EXECUTABLE ARCHIVES

Software integrity for data readability and validation of archived studies

Natasa Milic-Frayling

Intact Digital Ltd

United Kingdom

natasamf@intact.digital

0000-0003-1244-1973

Marija Cubric

University of Hertfordshire

United Kingdom

m.cubric@herts.ac.uk

0000-0001-6699-3576

Abstract – Long-term readability of electronic data is a key regulatory requirement for archived data integrity in life sciences and pharmaceutical research. However, this has been difficult to achieve within current data and software preservation practice due to data dependence on specialty software which becomes unusable as a result of rapid obsolescence of hardware and operating systems. This paper introduces a novel Executable Archive framework that extends traditional data archives with a platform for hosting legacy software and with processes for installing, use, and long-term maintenance of the software. Through a case study of a scientific software de-commissioning, we demonstrate the use of the framework for designing a solution for GLP-compliant software transition from operational to archival use and a secure processing of raw archived data to reconstruct past research studies. The framework is flexible and opens up opportunities for preservation planning and action that consider both data access and software management together, ensuring that the archived data integrity is fully supported by the long-term software integrity.

Keywords – data integrity, software integrity, study reconstruction, significant properties, executable archive

Conference Topics – Exploring the New Horizons; Scanning the New Development.

I. INTRODUCTION

The ever increasing diversity of digital technologies and use scenarios is continuously challenging digital preservation practices and constantly moving the goal posts for preservation action. In this paper we present a case study that required us to revisit the two fundamental notions in digital preservation: the preservation of significant properties and the management of access and reuse.

Originating from a highly regulated sector that involves pharmaceutical, life sciences and bioanalysis organizations, the use case includes strict guidelines on data retention and the reproducibility of archived studies [23, 5]. Similar to other archiving practices, long term archiving of digital records is managed through a combination of format standardization and interoperability of both digital record formats and content management systems. However, the raw data that arise from research experiments have to be stored in the original format supplied by specific instruments (Fig. 1).

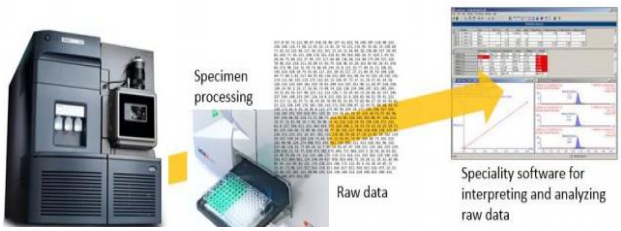

Figure 1 Raw data are produced by specimen processing and processed using software designed to support specific data analyses. The instrument and software installation are subject to extensive calibration and system validation process.

The collection and handling of research data during the operational phase are subject to strict data integrity

17th International Conference on Digital Preservation

iPRES 2021, Beijing, China.

DOI: 10.1145/nnnnnnn.nnnnnnn

regulations that, in the archiving phase, translate into well-defined procedures for data deposit, meta data management, and regular file fixity checks. Raw data must be immutable (Fig. 2). The unresolved issue, however, is the reproducibility and validation of the reported study results.

Reliable reconstructions of studies depend on the integrity of the software installations used to perform data analyses. Thus, both the data integrity and the software integrity requirements affect the preservation practices as they must enable the organizations to meet evolving regulations and support regular compliance audits (normally every couple of years). However, there is another layer of complexity. While the study records and raw data are stored in the archive, the operation of the software lies outside the area of an archivist's competence. Indeed, the studies are reconstructed by scientists. Similarly, the management of the software installations, particularly software reliant on legacy operating systems, lies outside the area of an archivist's or a scientist's competence and must be addressed by IT specialists in a principled and well documented manner.

OECD SERIES ON PRINCIPLES OF GOOD LABORATORY PRACTICE AND COMPLIANCE MONITORING
Number 17

Appendix 2: Glossary

Data (derived data) Derived data depend on raw data and can be reconstructed from raw data (e.g., final concentrations as calculated by a spreadsheet relying on raw data, result tables as summarized by a LIMS, etc.).

Data (raw data) Data (raw data) may be defined as measurable or descriptive attribute of a physical entity, process or event. The GLP Principles define raw data as all laboratory records and documentation, including data directly entered into a computer through an automatic instrument interface, which are the results of primary observations and activities in a study and which are necessary for the reconstruction and evaluation of the report of that study.

Figure 2 Definitions of derived and raw data specified in the glossary of the OECD guidelines [5] (p31).

This separation of concerns and roles raises a research question:

What would be an effective design of a system and services to extend the current electronic data archiving solutions and practices, recognizing (1) the fundamental dependence of electronic data access on software and (2) a need for long-term software installation support.

We propose to explore such designs through the *Executable Archive* framework that extends a traditional data archive with a complementary technology component, the software hosting platform, and with corresponding processes for managing installation, validation, maintenance and use of the software. These two aspects, ensure that data archives are resistant to technology obsolescence and effective in supporting long-term preservation of data.

