



Preserving the Legacy of IT Innovation

Doron Swade

► To cite this version:

Doron Swade. Preserving the Legacy of IT Innovation. Leon Strous; Roger Johnson; David Alan Grier; Doron Swade. Unimagined Futures – ICT Opportunities and Challenges :, AICT-555, Springer International Publishing, pp.162-176, 2020, IFIP Advances in Information and Communication Technology, 978-3-030-64245-7. 10.1007/978-3-030-64246-4_13 . hal-03194541

HAL Id: hal-03194541

<https://inria.hal.science/hal-03194541>

Submitted on 9 Apr 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Preserving the legacy of IT innovation

Doron Swade

Kingston upon Thames, Surrey, United Kingdom
doron.swade@blueyonder.co.uk

Abstract. IT challenges museological conventions. There are issues of physical scale at both extremes – large mainframes on one hand and microscopic devices on the other. The social utility of collections is widely measured by public exhibition. But ‘Black-box’ syndrome is a nightmare for exhibition designers and content developers: in the case of computer cabinets and integrated circuits there is no transparency of function to speak of and this stresses established techniques of display and interpretation. Software poses a near-intractable challenge for traditional object-centred museums and it is unclear upon whom, and on what institution, custodial responsibility falls. The impermanence of storage media is incompatible with archaeological timescales. Whether software has meaning without the capability of running it is an open question. Yet running legacy software has formidable resource implications: the need for working historic machines or their physical surrogates, and/or expertise to migrate applications to contemporary platforms. We look to museums to preserve for posterity the material culture of these transformative technologies. There are formidable challenges. How are we to meet them?

Keywords: Collecting IT, Museums and IT, Software preservation, IT Innovation legacy, Computer preservation, History of Computing

1 Introduction

Information technology represents a substantial human endeavour. The intellectual, economic and material resources involved in production, distribution and use represent major social, cultural and technological movements. There is a prevailing sense that, given the scale and transformative influence of these technologies, their social and technical histories should be preserved.

The mandate of museums is to preserve a material record of change. So it is to museums and museum-like institutions that we turn as repositories and custodians of the technological narratives of our times. What privileges physical objects in preservational practice is the 18th-century concept of ‘permanence of substance’ – that objects are durable and unchanging and are therefore appropriate evidentiary sources for the preservation, in perpetuity, of meanings [1]. In the case of IT, it is to museums of science and industry, and to specialised computing museums, that we look to preserve a permanent record of technological change.

2 Material Culture

IT hardware artefacts fall within traditional collecting criteria, and there are well-established protocols and practices for their acquisition and preservation. In the case of computers, historical significance is framed in different ways. We collect ‘world firsts’, technological breakthroughs and major innovations – Fleming’s diode, the first transistor, the first sub-micron integrated circuit. We collect physical examples that are typical of their kind or representative of widespread practice – PCs, smartphones, electronic calculators – mass produced artefacts where what is distinctive about them is, paradoxically, their uniformity. We collect objects with special provenance – an original lens from Colossus’s optical tape reader, a console clock from a Ferranti *Pegasus* (a signature feature of several though not all Ferranti machines), the distinctive red logo from an Elliot 803, Alan Turing’s pen – artefacts that have actual or mythical associations with legendary inventors, designers, makers or systems. We collect failures, real and virtual, that stumbled for whatever reason before viable realisation or, if realised, that failed commercially or simply were not favoured by the roulette of success (Apple Newton). We collect objects at the extremes of physical scale, performance and history – substantial chunks of mainframes; supercomputers (Cray-1, the Russian BESM-6), electromechanical race-track Totalisators [2, 3]; and the Harwell Dekatron aka WITCH that dates from 1951 [4], respectively. We collect objects as cultural artefacts without any necessary technical distinction – the National Savings ERNIEs which generate Premium Bond winning numbers [3, 5], acoustic hoods for daisy wheel printers, product packaging, and ephemera - trade literature, an Apple lapel badge [figure 1].



Fig. 1. Apple lapel, badge (1982). Picture credit: Doron Swade

3 Part for the Whole

Curators, archivists and museum store managers are said to have two recurring nightmares. The first is that nothing will survive. The second is that everything will survive.

