Sustainable Computation—Foundation for Long Term Access to Digital

Natasa Milic-Frayling
Microsoft Research
21 Station Road
Cambridge, United Kingdom CB1 2FB
+44 1223 479 700
natasamf@microsoft.com

ABSTRACT
Digital assets, such as digital documents, databases, and interactive games, can be instantiated and used only through computation by executing the software programs that enable users to experience and interact with the digital content. Prior efforts to secure access to digital assets focused primarily on the persistence of data and program files and less on ensuring that the software can run in the contemporary environment. In this paper we emphasize the importance of computation. In fact, we argue that *the task of preserving digital assets is the task of enabling computation by which digital content can be used in the contemporary computing environment*. We discuss three approaches which, to various degrees, rely upon the original program and data files to provide access to digital content. We expect that all three will be required to cover a range of scenarios and requirements. As we depart from the use of original programs and files we are faced with the issue of assessing the results of the replacement programs and files. That, in turn, requires a clear understanding of the value that the original assets entail and the metrics and methods to assess how that value changes when alternative computation and files are used. In order to address these issues, we propose a research agenda that focuses on the principles of sustainable software engineering to enable computation and preserve the value of the digital assets.

Categories and Subject Descriptors
H.3.0 [**Information Storage and Retrieval**]: General.

General Terms
Algorithms, Reliability, Standardization, Theory, Verification.

Keywords
Digital preservation, computation, files

1. MOTIVATION
Design of modern computing systems and applications have detached the user from the core computing processes. Indeed, the design has been intentionally optimized to hide the complexity of computation required to support interaction with programs and digital content through visual, audio, or tactile feedback. At the same time, the users have been provided with facilities such as file system features to manage their digital assets focusing on the organization and access to files created through computation. Ensuring that such files are safely stored and available for reuse is an important aspect of protecting the value that the user generated through the application. However, equally important is to ensure that the corresponding program can be used within the contemporary computing environment

The notion and the perception digital content have been strongly shaped by the advances in human-computer interactions. Menu-based WYSIWYG interfaces have enabled easy use of applications through a set of pre-defined functions, accessible through buttons or menu options. Furthermore, the display and rendering technologies have made the system responses to the user's actions instantaneous. This tight integration of the user's actions, stored data, and presented content gives an impression that the displayed content is, in fact, stored and, therefore, it suffices to manage data files to ensure its future use. Yet, the rendered content is never persisted. By its nature, it lasts only while the application is running. It is ephemeral and completely determined by the computation of the involved programs.

The characteristics and, therefore, the value of a specific digital asset are determined by the properties of the computation and the stored input and output files. During computation, they jointly provide utility to the user. With this understanding, we consider three fundamental approaches that, to a various degrees, involve the original programs and files of a digital asset:

1. *Encapsulation* of the original computation in a virtual machine environment where both the original data files and program files can be executed.
2. *Porting* of the software application to a new version of software that can run in the contemporary environment. In this manner we can use the original data files and focus on ensuring that the new software meets the requirements and specifications of the original software.
3. *Replacement* of the software application by a contemporary application and *conversion* of the data files into the file format required by the new application. In this approach, reproducing the user experience and the value from the digital asset depends on the features of the replacement software and our ability to map the original file format onto the format required by the new application.

In all three cases, we may still have an issue with the supporting software that runs the computing hardware and the peripherals. Indeed, the ability to persist and use the data files and program files is highly dependent on the availability of adequate hardware and supporting software, e.g., the operating systems, device drivers for input and output, etc. Thus, encapsulation, for example, may also require emulation of the relevant hardware components and simulation of the supporting software.

The outlined approaches of software porting and replacement depart from the strict use of the original files and programs. This raises a question of characterizing and comparing the results of the original and the substitute computation and files. In practice, it has been assumed that there are inherent properties of the digital assets that are deemed relevant and therefore should guide the preservation strategies [2][5]. However, arriving at metrics and methods for assessing that such properties are preserved is a challenging issue.

Indeed, by observing the user practices around digital content, we note that the value of digital assets is not derived simply from running the original software but from the interaction with a broader ecosystem of applications and services. In particular, it is typical for users to diversify both the computation and the file format used to present the particular content. For example, the users may author a document using office productivity tools and then derive additional value by publishing and broadly disseminating the document through conversion into several formats, e.g., .html, .xps, and .pdf. In that manner, they make the content accessible through a range of applications. Each file format and corresponding software program is likely to deviate from the originals in many respects. Nevertheless, they meet the user's specific objective. The challenge, therefore, is to consider the value of the digital assets in the context of use and relationship to other applications and services in the ecosystem.

