https://doi.org/10.2352/issn.2168-3204.2023.20.1.24

Archiving: Vendor lock-in or "complicated" conformance?

Anssi Jääskeläinen; South-Eastern Finland University of Applied Sciences / Digitalia research center, Mikkeli, Finland. Karin Oolu; Tallinn University / Estonian Academy of Arts, Tallinn, Estonia. Andres Uueni; MUKOLA Lab, Estonian Academy of Arts, Tallinn, Estonia

Abstract

National archives set and implement national policies, recommendations and legislation for archiving standards which governmental and public actors need to obey. However, smaller actors in the field do not have this obligation and often they do not possess the resources or know-how to follow the set guidelines or to implement their own digital preservation workflows. This easily leads to the utilization of commercial information management solutions, which might lead to a vendor lock-in situation. In the worst-case scenario, the software is merely a CMS or ERMS without any adherence to digital preservation standards. The OneClick SIP creator presented in this paper responds to this challenge. As a very welcome side effect, it also makes it easier to disengage from vendor lock-in situations by simplifying the creation of compliant information packages.

Motivation

Even nowadays it is not uncommon to encounter IT systems, such as CMS (Content Management System), CRM (Customer Relationship Management), ERMS (Electronic Records Management System) which do not have sophisticated data export functionalities. This exact situation is still going on at the South-Eastern Finland University of Applied Sciences. Users with normal access rights can only download payload files without the connected metadata and a plain data without the relevant metadata is close to useless. Furthermore, that system does not even allow the administrators to export data with the related metadata easily. Situation can be even worse, there are systems out there that lack all forms of data export functionalities. For such cases, the only general solution is money. This was the case with our project partner EKA (Eesti Kunstiakadeemia, Estonian Academy of Arts), ERMS and CMS systems. Significant amount of time and money was spent before the data was successfully exported from the systems.

The above-mentioned situations are called vendor lock-ins where the user or client is fully dependent on the products or services of a vendor without an easy exit. Such vendor lock-in situations can have a long history. Helein [1] for instance, described in 1984 that "Software lock-in allegedly occurs when a customer creates applications programs tailored to a particular operating system". Information sharing between different systems in such cases is commonly done with custom integrations, which leads to an even tighter vendor lock-in situation. Shaanika et. al [3] describes this situation within the Namibian government. They have studied the implications of information systems vendor lock-in to the governments' digital transformation efforts. They conclude that there can be implications on interoperability between systems and a negative effect on the efficiency of service delivery within the public sector. Thus, such a vendor lock-in situation can happen at the governmental level, it can happen anywhere.

If taken seriously, Archiving by design (Abd) approach [2] is one of the possible future solutions for this problem. However, the field outside digital preservation does not necessarily recognize its importance, thus archiving is something that is traditionally considered at the end of the lifecycle. This attitude is something not easily changed. The basic principle of the Abd approach is to include a digital archiving consultant as part of the team designing the actively used system. However, Abd approach cannot help with existing proprietary systems that already contain information. When tendering or negotiating with the vendors, requesting the utilization of OAIS (Open Archival information System) model and conformant IPs (Information Package) are also good starting points.

Challenge after another

The first challenge relates to a simple data export functionality. There seems to be a contradiction between vendor and customer opinions. For example, Opara-Martins et al. [4] lists export functionality as one of the key metrics when measuring customer retention and engender trust. They also state that customers need to understand how data can be imported and exported. However, some proprietary solution providers still seem to consider export functionality as a way to lose customers. Therefore, getting the data out from a proprietary vendor locked system might be a difficult and time-consuming challenge.

This was the situation with the OneClick eArchiving project partner EKA who partners with Estonian vendors. EKA collections were inside very well-functioning proprietary systems, and everything was running smoothly. The challenge became obvious when a need to archive the collections properly arisen. This was realized in a form of missing export functionality. During the OneClick project, vendors were subcontracted for EKA and they provided the exports of the requested data on a case-by-case basis. In each case, they had to implement some extra workflows and features into their system to be able to get the requested data out. This naturally caused expenses.

One of the targets of the OneClick project was to be able to migrate these EKA collections exports into E-ARK/OAIS conformant SIPs (Submission Information Packages) without compromising any metadata or content. This formed the second major challenge; proprietary systems tend to use their own unstandardized data structures. EKA systems were not exceptions. The bulleted list below presents the general structure of the ERMS Webdesktop export package as it was received from the vendor.