While the mandate to collect items of IT hardware is relatively unproblematic, in practice IT hardware presents dilemmas for curators and collections managers. Mainframes from the ‘Big Iron’ era are impractically large to collect in their entirety and there are few complete systems to be found in protective custody that include duplicated peripherals, console furniture and a full set of processor cabinets. It remains unclear whether there is a complete IBM System/360 to be found in a museum collection. Computing is not alone in this. The last remaining hot-metal printing press used in Fleet Street, the Woods press dating from the 1930s, measures nine metres tall, nine-and-a-half metres long and weighs 140-tonnes. The Press was acquired as part of the Science Museum’s printing collection and is stored in an aircraft hangar outside London [6]. Locomotives and aircraft pose similar problems of physical scale.

Faced with the improbable size of first-generation computing installations, and later, mainframes, curators resorted to what might be called ‘artefactual synecdoche’ – acquiring parts to stand for the whole. The struggle for increased speed and power was a perennial priority especially in the pre-chip era. Given the mismatch between physical scale and available storage space, curators and collectors chose the central processor or processor cabinets as placeholders for what was seen to be the essential significance of the whole, resonating perhaps with the idea of preserving only a brain as the pre-eminent organ of the human body. The same was largely true of memory and the wish to reflect the challenges of storage capacity. Magnificent examples of magnetic core memory, with the visually arresting layered arrays of minute ferrite beads, survive, torn, as it were, from their operational environments in a process we might call ‘chainsaw acquisition’ where only the severed part survives. [figure 2]

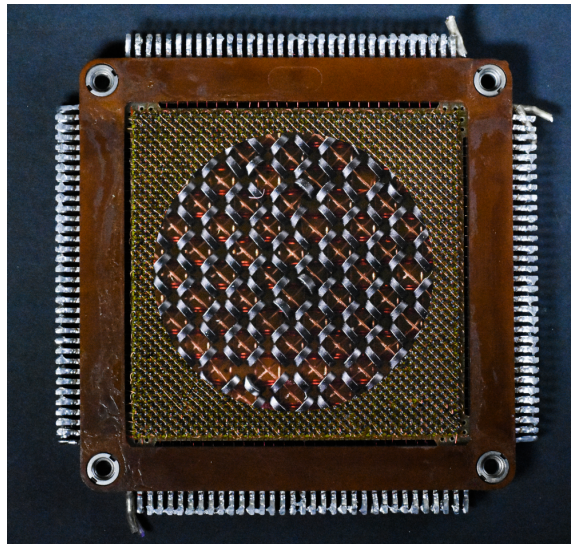


Fig. 2. Magnetic Core Memory (Digico 1972)) (Inset: detail). Picture credit: Doron Swade

The downside of reduction by selection was to lose an artefactual record of the operational environment and physical context – office furniture, for example, and enough

kit, not always visually distinguished and often many times duplicated, to convey the physical scale of the original installation. In the case of Olivetti's computers – cabinet design, floor layout, colour and overall appearance. To collect anything less than the machine in its entirety loses an historic convergence of design, technology, ergonomics and human machine interaction – an expression of Adriano Olivetti's personal philosophy about industry and society and the company culture of elegance and ergonomics in product design [7]. A legacy of the prohibitive size of early systems is that none of the great first-generation machines survives complete, and some are barely commemorated through even partial relics – ENIAC, EDVAC, EDSAC, UNIVAC, SAGE, Whirlwind, and the Manchester 'Baby' (SSEM), LEO.

Another selective reduction was to acquire control consoles which straddled technological and cultural meanings. SAGE, a radar command and control air defence system, built in the US, in response to the threat of Soviet nuclear attack during the Cold War is an example. In cultural terms the sober grimness of its consoles, a small fraction of this vast system, conveys vigilance, the seriousness of the threat it was intended to neutralize, and the high stakes of Cold War tensions [figure 3].



Fig. 3. SAGE Console (1960) Picture credit: Computer History Museum

Increased miniaturisation came to the rescue of despairing museum store managers. Starting with the mini-computer era of the 1960s, when complete machines were modestly sized and could be manageably stored, the problem of prohibitive space requirements for 'Big iron' acquisitions was for the most part self-resolving. For the first three generations of computers, machines and systems were defined localised entities that happily conformed to traditional categories of artefact. With the challenges of distributed processing, embedded devices and especially networks, the earlier era of collecting seems, in comparison, to be blithely innocent.

4 Exhibition and Interpretation

Maintaining a material of technological change through collections can legitimately be regarded as the core mandate of museums. Yet the social utility of collections is widely measured by public exhibition, interpretation and display. ‘Inform/inspire, educate and entertain’ is a tired but typical aspirational mission. In presentational terms the importance and transformative influence of early rail transport is conveniently symbolized by the massiveness of steam locomotives. In much the same way, up to the early 1960s, the fact that computer systems were big, conveyed significance and complexity, and there was a happy congruity between meanings and external form. As a free-standing object, function was largely opaque but this was masked by wonder at their size which supported the message of technological and historical significance. If it’s big, it matters, is the perceptual equation.