Through a case study of software transition from operational use to a 'data reader' we illustrate the use of the framework to design the Software Library platform and services in order to (1) host the collection of validated software installations, (2) provide secure connections to data repositories, and (3) enable access to software and data to meet Good Laboratory Practices (GLP) regulations in bioanalysis research. The software transition puts an emphasis on both (a) the process of software installation and validation, i.e., the data reader set-up and (b) the expert inspection of the data processing outcomes. Thus, the specification of the significant properties is split across the software preparation process and the characteristics of the data analyses that the software enables.

We expect that our work will motivate researchers and practitioners to revisit the notion of digital objects and their significant properties and recognize the importance of software validation and software integrity for digital content reuse and research reconstruction, as illustrated in the case study.

The rest of the paper is organised as follows: In the following section we introduce the necessary background and definitions from data and technology management and related regulations. After that, in Section III we reflect on the related work in digital preservation and management of software. In section IV we describe the Executable Archive framework, its components and main processes. In Section V we demonstrate the use of the framework through a case study on the long-term maintenance and validation of Analyst 1.4.2 (Sciex) software required for accessing and validating pre-clinical study data in bio-sciences. We conclude with a summary of the paper contributions and areas of future work.

II. BACKGROUND

A. *Data Collection and Technology Management*

The process of data gathering and analysis starts with instruments and specimen processing (Fig. 1). Interaction with the raw data is facilitated by specialized software, a key enabler of the data interpretation and analysis. Reports from the experiments are stored as evidence of observations, findings, and conclusions. Any changes to the software or the environment within which the software operates may affect the results. For that reason, the technology vendors are concerned with both (1)

the implementation of the software and (2) the environment in which the software runs. It is common for vendors to supply a dedicated PC with pre-installed software to be used for processing data in the lab. They provide extensive service support and software upgrades that must be tested when deployed. The problems arise when the instrument and the software are no more in operational use either because the technology is discontinued or because the organization has changed the technology provider. In both cases, the instruments and the software are decommissioned. That leaves the archived data without supported software.

B. Regulations

The importance of raw data and validation of research outcomes is emphasized by the *Good Laboratory Practices* (GLP) [5] that the organizations must adhere to. The *Organisation for Economic Co-operation and Development* (OECD) works closely with the professional community on the guidelines for complying with GLP regulations. Two aspects are particularly key to our discussion: the requirement for reproducibility of research directly from raw data (Fig. 3) and a recognition that the software is important for the readability and validation of archived data and therefore must be managed as part of the archiving practices (Fig. 4).

While we illustrate the Executable Archive framework using practices within a specific sector, the need for regulatory compliance and research reproducibility are broadly recognized. Data retention and reproducibility requirements are present across industry sectors, from fintech to aerospace [1,16]. While the General Data Protection Regulation (GDPR) expects organisations to create data retention policy, it does not specify the retention periods and those will vary across industries and type of data (e.g., 3-10 years in financial sectors [1] to 50 years for the design data in the aerospace industry [16]. Here we use a generic attribute 'long-term' to mean the longest retention period required in any specific sector. At the same time, government funding agencies are promoting open research data repositories and research hubs to enable reusability of data and maximize the impact of research investment [4,17]. Such initiatives typically provide tools for ingest, documentation and search of research data but still lack clear guidelines and requirements on validation and reproducibility of results.

C. Data Integrity and Software Integrity

In order to support organizations in meeting regulatory requirements, it is important to consider operational practices that led to the production of data and archived studies. These practices are shaped by concerted efforts to maintain the data integrity throughout all the aspects of the research work. For data produced using computerized system that inevitably means rigorous management of hardware and software to ensure the quality of collected data. It is therefore helpful to consider data integrity and software integrity together (Fig. 5).

OECD SERIES ON PRINCIPLES OF GOOD LABORATORY PRACTICE AND COMPLIANCE MONITORING
Number 15

Advisory Document of the Working Group on Good Laboratory Practice

Establishment and Control of Archives that Operate in Compliance with the Principles of GLP

The archiving of records and materials generated during the course of a non-clinical health or environmental safety study is an important aspect of compliance with the Principles of Good Laboratory Practice (GLP). The maintenance of the raw data associated with a specific study and the specimens generated from that study are the only means that can be used to reconstruct the study, enabling the information produced in the final report to be verified and the compliance with GLP of a specific study to be confirmed.

Figure 3 Excerpt from the OECD guidelines for establishment and control of archives and raw data storage for compliance with Good Laboratory Practices (GLP) [5] (p9).

OECD SERIES ON PRINCIPLES OF GOOD LABORATORY PRACTICE AND COMPLIANCE MONITORING
Number 17

Advisory Document of the Working Group on Good Laboratory Practice

Application of GLP Principles to Computerised Systems

3.2 Data and storage of data

75. Hardware and software system changes must allow continued access to, and retention of, the data without any risk to data integrity. When a system or software is updated, it must be possible to read data stored by the previous version or other methods must be available to read the old data. Supporting information (e.g. maintenance logs, calibration records, configuration etc.) which is necessary to verify the validity of raw data or to reconstruct a whole study or parts of it should be backed-up and retained in the archives. Software should be retained in the archive if necessary to read or reconstruct data.

Figure 4 Excerpt from the OECD guidelines for application of GLP principles to computerized systems [23] (p20).

