us to be a very questionable position. Other communities take a very different view. For example, during the IFIP WCC2010 conference,¹ there was widespread recognition of the continuing value of computing museums, including an endorsement of the role they have to play in digital preservation (e.g., Ainsworth, Avram, and Sheard, 2010; Demant, 2010).

Rothenberg (1999) does present various concerns about computer (hardware) museums but they essentially come down to a single point: hardware (including media) will deteriorate over time to the stage where old machines will not be able to access the software written for them. The inevitability of physical deterioration of systems will not come as a surprise to anyone familiar with the 2nd Law of Thermodynamics.² Rothenberg does concede two “limited roles” for computer museums: performing “. . . heroic efforts to retrieve digital information from old storage media” and verifying the behavior of emulators. Rothenberg’s characterization of these roles as “limited” notwithstanding, they are, in fact, essential activities; the first is arguably the only way in which future generations will gain access to some important historical material which would otherwise be lost, and the second is vital if we are to verify that the behavior of emulated computer platforms is faithful to the original. It is crucial to establish if, for example, some unexpected behavior exhibited by a digital object is the result of a defect introduced during preservation or was originally present.

Bearman, writing about computer (software) museums, concedes that:

. . . by documenting standards and widespread operating functions, software archives preserve a record of the fundamental structures of the software environment which will contribute to future understanding of more specialized software (Bearman, 1987).

Similar considerations apply equally independently of which digital preservation approach is preferred.

¹September 20th-23rd 2010 Brisbane, Australia

²The 2nd Law of Thermodynamics asserts the universal principle that absolutely everything decays.

All computers have undocumented features, and preserving original hardware in working condition for as long as reasonably possible is an important aspect of digital preservation. It is not possible to preserve computer systems forever, which only lends urgency to the need to gather as much information as possible from machines while they are available. The cost of funding computer museums, particularly when viewed from the national or international perspective, is not high. Rothenberg somewhat mistakes the position when he asserts that:

It is unlikely that old machines could be kept running indefinitely at any reasonable cost, and even if they were, this would limit true access to the original forms of old digital documents to a very few sites in the world, thereby again sacrificing many of these documents' core digital attributes (Rothenberg, 1999).

The fact that preserving machines in working order is subject to the law of diminishing returns is a powerful reason why we should endeavor to make the best use of old machines while it is still feasible.

Computer museums have a continuing role, broadening their collections as more and more machines become obsolete. Their custodial remit should run further than simply maintaining the oldest machines in their care. For each machine in a museum's collection, there will come a point when the device can, at best, be preserved in a non-working condition. This does not diminish the importance of computer museums, which could and should become repositories of collective knowledge like any other memory institution. The 2nd Law of Thermodynamics is unavoidable and if arguments against computer museums based on it were sound, they would apply with equal force to every sort of museum or library; all statues will eventually crumble and every book will, given sufficient time, turn to dust. Despite this, memory organizations, including computer museums, have a major role to play in preserving our cultural, technological, and scientific heritage.

Emulation or Migration?

The major proponents of emulation and migration often express themselves in quite stark terms:

In the long run, migration promises to be expensive, unscalable, error-prone, at most partially successful, and ultimately infeasible (Rothenberg, 1999).

[an] “emulation” approach does not adequately address the problems of maintaining electronic records, won’t work as a strategy, and may encourage potentially dangerous wishful thinking (Bearman, 1999).

There are a number of questionable assumptions that appear to underpin this disagreement. There is also a widespread failure to recognize the difference between logical possibilities and practical realities. For example, it is logically possible to migrate complex interactive games piecemeal from platform to platform and for this process to be extended indefinitely far into the future. However, the practicalities involved are such that they exclude from serious consideration the idea that the digital preservation community will actually accomplish this. A solution is required that permits as many as possible of the digital objects dependent on a particular obsolete hardware platform to be made available “en masse” on the replacement platform. This could be by means of emulation or by other means (Gladney, 2008). Piecemeal migration is not the answer.