But from the early 1960s, developments in solid-state physics began to stress traditional exhibit presentation in a new and distinctive way. More and more was happening at component and device level, and there was less and less to see. What was of significance in an integrated circuit was invisible, whether live or not. ‘Black-box syndrome’ was the exhibitor’s dread where external form and internal function, however innovative, agreed to amicable divorce. With diminished visual appeal, long-accepted display conventions were challenged in an unprecedented way. New techniques were called for to unpack meanings from small items with no functional transparency to speak of. While difficult, the problem of the chip, as museum artefact, is not intractable. Creative use of existing techniques – panel graphics, illustration, imaginative comparisons, screen-based layered interactive exposition – provide solutions. No such obvious relief is available for software.

5 Software in an Object-centred Culture

The acquisition, preservation and interpretation of software poses near-intractable practical and philosophical difficulties for museums. Traditional museums are part of an object-centred culture, and privileged in this culture are physical artefacts, their meaning, significance and their care. Items of hardware, the physical stuff of IT systems are, by the conventions of traditional curatorship, unproblematic candidates for acquisition—monitors, keyboards, disk drives, system boxes, printers and the like. Software, a term in general use by the early 1960s, was defined negatively. *The Oxford Dictionary of Computing* defined software as ‘a generic term for those components of a computer system that are intangible rather than physical’ [8]. Prentice Hall’s *Illustrated Dictionary of Computing* severs any link with the physical by noting that that ‘software is independent of the carrier used for transport’ [9]. We habitually refer to CD distribution disks, and previously to floppy disks, as ‘software’. However, this fudges the distinction between the carrier, or medium of record, and the program. Strictly the software is a non-material logical abstraction, and the disk, PROM, mem-stick, punched cards, punched paper tape, are carriers [10].

We do not collect polynomials or prime numbers, but we do collect mathematical instruments, physical models, mathematicians' workbooks, drafts of seminal publications, first editions and, in the case of prime numbers, hardcopy printouts. The mathematics gallery at the Science Museum, London completed in 2016 [3] does not display number or mathematical operations, but computational machines and devices used in various arenas of human affairs, trade, war, finance, vital statistics, and structures i.e. ways in which number, an abstraction, has been physicalised for computational purposes. In this way abstract meanings are abducted by material artefacts. Similarly, when we 'collect software' we are not strictly collecting algorithmic logic but the physical media of creation representation, distribution, and storage – coding sheets, flow charts, written deliberation of designers and programmers, manuals, publicity literature, punched cards, and carriers in various forms – optical, magnetic, solid-state, paper. The distinction is not an entirely pedantic one. If software falls outside the mandate of object-centred museums then upon whom, and on what institution, does custodial responsibility for preservation fall?

Even when we collect, say, shrink-wrapped software products, we do so, at least partly, in their capacity as cultural artefacts – historical placeholders – rather than as archival records of a working product. The model for Museum preservation practice is essentially an archaeological one in which we retard physical degradation by passivating the physical environment – protection against physical trauma, low light, temperature and humidity control. The effectiveness of this model is limited when it comes to ensuring the data integrity of software carriers. Optical media and, *a fortiori*, magnetic media are notoriously impermanent, and manufacturers are loath to commit to figures for service lifetime. At best we are talking about decades, which is negligible in the archaeological timescales we envisage. Software is not alone in this. A curator of medicine might acquire a pack of Aspirin. There is significance in the branding, advertising graphics, tablet form, blister-pack press-through dispenser, logo and packaging – cultural content. But a pack of Aspirin would not be acquired and preserved as a permanent record of its chemical structure which it is accepted will degrade, and with it, its therapeutic potency. The inventoried Aspirin pack in a medical collection is not preserved as a pharmaceutical record of the product as it would be by retaining its chemical formula and/or the recipe for manufacture [10]. Similarly, it is accepted that passivating the environment will not ensure the operational viability of software when stored on its native medium. In short, there is a basic incompatibility between life-expectancy of the medium-of-record and the long-term custodial aspirations of museums.

6 Software Preservation

A working party was set up by the Computer Conservation Society [11] in the early 1990s to address the issue of software preservation and take account of the impermanence of storage media and the resources required to ensure content renewal in perpetuity. The group disbanded after three months with the conclusion that the only credible way to preserve software systematically was to record the content as a bit stream on

paper and entrust this printed record to archivists who have well-proven techniques for the indefinite preservation of paper media.

