Monolith: Lessons Learned on the Way to the Market

Andreas Wassmer and Peter Fornaro; Image & Media Lab University of Basel and bitsave AG; CH-4056 Basel; Switzerland

Abstract

In 2006 the Imaging and Media Lab at the University of Basel started a first project called Peviar (Permanent Visual Archive) [1, 2]. The aim was to find a digital archival storage solution without the need of periodic migration. The basic approach was to store binary data as 2D barcode on micrographic film. The theoretical, promising results found in Peviar have been further developed. Two subsequent applied science projects concluded in a commercial product called Monolith, which was introduced on the market in 2008.

Monolith is a workflow for migration-less preservation of digital data on optical media. It combines the permanence and visual nature of photographic material and the strength of digital imaging technology. The binary information is stored as bit pattern in 2D barcode either on one or multiple film layers. The machine readable code is enriched by human readable metadata in order to describe the archived objects. It also provides detailed descriptions of how to recover the original files. Because of the image based approach, the recovery is not affected by change of technology; digital cameras certainly will be available in the future. Moreover, they will become better and cheaper.

Introduction

There are many products available promising to be a long-term archival solution. Most of them are based on IT means such as server farms and are elegant solution offering a highly automated ingest and archiving process. But they all depend on keeping a high-tech infrastructure up and running for years. Broken or outdated parts must be replaced and old versions of software updated constantly, generating costs as long as the data is archived. These solutions are not migration-less because the archived data is constantly threatened by the steady change of technology. This affects the digital file format as well as the data carriers and the necessary hardware to access them. Both become obsolete within five to ten years. If there is no periodical migration to new technologies and/or formats the data will be lost inevitably. Nevertheless, IT based solutions are widely used and well-known. This makes people confide in. If you want to enter into this market then you have to offer more than just migration-less preservation.

No migration needed - and more

Migration can be omitted if the storage media fulfils the following requirements: 1) It must contain human readable metadata in order to describe the archived object; 2) the information on how to recover the original file must be part of the metadata. This knowledge is the key to interpret the archived byte stream; 3) the digital data is stored in a hardware independent way as far as possible. Thus it is not affected by the change of technology. In fact, if a medium claims to be suitable for long-term preservation of digital data it has to fulfil more requirements. These are listed in a paper presented in 2012 by Lunt et al. [6]. They identified 7 characteristics which are particularly interesting to archivists regarding preservation of digital data. The first says, that there should be no active maintenance or migration required to preserve actual data. They continue with: 2) no special storage conditions are required to preserve the storage media; 3) a minimum lifetime of at least 100 years, preferably more; 4) no power is required to maintain the data; 5) the media is easily transported; 6) the format is widely adopted; 7) the medium has a large storage capacity.

These points can be used as a benchmark to test against for each storage media considered for long-term archiving. Doing so, solutions based on optical material such as microfilm perform very well. A lot of work has been done investigating their possibilities in archives. Among many candidates Monolith is a serious one. It is different but it is neither unique nor new (e.g. see [3], [4], [5]).

Key features of Monolith

Monolith combines the advantages of photographic material and standard digital imaging technology to create a long term migration-less archiving system. This is achieved by the hybrid characteristics of the optical carrier. The digital information, stored as 2D barcode, is put right besides the human readable

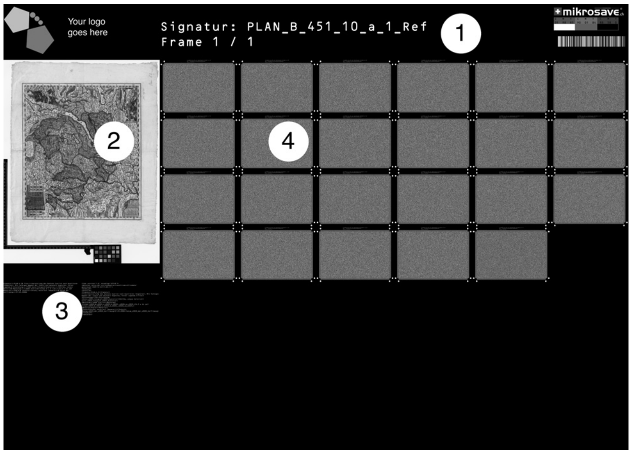

Figure 1. The Monolith Data Object comes in different layouts and formats. The fig. shows the micro fiche 10x15 cm format, which is the most commonly used one. This layout, used for archiving digitised photographies, consists of four section: (1) header with unique identification information for easy finding, (2) a high resolution (> 4000 dpi) overview of the Information Object which can be used as a fallback in case decoding the patterns fails, (3) metadata of relevant information (such as photographer, location, etc.) both as human readable text and in XML format for machine processing, (4) the 2D barcodes (patterns) of encoded digital image data (the Information Object). Section 2 and 3 are part of the Representation Information as defined by the OAIS model.