Emulation offers the logical possibility of perfectly reproducing the behavioral characteristics of an obsolete hardware platform on a current system. Something of an unspoken, but critical, issue at the heart of digital preservation concerns input-output peripherals. Many of these hardware devices have been produced, both third-party and generic, and custom made. They range from the relatively simple, such as mouse/keyboard, to the complex, such as one-off devices like instruments for Rock Band (Harmonix Music Systems, Cambridge, MA, 2008) or Guitar Hero (Harmonix Music Systems, Cambridge, MA, 2006). We are also entering the era of touch screen and natural interfaces such as the Wiimote, EyeToy, and Move controllers. All of these present enormous challenges that remain, as yet, unexplored in the literature. I/O devices have a significant bearing on the experiential character of using computer systems, preserving access to which is important, not least, because it casts light on the cause for motivation in subsequent developments. Some areas, such as the Amiga demo scene, cannot be understood without an appreciation of the limitations (and undocumented features) of the original hardware system. This sort of socio-computational information can easily be lost.

The notion, whereby emulation and migration are competing and independent approaches, represents something of a professional mythology which ignores the fact that emulators are, intrinsically, digital objects. When a hardware paradigm shift occurs there are just two strategies that an emulation-based digital preservation approach can follow:

1. Produce from scratch a new emulator (or emulators) to run on the latest hardware to provide access to digital objects dependent on previous platforms; or
2. Migrate the existing emulators.

Developing emulators is an extremely complex and time-consuming activity demanding substantial human and technical resources. In cases where it is proposed to produce an emulator that reaches back over several hardware paradigm shifts, there are extra problems to consider. The greater the distance in time from the original hardware platform to the emulator which seeks to recreate it, the greater is the likelihood that the knowledge required to produce it will be unavailable. Computer museums have a role to play here, but the danger remains that given enough time it will be impractical to produce from scratch new emulators for the oldest hardware platforms. In either case, we are presently not in a position to feel justifiable confidence that an emulation-based digital preservation strategy predicated on the idea that new emulators will be produced for each of the target hardware platforms every time there is a new hardware paradigm shift is a practical possibility. From today's perspective, this looks every bit as impractical as attempting piecemeal migration of complex digital objects.

The KEEP approach is a hybrid representing a form of emulator migration. We aim to develop a virtual machine as the platform on which emulators will be written (or ported) to run. This virtual machine is designed, in such a way, as to be migratable without difficulty to any conceivable hardware platform on the future. Thus, we plan to ensure that emulated environments, once written, can be kept portable. We regard the notion that emulation and migration are diametrically opposed approaches as a false dichotomy preferring to see

migration and emulation as complementary and mutually supportive. A future in which emulation completely displaces migration is not one that we take seriously.

Significant Properties

Migration purports to concentrate on the information content of the digital object itself, thus, preserving future access to that content despite constantly changing technology. Underlying this, is the very Socratic notion of “significant properties” that was developed in the CEDARS project and built on by the InSPECT project (Knight and Pennock, 2009) and the British Library (Dappert and Farquhar, 2009). Significant properties purport to represent the very essence of a digital object: its intellectual content. The argument states that if the significant properties of a digital object are retained, the object’s intellectual content will have been preserved in spite of any “superficial” changes in form or appearance. A property of a digital object that is not held to be significant can be ignored.

This presupposes the information content of digital objects is both fixed and discernable and remains intact despite the changes to form and structure necessitated by migration processes. However, there is little reason to suppose that any of this is true in general. Abundant examples exist showing that meaning and significance changes over time and place. Objects that were originally seen as having little significance develop much greater importance for future generations. For example, the Rosetta Stone, which at the time of its construction in 196 BCE, served to record a decree issued at Memphis on behalf of Ptolemy V, came in the 19th Century to have a different (and much greater) significance in unlocking our understanding of Egyptian hieroglyphics. The Rosetta Stone should serve to remind us, not only that the significance of objects changes over time, but that the information content of objects is a derived rather than an intrinsic property.

For digital objects, the situation is more complex because merely preserving the bitstream of original object will not assure future generations’ access to the informational content, unless further steps are taken to protect some route back to the original technology platform.

The Emulation Metadata Model

Even under favorable conditions, post hoc determination of suitable hardware environments to access digital objects can be tricky. Therefore, it is imperative to store appropriate environment metadata at the time of ingest into a preservation system. When this has not been done, external technical registries (e.g., PRONOM, UDFR) may be of use, but the process is far from simple.