There is a variety of initiatives that have risen to meet the formidable challenges of software preservation. Perhaps the most broadly-scoped programme is that of the Computer History Museum (CHM), California, where the collecting emphasis is on source code – a form more revealing than most, of programmers’ intent and approach [12]. Enviably, the Museum supports a dedicated Software Curator, Al Kossow, who has developed protocols and practices to ensure the integrity of stored content including measures to prevent and detect corruption. In the UK David Holdsworth heads up the software conservation activity of the Computer Conservation Society, London. Here the mission is to preserve software in machine-readable form with a view to current or future capability to execute [13]. There are countless software preservation initiatives by societies, by national libraries and archives, and through the efforts of enthusiasts some dedicated to a specific class of products, computer games, for example. All face a variety of common issues, principal amongst which are the limited life of storage media, the impermanence of contemporary hardware platforms, and the portability of virtual platforms for emulators and simulators.

7 Software Recovery

As with hardware, the historical significance of software was rarely evident at the time of its creation or implementation and, if it survived at all in whatever form, it did so for the most part by accident or, occasionally, through the agency of a conscientious practitioner. There are cases where the original medium survives intact but nothing is known of its contents. There are other cases where the original medium did not survive but there is a non-electronic record which may or not be complete. And yet other cases where software is known to have existed but nothing at all exists in material form. Only in the last of these cases can the software strictly be considered ‘lost’. In each of the other cases software, taken as the logical algorithmic content, survives, whether or not in executable form. However, in all these cases, before the software can be preserved for future use, it needs first to be recovered, whether retrieved from the original medium, translated from a non-electronic medium, or reconstructed from scratch.

An example of retrieval from an original medium is the restoration to working order of the Elliott-NRDC 401, an experimental vacuum-tube machine dating from 1952-3. Only one 401 was made and the restoration, begun in the early 1990s by the Computer Conservation Society, was obstructed by the absence of any documentation for the drum store – no data, metadata, or information on the file structure. The restoration team headed up by Tony Sale and Chris Burton built read heads, ran the drum and captured the bit stream on a PC where it could be inspected, analysed and decoded. That the data was readable from a 40-year old drum puts in a slightly more favourable light the notorious impermanence of magnetic media. Len Shustek at the Computer History Museum has recovered source code from degraded magnetic tapes from Whirlwind – this by digitising the streamed analog waveforms and using software to reconstruct the original data [14, 15].

There are more extreme examples involving retrieval from other than native media. David Holdsworth recounts the recovery of software for the KDF9 computer from line-printer listings [16]. The texts were copy-typed after OCR proved inadequate as a recovery tool for the 40-year old hardcopy. Two assembler language programs were recovered in this way and the code runs on an emulator.

8 Love Letters

David Link's reconstruction of Christopher Strachey's program to generate love letters is an example of software recovery that combines computer archaeology and art [17]. Strachey wrote his love letter generator for the Ferranti Mark 1 (1952) at Manchester. Nothing of the original storage medium appears to have survived. Link located the source code in the Bodleian Library in Oxford but found it impossible to decode the algorithmic structure or internal dynamics of the program. He concluded that the only way to analyse the program was to reconstruct it and run it live – this by first constructing a software emulator, and later a functional physical replica using contemporary parts including simulations of Williams-tube memory using WWII radar tubes. He re-wrote a critical missing subroutine (the original was subsequently found) and the system formed the basis of an art installation which displayed continuously generated love letters on a screen, and also provided some interactivity with the visitor [18].

It is clear from these examples that there are substantial, ingenious, and significant initiatives to recover and preserve software. It is also clear that such initiatives, which require high levels of specialised expertise, are not centralised or co-ordinated and do not have a shared mission. Perhaps most importantly, they do not have the long-term security conferred by statutory responsibility as is the case for state-funded national museums with its mandate defined in law, the UK government's 1983 Heritage Act, for example. This is not a criticism. Quite the reverse – let a thousand flowers bloom. The observation is intended simply to flag that if our concern is preservation in perpetuity then the long-term future of these remarkable efforts is not secure.

The creation of new institutions to preserve new informational media is not without precedent. The National Sound Archive (NSA) was founded in 1955 as the British Institute of Recorded Sound and was absorbed into the British Library in 1983. As part of the British Library the sound archive is state funded. The National Film Archive was founded in 1935. As part of the British Film Institute (BFI) the film archive is sponsored by the government's Department of Digital, Culture, Media and Sport (DCMS). Both archives are secure for as long as society continues to value these media as part of its cultural heritage. If it is time for institutionalised software preservation with the security of state funding, how do we articulate answers to the questions, what, and who is it for?

