

# Building open source audiovisual collections management systems on Samvera

Received (in revised form): 7th May, 2020



## Karen Cariani

David O. Ives Executive Director, WGBH Media Library and Archives & Project Director, American Archives of Public Broadcasting, USA

Karen Cariani is the David O. Ives Executive Director of the WGBH Media Library and Archives and WGBH Project Director for the American Archive of Public Broadcasting. Karen has over 30 years of television production, project and archival management experience. She has served on numerous Association of Moving Image Archivists committees and two terms on the board. She serves on the National Stewardship Digital Alliance Coordinating Committee.

WGBH Media Library and Archives & Project Director, American Archives of Public Broadcasting, 1 Guest St, Boston, MA 02135, USA

Tel: +1 617 300 4286; E-mail: karen\_cariani@wgbh.org



## Richard Green

Operations Adviser, Samvera, UK

Richard Green is an independent consultant who worked with the University of Hull's repository team in the UK for many years. He was a founder member of the Hydra Repository Project, now known as the Samvera community, and was Chair of their Steering Committee for 2018/19. Richard is currently acting as Samvera's Operations Adviser. Among other related activities, Richard has also served as Vice-Chair of the international Open Repositories conference.

E-mail: r.green@hull.ac.uk

**Abstract** Building and using an open source community-supported system to manage audiovisual materials for a digital archive/library has many advantages and challenges. Being able to dictate the features, have a system speak to specific needs, and have staff on hand that can change or fix problems is certainly appealing. Embarking on this effort with an established open community, such as Samvera, has the advantages of a robust community of developers and service vendors to turn to for help. Managing needed customisations to core code base, keeping track of updates and contributing to the community, however, is challenging. For WGBH, a public television station with a robust 60-year archive, most customisations are due to the use PBCore to structure the metadata of the audio-visual items. This paper focuses on WGBH's efforts to build a system for its Media Library and Archives based on the Samvera digital repository framework and its Hyrax and Avalon Media System 'products'.

**KEYWORDS:** open source, metadata modelling, PBCore, Samvera, Hyrax, Avalon

## INTRODUCTION

Digital media management systems are available in all shapes and sizes with all sorts of features. There are many companies

available to license software, host systems or contract to build. Some systems are customisable for use with audiovisual (AV) materials and several specialise in the

management of A/V files. Building and using an open source community-supported system to manage audiovisual materials has many advantages and challenges. Being able to dictate the features, have a system speak to specific needs, and have staff on hand that can change or fix problems is certainly appealing. This paper focuses on WGBH's efforts using the Samvera digital repository framework and its Hyrax and Avalon Media System 'products'.

## THE SAMVERA COMMUNITY: AN OVERVIEW

Samvera (<https://samvera.org/>) is a grassroots, open source community creating best-in-class digital asset management solutions for libraries, archives, museums and others. Samvera is based on the premise that no single system can provide the full range of repository-based solutions for a given institution's needs and that no single institution can resource the development of a full range of solutions on its own. Working together, the Samvera community creates sustainable solutions using a common infrastructure within which there is the flexibility to tailor solutions to local demands and workflows.

Samvera software is all free and open source, available under an Apache 2 licence, and based on four major components:

- Fedora repository software to provide a robust, durable repository layer for persisting and managing digital objects;
- Solr indexes to provide fast access to information about an institution's repository content;
- Blacklight — a Ruby on Rails plugin that sits above Solr and provides faceted searching, browsing and tailored views on objects; and
- Samvera gems — Ruby on Rails components that integrate the building blocks to form a complete, flexible and extensible digital repository solution.

In the last few years, the Samvera community has seen an emerging need for some off-the-shelf 'solution bundles', addressing particular needs, that can be installed and maintained with fewer local resources — or that can be deployed as a hosted, cloud service. The community has responded to this need with two such bundles, Avalon and Hyku, in addition to the more general repository-building engine, Hyrax. Institutions can use these components to develop their own completely customised local solutions; the components are also used in three major community-supported applications:

- *Avalon*: The Avalon Media System is an open source system for managing and providing access to large collections of digital audio and video. The freely available system enables libraries and archives to easily curate, distribute and provide online access to their collections for the purposes of teaching, learning and research. It has been developed by Indiana University and Northwestern University.
- *Hyrax*: Hyrax is a Ruby gem front-end based on the robust Samvera framework, providing a user interface for common repository features. Hyrax offers the ability to create repository object types on demand, to deposit content via multiple configurable workflows and to describe content with flexible metadata.
- *Hyku*: Hyku (formerly Hydra-in-a-box) has been developed by the Digital Public Library of America (DPLA), Stanford University and DuraSpace to provide a hosted solution based on Hyrax. A number of service providers are offering cloud-based hosted versions.