For example, PRONOM³ carries information covering fifteen different versions of the .pdf format. In addition to this basic description, further information is available. In terms of software, there are multiple pieces of information in the “Related file formats” field, providing priority and previous version information. In terms of hardware, while there is currently nothing in the “Technical Environment” field, a hint is given of a computer hardware relationship/dependency; the “Byte order” field indicates the file is “Big-endian (Motorola).” The “(Motorola)” reference here is to the historical difference in hardware approach between Motorola and Intel. However, this “endianness” does not appear to impose any real restriction on the hardware in terms of current emulation practice,⁴ and so is of historical value only. However, as we cannot know what decisions might be made regarding future emulation practice, it is helpful that it is recorded in this way. It is noteworthy that PREMIS refers to byte order as being hardware dependent (big-endian vs. little-endian), and recommends that it should be included as technical metadata (OCLC, 2006, pp. 4–9).

The basic description given for the ninth .pdf version; Acrobat PDF/X - Portable Document Format - Exchange 1:1999, shows this format incorporates a “Graphics Art Technologies”

³See <http://www.nationalarchives.gov.uk/PRONOM/Format/proFormatSearch.aspx?status=detailReport&id=613> [accessed 22nd March 2010].

⁴“Hardware specificities such as endianness of the processor have to be taken into account during the process of developing software (e.g. a PDF reader). When the software is finished and works, the technical details pertaining both to the hardware platform and to the file format that is handled by this software are irrelevant to its exploitation to render files. Indeed, endianness is only one aspect of computer hardware and software in general, and much more technical information is required to develop a software to interpret such a complex fileformat as PDF”. Comment by Vincent Jogun, Jogun SAS, 9/4/2010 (Personal Communication with Janet Delve).

standard. Note that for this version, text, and so forth, can no longer be included in the document. Hence, the .pdf versions cannot be used interchangeably. The “Related file formats” field now presents previous version and subtype information; whereas, there is no mention of “Big-endian (Motorola)” in the Byte order field.

A further search of PRONOM reveals which software is currently available with which to run the .pdf file, and this is broken down by file type version and software version. (Note that according to PRONOM there is currently no way of implementing the PDF/X versions—it is necessary to search the Acrobat site in order to ascertain this.) Taking one of these, CorelDRAW 11, we find that for this software, the Media format field is specified as unknown, and the Operating systems, Software requirements, and Hardware requirements fields are all blank.

What information concerning the technical environment might an archivist, a librarian, or a library technician need to establish the contents of this .pdf file to determine a suitable technical environment in which to run it?

- Precisely, which version of .pdf is this file?
- Are there any hardware issues due to byte order?
- Which version of software will run this file type version?
- Which operating system is needed?
- What hardware is needed?

These questions give an inkling of the process necessary in order to establish the technical environment required to emulate each digital object (and by extension, each category of digital object). An initial attempt at a data model would comprise several “many-to-many” relationships; for example, many filetype versions run on numerous software versions. What is required is, at least, one pathway through this technical maze; for example, providing available software versions for a given filetype. An example of a possible pathway⁵ is a .pdf version 1.4 file that would run on

⁵In practice, the pathway should provide more extensive detail to include: software patches; system libraries; plug-ins; fonts; plugged hardware devices with their corresponding drivers (together with driver versions); BIOS revision; network/internet connectivity; and so on.

Adobe Acrobat 5.x software on a Mac OS X 10.5.6 (9G55) operating system on a hardware platform in the form of a MacBookPro 5,1.

Finally, it is worth noting that .pdf files are covered by an Adobe XMP metadata standard: the eXtensible Metadata Platform. A .pdf file contains an advanced metadata section detailing XMP core properties, PDF properties, Dublin Core properties (DCMI, 2003), Adobe properties, and, lastly, TIFF properties. Hence, the archivist/librarian could have found information regarding the file version and software on which to run it among this metadata. JHOVE (JournalSTORage / Harvard Object Validation Environment)⁶ is another characterization tool that could be used to identify file type/format in addition to PRONOM / DROID (Digital Record Object Environment).⁷ A characterizer has also been developed as part of the Planets project (Thaller, 2009).