metadata, which is used for description and recovery to form an entire archiving object. Figure 1 shows an example of a Monolith data object. The 2D barcode has built-in error correction and therefore ensures lossless recovery. If microfilm is chosen as storage media, then there are no special storage conditions needed. Just store it in a dry and cool place. And obviously, you do not need any energy to keep the data on the microfilm. This fact seems trivial nowadays but it will become more important in the future as a lot of effort is directed towards sustainable IT services and ”Green IT” infrastructure. Last but not least microfilm has very good reputation among archivists and there is an agreed lifetime of 500 years.

But Monolith does not meet requirement 6 because it is new to the market and therefore, the format is not widely adopted yet. But due to the hybrid character its entire specification can be put on the film next to the data for future reference. This allows for data recovery without the existence of the manufacturer. And point 7 is also not fully met. If used with 35 mm film we can reliably store 66 megabits per meter. This may be a factor 3 less than what is achieved by others, e.g. [5]. But we learned, among other things, that this is not a crucial issue for customers. There were other challenges which will be covered in the rest of this paper.

Challenges

When launching Monolith to early adopters we thought they may be sceptical about using it for the preservation of their digital assets. Although they would see that the preservation is migration-less, they may feel uneasy with the optical character of the medium. Storing digital data on film takes it out of its native digital environment and it would take the additional step of digitising in order to access it. But this was not the case. The archivists were used to handle microfilm and had a infrastructure up and running. They welcomed the idea of using it for the archiving process of digital data. Thus, the main challenge left was to define an OAIS compliant archiving container, i.e. to optimise the structure of the Monolith Data Object and the Representation Information on it (see fig. 2).

Once we had set the coverage of the Representation Information we needed to clarify the workflow in order to validate its correctness and completeness. And last but not least all the information had to be merged with the encoded Information Object to a single Data Object. We saw that this will give quite a complex structure (see fig. 1). The conclusion was that in order to be competitive the Monolith workflow must be automatised to a great extend.

Defining the Data Object

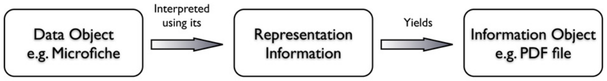

Figure 2. The workflow according to OAIS model to retrieve the archived digital data (the Information Object) from the carrier (Data Object). Schematic adapted from [8].

The Representation Information is crucial if the Information Object (the file containing the digital data) is to be preserved successfully. It must clearly identify the file and its structure down to the bits. But there is more. Let us look at an example. An important Swiss archive digitised a vast collection of maps from all ages. They use Monolith to preserve these digital assets, which are JPEG 2000 image files. This format was chosen because it allows for a high compression rate with minimal or no loss of quality. The Representation Information clearly needs to describes the JPEG 2000 image file format. But it also has to describe the original physical object, the map. The challenge is to define what will be relevant information worth preserving. In this particular case the data notes that it is a map of ancient Zurich drawn on paper with the flags of the quarters, its scale, the creation date and the author, and the digitalisation date. Of course, this set of metadata was defined specifically for those assets. If there were audio file to archive the set would look differently. Monolith would easily allow that.

Automatisation

In order to be competitive the whole Monolith workflow must be highly automated. This is not limited to compositing the Monolith Digital Object to the final image for recording. It also involves an automated validation of the ingest and the creation of the Representation Information. And last but not least there must be a proof-reading of the final Digital Object. This means that the 2D barcodes must be scanned and decoded to get the original file. Let us look at these steps in more detail.

Validation

A file (the Information Object in OAIS model) that is to be archived must be checked if it is a valid file format. This includes validation of the file extension and the integrity of its structure as well. This step may seem trivial but is essential. If a corrupted file is archived one will not be able to access the information in the future. These defects may occur in some cases especially if the software used to create the file crashes or hangs. There are tools such as JHOVE2 [9] to automate this process.

Next, the metadata describing the object is validated. The Monolith Data Object contains the metadata in two formats. Once in XML format for machine processing and once as human readable ASCII text. The XML data uses the Dublin Core convention and thus validation of this section needs to make sure that all the required fields are present and contain the correct data (validation for correctness and completeness). The latter also includes spell checking, of course. The values are compared to the human-readable section. These steps cannot be fully automated. Of course, the check for completeness of the XML can be done with the software and also the extraction of the ASCII text. But correctness of the collected data must be done manually by comparing it to the file.