Data Integrity is of ongoing concern and a matter of constant improvement, from increased security and interoperability to a reliable management of data provenance and digital signatures. The community is actively pursuing interoperable XML-based formats for representing raw data and data analysis in order to automate encryption/decryption of data files as the data are moved between different applications for various types

of analyses. That work is ongoing [12]. Once a study is completed, the researchers transfer data for archiving and preservation to a central archive. The data are regularly checked for bit-rotting issues by conducting check-sum validation of data samples on a monthly basis.

Software Integrity, on the other hand, has not been of much concern since operations are supported by a careful and comprehensive validation of instruments and software at the time of the technology deployment and upgrades. That ensures that the software stays performant, secure, and consistent. However, when the software is decommissioned the software care stops and that leads to various *ad hoc* approaches to ensure its sustained use, from creating an image of the full computing environment to re-installing the required software within a suitable computing environment. No principled ways of managing the software in the archiving phase have been established.

Data Integrity

Data Integrity is the extent to which data are complete, consistent and accurate throughout the data lifecycle.

- **Complete** – All the required data needs to be collected and stored
- **Secure** – Data must not be destroyed
- **Unaltered** – Data must not be changed
- **Confidential** – Data must not be disclosed to unauthorised individuals
- **Usable** – Study metadata needs to provide context for experts reproducing study results.

Software Integrity

Software Integrity is the extent to which a software installation is functional, reliable, and usable throughout its lifecycle from operational use (i.e., applied in new studies) to study reproducibility (i.e., applied to archived data).

- **Functional** – Software installation must stay operational
- **Unaltered** – Software must not be changed
- **Secure** – Software installation must not present risks from cyberattack and confidentiality breaches
- **Accessible** – Software installation needs to be easily accessed for repeated use.

Figure 5 Data Integrity and Software Integrity definitions.

D. Summary

The bioanalysis research use case highlights two key issues:

1) The success of the preservation process is dependent on the data file fixity but the preservation and demonstration of the significant properties are subject to the software integrity, i.e., ability to re-compute the data and reliably reproduce the results.

2) The regulatory requirements mandate the archiving of original software alongside the data, clearly recognizing that the capability of data presentation and data analyses is not in the file format but in the computation of the raw data files.

One may argue that the preservation of final study reports, e.g., using standardized rich file formats with imbedded data, should be an alternative approach, assuming that there exist reliable and regulated standardized readers. Unfortunately, normalization of raw data and data analysis formats across instruments is difficult to achieve, if not infeasible. Furthermore, one cannot underestimate the challenge of proving that a substitute software (reader) can reliably produce the same results as the original, nor can we easily determine the impact that invalid results may have. The latter was recently illustrated in a highly reported case of Public Health England, failing to account for thousands of Covid cases due to a software versioning problem [2].

III. Related Work

The importance of digital objects authenticity and preservation of software has been recognized by the digital preservation community and led to research efforts dedicated to developing effective methods. Here we provide a brief overview of the past work relevant for framing our research effort and contributions.

A. Preservation of Significant Properties

The term 'significant property' has different interpretations in literature. Open Archival Information System (OAIS) standard [20, 21]) defines it as an information property that is necessary for preserving the information content across any non-reversible transformation, while PREMIS [11] refers to it as a specific set of meta-data attributes required for rendering a file or a digital object. Both definitions emphasize the link between significant properties and authenticity of digital artifacts, but also the subjectivity of their choices.

The subjectivity is a result of a specific domain's assumptions of what is necessary or worth preserving. For example, preserving colors may seem important for an art eBook but not necessary for a history eBook in which case it is sufficient to preserve words, punctuation, and paragraph separation. Moreover, in Digital Arts, the definition of significant properties is expanded outside of the file-related attributes to include behaviors, rules of engagement, and visitor experience amongst others [2].

In the context of our use case, the preservation of significant properties relates to the ability to reproduce a scientific study rather than a digital object. The data analysis is instantiated by re-computing the raw data. One may thus argue that, according to the OAIS

interpretation, the only significant properties are the stored results of the study or their selected subset; more precisely, the input- output dataset of the archived study. However, this interpretation does not take into account the requirement of preserving the operational environment. In that context the PREMIS meta-data interpretation of the significant properties is more suited, with relevant attributes spanning the characteristics of data, network, and software components of the preservation environment.

As suggested by Matthews et al [9], besides the significant properties of the input dataset, e.g., attribute-value pairs and instance numbers, one needs to consider additional data such as characteristics of the network (e.g., the security protocol) and the software (e.g., functionality, composition, ownership, and other properties defined in [9]). In our use case, the necessary meta-data about the software are included and verified through specific 'qualification' procedures (Fig. 9), before the software is transitioned to the Software Library platform. The qualification procedures are closely linked with the practices of maintaining software during its operational use when it was critical to ensure that the operational processes (e.g., pre-clinical research or manufacturing) produced quality data. The goal of the qualification procedures is to guide the installation process so that the archived software installations produce outputs consistent with a predetermined quality. The choice of significant properties remains a major research question for the preservation community in various domains, including digital games [7], and is a pre-condition for selecting an optimal preservation strategy.

