Design Patterns In Digital Preservation

Understanding Information Flows

|  |  |  |
| --- | --- | --- |
|  | **Andrew N. Jackson** |  |
|  | *The British Library*  *UK*  *andrew.jackson@bl.uk*  *0000-0001-8168-0797* |  |
|  |  |  |

**Abstract – This paper proposes a framework to help understand the different ways digital preservation goals can achieved, and the contextual factors these choices depend on. This is done through a worked example: three different design patterns representing the three possible modes of archival information flow, each illustrated with realistic examples and practices. By helping a wider audience understand these different approaches, we can ensure implementation choices are informed by practice, rather than by default, or without even realizing a choice has been made.**

**Keywords – OAIS, design patterns, risk management.**

**Conference Topics – Community; Innovation**

# Introduction

To a newcomer to digital preservation, it must seem like every question they have has one of two answers: *“OAIS”*, or *“it depends…”*. Standards like the Open Archival Information Systems reference model can provide useful high-level frameworks, but cannot provide concrete guidance when it comes to how to actually implement a digital preservation program. At the other extreme, over the years the digital preservation community has done a good job of sharing information on local implementations and workflows (through conferences like [iPres](https://ipres-conference.org/), and via things the [Community Owned Workflows](https://coptr.digipres.org/index.php/Workflow:Community_Owned_Workflows) wiki [1]). These individual reports are a rich source of information, but synthesising this information is a very difficult challenge for a newcomer to take on, as they are least well equipped to know how relevant or current any particular piece of work might be.

The [NDSA Levels of Digital Preservation](https://ndsa.org/publications/levels-of-digital-preservation/) [2] have helped bridge this gap from the top down, by providing a concrete roadmap of goals. But any such universal path necessarily must remain abstracted away from the details that depend on context. And working from the bottom up, summaries of recent work, like the [Digital Preservation Coalition’s Technology Watch Publications](https://www.dpconline.org/digipres/discover-good-practice/tech-watch-reports) [3] provide very helpful in-depth summaries of known areas of interest. But this is still a lot to work through, and it remains difficult to quickly find and filter through the wide range of information on different approaches, and understand how the details depend on your context.

Is there some complementary format or framework that would allow us to navigate this gap, providing some kind of map that helps us find our place, and understand how our work should adapt to our context? In short, if the answer is *“it depends…”*, then the next question is *“what does it depend on?”*.

Here, I propose that a practice-driven approach is needed in order to answer this question. Specifically, one where concrete experience of implementing digital preservation processes are drawn together based on common patterns, thereby allowing the common contextual factors to be revealed. What follows are the results of analysing one aspect of digital preservation – the overall flow of information through the archive – within this *“grassroots”* framework. By publishing this here, I hope to get constructive feedback on whether this approach is useful, especially from newcomers to digital preservation. This will help shape future work on completing the analyses of a wider range of aspects of digital preservation.

# Patterns of Information Flow

The [Open Archival Information System (OAIS)](https://en.wikipedia.org/wiki/Open_Archival_Information_System) reference model [4] is generally very high-level. There are a *lot* of different ways of running an archive that would still fit.

But one thing it is very clear about is the overall flow of information through the archive.

## Flow 1: The Line

*OAIS §1.4 - CONFORMANCE: “A conforming OAIS Archive implementation shall support the model of information described in 2.2”*

*OAIS §2.2.3 - INFORMATION PACKAGE VARIANTS: “Within the OAIS one or more SIPs are transformed into one or more Archival Information Packages (AIPs) for preservation. […] In response to a request, the OAIS provides all or a part of an AIP to a Consumer in the form of a Dissemination Information Package (DIP). “*

Under OAIS, Submission Information Packages (SIPs) come in from Producers, are managed as Archival Information Packages (AIPs), and these AIPs are then used to generate the Dissemination Information Packages (DIPs) that serve the needs of the archive’s user community a.k.a. Consumers. This fundamental design pattern for digital archives – this way of describing the overall flow of archival information – can be visualised via this simple diagram:

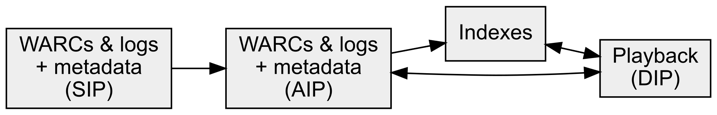


This linear flow is one of the strengths of the OAIS model, because by generating the DIP from the AIP, we ensure that all the information we might need to (re)create a new DIP is in the archive.