Figure 1 gives an initial basic conceptual data model with the Entities shown as rectangles and Relationships as diamonds. Attributes are given as ovals in Figure 2 and Figure 3. An Enhanced Entity Relationship Diagram (EERD) is used to incorporate additional semantic concepts (Connolly and Begg, 2005, p. 371). In this figure, the many-to-many relationships have been properly decomposed to provide pathways, and a Library entity and a Library Version entity are included to allow for the use of different OS and Software libraries.

Of course, it should be said that automatic systems such as Kopal and e-Depot do exist to migrate static files such as .pdfs (Anderson, Delve, Pinchbeck, and Alemu, 2009). The workflows for such migration systems are well-defined and streamlined, so the process described previously does not represent an improvement on current practice.

Figure 2 provides an expansion of the hardware architecture type, showing some of the core details needed for two architecture types: the PC, and the Apple II. Figures 1 and 2 are suitable for modeling video games that run on a computer, but a slightly

⁶<http://hul.harvard.edu/jhove/>

⁷DROID (Digital Record Object Identification) is a software tool developed by The National Archives to perform automated batch identification of file formats. See: <http://www.cctmark.gov.uk/CCTMAwards/TheNationalArchives/tabid/105/Default.aspx>

FIGURE 1 Emulation Metadata Model with Pathways and libraries.

FIGURE 2 Emulation Metadata Model for Hardware Architecture Type.

different model is required for console video games, as shown in Figure 3, where the console contains an integrated operating system. In these three figures, the digital object is referred to as the more generic digital publication.

Semantic interoperability via linked data clouds is strongly advocated by Dublin Core (DCMI, 2003). But, how could the linking data set cloud be useful to KEEP in practice? For example, technical data needed to run the computer game “World of Warcraft” can be found on the data cloud: http://dbpedia.org/page/World_of_Warcraft. This data will help to support the task of filling the KEEP metadata database with technical data for emulation. Data will undoubtedly become available in this way on software versions, OS versions, library versions, and so forth. While recognizing, in the long-term, that a truly portable emulation system cannot depend on the Internet, in the short to

FIGURE 3 Emulation Metadata Model for Console Game.

medium term, it would seem a sensible strategy to consider using these linked data clouds, as outlined in terms of Dublin Core formal semantic interoperability, to obtain the required data for the emulation metadata database and to inform the development of the emulation metadata standard.

Other helpful initiatives have been highlighted in (Anderson et al., 2009). MobyGames (pp. 4, 61–65) is a community-driven dataset with extensive descriptive metadata and limited technical metadata that could perhaps be expanded in cooperation with the games’ user community in order to capture the technical environment metadata necessary for KEEP. Initiatives such as German games publishers sending their metadata to a central computer games metadata registry database in order to obtain their games age rating (p. 60) could be expanded as necessary to include all necessary technical metadata and copied elsewhere. Other developments in games metadata are now considered in greater depth.

Issues for Metadata Developed for Game Preservation

The International Game Developers Association (IGDA) Special Interest Group on Preservation recently published a white paper detailing many of the specific issues facing game preservation. These include rapid obsolescence of both soft and hardware: media decay, access to material, legal constraints, lack of understanding or impetus, breadth of material surrounding any given title, and loss of cataloguing/descriptive information. They can be broadly grouped into three major problem areas: legality, cultural complexity, and technical complexity. All of these require representation in any metadata schema aimed at the robust preservation of games. They will be dealt with here in reverse order.

Migrating the quantity of code required to ensure runtime viability of a modern game is simply impractical (Pinchbeck et al., 2009). Dondorp and van der Meer (2003) conclude that of Quake (id Software 1996) that “Rebuilding such a game is a gruesome operation that might easily compare to the complexity of emulation of the computing platform. To preserve highly interactive objects such as games, emulation is probably the only solution” (p. 40). It is worth noting that they are referring to a game nearly 15 years old and, thus, comparatively simple by contemporary standards.

Guttenbrunner (2008) argued that emulation at a hardware level is the ideal solution for console games. The split of emulators into hardware and software is one also made by Tijms (2000) and by Conley et al. (2004), who described a “window of opportunity” created by the lag in new console developments and PC hardware improvements that may even allow predictions to be made of when emulation technology will emerge for a given platform.