- *57835/*
  - *unit_58119/*
    - *object_type_name/*
      - *XML/*
        - *XXX.xml*
        - *ABCD.xml*
      - *files/*
        - *1234*
        - *98766*

In this structure 57835 is the main client ID from the system perspective and the unit_58119 is the organization ID. Object_type_name folders are the topic root folders which were named according to their content. Each folder was populated with XML and files folder which both contained thousands of files. The only linkage between the XML metadata file and the actual file was a reference number or multiple reference numbers inside the XML file. In addition, folders outside the topic root folder might also contain files related to this certain topic. Furthermore, the pilot export contained all the payload files without their names and endings; in other words, the payload "filenames" were just a series of numbers. IDs, file names and file endings as well as other related metadata was described inside an XML metadata file.

If this structure had been preserved as it was, it would have been quite cumbersome to try to figure out the structure for example after 20 years of preservation. Therefore, the third challenge was to normalize the structure and naming. This step was done by EKA with a Linux shell script. After this step the structure was as follows, which already looks like a preservation worthy.

- *object_type_name/*
  - *XXX.xml*
  - *ABCD.xml*
  - *XXX/*
    - *real_file_name_1.ext (was 1234)*
  - *ABCD/*
    - *real_file_name_1.ext (was 98766)*

EKA CMS export package of course had a different structure thus being a product of another company. The bullets below present the structure of this CMS export package.

- *category_object.sql*
- *category_object.xml*
- *description.sql*
- *description.xml*
- *event_objects.xml*
- *Metfond.xml*
- *object_event_1.sql*
- *object_event_images.sql*
- *object_event_instructors.sql*
- *object_event_pub_authors.sql*
- *object_events.sql*
- *object_events.tsv*
- *object_events.xml*
- *AAA/*
  - *object_event_1.tsv*
  - *object_event_images/*
    - *BBB/*
    - *CCC/*
      - *filename.ext*
  - *object_event_images.tsv*
  - *object_event_pub_authors.tsv*
  - *title_image/*
    - *filename.ext*

During the project it was agreed that this structure is tidy enough to be preserved. Within the structure, Metfond.xml contains the export package-level metadata, including a description of the collection and a few details about the contents. The SQL files are included for reference and the TSV files contain all deeper-level metadata such as general events and object related events. In the structure AAA is the main object/event ID, CCC is the payload foldere ID. BBB was used to match the file location to the CMS web server.

The fourth part of the challenge relates to migrating the exported data into a preferred ingest format, which in this case means conformant SIP packages. It needs to be clarified that it was decided among project partners that the SIP creation stage does not convert the payload files into archival graded files. PDF files for instance will not be converted into PDF/A files. This decision was made thus generally this step is taken care of when SIP packages are ingested into OAIS conformant digital repository. Even with this simplification, creating conformant SIPs is not a simple task and smaller actors in the field rarely have the resources or know-how to implement SIP creation workflows. Furthermore, adding those into a proprietary system would cost money and break the vendors' precious lock-in.

Yet another possible challenge is related to the variety of export formats. In the best-case scenario, the export package is based on open standards and specifications which means that the migrations and conversions are simpler. However, in the worst-case scenario the exported content could be some vendor and system specific proprietary binary format without instructions on how to extract the actual payload and metadata. In such cases there are only a few possible solutions; reverse engineering or again a cash flow to the vendor.

These challenges can together easily lead to a situation where valuable information is "preserved" in a variety of actively used everyday systems. Such systems include but are not limited to CMS, CRM, ERMS and cloud drives where there is little or no adherence to any digital preservation standards. In such systems, the digital content evolves continuously, hardware and software components change and content file formats can become corrupted, inaccessible, obsolete, or even accidentally deleted. No matter what the vendors might claim, these systems are not suitable for preservation purposes without a proper background archival extension that is seamlessly integrated into the active system. Some vendors offer these true validated repositories, but others don't and then the only practical solution is export and preserving elsewhere.

Approach

In EKA, there is no specific digital preservation policy document that provides guidance and authorization on the preservation of digital materials to ensure their authenticity, reliability, and long-term accessibility. Content is being used and stored mainly within two digital systems that are both in active use. ERMS Webdesktop and CMS Digiteek. These comprise a broad range of content, including digitized materials and born-digital resources. These are the two biggest collections of EKA containing different varieties of materials, some of which require urgent long-term preservation. Types of digital materials include textual documents in both systems and multimedia content in Digiteek. Both collections, but especially Digiteek, will likely acquire materials in additional formats in the future.

Therefore, it was a mandatory step to create a digital preservation plan which describes the workflow from the active systems to a long-term preservation system by utilizing OAIS

conformant SIP packages. Preservation policy was designed to be format and type independent to be able to accommodate new formats with the minimum possible effort.

Based on the digital preservation plan the following simplified workflow for migrating EKA materials was designed and implemented

- Extract the content from the systems with aid from the vendor
- Create SIPs by using OneClick SIP Creator
- Create AIPs (Archival Information Package) and DIPs (Dissemination Information Package) by using EARK Archivematica AIP and DIP Creator

The rest of this paper focuses mainly on the second bullet point and describes the SIP creation workflow. More information about the Archivematica AIP and DIP creator can be found via YouTube[1] tutorial.