## A SHORT HISTORY

Samvera began in 2008 as the Hydra Project, emerging out of discussions between the University of Virginia, University of Hull (UK), Stanford University and Fedora

Commons (later absorbed into Duraspace which, in turn, is now part of LYRASIS) regarding how they might jointly produce modular, open source software catering to a number of repository needs but using a single Fedora store. This idea of 'one body' (Fedora) with 'many heads' gave rise to the project name 'Hydra'.

It was decided very early in the planning process that the project should utilise existing, 'best of breed' open source software alongside its own work. Accordingly, Blacklight and Solr were adopted alongside Fedora. The group identified a core of common needs and set themselves the target of producing within three years 'a reusable framework for multipurpose, multi-function, multi-institutional repository-powered solutions'. In autumn 2012, Hull was the first of the universities to go live with a Hydra repository.

There was an early recognition that any output from the project would only work sustainably if it could be built on and further developed by others as well. From the beginning, an essential aim was thus to enable others to join the open source Hydra Project as and when they wished, and to establish a framework for sustaining the community as much as any technical outputs that might emerge. A formal governance structure was adopted by the initial partners in 2012 and by August 2019, Samvera had 36 partners in the USA and Europe (see appendix).

By 2015, the project partners decided that Hydra was sufficiently established that both the name and the logo should be trademarked. It transpired that 'Hydra' could not be used and so the community renamed itself 'Samvera' (an Icelandic term meaning 'togetherness'). Rather than seeing the rebranding work as a setback, the community treated it as an opportunity to re-evaluate its aims and its working practices. Part of this analysis resulted in the formation of the Governance Working Group to make recommendations for the future. These recommendations were accepted and underpin today's community structure.

At the heart of the modern Samvera community are its partners, currently comprising institutions, companies and other groups that have formally committed to contributing to the Samvera community. The community is defined by a memorandum of understanding, to which all partners agree, and its legal and financial affairs are handled through a fiscal sponsor, DuraSpace — now part of LYRASIS.

The community puts considerable effort into ensuring that its code is robust, reliable and sustainable. Samvera's developers use continuous integration tools to ensure that new code does not 'break' anything else in the system. No code is accepted into the Samvera core that does not have adequate, in-built testing. All Samvera code is distributed under the Apache 2.0 open source licence and, to ensure compliance, contributor licence agreements (CLAs) based on the Apache Foundation CLAs, are required from contributors, both individual and institutional, before their code can be added to the community's code base. Many of Samvera's adopters adapt the core code to address local use cases and needs. Institutions that do this are encouraged to contribute the code back if it has more general applicability.

The organisation maintains a website at <https://samvera.org> and a wiki at <https://samvera.org/wiki>. In addition, there are various mailing lists, each serving a specific focus within the community, and a widely used Slack channel. Samvera strives to project a professional image and is mindful of its responsibilities to community members as well as to its adopters and potential adopters. The community operates a publicly posted code of conduct and an anti-harassment policy.

## WGBH AND AVALON<sup>1</sup>

WGBH has been the Public Broadcasting Service (PBS) station in Boston, MA since 1951. It produces roughly 30 per cent of the content that airs over PBS nationally and

maintains several radio stations. To manage and preserve its programming — both current and archival — it requires a digital media system. WGBH's Media Library and Archives (MLA) establishes the policies and procedures for the access, acquisition, intellectual control and preservation of the organisation's physical media and digital production and administrative assets. The MLA's goal is to promote the efficient and effective use and re-purposing of WGBH's physical and digital assets. The MLA is responsible for acquiring, creating access to and preserving the programme media materials and administrative records generated during the production of radio and television programmes for WGBH.

As a production and broadcast organisation, WGBH must manage production work and broadcast files in addition to a library/archive system. In an ideal world, this would be done via a single system; as experience has shown, however, this is not always the best solution. The key issue here is that an archive system must track as much descriptive data as possible about the material held — provenance, technical information, preservation data, and so forth. For production systems, however, this volume of metadata can soon become cumbersome. Thus, broadcasters tend to use separate systems to manage broadcast files and internal production files.

WGBH MLA tried using a single vendor-licensed system for production tracking and archiving but became frustrated. The solution required customisations to accommodate the MLA's specific needs, and the MLA was dependent on the vendor to make these changes. When requested changes fell off the product marketing track, however, the requests took longer to fulfil, and WGBH's needs were not met. Concurrent with this, WGBH also fell behind on upgrades and migrations were becoming increasingly expensive. In concert with the licence fee, it all added up to an expensive system that did not fulfil all the necessary functions.

WGBH looked for other solutions. A system was cobbled together that could be managed internally, but something more robust was still desired.

Here, it is worth underscoring that the present authors work with the WGBH archive, rather than production or IT, hence sought a solution that would serve archive/library needs.

The MLA became involved with the Samvera community because it was a community of longstanding, well-resourced academic libraries that were not chasing a marketable value. Furthermore, the various organisations that make up the Samvera community all need to manage large digital libraries with similar features and needs, even if most of them do not deal with the same volume of media files as WGBH.