The technical issues surrounding preserving games do not stop at code complexity. Subtle aspects such as minor alterations in processing speed can affect qualitative experience. Tijms (2000) noted there is not a direct correlation between the actual processing power of an emulated system and the requirements of the emulating system, due to specificities and peculiarities of the underlying hardware components of the former. Many games include middleware and DirectX components, and an increasing number rely on internet connections for patches, updates, and authorizations, some even requiring this to be active during play. Online multiplayer and LAN gaming all present further technical challenges as, in these instances, it is not

only a client-side emulation that is required, but emulation of the outlying environments with which the game may engage.

Equally problematic are the additional digital objects that develop around a commercial game. Lowood recognizes this when he states that “Capturing the history of community-generated content and the mod scene is a huge challenge, and it will require special attention to the variability and modifiability of software, including provisions for carefully documenting version history through metadata” (Lowood 2004, p. 5). We need to preserve not just a first or final version of an object, but its evolution through official patches and updates. But, as well as these official add-ons, we need to ensure we are also ingesting unofficial fan-community work. To put this into context, we should remember perhaps that *Counter Strike* (Valve Software, Bellevue, WA, 2005), recognized as an important game in the history of online multiplayer shooters, spent its early life as a fan-community mod⁸ of the commercial game *Half Life* (Valve Software, Bellevue, WA, 1998). Likewise, Media Molecule’s *Little Big Planet* (2008) is less a game than an engine for the construction and sharing of user-generated content. More discussion of associated objects can be found in Pinchbeck et al. (2009) and Lowood et al. (2009).

Barwick (2009), Lowood et al. (2009), Gieske (2002), and others have begun the process of understanding why games have been ignored by preservationists for so long, but at least the need is now generally recognized. However, the relatively late realization of what Gooding and Terras (2008) call the “current preservation crisis” facing games means that we are left with the task of either retrofitting the specific preservation metadata, or creating new metadata structures that can be easily assimilated into existing schema. This introductory discussion makes clear that this specific metadata falls into three major areas not necessarily covered by existing schema:

- technical metadata is required to describe the original runtime environment, middleware, add-ons, and aspects of performance

⁸Mods (or modifications) are common in the world of PC games. They can be quite extensive more or less constituting entirely new games in themselves, but require the user to have the original game release in order to run. They typically include new items, weapons, characters, enemies, models, textures, levels, story lines, music, and game modes.

that may affect the experiential qualities of the game, such as sound capabilities, processor speed, data delivery from disc in run-time, video outputs, I/O devices, and so forth.

- Additional metadata and a robust network of associated objects (both games and other digital objects, such as video, audio, code, image and text) may be required to capture the cultural experience of a given game.
- Legal information clearly describing the status of the disc image, runtime rights, and potentially legal status of embedded third-party requirements or associated material may be important given the highly charged nature of the field.

Game Specific Preservation Metadata Schema

At this time, only Huth's (2004) thesis⁹ has attempted to develop a systematic, game specific schema. Current metadata on games tends toward simple cataloguing information, both in repositories and commercial/community sites such as MobyGames. While the latter are clearly vital in the preservation of games, the information contained is both patchy and mainly descriptive: title, date, platform, and sometimes system requirements. This falls far short of the level of detail, robustness, and interoperability required by repositories.

Commercial organizations such as PEGI, who maintain their own archive of games passing through the certifying process, are in a similar state, holding descriptive data to catalogue the titles, but little complex metadata that would allow a runtime environment to be selected or recreated. A recurring issue with descriptive data schemes is classification by type or genre, which has generated a substantial quantity of literature. Dahlskog, Kamstrup, and Aarseth (2009) and Bjork and Holpainen (2005) have both proposed alternative approaches to classification for example.

Huth's solution is to draw from existing metadata schemas and supplement them with additional, self-generated, fields. The schemata he considers are:

- OCLC Metadata Elements (OCLC, 2003)
- Dublin Core Metadata Element Set (DCMI, 2003)

⁹We acknowledge the contribution of Andreas Lange and Dr. Winfried Bergmeyer of the Computer Spiele Museum, Germany, towards the translation and analysis of Huth's work.