And last but not least the final revision ensures that the metadata are associated with the respective file. This can be done programmatically if the metadata is also present in the header of the file. This is possible for TIFF and JPEG 2000 files.

Composition

The layout of the micro fiche shown in figure 1 is quite complex. It is based on a defined 8x8 grid and it takes several steps to create. First, you must encode the data, i.e. the image in the example. Then, the overview image must be scaled and fit into the

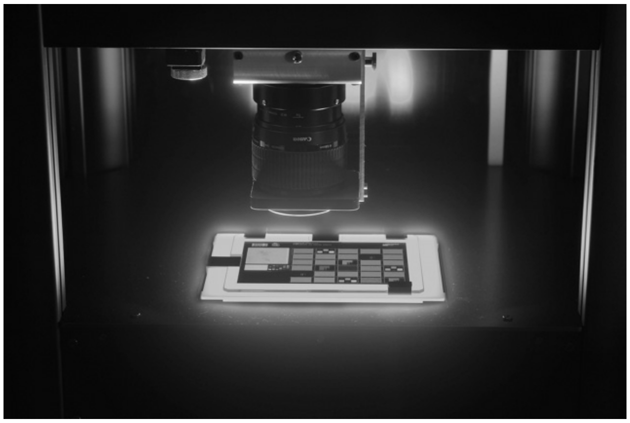

Figure 3. This figure shows a scanner proof-reading a fiche. The scanner was designed to scan the grid layout of Monolith fiche. It is based on a 12 megapixel CCD camera which can be placed over each pattern. In combination with a 2x macro objective it can be used to scan patterns with a structure size of 15 μm. (Courtesy Swiss-Mikrosave AG)

grid. Next, all the metadata must be collected and entered. You can use available software such as Photoshop for the composition. But this would be too much work. Even an experienced user may need several minutes to compose the fiche. If the task includes many files the time needed to create all the fiches would raise the production costs considerably. It is not surprising that the possibility to both encode the data and compose the image unattended was the most asked feature by the service provider. We spent quite a time writing a performant encoding software that can do the job automatically. It takes a comma delimited values file (CSV; see [7] for specification) as job description. The file contains all the information needed, e.g. file names and metadata. It is used as input to the software which then composes fiche after fiche, eventually running for days. And there is a lot of output. Compiled for the Fluck Eternity 105 recorder each fiche is a 41,800 x 30,000 pixels wide and a size of 3.7 Gb.

Proof reading

Up to now two thirds of the way to the final Digital Object is completed. There is one final step to do. The patterns on the Digital Object must be read back and decoded. This results in a byte stream that has to be assembled to the final file. When encoding the data the encoder calculated a md5 hash for each stream[1]. We chose the md5 because it is very sensitive on bit level. One altered bit gives a completely different checksum and is thus easy to detect. This number is encoded into each barcode and is used ”on-the-fly” for error detection when decoding. Once all the individual streams are merged to the final file the overall checksum is calculated and compared to the one of the original file which is written in the Representation Information. Not before this match has been confirmed the archival process can be considered successful.

Decoding a pattern is not as straight forward as encoding. It involves some image processing and should account for three major problems:

- The images may be under- or overexposed. This happens occasionally in a series of scans, even if the the scanner was initially set up correctly.
- There may be scratches or dust particles covering part of the barcode.
- Parts of the barcode may be unsharp. This is mostly due to misaligned optics or the film material nor laying flat on the scanner

The decoder software can correct some of those problems to a certain extend. Even if a part of the marker lines is covered by a fibre as seen in fig. 4 the decoding will not fail. The software can still recover the co-ordinate system and find the pixel positions. Furthermore, the software is designed to assist the user in the decoding process. It can give hints if decoding of a pattern should fail.

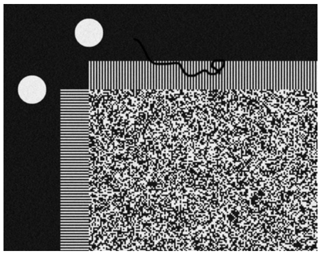

Figure 4. The marker lines around a barcode. They are used to locate the pixels, i.e. bits in the pattern. There is a small fibre covering part of the markers.