B. Validation of Software Installations

The efforts required to enable stable installations and provide ongoing maintenance, to keep the software operational, results in a significant cost. While in other industries the maintenance cost is estimated to be between 10 and 25 percent of the total operating costs [14], software maintenance contributes to a much higher percentage of the total software life cycle cost (e.g., 66% quoted in [16]). In fact, the high cost of maintenance has been identified as one of the key external factors that contribute to software aging [25]. According to the same study, software aging metrics include not only performance, usefulness, business demand, environment, and technology change but also a need to retain and train experts.

The same applies beyond the typical software use period, i.e., when both the data and the software need to be archived. This need is heightened with premature software aging as software release cycles are becoming shorter and shorter [1].

Development of service-based software models, replacing the product view of the software, has been recommended in late nineties [24] as a step forward in reducing the cost of ownership. Since then, various 'as a Service' models have emerged such as SaaS, PaaS, IaaS to mention a few. The Executable Archive framework is, in effect, a software-as-a-service model, with fully managed hosting of virtualized software that belongs to the user, i.e., the user's organization.

C. Long Term Software Management

Aging of software typically involves two technical factors: the deteriorating hardware and unsupported (i.e., non-secure) operating system. Virtualization can assist with both. The technique allows a user to execute their software application in an operating environment different from the host system, taking advantage of the host hardware. This has a broader applicability, addressing the incompatibility of software with different operating systems. For example, software such as Microsoft Project that does not have MAC OS binaries can be run on top of a VMWare virtual machine on a MAC machine. By reducing hardware/software dependencies, virtualization enables cloud-based provision of services and more efficient and productive software maintenance [9]. In other scenarios it assists with prolonging the life of installations that involve software, such as modern sculptures and digital arts, where software is an integral part of the artefacts [2].

The term virtualization is sometimes used interchangeably with emulation. Emulation-As-A-Service Infrastructure (EaaSi) [14] is a service that enables preservation practitioners in memory institutions to install software in virtual machines, pre-configured with required legacy operating systems.

Both methods allow the software code originally developed for one system to execute on another. However, they differ in key technical points:

- Emulators interpret the source code into the CPU instructions of the host machine, while in virtualization, the original code (binaries) is executed in a 'container' process that provides a bridge between two operating systems.
- Emulators are slower compared to virtualized applications.

From our perspective, an important difference is that virtualization aims to provide a generic execution

environment for any application (e.g., enables any application that requires Windows environment to run on a MAC server). Emulation, on the other hand, provides a bridge between a specific application and the host hardware, e.g., enables an old Atari game to run on a Windows laptop.

Complementary to software virtualization and format migration, is software modernization, i.e., software porting to new coding platform. Software Sustainability Institute [24] has focused on techniques and methods for creating research software to enable effective and sustainable use of software code that support research tools. This approach is effective in coordinated community initiatives but does not scale to a full ecosystem with a large proportion of commercial and proprietary software technologies.

D. Software Use

Perhaps, the most important characteristics of our use scenario, in contrast to ongoing software preservation initiatives, e.g., EaaSi and Internet Archive (www.archive.org), is the active use of software to accomplish a specific task with data from secure data repositories. The Executable Archive framework supports a complete workflow from virtualization and validation of software to secure data import and remote access to virtual machines running non-secure software.

Furthermore, software virtualization techniques are also subject to aging, i.e., lack of support. Thus, it is key to put in place processes for ongoing risk assessment and installation updates. We adopted Xen virtualization technologies provided by Citrix and carefully manage a range of aspects [6]:

- Licensing and cost issues, as the license is required for all virtualized operating systems. A suitable range of host platforms and operating systems need to be supported.
- Performance might be an issue in the environments where near real-time performance is expected.
- Aging and maintenance of the virtual platform itself need to be carefully monitored and planned for.

IV. EXECUTABLE ARCHIVE FRAMEWORK

The Executable Archive framework is intended to support design of solutions for access and use of archived digital data. It includes (1) technical components for managing and using software and (2) processes and procedures for platform, installations, and user management, as shown in Fig. 6. We used it to design and implement the Software Library platform and services for the reconstruction of archived research studies by bioanalysis researchers in order to pass GLP compliance audits [23] (Fig. 8).

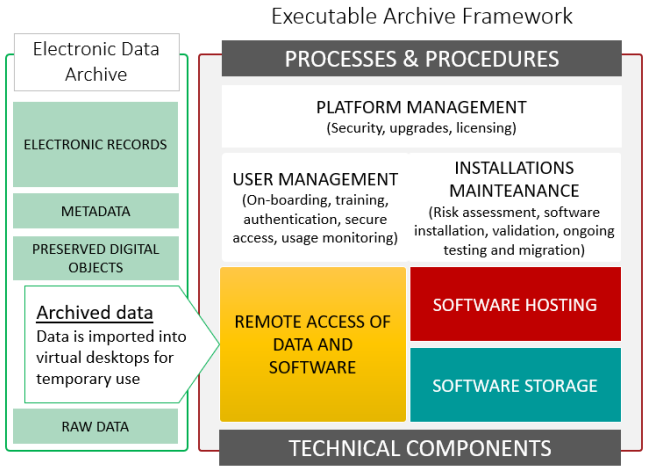

Figure 6 Executable Archive framework comprises technical components and processes and procedures that complement traditional electronic data archives.