- Manual of Archival Description (Procter and Cook, 2000)
- DIN (Deutsches Institut für Normung [German Institution of Standards]) 66230
- Projekt Metadaten (DiGA, 2003)

Additionally, Huth restricted his work to very early systems: the Atari2600 and the Commodore 64 (one console, one computer). This means that his schemata may not be particularly suited to dealing with some of the more complex issues noted previously as it predates them considerably. For example, user-generated content in the form of community “mod” culture (literally, modification of a commercial game that is freely available for users to distribute, usually with the support or at least blessing of the developer) only began, in earnest, following the release of Doom (id Software1993), as did network-based multiplayer gaming. Massively multiplayer online role-playing games (MMORPGs) such as EverQuest (Sony 2000) or World of Warcraft (Blizzard 2005) have their roots in the MUD systems first developed by Richard Bartle in 1991, but only really began to gather momentum in the late 1990s. However, even this falls nearly a decade after the C64’s heyday. Thus, although Huth’s work is of critical importance to KEEP, it is important to recognize its potential limitations. At the same time, the new technologies and game types that emerged in the mid–late 1990s raise a host of incredibly complicated problems for preservationists in general. At a recent summit meeting for the National Videogames Archive, it was agreed by participants that, alongside the plethora of control devices, the question of how to preserve MMORPGs was “the thing nobody wants to talk about.”

Huth splits his fields into five groups: Representation, Reference, Provenance, Fixity, and Context.

Representation (22 fields) contains basic identification information and some technical details, including input/output devices, storage requirements and Operating System. Instructions for installation and control are also held in this group, as is a field for software requirements that could be co-opted for middleware, patch, and plug-in associations. Huth does not include version number in this group, which would suggest that according to his system, new versions would be held as separate entities. While this seems appropriate, this is a case where the age of games Huth chooses provides restrictions that are far less likely to include a

large body of important associated objects, and this probably requires a separate field. The rather generic Tools and Utilities (DIN66230 1.6.3) is the most likely candidate for expansion. This issue also relates somewhat to Control Instructions (DIN66230 4.2.2.3)/Cheats (DiGA, 2003), the latter of which would now relate to both classic button-combination/secret word cheats and more generic console instructions. Console codes are important, as they allow the player access to the development console in-game, in run-time, which enables not just cheats, but access to vital information about performance and options to adjust the game. In terms of historical data, this is an unprecedented window into some of the surrounding cultures and associated knowledge about a given object and may be worth highlighting further (although again, it needs to be noted that this is more recent aspect of a game to be generally available). Finally, it is important to note that Huth explicitly references emulation under Application Software (DIN66230 1.6.3 / Application (OCLC 1 2003).

Reference Information (16 fields) is a predominantly descriptive group dealing with standard cataloguing data: Title, Platform, Version, and so on. Of these, the Creator (DC2) field requires expansion—it would seem to refer to Developer, as Publisher is separated out (DC5). Given the large numbers of staff employed on games, many of whom are not credited, this would be the simplest option although, it may then be sensible to include an additional field allowing Project Leads to be identified and searched against. There are key figures in gaming history that span developers, publishers, and even genres, and really require identifiers for proper historical analysis.

Provenance Information (53 fields) does include a credits field and other descriptive data, but it is the technical descriptors that make up the bulk of this group that we are really interested in (although the single field for Conventions [DIN66230 1.10]/Rights (DC15) should be noted as it is used for legal information). Huth opts for an exhaustive list of fields, not all of which may be necessary for identifying an appropriate emulation environment; although, given Hedstrom and Lampe's research, they may arguably be required to avoid the subtle breakdowns in experience noted by some. Following basic original technical environment fields, there is a repeat of the information contained in the Representation group. After this, the level of technical detail rises

sharply. Huth includes compilers, programming languages, a full list of modules, overview of program architecture, and data flow. While this is no doubt important information, it is not technically required for the level of patching object to emulators required by KEEP, for example. There is also a field for Source Code, which in the case of more modern games may run to millions of lines and, thus, may not be practical or as important as retaining the code. The Program Handling fields largely repeat earlier fields from Representation as well, so the question is less one of the time spent ingesting than sourcing this level of technical information in the first place. Certainly, it is secondary in importance to the general run-environment details. One potential omission here is the lack of separation between application and engine. Alongside the need to reference multiple middleware applications within a single-object, many contemporary games are built from modified variations of generic engines. For example, *Gears of War* (Epic Games 2006), *Mass Effect* (Bioware 2007) and *Mirror's Edge* (DICE 2008) all use Epic's Unreal Engine 3. This may have an impact upon necessary information for recreating a run-time environment as it adds an additional layer of requirements around the OS level.