9 Functionality

Whether the preservation of software, without means to run it, is meaningful is a fraught question. Gerard Alberts argues that software is a dynamic artefact and not to preserve user experience and interaction with it is to lose essential elements of its meaning [19].

Web archiving, which preserves ‘snap shots’ of static screen pages, has a very different outcome to preserving the experience of web-surfing in real time. David Holdsworth is of like mind in writing that ‘a basic tenet of the CCS’s software conservation activity is that software is only truly conserved when it can be run on current hardware and has the prospect of being run on hardware in the foreseeable future’ [16]. This is a judicious formulation in that it sanctions and encourages responsible preservation without making a contemporary hardware platform, or an operational equivalent, a prerequisite. We are enjoined to preserve, as a holding operation, before the material is lost and while specialised expertise survives.

Two areas come to mind where running software might be thought to be indispensable to meaning. The first is computer games. Gaming has significantly driven innovation and it is questionable whether the preservation of only the physical paraphernalia of gaming can meaningfully replace the ability to run these programs live on contemporary platforms [20]. A second example is that of art installations. Computer-enabled or computer-dependent art installations are largely meaningless unless run live. David Link’s *Poetry Machine* and his self-regenerating book, *Meditationes*, are examples [21, 22, 23, 24] [figure 4].



Fig. 4. Poetry Machine by David Link. Picture credit: David Link

The demands of exhibition are usually time-bound and any given platform will last the duration before terminal obsolescence. *Poetry Machine*, which involves sophisticated software and interactivity was revived after twenty years by migrating the software to a Windows platform and refreshing the hardware – a non-trivial process even for Link, its original creator. Archiving an installation scarcely qualifies as meaningful preservation without the resources to renew it and without the security of continuity. A related context in which functionality impacts preservation is the art market where the installation is seen as product. If an installation is sold what is it one owns without the

resources to maintain in perpetuity the original system, or migrate the software to a later platform? Paintings and sculpture are not plagued in this way. They do not require the mediation of complex technology to render them intelligible.

10 Platforms

There are countless initiatives to provide contemporary hardware platforms or functional facsimiles, primarily through the restoration to working order of surviving machines, or through the reconstruction of historic machines of which there are no surviving examples [25, 26]. The hallmark of such initiatives is the expertise, ingenuity and commitment of largely volunteer effort. Teams at the Computer History Museum have successfully restored a DEC PDP-1 (1959) and an IBM 1401 (1959) [27]. At The National Museum of Computing (TNMoC) at Bletchley we find working restorations of the Harwell Dekatron ('WITCH') a hybrid relay-valve machine dating from 1951 [4], a Marconi TAC (1959), the discrete component germanium transistor Elliott 803 (early 1960s), an Elliott 903 (late 1960s), an IBM 1130 (1965), an ICL 2966 (1980s), ICL System 25 (1980s) and ICL DRS 6000 (1990), a Cray EL98 (1993), and a variety of classic personal computers [28].

In instances where no original exists to restore or replicate, reconstructions have been resorted to. Prominent amongst these is the Manchester SSEM ('Baby') at the Science and Industry Museum in Manchester. At TNMoC there are several substantial working reconstructions of WWII machines [28] – Colossus, Tunny, Heath Robinson, and the Turing-Welchman Bombe, as well as the post-war Cambridge EDSAC [29].

Restorations and reconstructions provide contemporary hardware platforms, or functional equivalents, for running legacy software as well as new programs. Through these restorations and reconstructions, the fidelity of programming practices, performance, interactivity, and the user experience are preserved. These initiatives have other historiographical as well as social benefits: the practicalities of restoration or reconstruction draw one into a level of intimate detail with the machine that rarely occurs by other means; physical realisation almost invariably results in contingent or unexpected findings not foreseen by analysis or theory; operating the machines gives insights into contemporary ideas and practice and captures generational experience that would otherwise be lost; tacit knowledge is regenerated; documentary completeness is a contingent outcome; and the working system provides an authentic benchmark for simulation [25].

In addition to the historical and historiographical value of restorations and reconstructions there is the social capital they create i.e. their value to those who make up the project teams, to visitors who view and experience them on public display, and to museums and organisations that host these exhibits. The projects provide meaningful engagement within an organisational context for veteran experts to share expertise, exercise their craft, and extend their professional activities in historically and educationally meaningful ways. Demonstrating the machines has pedagogic value to users and visitors. Large machines on public display act as foci for visitor attention and provide a platform for storytelling through live demonstration and conventional exhibition graphics. Finally, large systems publicly displayed memorialise important episodes and

practices for which they act as both placeholders and monuments that serve as generational bridges to an otherwise lost past.

