Real-time Digital Archaeology?

written by Jeremy Huggett | 14/03/2015

Vint Cerf, co-designer of the TCP/IP protocols that make the Internet work and vice-president and Chief Internet Evangelist for Google, warned last month (for example, here, here and here) about an information black hole into which digitised material is lost as we lose access to the programs which are needed to view them. Somewhat ironically, Google's own priorities recently seem to have been to increasingly withdraw from information projects which preserved the past – killing off archives, slowing down digitisation activities, removing the Timeline and increasingly prioritising newness over older more established sources in search results (Baio 2015).

Responses to the reporting of Cerf's warnings were mixed. Some seemed relatively complacent: after all, we're already preserving data and information in libraries and archives, aren't we, while using open file formats will mean that bit rot is not a problem? In the process, many seemed to overlook part of Cerf's argument – that there was a need to preserve old software and hardware so that we retain the ability to read files in their original formats: what he characterised as 'digital vellum'.

Archaeologists have long recognised the importance of archiving old hardware and software. The Archaeology Data Service ran a 'computer museum' from the late 1990s and famously used a mix of old hardware and software (I recall emails circulating around 1998 seeking an early version of TurboCAD, for instance) to recover some of the files deposited in the Newham archive, files which are still downloaded today. Some years ago the ADS donated its collection of hardware, software and documentation to the Jim Austin Computer Collection, although none of it seemingly appears in the collection's catalogue at present.

The standard archival recommendation is to retain data files in their original format, and, at the same time, migrate them into widely supported and openly documented formats – especially if the files are in proprietary formats. Retaining the original format recognises that export or migration routines may not fully capture all the nuances of the original data – sometimes deliberately so. For instance, Alex Ball observed that providing commercial CAD systems with high-quality export routines would make it too easy for customers to migrate to competing products and that the CAD files themselves are more like software recipes than exhaustive descriptions of the model, consequently

> "… even later versions of the same piece of software, ostensibly using the same file format, might bring up somewhat different models on reading the same CAD file." (Ball 2013, 10).

Leaving copyright issues aside, emulating software directly, or running the software within a virtualisation of the original operating system, offers the prospect of continuing to view files in their original format. Creating such systems is far from straightforward, and from an archive preservation perspective its success is dependent on the accuracy of the emulation/virtualisation employed. This is because emulators and simulators may be compatible enough to appear to run successfully

although not with 100% accuracy. For example, in the context of SNES game emulation, Byuu asks:

> "if an emulator appears to run all games correctly, why should we then improve upon it? The simple answer is because it improves the things we don't yet know about."

Games, with their high level of graphical output and interactivity are a good testbed for developing emulators, as is evident in projects such the Internet Archive's Historical Software Collection and the Olive (Open Library of Images for Virtualized Execution) Archive from Carnegie Mellon who are making strides using very different approaches, and not just with games, but with software such as VisiCalc, WordStar, Mosaic, Hypercard etc.. Some of the results are distinctly impressive: for instance, in January 2015 James Baicoianu managed to get Netscape 1.0 (released December 1994) running under Windows 3.11 (released August 1993) running within a Javascript emulation of MS-DOS within a modern browser window and viewing a current website – a virtual time machine! (It's interesting that in the process it demonstrated that some of the fundamental aspects of the way Windows communicated with the Web have changed and no longer work – who else remembers having to install Trumpet Winsock?).

From an archaeological perspective, though, what is particularly interesting about these developments is that although the the need to archive software is seen primarily as a means of retaining the ability to open files in their original formats, retaining access to the original software means it becomes possible to reproduce the technical environment within which the data were produced. As a result, it offers the prospect of a Digital Archaeology which goes beyond the retrieval of information from archaic machines or damaged data resources such as that outlined by Ross and Gow (1999). This real-time Digital Archaeology incorporates the reconstruction of aspects of the socio-technical circumstances surrounding the creation and manipulation of the data using the original software tools themselves (Leighton John 2012, 22). For example, the reactions of present-day Photoshop experts to the experience of working with Photoshop 1 released twenty-five years ago is revealing, as well as amusing. Furthermore, maintaining the working software and its associated data files means we can to some extent also retain the embedded knowledge encapsulated in the software itself, although accessing the underlying code for inspection would be preferable if improbable in the case of commercial software. An argument for the use of Open Source in archaeology (see Ducke 2012, for instance).

References

Andy Baio 2015 'Never trust a corporation to do a library's job', *The Message* (January 28 2015) https://medium.com/message/never-trust-a-corporation-to-do-a-librarys-job-f58db4673351

Alex Ball 2013 *Preserving Computer-Aided Design (CAD)* (Digital Preservation Coalition Technology Watch Report 13-02) http://dx.doi.org/10.7207/twr13-02

Benjamin Ducke 2012 'Natives of a connected world: free and open source software in archaeology', *World Archaeology* 44 (4), 571-579.

Jeremy Leighton John 2012 *Digital Forensics and Preservation* (Digital Preservation Coalition Technology Watch Report 12-03) http://dx.doi.org/10.7207/twr12-03

Seamus Ross & Ann Gow 1999 *Digital Archaeology: Rescuing Neglected and Damaged Data Resources* (JISC/NPO Study within the eLib Programme on the Preservation of Electronic Materials: Library and Information Technology Centre, London) http://eprints.erpanet.org/47/01/rosgowrt.pdf