Fixity (10 fields) records descriptive functionality, which goes some way towards the cultural complexity issues discussed earlier, in combination with Context (22 fields). It also contains two fields pertaining to Copy Protection added by Huth and detailing both the Protection associated with the object and any bypassing software or techniques. It is likely this may need expanding for more contemporary games and also replicating in the Installation fields of the Representation group. Otherwise, both Fixity and Context are fairly standard groups drawn mainly from Dublin Core (DCMI, 2003).

Conclusion

To conclude, the emulation/migration debate needs to mature so that the important steps taken in developing technical environment metadata needed for emulation, especially that originating from the computer games community, can start to filter through to the various stakeholders.

References

- Ainsworth, A. B., C. Avram, and J. Sheard. "The Monash University Museum of Computing: Ten Years On." *History of Computing: Learning from the Past*. Ed. A. Tatnall. Germany: Springer, 2010. Print.
- Anderson, D., J. Delve, D. Pinchbeck, and G. Alemu. *Preliminary document analyzing and summarizing metadata standards and issues across Europe* (Report No. D3.1: KEEP Project). Portsmouth, 2009. Print.
- Barwick, J. *Where have all the games gone? Explorations on the cultural significance of digital games and preservation*. Paper presented at the conference - DiGRA2009: Breaking New Ground: Innovation in Games, Play, Practice and Theory., London: Brunel University, September, 2009. Print.
- Bearman, D. *Collecting Software: A New Challenge for Archives & Museums*, 1987. Web. 15 Oct 2010. <http://www.archimuse.com/publishing/bearman_col_soft.html>
- Bearman, D. "Reality and Chimeras in the Preservation of Electronic Records." *D-Lib Magazine* 5.4 (1999). Print.
- Bjork, S., and J. Holopainen. *Patterns in Game Design*. Hingham, Massachusetts: Charles River Media Inc, 2005. Print.
- Conley, J., E. Andros, P. Chinai, E. Lipkowitz, and D. Perez. "Use of a Game Over: Emulation and the Video Game Industry." *Northwestern Journal of Technology and Intellectual Property*, 2.2 (2004). White Paper.
- Connolly, T., and C. Begg. *Database Systems: A Practical Approach to Design, Implementation and Management*. Harlow: Addison Wesley, 2005. Print.
- Dahlskog, S., A. Kamstrup, and E. Aarseth. *Mapping the Game Landscape: Locating genres using functional classification*. Paper presented at the conference - DiGRA2009: Breaking New Ground: Innovation in Games, Play, Practice and Theory., London: Brunel University, September, 2009. Print.
- Dappert, A., and A. Farquhar. *Significance is in the Eye of the Stakeholder*. ECDL, 2009. Paper Presentation.
- DCMI. Dublin Core Metadata Element Set, Version 1.1: Reference Description, 2003. Web. 15 Oct. 2010. <<http://dublincore.org/documents/2003/06/02/dces/>>
- Demant, D. "Why the Real Thing is Essential for Telling our Stories." *History of Computing: Learning from the Past*. Ed. A. Tatnall. Germany: Springer, 2010. Print.
- DiGA, e. V. Projekt Metadaten: Ein Blick auf verschiedene Softwarearchive und deren Datenstrukturen, 2003. Unpublished report.
- Dondorp, F., and K. van der Meer. "Design Criteria for Preservation Repositories." *Proceedings of Conferentie Informatiewetenschap 2003*. Ed. D. B. P. van der Meer. Technische Universiteit Eindhoven. Web. 15 Oct. 2010. <<http://alexandrio.tue.nl/extra1/wskrap/publichtml/2003ll.pdf>>.
- Gieske, J. *Avoid Missing Ball for High Score*. Salford, UK: University of Salford, 2002. Print.
- Gladney, H. M. "Durable Digital Objects Rather Than Digital Preservation." [Electronic Version]. *ErpaePrints*. ErpaePrints, 2008. Web 16 Jul. 2009. <<http://eprints.erpanet.org/146/01/Durable.pdf>>.