For an implementation to follow this pattern, it must ensure that *only* information that is in the AIP was used in the creation of the DIP, and there is no dependency on the SIP. OAIS does this by specifying that the DIP is generated from the AIP held on Archival Storage, after the SIP has been discarded. This could be immediately after ingest, or some time later, or on demand – the important thing is that it’s post-ingest. To understand what this means in practice, it is useful to look at some examples of digital archives that work in this way.

### The UK Web Archive

Like most web archives, the [UK Web Archive](https://www.webarchive.org.uk/) generates access copies on demand. Before that can happen, the archived web resources are captured in [WARC](https://en.wikipedia.org/wiki/Web_ARChive) files, which are then placed on archival storage before the necessary indexes are generated from them in order to enable access. When individual items are requested, the relevant records are looked up in the indexes before being copied from the AIP, and the access version (DIP) is generated from the archival version during the playback process.



Because the transformation is done on demand, this makes it easier to modify how playback is performed in response to changes in web technology. In essence, we preserve the software that gives us the ability to generate access copies, rather than preserving the access copies directly.

### The Internet Archive

While their web archive generates derivatives on demand (as above), the [Internet Archive](https://archive.org)’s documentation makes it clear that their archival system generates various derivatives for items after they have been ingested/uploaded, depending on the content. See for example this [documentation on type of derivatives they generate](https://help.archive.org/files-formats-and-derivatives-a-basic-guide/) [5], and this [summary table showing all the format conversions they do](https://archive.org/help/derivatives.php) [6]. The derivatives are stored rather than re-generated on demand, but doing this post-ingest means the derivation process can’t block the process of ingesting items and ensuring they are safely replicated.

## Flow 2: The Stop

The linear workflow described above may seem obvious, perhaps even inevitable, but it’s not. Indeed, it’s not unusual to find archives where there is *no outflow at all*. This corresponds to so called “dark archives” - ones that cannot be accessed except by the people who manage the archive. Usually, this is because the content is available elsewhere, and the archive is acting as a ‘backup copy’ in case the original goes away.



In these cases, the focus tends to be on ingesting the content, and not on how that content might be used in the future. There may be DIPs, but they are not well tested because no-one really uses them much. Or there may be no DIPs at all, or just some theoretical DIP based on an imagined future. The danger here is that it’s possible to miss some information you need from the SIP, or from the wider context, because you’d only realise you needed it when your user community started trying to access the material. Ideally, to mitigate against this risk, it is necessary to either encourage real usage, or at least simulate it, so that access problems can be identified while there is still some hope of resolving them. Otherwise, your AIPs are like a backup that’s never been tested.

### The UK Web Archive

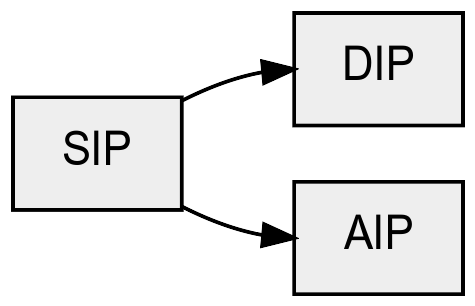
While the UK Web Archive has thousands of open-access websites, we also archive many millions of sites that can only be accessed under Non-Print Legal Deposit terms. This places heavy restrictions on access to those sites, and this combined with our limited capacity for manual quality assurance means it is unlikely that many of these archived pages will be viewed quickly enough to pick up any issues while we can still resolve them. One way we are trying to mitigate this risk is by starting to automate some of the manual QA processes. For example, for many years we have been collecting screenshots of how the websites looked when the web crawler originally visited them. The next step will be to add a post-ingest process that takes a screenshot of the *archived* web page in the same way as the crawler, allowing the images to be directly compared. This should highlight any issues and focus our efforts on where things needs to be improved. We have also been working on publishing non-consumptive datasets based on our holdings. These are no substitute for the original web pages, but the surrogates are rich enough that meaningful research can be done with them, and this can help identify gaps in our collection.