We note that the archiving of study data follows a well specified procedure and shared practices adopted by researchers and archivists involved in the study deposit process. The deposited data includes metadata that enables researchers and archivists to locate the specific study very efficiently within the record management system (Fig. 7). The system includes contextual information of the study and the accompanying documents in a standardized format, most often PDF.

The reported graphs and statistics, derived from the raw data analysis, need to be reproduced. Reports, typically in the PDF file format, are different digital objects from the raw data files. Data characteristics are revealed only through the computation, i.e., the use of the analysis software that renders the analysis results on the screen (Fig. 8). Thus, the emphasis is on the properties of the software and, therefore, on the well-controlled process of software installation and validation. This required special care as the required operating system may be unsupported (e.g., Windows XP SP2, Windows 7), thus non-secure.

Further important aspect is the separation of the virtualized software, hosted on the Software Library platform, from the archived data repository. Since legacy software installations cannot be exposed, i.e., connected to the organizational network, one has to isolate both the archive and the software installation, or extract data from the data repository and bring it into the environment with virtualized software. The latter approach was

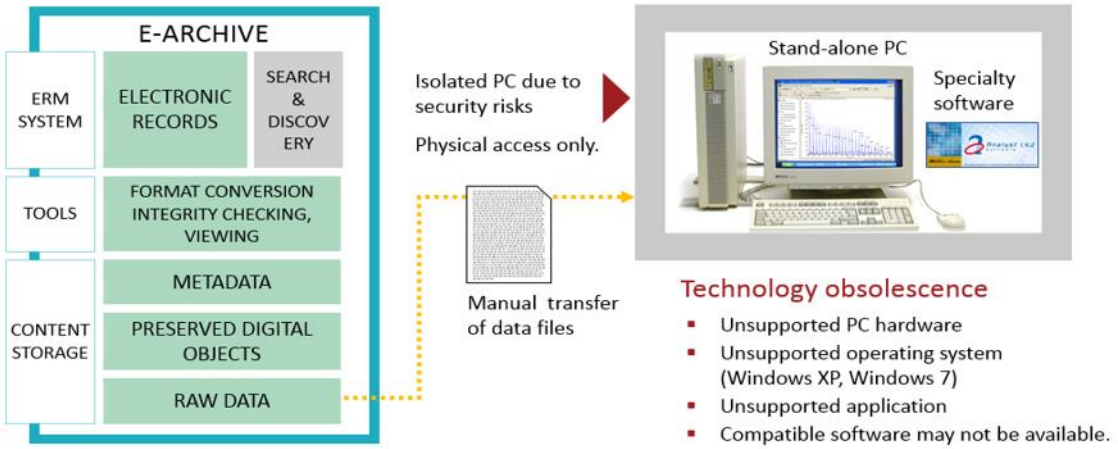

Figure 7 Components and data access in traditional 'PC with software installation' preservation case.

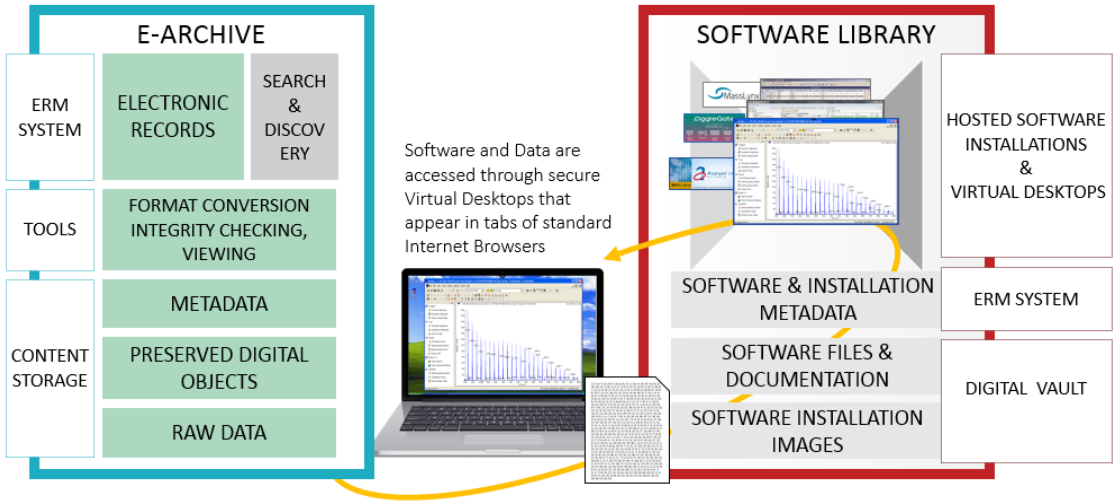

Figure 8 Components and data access in the proposed preservation framework with a Software Library hosting virtualized software.

deemed more appropriate. Thus, support for the data export and transfer had to be carefully designed and implemented.