- Gooding, P., and M. Terras. "Grand Theft Archive": A Quantitative Analysis of the State of Computer Game Preservation." *International Journal of Digital Curation* 3.2 (2008): 19–41. Print.
- Guttenbrunner, M. "Preserving Interactive Content: Strategies, Significant Properties and Automatic Testing." *Proceedings of Workshop on Data Analysis WDA'2008*. Eds. F. Babi, J. Parali, and A. Rauber. Dedinky, Slovakia:, 2008. Print.
- Hedstrom, M., and C. Lampe. "Emulation vs. Migration: Do Users Care." *RLG Diginews*. 5(2001). Web. 15 Oct. 2010. <<http://webdoc.gwdg.de/edoc/aw/rlgdn/preserv/diginews/diginews5-6.html>>
- Huth, K. Probleme und Lösungsansätze zur archivierung von computer programmen—am beispiele der software des ATARI VCS 2600 und des C64. Berlin: Humboldt-Universität zu Berlin, 2004. Print.
- Knight, G., and M. Pennock. "Data Without Meaning: Establishing the Significant Properties of Digital Research." *International Journal of Digital Curation*, 4.1 (2009): 159–174. Print.
- Levy, D. M. "Where's Waldo? Reflections on Copies and Authenticity in a Digital Environment." *Authenticity in a Digital Environment*, pp. 24–31. Washington D.C.: Council on Library and Information Resources, 2000. Print.
- Lorie, R. A. *The UVC: a Method for Preserving Digital Documents - Proof of Concept*. Amsterdam: IBM Netherlands, 2002. Print.
- Lowood, H. *Playing History with Games Steps towards Historical Archives of Computer Gaming*. Paper presented at the Annual Meeting of the American Institute for Conservation of Historic and Artistic Works. 2004. Web. 15 Oct. 2010 <<http://aic.stanford.edu/sg/emg/library/pdf/lowood/Lowood-EMG2004.pdf>>.
- Lowood, H., A. Armstrong, D. Monnens, Z. Vowell, J. Ruggill, K. McAllister et al. "Before It's Too Late: Preserving Games across the Industry / Academia divide." *Proceedings of DiGRA2009: Breaking New Ground: Innovation in Games, Play, Practice and Theory*. London: Brunel University, 2009. Print.
- OCLC. *Digital Archive Metadata Elements* (Policy Document). Dublin, Ohio: Online Computer Library Center, 2003. Print.
- OCLC. *OCLC Digital Archive Preservation Policy and Supporting Documentation* (Policy Document). Dublin, Ohio: Online Computer Library Center, 2006. Print.
- Pinchbeck, D., D. Anderson, J. Delve, A. Ciuffreda, G. Alemu, and A. Lange. "Emulation as a strategy for the preservation of games: the KEEP project." *Proceedings of DiGRA2009: Breaking New Ground: Innovation in Games, Play, Practice and Theory*. London: Brunel University, 2009. Print.
- Procter, M., and M. Cook. *Manual of Archival Description* (British Library R&D Report No. 5965 ISBN 978-0566082580). Gower Publishing Ltd., 2000. Print.
- Rothenberg, J. "Avoiding Technological Quicksand: Finding a Viable Technical Foundation for Digital Preservation." *Council on Library and Information Resources*, 1999. Web. 15 Oct 2010. <<http://www.clir.org/pubs/reports/rothenberg/contents.html>>.
- Stawowczyk Long, A. *Long-Term Preservation of Web Archives – Experimenting with Emulation and Migration Methodologies*. International Internet Preservation

- Consortium. IIPC, 2009. Web. 15 Oct 2010. <http://www.netpreserve.org/publications/NLA_2009_IIPC_Report.pdf>.
- Stawowczyk Long, A., and D. Pearson. *I Say Emulate; He Says Migrate – Are Emulation or Migration Feasible Preservation Strategies?* Paper presented at the IIPC (International Internet Preservation Consortium) Open Day, 2009. Web. 15 Oct 2010. <<http://www.nla.gov.au/openpublish/index.php/nlasp/article/download/1475/1803>>.
- Thaller, M. *The eXtensible Characterisation Languages - XCL*. Hamburg: Verlag Dr. Kovac, 2009. Print.
- Tijms, A. Binary Translation: Classification of Emulators. *Leiden Institute for Advanced Computer Science*, 2000. Print.