11 The End of the Road

However successful are such efforts to extend the operational life of historic machines or to provide historically authentic reconstructed platforms, we have to accept the eventual demise of such systems if our aspirations are to extend to archaeological time spans. While degraded wiring can be replaced, and any part, vacuum tubes for example, can, in principle at least, be refactored as replacements, a point will arrive where the effort and cost will become prohibitive even if the relevant expertise can be migrated inter-generationally. It is anyway the case that any replacement programme compromises the physical integrity of the original artefact and, if the machine is unique, violates the central primacy of the original in the central tenets of museum practice. There are also political insecurities. The priorities of host institutions change. At the Science Museum, London, a Ferranti *Pegasus* (1959), a vacuum-tube computer with magnetic drum storage, was restored to working order by the Computer Conservation Society and successfully run and demonstrated from the early 1990s for over 25 years. There was no technical reason preventing further extending its operational life. Its maintenance and public demonstration were ended following a policy shift away from working exhibits, and the machine was moved off public display into storage.

The construction of an historic machine is essentially an act of physical replication. There is of course logical replication in the form of emulators and simulators. Emulators allow us to run contemporary software on a non-native platform and this offers liberation from the need for working original or non-original hardware. Simulation captures the operational persona of the machine in providing a virtual functional equivalent that allows a user to program, run, and interact with an operational facsimile. While the logic of virtual machines may be faithful to the original there are performance issues that are not automatically portable, real-time execution being one.

Emulation and simulation offer the promise of indefinite migration to new platforms and, as the languages used for the simulations become increasingly machine independent, indefinite migration from one generational platform to the next appears to offer a form of logical immortality. In practice, however, the independence of software from hardware is rarely absolute and there are interdependences that compromise the prospect of indefinite preservation by these means. There is also the question of the longevity of the languages that host the emulators. When these are superseded, we have a new migration problem.

12 Justification for Working Software

In a bid for state or institutional funding for software preservation a question that will inevitably be asked is, to what purpose i.e. what benefits would accrue from having an archive of software whether running or not? The tendency is to begin to stutter. The point is that we have little idea how these artefacts will be interrogated by those who

follow. We cannot foresee what features may be of interest or significance. A researcher may be interested in the historic use of subroutines, the practice of altering instruction codes under program control, trawling early programs for forgotten techniques that could inventively be repurposed, algorithmic structures and precedents, the first use of nested looping, the first use of number as symbol rather than quantity, response and execution times, the speed of a compiler, latency . . . a gallimaufry of practical, philosophical, logical and technical inquiry. We do not ask this question of traditional museum objects nor of the specific utility of sound or film. It is accepted that these have cultural and historical value and are saved to be available for unforeseen and unforeseeable interrogation, and cultural appreciation. I have argued elsewhere [1] that what underpins the primacy of the original in museum culture is that only the original embodies predicates, properties and features the significance of which we cannot fully know nor anticipate. It is precisely this epistemic incompleteness that is responsible for the elevation of the original above object surrogates, replicas, or imitations. What underpins the value of the original, and is responsible at least in part for its mystique, is the sense that understanding is never complete, that knowledge is never total, that the original artefact embodies evidence the meanings which may not be accessible by other means, and whose significance cannot be foreseen. Software, as a digital object, is no different.

Exhibiting software poses daunting difficulties, and very few museum-like organisations have embraced the challenge. The deterrents are already familiar: abstractness, arcane complexity and the absence of anything physically meaningful upon which traditional object-based exhibition routinely relies. A bold exception is the exhibition *Make Software: Change the World* that opened in 2017 at the Computer History Museum [30]. The exhibition features seven exemplars of software products or classes of software identified as transformative: Photoshop, MP3, MRI, Car Crash Simulation, Wikipedia, Texting, and World of Warcraft i.e. applications regarded as ‘game-changing’ in computer-mediated virtual reality, compression, simulation, gaming, and crowd-sourced knowledge. This treatment leapfrogs attempts to treat software as a traditional museum object, dumps otiose agonising about the relationship between software and hardware, and confidently abandons much else that has so far daunted and paralysed wishful exhibitors. The subtext of the exhibition is that these are wondrous objects and they are created or enabled by software which is demonstrably powerful, important and ingenious. This treatment uses widely known outcomes to raise interest in the generative process. This turns on its head the traditional approach that favours focusing on the generative process (programming, compiling, software engineering . . .) in an attempt to provide some kind of explanatory account for the outcome.