More Lessons Learned

Unlimited Access to Software

The 2D barcodes developed for Monolith are optimised for data safety. The bits are randomly distributed over the pattern and error correction bytes are added in order to guarantee a high fault tolerance when decoding. But as with any barcode one needs software to decode it. This is customer’s main concern. The data on the micro film is known to last for many decades, but what about the software. How can we guarantee unlimited availability of the software, no matter what happens to our company. That is why the source code for the decoder is OpenSource and is given to the customer for free. But this is not enough. The code is kept simple and is written in standard C, a popular and widespread programming language nowadays. It is easy to find a compiler and make the code run on almost any machine. But what about in twenty years? We cannot predict the future. But looking back in time we have witnessed many changes in hard- and software. And we assume this going on. It is most likely that the C programming language will become obsolete and then will be forgotten entirely. The only solution to decode the pattern then is to write a new code in a way suitable for future computers. That is why we have an OpenEverything strategy. The customer also gets what we call the ”White Book”. It describes in full details the structure of a pattern and how it was created. It also documents the software with help of pseudo-code. This a common tool for describing algorithms because this code does not depend on any programming language. Everything, the source code and the White Book are recorded to

[1] The author agrees to the possible objection that the md5 hash is not a secure indicator. Exploits exists that can manipulate the original stream without changing the hash. But used as an indicator of correctness we consider it safe.

microfilm as well and are part of the Monolith Representation Information.

The system on which the software runs is also important. We developed the first release on Apple Computers and Mac OS X[2]. We decided to use these systems many years ago. This operating system has a very powerful image processing API (Advances Programmers Interface). You can do a lot with few lines of code. But Apple's OS is still not so popular as its competitor from Richmond, Microsoft Windows[3]. It is clear that no one just buys another system only to use a dedicated software. And even if someone was willing to do this he would not be allowed to connect the computer to the companies network because network administrators do not like heterogeneous networks. Thus we had to come up with a Windows version.

There is an important note: The decoder software is what the OAIS model calls Access Software. Thus it is a vital part of Monolith and is given to the customer for free.

Conclusion and Outlook

Monolith has been successfully established in the Swiss market since its launch in 2008. In these years numerous improvements to the workflow and the product were made. This would not have been possible without all the lessons we had to learn. And there is still room for enhancements, especially in the validation workflow. The ingest process and the extraction of metadata must still be done manually.

References

[1] David Gubler, Lukas Rosenthaler and Peter Fornaro, The obsolescence of migration: Long-Term storage of digital code on stable optical media, Proc. Archiving Conference, IS&T, pg. 135, (2006)

[2] Florian Müller, Peter Fornaro, Lukas Rosenthaler and Rudolf Gschwind, PEVIAR: Digital Originals, J. Comput. Cult. Herit., 3, 1 (2010)

[3] ArchiveLaser Project: Accurate Long-term Storage of Analog Originals and Digital Data with Laser Technology on Color Preservation Microfilm, Proc. Archiving 2005, IS&T, pg. 197-200 (2005)

[4] Christoph Voges, Jan Fröhlich, Tim Fingerscheidt, Long-Term Storage of Digital Data on Cinematographic Film, Proc. Archiving 201, pg. 158-161. (2011).

[5] Oscar Plata, Rune Bjerkestrand, The ARCHIVATOR - A Solution for Long-Term Archiving of Digital Information, Proc. Archiving 2012, IS&T, pg. 70-74. (2012).

[6] Barry M. Lunt, Matthew R. Linford, Robert Davies, Research on Another Permanent Data Storage Solution, Proc. Archiving 2012, IS&T, pg. 19-21 (2012)

[7] http://tools.ietf.org/html/rfc4180

[8] The Consultative Committee for Space Data Systems, Reference Model for an Open Archival Information System(OAIS); CCSDS 650.0-B-1, Magenta Book, Washington DC, 2012

[9] Stephen Abrams, Sheila Morrissey, Tom Cramer, "What? So What?" The Next-Generation JHOVE2 Architecture for Format-Aware Characterization, The International Journal of Digital Curation, 4 (3), pg. 123-136 (2009)

Author Biography

Andreas Wassmer received his M.Sc. in Physics from the University of Zurich (2000). He has been working in the Imaging & Media Lab at the University of Basel as a Software Engineer. His work focusses on the field of image processing and Computational Photography. Since 2008 he has worked as Senior Software Architect at bitsave AG, Switzerland. His research interests are Computational Photography, High Performance Computing and the Internet of Things.

[2] Mac OS X is a trademark of Apple Inc., Cupertino.

[3] Microsoft Windows is a trademark of Microsoft Corporation, Richmond.