### Electronic Journals

Due to their importance and economic value, eJournals are often ‘backed up’ in dark archives like CLOCKSS or Portico. Usually, access to eJournals is via the publisher sites, and the archived copies only become active if the publisher shuts down. Here, AIPs usually contain all the SIP information, but ensuring the completeness of these packages is highly dependent on how fully the structure and content can be validated. This is particularly challenging when it come to the handling of supplementary material, as this can come in a wide range of formats, or may only be supplied as a reference to resources held elsewhere.

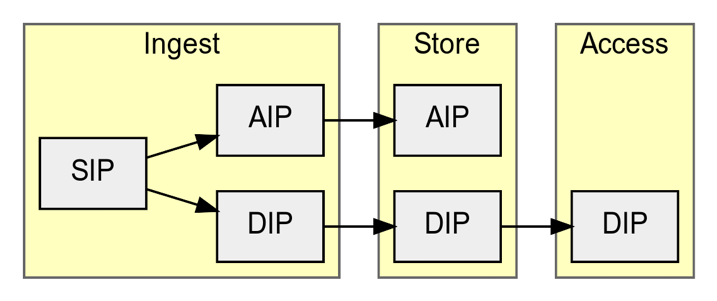
## Flow 3: The Fork

The final class of information flow breaks the linear model entirely: while there is always a line from ingest to access, there is sometimes a fork in the road.



In this kind of split workflow, both the AIP and the DIP are generated directly from the SIP. In contrast to the linear flow, this design pattern *does not automatically ensure* that the AIP contains all the information we need to generate the DIP. This is not to say that this risk cannot be addressed, rather the point is that the implementation has to do *additional work* to make sure that this is the case.

One of the reasons why this can happen is that, during the design and implementation of archival workflows, it is common to try and perform a lot of processing up-front. Schematically, the result can look something like this:



Here I am using *Ingest*, *Store* and *Access* to identify processing contexts, covering all the functions associated with that phase of the workflow. During the *Ingest* phase, the incoming packages have been used to generate the DIP before the AIP has been transferred to the archive store. This means it is *possible* for the AIP and DIP to diverge.

In many cases, especially those where the critical payload of each package is a single file, this is a modest burden, because when the number of files involved is small, it’s easier to keep track of individual files, their checksums, and their relationships. As things get more complex, it becomes easier for errors to creep in, especially if some network or system outage occurs while an item is being processed. These potential risks are certainly manageable - the point is simply that these risks arise because of the decision to perform more processing up front. This does not mean “The Fork” is a *bad* pattern, indeed in those cases where we can confidently verify the packages are complete, this means we can pick up problems before items make it to the archival store. The downside is that if there are problems during ingest, this can lead to a backlog of material that spends far too long stuck on non-archival storage.

### An Outsourced Digitization

This example involves a digitization project where much of the work had been outsourced to a third-party. The decision had been made to get the external partner to generate “access copies” as well as the long-term “preservation copies”. The preservation version consisted of high-resolution, full page TIFF files and associated metadata tying each set of TIFFs together, mapping back to the original publication. However, the access versions were not simply JPEGs of the preservation versions, because the items were broken up at the level of individual articles. This worked well for access purposes at the time.

The problem was the article segmentation process had not been properly documented. The coordinates on the TIFF version that corresponded to the positions of the JPEG versions had been lost. This meant that when the access system became obsolete, it was not be possible to replace it like-for-like, i.e. while still preserving the same article-level experience.