13 Summary and Conclusions

Non-working hardware devices and systems fall within the traditional collecting mandates of object-centred museums which have well-established preservation practices and protocols. Exhibiting these as static objects is amenable to existing interpretative

display techniques. In the case of smaller devices, especially solid-state devices, opacity of function, uniformity, and physical size stretch conventional techniques of interpretation, but not unmanageably so.

Working historic machines and systems are heavily concentrated in the independent sector. Private collections and independent smaller museums are the primary hosts of restoration and reconstruction activity that heavily relies on the expertise, interest and commitment of volunteers. There tends to be little co-ordination between these groups for acquisition, exhibition or restoration activity, and security of support, not to mention their artefactual holdings, are perpetually at risk.

Prominent in the independent sector are subject-specialist museums dedicated to computing. These include the Computer History Museum, California, the Heinz Nixdorf MuseumsForum (HNF) in Paderborn, Germany, The National Museum of Computing (TNMoC) at Bletchley. With company funding, philanthropic and sponsorship support these collections are relatively secure for the time being. There are many regional museums with local focus, small departmental collections in universities, and small museums funded by universities, where financial security is partly assured by their organisational hosts. Overall, the future of smaller private collections is precarious especially when viewed in the context of archaeological time scales.

Running historic software requires functional intactness in the form of bit-perfect records as well as contemporary hardware or a physical or virtual functional equivalent.

Software preservation is a growing activity though still tends to be localised and fragmented. The levels of specific expertise for effective preservation through capture and emulation are high, and dedicated volunteers, many retirees, tend to be the drivers here. When it comes to storage methods, formats, and media there is little standardisation.

The resource implications of running software on contemporary platforms, or on functional equivalents, act as formidable deterrents for traditional museums. High levels of expertise are required in skills areas that have high market value outside the museum sector. Organisational commitment is required to ensure continuity, and social, technical and historical utility is difficult to justify through cost-benefit arguments. The resources required to sustain an indefinite programme of copying, renewal, and integrity-checking are non-trivial and few organisations have yet been willing to commit to preservation programmes of this kind.

The dream is for an independently resourced organisation with a mandate to proactively acquire software, preserve executable versions in perpetuity, to maintain working contemporary hardware platforms, develop emulators and simulators, and migrate this material to new generational platforms on an indefinite basis, this to provide a gene pool of software evolution. Systematic pro-active collecting of this kind is rare even as an agenda for object-based state-funded museums. What tends to survive are artefacts seen to be important enough for people to care about. Advocacy for much of this collecting comes from outside the museum sector, from enthusiasts, practitioners, specialists and professionals. What survives for the most part is through the happenstance of history. Software, so far, is no exception.