Based on the presented insights, we propose a concerted research effort to investigate principles of software architecture design and engineering to

- Support a range of strategies for sustaining computation in relation to the rapidly changing computing ecosystem
- Preserve the value of the digital assets, defined through both the intrinsic properties of the digital assets and the interaction with the broader ecosystem.

In the following sections we discuss the state-of-the-art approaches to long term access to digital and illustrate the issues that require in-depth consideration by the research community.

2. CURRENT STATE OF THE ART

Digital preservation subsumes a notion of consistency in the user experience with the digital content as an important aspect of maintaining the value of a digital asset. For example, maintaining and proving authenticity of a digital document is a frequent requirement and leads to an important question of how to characterize these notions for documents in the digital form.

One may postulate that using the original application with the original files would lead to the authentic view and interaction with the document and, thus, it suffices to ensure that neither the program nor the data files have been tampered with. Thus, much of the effort has been placed on ensuring that the program files and data files are stored in the medium that is not prone to mechanical errors. A tacit assumption is that the digital authenticity is defined by the computing encapsulated in the specific software application and the data files, and that the generated content view is immutable with regards to computer hardware and supporting software that execute the programs to instantiate the content. It has led to a concerted effort in preventing bit rotting, i.e., ensuring the preservation of bits.

Indeed, reliable storage of digital encoding of data and programs has been a long-standing concern of the IT industry [1][3][4]. This has led to research in techniques for bit preservation and to commercial solutions for long term archiving. Such techniques are critical for ensuring that the digital information is not lost and is a pre-requisite for reproducing the digital content. As new computing paradigms appear, such as cloud computing, these techniques are re-evaluated and further expanded to take into account additional factors that may affect the quality of the bit storage and management of the digital assets. We propose that the bit preservation work is extended to include, not only the management of bits within the storage media but the accuracy of bits in relation to the program and data execution.

A special case of the consistency requirement relates to preserving perceptive aspects of the digital content such as the layout of the rendered documents. The concern with the presentation consistency of office documents has led to approaches that give preference to the layout of rendered content on over the richness of its digital encoding. For example, it is a common practice to transform rich XML based document formats such as .odt and .docx into .jpg or .pdf that can be used to show, with high fidelity, the content as originally viewed on the screen.

Visual properties, such as the layout, are examples of *significant characteristics* of digital assets that are deemed valuable and thus worth preserving. With the diversity of new digital assets, such as dynamic and interactive content, the set of such characteristics is increasing. This demands new approaches for evaluating the quality of preservation outcome and also forces us to re-examine the existing practices. For example, in the case of the layout characteristics of a document, the quality assurance methods often reflect the belief that the layout is the property of the persisted file, disregarding the role of the rendering software. This is despite many counter-examples such as different rendering of the same .html file in different Internet browsers. In order to consider significant characteristics of the digital content we need to consider together the data file and the software computation that instantiate the digital content. This means that we need the means of analyzing the computational as well as persistence aspects of the digital assets.

3. RESEARCH CONTRIBUTIONS AND BENEFITS

Considering that computation is the fundamental aspect of digital and that the IT ecosystem evolves continuously, it is not surprising that the legacy digital assets cannot be instantiated in the contemporary computing environment. Namely, software applications are dependent on an elaborate stack of supporting software that runs the computing hardware. Elements of this stack evolve at different rates and in a loosely coordinated manner. Thus, even the contemporary applications and services require continuous updates in order to stay current and functional. This is reflected in the standard service and maintenance provisions when computing systems are deployed. In essence, the problem of *persisting digital assets is the problem of enabling computation of these assets in the contemporary environment*.

Assuming that the IT ecosystem will continue to evolve through innovation in software architecture design and development practices, we propose to consider how related aspects of digital preservation should inform future systems and the development

efforts required to alleviate the consequences of software obsolescence.

3.1 Software Development and Sustainability

When considering access to legacy digital assets, we discussed encapsulation, porting, and replacement of computation as ways to instantiate the digital content. They reflect three different levels of retaining data files and programs in their original form. Once we make a decision which digital objects, i.e., data files and program files, are treated as immutable, we can focus on building the scaffolding around these objects to enable their instantiation within the contemporary IT environment.

In essence, we assume that, at any given time, there a single, all-encompassing computing ecosystem which determines whether a digital asset is compatible or obsolete. This is a simplification in the sense that one could maintain an isolated computing system that is not connected and, therefore, unable to generate or derive value from the rest of the ecosystem.

The amount of scaffolding required to sustain computation of a digital asset varies across different strategies. However, some aspects are common to all:

- The scaffolding itself incurs ongoing cost of maintenance and updating that will cease only if the ecosystem stops evolving or the access to the digital assets is not required any more.
- In order to maximize the value of the digital assets one needs to provide connectors with the rest of the services and applications. These *bridging technologies*, again, incur the cost of development and maintenance.