SIP creation part of the project was divided into five different tasks from which the first and second tasks are related to the first bullet point and numbers 3 to 5 are related to the actual SIP creation process.

1. CMS/CRM/ERMS exports investigation
2. Export extraction into file/folder structure
3. Metadata retrieval
4. Creation of SIP from file/folder structure
5. Verifying the SIP

Results

The project started with an examination of different export packages and formats that generally used active systems produce. This report was delivered to project funder HaDEA (European Health and Digital Executive Agency). The main finding of this investigation revealed that export packages and structures vary greatly. This finding confirmed that the utilization of CITSs (Content Information Type Specifications) would have required enormous amount of work for each file type. This is a kind of work that is best to leave for bigger projects and consortiums, such as the ongoing EARK-CSP. Furthermore, it was already defined in the funding application that the SIP creator will be simple to use and universal one click solution. Therefore, it was decided that the exported content, files, folders, etc. will not be altered during the SIP creation and no CITS will be used. Possible structural modification must therefore be conducted before the SIP creation stage.

In the EKA export case the above-described modification to the exported content was made. After this extra step tasks three to five happen seamlessly together with a tool called Oneclick SIP Creator. This tool can be found at xamkfi[2] Github and tested via Xamk Digitalia demo site[3]. Taking the dockerized SIP creator into use is probably the most difficult part of the process and might require an assistance from your IT support. After this step everything else can be done via simple browser UI which is presented in Figure 1. In this solution, the upload is done with PHP and the backend processing happens with Python. Everything is placed inside a docker container for simple transferability and taking into use. The installation instructions and tutorial videos linked on the GitHub page helps to get started with SIP creator.

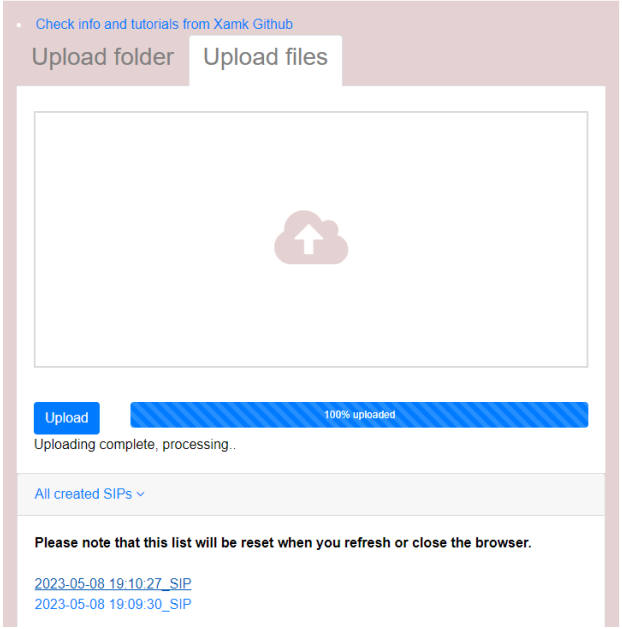

***Figure 1**. Oneclick SIP creator UI on Edge browser*

The SIP creation process, which happens hidden from the user, starts with a successful data upload This can happen via any available method as long as the uploaded content ends up inside the folder watched by the Python watchdog library. The creation event starts a subprocess which monitors the actual upload and does not proceed until the upload is completed. The first step of the actual SIP creation process is to check the uploaded content for viruses, malware, etc. This is done with the open source Clamav antivirus engine. After a successful check, the basic SIP structure with the required universal identifiers is created.

Gathering the required SIP metadata takes place next. OAIS SIP package supports a few different metadata schemas. The OneClick project has chosen Metadata Encoding and Transmission Standard (METS) and Dublin Core (DC) due to simplicity. Metadata gathering task simply goes through the uploaded content, seeks metadata files and uses information within those files to populate the METS.xml and DC.xml file fields. This task also includes many invisible, but mandatory steps such as checksum calculations, file modification time, size and mime type retrieval, date format fixes, and finally content language detection if requested. The ultimate idea is to provide as much information as possible about the payload content within the SIP package metadata files without overloading the files with unnecessary originating system related metadata.