References

1. Swade, D.: Virtual Objects: Threat or Salvation? In: Lindqvist, S. (ed.) *Museums of Modern Science*. Watson Publishing International, Canton MA (2000).
2. Swade, D.: Forgotten Machines: the need for a new master narrative. In: Haigh, T. (ed.) *Exploring the Early Digital*. Pp 41-68, in particular pages 42-56. Springer, Heidelberg (2019).
3. Rooney, D.: *Mathematics: how it shaped our world*. Pp 89-90. Scala Arts and Heritage Publishers Ltd., London (2016).
4. Harwell Dekatron Computer aka W.I.T.C.H. The National Museum of Computing. <https://www.tnmoc.org/witch>, last accessed 26 February 2020.
5. Swade, D.: The three ages of windfall fantasy. *Guardian*, 6 October 1988.
6. Good news for historic press. BBC News, 20 November 2001. <http://news.bbc.co.uk/1/hi/sci/tech/1665086.stm>, last accessed 25 February 2020.
7. Mori, E.: The Italian Computer: Olivetti's ELEA 9003 Was a Study in Elegant Ergonomic Design. *IEEE Spectrum* (2019). <https://spectrum.ieee.org/tech-history/silicon-revolution/the-italian-computer-olivettis-elea-9003-was-a-study-in-elegant-ergonomic-design>, last accessed 26 February 2020.
8. Software. In: Illingsworth, V., Glaser, E.L. & Pyle, I.C. (eds.) *Dictionary of Computing*. 2nd edition. Pp 352. Oxford University Press, Oxford (1986).
9. Nader, J. C. 1992. Software. *Prentice Hall's illustrated dictionary of computing*. Pp 412. London: Prentice Hall.
10. Swade D.: Collecting Software: Preserving Information in an Object-Centred Culture. In: Hashagen U., Keil-Slawik R., Norberg A.L. (eds) *History of Computing: Software Issues*. Pp 227-235, in particular page 230. Springer, Berlin, Heidelberg (2002).
11. CCS: Computer Conservation Society. <https://www.computerconservationsociety.org/about.htm>, last accessed 29 May 2020.
12. Software Preservation. Computer History Museum, Software History Centre. <https://computerhistory.org/software-history-center/>, last accessed 26 February 2020.
13. Holdsworth D.: History, Nostalgia and Software. In: Tatnall A., Blyth T., Johnson R. (eds) *Making the History of Computing Relevant*. HC 2013. IFIP Advances in Information and Communication Technology, vol 416, pp 266-273. Springer, Berlin, Heidelberg (2013). https://link.springer.com/chapter/10.1007/978-3-642-41650-7_24.
14. Shustek, L.: Decode the analog waveform extracted from old computer mag tapes (2019). <https://github.com/LenShustek/readtape>, last accessed 26 February 2020.
15. Stachniak, Z.: Software Recovery and Beyond. *IEEE Annals of the History of Computing*, 41 (4), pp 110-118. IEEE Computer Society, Washington D.C. (2019).
16. Holdsworth, D.: Rescuing Software from Lineprinter Listings. *Resurrection: The Bulletin of the Computer Conservation Society* 57, CCS (2012). <https://www.computerconservationsociety.org/resurrection/res57.htm#e>, last accessed 26 February 2020.
17. Link, D.: *Machine Heart (das Herz der Maschine)*, Hatje Cantz Verlag, Ostfildern (2011).
18. Roberts, S.: Christopher Strachey's Nineteen-fifties Love Machine. *The New Yorker*, 14 February 2017. <https://www.newyorker.com/tech/annals-of-technology/christopher-stracheys-nineteen-fifties-love-machine>.
19. Alberts, G., Went, M. & Jansma, R.: Archaeology of the Amsterdam digital city; why digital data are dynamic and should be treated accordingly. *Internet Histories: Digital Technology, Culture and Society* 1(1-2), pp 146-159. Taylor&Francis Online, London (2017). <https://doi.org/10.1080/24701475.2017.1309852>.
20. Guins, R.: *Game after: a cultural study of video game afterlife*. MIT Press, Cambridge, Massachusetts (2014).

21. Link, D.: Poetry Machine. In: Brouwer, J., Fauconnier, S. & Nigten, A. (eds.) *aRt&D: Research and Development in Art*. Pages 196-205. V2 and Nai Publishers, Rotterdam (2005)
22. Link, D.: Poetry Machine (2005). http://alpha60.de/art/poetry_machine/, last accessed 27 February 2020.
23. Link, David, and F. M. Einheit. "Radar Angels: Art and Artificial Text Generation." *The Turing Conversation*. ETH Zurich. <https://www.turing.ethz.ch/the-turing-conversation/contribution-8.html>, last accessed 12 August 2020 2020.
24. Vega, A.: *Meditationes: A metaphysical laboratory* by David Link (2016). <http://alpha60.de/art/meditationes/>, last accessed 25 February 2020.
25. Swade, D.: The Historical Utility of Reconstruction and Restoration. In: Campbell-Kelly, M. (ed.) *Making IT Work*, pp 7-21. British Computer Society and The National Museum of Computing, London (2017). <http://www.computerconservationsociety.org/miw/Proc%20MIW%202017.pdf>.
26. Swade, D.: Historical Reconstructions. *IEEE Annals of the History of Computing* 27 (3), page 3. IEEE Computer Society, Washington D.C. (2005).
27. Garner, R.: Restoring and Demonstrating 1960s Vintage Computers at the Computer History Museum. In: Campbell-Kelly, M. (ed.) *Making IT Work*, pp 54-67. British Computer Society and The National Museum of Computing, London (2017), <http://www.computerconservationsociety.org/miw/Proc%20MIW%202017.pdf>.
28. Explore the Galleries, The National Museum of Computing, <https://www.tnmoc.org/galleries>, last accessed 27 February 2020.
29. EDSAC - Electronic Delay Storage Automatic Calculator, The National Museum of Computing, <https://www.tnmoc.org/edsac>, last accessed 27 February 2020.
30. Make Software: Change the World, Computer History Museum, <https://www.computerhistory.org/makesoftware/exhibit/>, last accessed 26 February 2020.