Both the technical design and the procedures supported by the Executable Archive framework present novel contributions to the preservation practices in general, and solve a critical issue in the preservation of scientific results in particular. Implementation of the Software Library platform and services follows a software-as-a-service model (SaaS) with a fully managed and remotely used collection of virtualized software installations. Access to the archived data repository is configured for secure transfer and use within the running software sessions. The concept is applicable in general by archives that have data integrity and long-term data access requirements.

It is important to note that our framework supports a systematic capture of knowledge about operation and use of software installations in two ways: (1) through regular training and on-boarding of users (e.g., regulatory inspections are every 2 years and staff cannot use the software library without going through training) and (2) through installation and maintenance processes that include detailed documentation on operational and performance monitoring procedures.

Furthermore, the Software Library design is informed and optimized to address user productivity issues caused by technology obsolescence. With modernization of research facilities, new instruments are adopted and previous ones are decommissioned, including the corresponding analysis software. Since software is needed to read the archived data, the old hardware with software installations must be kept usable (Fig. 7). Such devices are isolated from the main organizational networks due to the non-secure operating system. That, in turn, affects the way the software and the archived data can be used in research and during compliance audits. If the data

archive is on the main organizational network, the old PCs should not directly interact with it. The archivist thus needs first to export data from the archive and place it on a medium that can be read by the PC, e.g., a USB stick or external hard drives. Besides the impact on the users' productivity, this transfer of data raises two key concerns: (a) one has to guarantee that the data are not changed during the transfer and (b) data should not be left on the portable devices or on any other PC. Considering the data readability and Software Library design, that means that we must (1) create software installations that are for all practical purposes equivalent to the original PC installation and (2) provide a mechanism for easy input of data into the virtual machines with legacy and unsupported operating system (Fig. 9). Both of these are achieved and illustrated in the case study that we describe in the next section.

V. CASE STUDY: REPRODUCTION OF ARCHIVED RESEARCH STUDIES IN BIOANALYSIS

In our case study, the software package Analyst 1.4.2 produced by Sciex had to be decommissioned as the organization stopped using the corresponding lab instrument. The studies were produced in the period from 2006 to 2015 at which point a different product was adopted. Thus, readability of all the studies over the period of 9 years is affected if the software is not in use anymore. Up to 2020, the data readability was achieved by maintaining an isolated PC with the original copy of the Analyst 1.4.2 installation. This is a common practice but not sustainable due to possible hardware failures. Thus, the decision was to eliminate the dependence on the hardware component and adopt virtualization.

A. Software Installation and Validation Approach

In a private data centre, we

- Create a sandboxed VM environment to enable installations of Analyst 1.4.2 software with WinXP SP3 operating system. (~2 hour work)
- Enable upload of the software into the Software Library environment. (~1 hour work)
- Follow the original installation instructions, applied to the installation of the software on the lab PC. These instructions are referred to as Installation Qualification (IQ). (~2 hour work)
- Document the process of installing the software in the VM. The new documentation is referred to as Software Library IQ (SL-IQ) indicating that the installation is virtualized. (~2 hour work)

This first part of the installation process represents a critical phase of addressing and documenting all the adjustments of the archived installation in comparison with the original installation, e.g., single-user vs multi-user installation, security settings for a stand-alone vs networked installation, user authentication, software activation, and related. If the rest of the process proves to be successful, SL-IQ becomes a blue-print for all other subsequent installations.

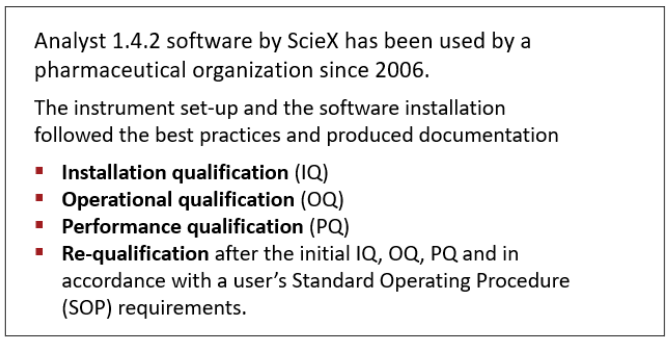

Figure 9 Software qualifying procedures for Analyst 1.4.2 installation.

The next stage requires researchers to test the features of the installed software in the VM. That involves specifying the task and configure a Virtual Desktop to support the user task. The researcher's effort then includes (a) a review of the documentation of the original software validation, referred to as Operational Qualification (OQ) documents and (b) a selection of the software features that support the study reconstruction task and must be tested. The result of this process is SL-OQ, i.e., operational qualification criteria for the evaluation of the virtualized installation of the software. The researchers

- Describe the study reconstruction steps in detail and select a sample data set. (~4 hour work)
- Perform the study reconstruction steps and compare with the OQ documentation and expected outcomes. (~2 hour work)

The researchers also create a short test that can be used just to test that the software has not changed between usage. Similar tests are performed on the original software installation from time to time and are referred to as Performance Qualification (PQ). Thus,

- Researchers decide on the minimal set of interactions with the virtualized software to establish that the Software Integrity is intact. The resulting set of actions is referred to as Software Library PQ (SL_PQ) and is applied every time the software is used, before importing the real data.

– Researchers document the outcomes of the SL-PQ based on the software screenshots. This document is used as a reference in all use scenarios, including the compliance audits.