[1] https://www.youtube.com/watch?v=aXgCv_Zlrps (2023/05/08)

[2] https://github.com/xamkfi/Digitalia-oneclick-full (2023/05/08)

[3] https://digitalia.xamk.fi/oneclickUploader/uploader-main.php (2023/05/08, note that this address is likely to be changed due to being a simple demo site)

The mime type detection method utilizes mostly Apache Tika and uses Exiftool and Python mimetypes libraries as a backup method. This method in some cases produces results that are too accurate and the created SIP package failed to validate correctly. This is because the utilized SIP validator uses the IANA media types lists[4], which does not contain all possible mime type extensions. Workflow identified mime types such as "image/x-canon-cr2", "application/vnd.ms-pki.seccat" and "Value audio/x-pn-realaudio" which cannot be found in the IANA list. These types were added to a list of replaceable mimes and during the runtime these are replaced with a generic "application/octet-stream" definition. This accuracy reduction then leads to a valid SIP package, however the authors would strongly advice in changing the validators and lists instead of reducing detection accuracy.

When all the mandatory and requested steps are taken and the SIP folder structure is completed, it is zipped into one container file which is the final SIP package. This package is validated by utilizing an existing SIP validator, CommonsIP[5]. At the time of writing the latest version is 2.3.1 but the development and the tests were made against earlier versions from 2.1 to 2.3. In addition to the commonsIP verification, the created SIP packages were verified by importing those successfully into two different compliant digital repositories, Roda and Archivematica. As a proof of concept, also two multi gigabyte export packages from EKA were successfully migrated into SIP packages. After a successful validation the SIP package and the validation report are zipped into one package for simple transferability. So if, and hopefully when, the demonstration version is tested by the reader, bear in mind that the downloadable package is not the SIP file, it is a container with the validation report and the actual SIP file.

Conclusions

Vendor lock-in is a common problem which is hard to tackle, but the wider utilization of conformant IP packages is one way to face this challenge. In wider scope, the availability of E-ARK/OAIS compliant repositories is a key requirement for interoperability and wider outreach. This paper demonstrated that creating conformant SIP packages doesn't have to be related to archival know-how or tight integrations with the existing proprietary systems. The main goal of the work behind this paper was to automate the SIP creation process by designing, implementing and testing an input independent SIP creator solution. This target was reached and the result is publicly available via above mentioned xamkfi Github.

It can be questioned why an existing SIP creator tool was not used for this project. Few open-source solutions for this purpose exist, such as Roda-in[6], ESSArch[7] and the earkweb[8] reference implementation. However, these applications are meant to be used by digital preservation professionals with adequate knowledge of how the packages are created, what is the workflow and what is expected format of the metadata. Furthermore, the UI and vocabulary of these applications can be somewhat difficult to understand for non-professional users. Finally, setting up such a system as on-premise installation requires technical skills which the smaller actors generally don't possess. These are the reasons why Oneclick eArchiving project got funded. The goal was to implement a simple to use one click solution for creating conformant SIPs.

References

[1] C. H. Helein, "Software Lock-In and Antitrust Tying Arrangements: The Lessons of Data General", 5 Computer L.J. 329 (1984)

[2] E. Saaman, "Archiving by design", nationaalarchief.nl. https://www.nationaalarchief.nl/en/archive/knowledge-base/archiving-by-design (accessed 07/02/2023)

[3] I. Shaanika, G. Nhinda and K. Amunkete. "Assessing the implications of ICT Projects Vendor Lock-in to the Namibian Government Digital Transformation", Nov 22, doi: http://dx.doi.org/10.2139/ssrn.4332737

[4] J. Opara-Martins, R. Sahandi, and F. Tian. "Critical analysis of vendor lock-in and its impact on cloud computing migration: a business perspective", *J Cloud Comp vol* 5, no. 4 (2016). doi: https://doi.org/10.1186/s13677-016-0054-z

Author Biography

Anssi Jääskeläinen has an IT MSc. (2005) and a PhD (2011) from the Lappeenranta University of Technology. He has an extensive knowledge of user experience, usability and programming. He is currently working as a research manager, being responsible for HPC infrastructure, international cooperation and project planning.

Karin Oolu (MA, 2008) is an experienced information professional with a history of working in the research industry. Currently working as a records manager at the Estonian Academy of Arts she is combining her expertise in PhD studies (Tallinn University) and consulting in projects focusing on digital preservation and curation, also the content of digital information.

Andres Uueni has worked in different memory institutions, designing, developing information systems and led many cultural heritage digitization and documentation projects. Andres is co-founder of EKA MUKOLA Lab and he is researcher and PhD candidate in the Estonian Academy of Arts, focusing on cultural heritage 3D documentation and multispectral imaging

.

[4] https://www.iana.org/assignments/media-types/media-types.xhtml (2023/05/08)

[5] https://github.com/keeps/commons-ip (2023/05/08)

[6] https://rodain.roda-community.org/ (2023/05/08)

[7] https://www.essolutions.se/essarch/ (2023/05/08)

[8] https://github.com/E-ARK-Software/earkweb (2023/05/08)