In the example of data and software encapsulation using a virtual machine (VM), it is the VM-ware that provides the scaffolding and the interface with the rest of the ecosystem. In essence, the VM provides a 'computational cradle' for the original software and data which stay unchanged but the VM-ware itself will change over time. The new versions of VM-ware will need to continue supporting the legacy computations.

In the case of program porting, it is primarily the storage of original data that needs to be supported, making provisions for possible changes in the storage technology, i.e., digital encoding and retrieval. Otherwise, the ongoing cost is associated with the software porting and updating over time.

Finally, the software replacement strategy assumes that, at any moment in time, there is a contemporary software application that provides functionality sufficient to sustain the value of the digital assets. The cost of providing compatible computation is not directly incurred. Instead, it is shifted towards the development and maintenance of the format translators and data format migration to ensure that the information stored in old data files can be used by the new application.

In all these approaches there is an ongoing effort and cost of software development, primarily related to three types of effort: support for scaffolding and its maintenance, upgrades of the original software, and development of bridging technologies that connect different computational processes through input conversion. Feasibility and sustainability of digital preservation will, therefore, depend on the availability of development skills and resources and the economic models that can support the incurred costs.

3.2 Value of Digital Assets

The software development efforts also play an essential role in establishing the value of the digital assets. As we already noted, assuming that the value of digital assets is realized only through their original application and data files misses to acknowledge the fundamental benefit of the digital media—content in the digital form can be easily repurposed and transferred from one context to another. In effect, *the value of a digital asset rises with the range of applications and services that can exploit it*.

Thus, regardless of which approach is taken to instantiate the digital content, further added value will be enabled though bridging technologies that connect the digital assets with the contemporary services. For example, we may implement a VM to access a corpus of Word Perfect 6.0 documents. While this enables us to view and interact with the documents, searching over the collection would involve an extra effort. We would need to incorporate a module that parses the text of the WordPerfect documents and provides the output compatible with the indexing engine that is running outside the VM.

3.3 Research Directions and Benefits

Framing of the technical aspects of digital preservation as a problem of sustained computation leads to a simple yet sufficiently broad framework for reasoning about effort, cost, and sustainability. We propose to undertake an in-depth analysis of the current software development practices and focus future efforts on sustainable software engineering that takes into account the complete life-cycle of computing systems, applications, and services. We argue that besides the adopted success criteria for software quality, such as reliability and usability, we need to include properties that pertain to the end-of-life of computing systems. These should include provisions for minimizing the expected effort and cost of sustaining digital assets produced by the system. A plausible approach could be highly modular architecture that allows effective and economical updates, modifications, and porting. By starting with the very core of software development principles, we expect to arrive at sound and effectives ways with long term benefits.

4. OUTLOOK

We are aware that the proposed framework for approaching the technical aspects of digital preservation is not a common place. However, it encompasses rather than deviates from the practices that are currently adopted. For example, a typical approach of using format migration to enable access to content is shown as one of the possible strategies that, in effect, abandons both the original software applications and the original data files.

We are encouraged by the emergence of the new computing paradigms, the cloud computing, that can potentially lead to the new models of creating and maintain computation. Namely, one essential value proposition of cloud computing is the lower cost of storage and computation as the cost is offset against the demand. We, however, see the potential for cloud environments to enable economically affordable development of software resources that are needed to ensure reuse of digital assets. This can be achieved through providing repositories of development resources (development tools, SDKs, API documentations, etc.) that could lower the barrier for acquiring skills and engaging in the software development. We are conscious that such an approach would inevitably raise the issues of commercial software licensing, reuse, source code access, and related. These would need to be taken

into account as the bounding factors. However, they should not stay in the way of research in the principles of software design and development that are critical to sustain access to digital.

5. REFERENCES

[1] Baker, M., Shah, M., Rosenthal, D. S., Roussopoulos, M., Maniatis, P., Giuli, T. J., & Bungale, P. (2006). A fresh look at the reliability of long-term digital storage. *ACM SIGOPS Operating Systems Review*, *40*(4), 221-234.

[2] Hedstrom, M., & Lee, C. A. (2002, May). Significant properties of digital objects: definitions, applications, implications. In *Proceedings of the DLM-Forum* (pp. 218-27).

[3] MacLean, M. G., & Davis, B. H. (Eds.). (1998). *Time and Bits: Managing digital continuity*. Getty Publications.

[4] Smith, M. (2005). Eternal bits [digital files preservation]. *Spectrum, IEEE*, *42*(7), 22-27.

[5] Thibodeau, K. (2002). Overview of technological approaches to digital preservation and challenges in coming years. *The state of digital preservation: an international perspective*, 4-31.