B. *Software Installation and Testing of Analyst 1.4.2*

DEV stage involves the Sandboxed VM, using SL-IQ instructions for Analyst 1.4.2 to ensure that the installation is as close to the original as possible. Controlling the installation process serves as assurance that even the features that have not been tested explicitly are likely to stay functional as with the original installation.

TEST stage involves Virtual Desktop access to the Analyst 1.4.2 and enables the user to use the data attached to the VM to apply SL-OQ and SL-PQ procedures. All the outcomes are compared with the same test run on the PC in the Lab which is still functional.

PROD stage involves the final release of the software for use on the Software Library platform (Fig. 11). The testing of the PROD environment is conducted by the IT staff to confirm the performance parameters that were already established in the TEST phase, e.g., the speed of upload, movement of data on to the Analyst 1.4.2 VM, decompression and checksum testing of the data.

C. *Study Reconstruction Test*

A test of the full study reconstruction with Analyst 1.4.2 (Fig. 10) was performed using SL_PQ procedure and data exported from the archive. Data contained a large collection of studies (1 GB compressed, 9GB decompressed) which could not be separated into individual study files due to the organization of files by the Analyst Software. Particularly important was to ensure that all audit files associated with the data could also be viewed with the virtualized software installation. The researchers advised that the complete audit files can be viewed only when stored on the C: drive of a device. Thus, it was important to consider the speed of data management: data upload into the Software Library platform, checksum verification of the zipped file, data copy to the destination C: drive, and data decompression. Data transfer to the Software Library platform took 1 min using MOVEit data transfer service [17]. Copying zipped file to C: drive of the Analyst Desktop took about 2 min. The checksum took <20 sec and the decompression about 10 min. Thus, within less than 14 min, a large data collection was ready for inspection. Changing the order of data preparation, e.g., uncompressing the zipped file before moving to the C: drive increases the time by ~ 40min.

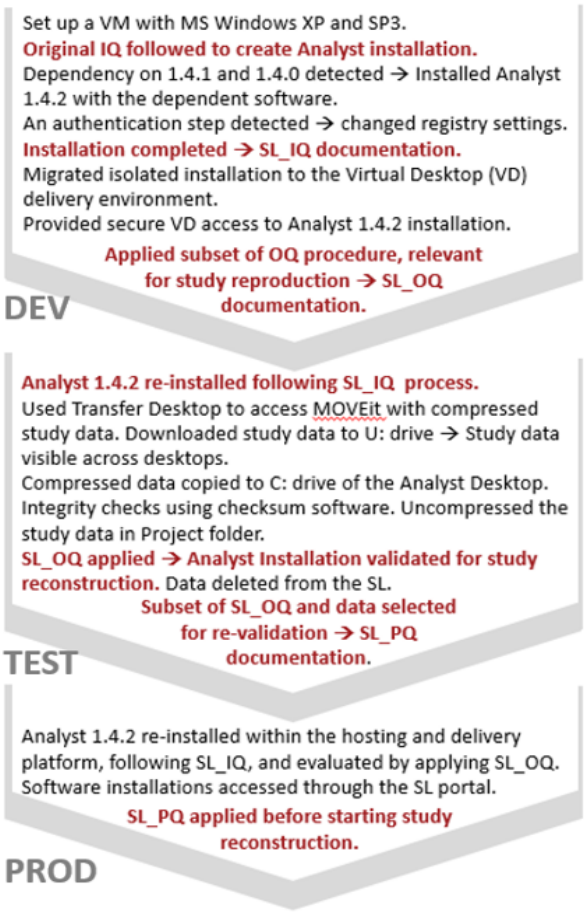

Figure 10 Phases in the installation and validation process (MOVEIt is a data transfer service [17]).

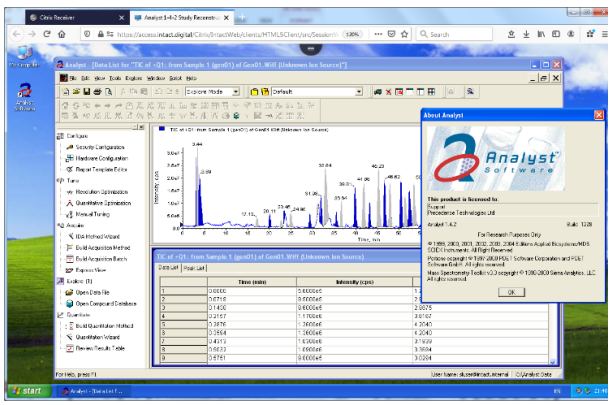

Figure 11 Reconstruction of a research study is enabled using a validated virtualized installation of Analyst 1.4.2. Shown data file is dated Dec 2001. It is part of the data sample that comes with the software release. The Analyst 1.4.2 version is from 2004.

VI. CONCLUSIONS AND FUTURE WORK

In this paper we presented a case study of archiving practice that requires a different approach to defining and enforcing the preservation of significant properties. Since research studies must be reconstructed from raw data, the reproduction of results requires re-

computation of the data. Thus, it is the software properties that determine the outcome. That, in turn, calls for introducing Executable Archives as an extension of the traditional archive with a Software Library platform that hosts virtualized installations of the required software, i.e., data readers.