### Preservation System Design

When looking at tools and services that implemented digital preservation processes, it can be very difficult to tell exactly what’s going on *under the hood* and therefore difficult to determine what kind of information flow is being implemented. However, both [Archivematica](https://www.archivematica.org/en/) and [Rosetta](https://exlibrisgroup.com/products/rosetta-digital-asset-management-and-preservation/) have a significant amount of documentation that is openly accessible online, so this allows is to gain some insight into how things are done.

Reading [the Archivematica documentation](https://www.archivematica.org/en/docs/archivematica-1.13/user-manual/ingest/ingest/#normalize) [7] it is clear that Archivematica *usually* performs all processing up front, prior to ingest: the DIP is clearly generated directly from the SIP, but the system takes steps to ensure nothing is lost by making the original SIP form the basis of the AIP. Archivematica does make it possible to derive access copies in a post-ingest process as part of the ‘re-ingest’ workflow, but this is not the normal mode of operation. Rosetta also supports post-ingest generation of access copies via a [*Create Derivative Copy Representation*](https://knowledge.exlibrisgroup.com/Rosetta/Knowledge_Articles/How_does_Rosetta_manage_Derivative_Copy_representations%3F) workflow [8], but like Archivematica, this appears to be seen as a secondary mode of operation, with ingest-time processing being the primary focus.

Note that this is not a criticism of these tools, which are merely implementing the workflow that their users have requested. Note also that it is not possible to draw comparisons with other preservation systems because this analysis is necessarily skewed towards those tools that provide detailed documentation online.

## Summary

These three distinct information flow patterns show that there are fully-functional and widespread archival information flows that are not strictly OAIS conformant. This is not a problem with the archives or the software, but just alternative modes of operation that are not fully represented by OAIS as it stands. Each information flow pattern has its place, but it’s important to be aware of the balance of benefits and risks of each.

# Conclusions

In this paper we have used the idea of design patterns as a way of capturing different choices that can be made when implementing a digital preservation process, focussing on overall information flow. The same tactic could also be applied to other aspects of digital preservation, such as:

* Communities: can we identify different classes of communities and environments, helping us understand how to engage with them?
* Ownership: is the archive part of the organisation that own the records, or do we hold records on behalf of others? How does this change what we need to do?
* Assessing Preservation Actions: what are the different meanings of Significant Properties, and what other methods can we used to assess our interventions?
* Archival Packaging Patterns: what are the different approaches to defining information packages.
* System Architectures: what are the different ways we can implement the OAIS functional requirements, i.e., which system or systems covering which functions?

This paper seeks feedback on this approach, and on whether future work along these lines would be of use to the wider community.

# REFERENCES

1. Community Owned Workflows wiki. https://coptr.digipres.org/index.php/Workflow:Community\_Owned\_Workflows
2. Phillips, Megan, et al. "The NDSA levels of digital preservation: Explanation and uses." *Archiving conference*. Vol. 2013. No. 1. Society for Imaging Science and Technology, 2013.
3. Technology Watch Publications, Digital Preservation Coalition. <https://www.dpconline.org/digipres/discover-good-practice/tech-watch-reports>
4. Reference Model for an Open Archival Information System  
   (OAIS); CCSDS 650.0-B-1; Consultative Committee for  
   Space Data Systems: Washington, DC, 2002.
5. Files, Formats and Derivatives – Basic Guide, Internet Archive. <https://help.archive.org/files-formats-and-derivatives-a-basic-guide/>
6. Table of derivatives for different types of content. Internet Archive. https://archive.org/help/derivatives.php
7. Archivematica 1.13 Documentation – Ingest – Normalise. <https://www.archivematica.org/en/docs/archivematica-1.13/user-manual/ingest/ingest/#normalize>
8. How does Rosetta manage Derivative Copy representations? Ex Libris Knowledge Base. <https://knowledge.exlibrisgroup.com/Rosetta/Knowledge_Articles/How_does_Rosetta_manage_Derivative_Copy_representations%3F>