The validation of virtualized software installations closely follows the software installation practices that are enforced by the companies deploying and maintaining the software during its operational time span. These procedures are adapted to the VM hosting environment and serve as key mechanisms for maintaining the integrity of legacy software installations over time. We demonstrated the technical feasibility of hosting and remote use of installations. The method is effective, fully compliant with organizational policies, and aligned with established validation practices. It does not require any changes to the data or software. In fact, it is devised to ensure both data and software integrity.

Going forward, we advise to optimize the process further by adding software to the Software Library at the time it is first deployed and subsequently upgraded. That has two advantages: (1) the validation process need not be performed (again) at the time of software decommissioning and (2) up-to-date Software Library is aligned with the archived data and content, providing validated software for data reading and processing.

REFERENCES

[1] Abdullah, Z. H., Yahaya, J. H., Mansor, Z. & Deraman, A. (2017). Software Ageing Prevention from Software Maintenance Perspective–A Review. Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 9(3-4), 93-96.

[2] Ashe, A., Falcao, P. & Jones, B. (2014) Virtualization as a Tool for the Conservation of Software-Based Artworks. In Proceedings of the 11th International Conference on Digital Preservation (IPRES), Melbourne, Australia, October 6-10, pp. 83-90.

[3] BBC News (2020) Covid: Test error 'should never have happened' – Hancock. Available at: https://www.bbc.co.uk/news/uk-54422505

[4] Bennett, K. H., & Rajlich, V. T. (2000, May). Software maintenance and evolution: a roadmap. In Proceedings of the Conference on the Future of Software Engineering (pp. 73-87).

[5] Celebi, I., Dragoset, R.A., Olsen, K.J., Schaefer, R. & Kramer, G.W., (2010). Improving interoperability by incorporating UnitsML into markup languages. Journal of research of the National Institute of Standards and Technology, 115(1), p.15.

[6] Consultative Committee for Space Data Systems. (2012). Reference model for an open archival information system (OAIS).

[7] Emulation-as-a-service (EaaSi) https://www.softwarepreservationnetwork.org/wp-content/uploads/2020/06/UI_Use-of-EaaSI-at-Uofl-Libraries.pdf

[8] FCA (2021) FCA Handbook. Available at: https://www.handbook.fca.org.uk/

[9] Gates Open Research, https://gatesopenresearch.org/

[10] Giaretta, D., Matthews, B., Bicarregui, J., Lambert, S., Guercio, M., Michetti, G. & Sawyer, D., (2009) Significant properties, authenticity, provenance, representation information and OAIS information. iPRES, San Francisco 5&6 October 2009

[11] Shamir, J. (2021) 5 Benefits of Virtualization, IBM Cloud, Available at: https://www.ibm.com/cloud/blog/5-benefits-of-virtualization

[12] International Aerospace Quality Group standards https://iaqg.org/

[13] Internet Archive https://www.archive.org

[14] Matthews, B., McIlwrath, B., Giaretta, D., & Conway, E. (2008). The significant properties of software: A study. JISC report, March.

[15] McDonough, J.P., Olendorf, R., Kirschenbaum, M., Kraus, K., Reside, D., Donahue, R., Phelps, A., Egert, C., Lowood, H. & Rojo, S. (2010) Preserving virtual worlds final report. Available at: http://www.ideals.illinois.edu/bitstream/handle/2142/17097/PVW.FinalReport.pdf

[16] McKinsey & Co., (September 1, 2012) Planning to fix: improving maintenance efficiency, Available at: https://www.mckinsey.com/business-functions/operations/our-insights/planning-to-fix-improving-maintenance-efficiency

[17] MOVEit www.moveitmanagedfiletransfer.com

[18] OECD (2007) OECD Series on Principles of Good Laboratory Practice and Compliance Monitoring No. 15: Establishment and Control of Archives that Operate in Compliance with the Principles of GLP, OECD, Paris, 2007.

[19] OECD (2016) OECD Series on Principles of Good Laboratory Practice and Compliance Monitoring No. 17: Application of Good Laboratory Practice Principles to Computerized Systems, OECD, Paris, 2016.

[20] PREMIS (2015), PREMIS Data Dictionary for Preservation Metadata, version 3. Available at: http://www.loc.gov/standards/premis/

[21] RECOMMENDED PRACTICE CCSDS (2012) RECOMMENDED PRACTICE CCSDS 650.0-M-. MAGENTA BOOK June 2012 CCSDS Secretariat.

[22] Software Sustainability Institute https://www.software.ac.uk

[23] The Open Research Data Task Force (UK) https://www.universitiesuk.ac.uk/policy-and-analysis/research-policy/open-science/Pages/open-research-data-task-force.aspx

[24] Yahaya, J. H., Abidin, Z. N. Z., & Deraman, A. (2015). Perspective and perception on software ageing: The empirical study. In 2015 10th International Conference on Computer Science & Education (ICCSE) (pp. 365-370). IEEE.

[25] Yip, S. W., & Lam, T. (1994). A software maintenance survey. In Proceedings of 1st Asia-Pacific Software Engineering Conference (pp. 70-79). IEEE.