Indiana University was undergoing a massive digitisation of its media holdings and building a solution called Avalon to access its digital library of media, which WGBH thought held a lot of promise.

WGBH undertook to develop Samvera related solutions for the preservation files (Phydo) and for the American Archive of Public Broadcasting (AAPB) metadata management system (AMS2.0).

Phydo was a joint project with Indiana University to develop a preservation system using Fedora 4. (Although the project did develop a repository using Fedora 4 to store preservation technical data, it has not yet been fully implemented at either WGBH or Indiana University.)

The AAPB is a collaborative initiated between WGBH and the Library of Congress seeking to preserve and make accessible significant historical content created by public media, and to coordinate a national effort to save at-risk public media before the content is lost to posterity. The collection currently holds more than 110,000 digitised and born-digital programmes and original material from over 130 public broadcasting stations and organisations.



The AMS2.0 manages the metadata for the collection and the public website. Through this project, WGBH MLA rebuilt a system that already existed under the name Archival Management System (AMS). The original AMS was a custom application built by AVP (through funding from the Corporation for Public Broadcasting) for the American Archive Project before WGBH and the Library of Congress were awarded stewardship of the collection. Participating organisations have access to the AMS, and station administrators can search and access their metadata records, export their records, import additional records, watch/listen to their low-res proxy files and download their low-res proxy files.

Almost immediately after WGBH MLA and the Library of Congress took on the management of the AAPB, the AMS became the backbone of the AAPB where all of the metadata about the collection were managed. However, the AMS had become outdated, needing dependency upgrades, and the code base was not in the wheelhouse of the WGBH MLA developers. While the system itself was integral to the maintenance of the collection, the MLA was not able to maintain the system itself. WGBH worked with AVP and Indiana University to build a new AMS (AMS2.0) that better fit the AAPB's needs based on the Samvera stack and code.

The AMS has some unique managing features that are important for AAPB operations. First, the AMS does not manage any of the files, thumbnails, proxy files or preservation files. It manages technical and preservation metadata about those files, and it includes identifiers so the files can be served up from their separate S3 and Sony Ci locations. Secondly, the AMS uses the hierarchical PBCore (<http://pbcore.org>) model that breaks out intellectual content into an asset, which can have multiple repeatable child instantiations, which in turn can have multiple repeatable essence tracks.

The AMS also needs to handle many batch operations, mainly in terms of import

and export. Data typically preexist in one of the contributors' systems. The AMS needs to import data quickly, rather than have records created or submitted one at a time. Finally, the AMS is the system used for both the AAPB staff to manage the metadata and where contributing organisations can access their own metadata. While this system needs to be geared for access in the sense of internal people seeing media items to better catalogue them, it is not geared toward anyone who would want to use the collection for research. The public website (<http://americanarchive.org>) is geared towards the public and harvests directly from the AMS.

WGBH MLA considered three options of moving forward given the updated system requirements/desires: (1) extend AMS1.0, (2) build a new system from scratch, or (3) extend and implement a Samvera-based Hyrax/Avalon solution.

At the beginning of the project, the MLA staff considered various options to build an improved AMS by taking into account the structural differences of the original repository that needed to change. A new system could be built from scratch, but such systems generally become so customised to the creator's needs that they are not as useful to others. Historically, the MLA has leaned toward using open source components to build products that can serve not only WGBH MLA but also the archive community at large. The MLA explored those Samvera-based applications that were

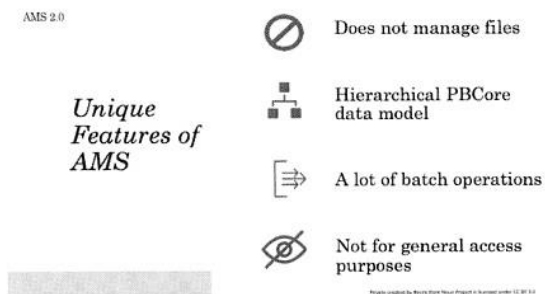


Figure 1: Unique features of the AMS

available at the time, including Avalon and Hyrax. Due to the MLA's previous experience with an expensive system that did not meet its needs, an off-the-shelf solution was not considered.

After comparing all of the options, the MLA team created a spreadsheet that included system requirements and wish-list features, and compared the current version of the AMS against Avalon and Hyrax. Working with Indiana University, Northwestern University (Avalon developers) and AVP, a gap analysis determined that Hyrax was the best option.

## **THE MOVE FROM AMS1.0 — WHAT NEEDED TO CHANGE**

### **PBCore data model**

The original AMS was very much centred on the contributing organisation. It was designed to handle the digitised copies of a tape, but not designed for multiple organisations to have a tape of the same content in a central repository. The PBCore metadata schema that WGBH uses has an asset level at the top of the hierarchy, designed to describe intellectual content only. Under the asset, a record can have as many child or nested instantiations as needed, which may include a tape, master tapes, dub, viewing copy and digital files.

AMS1.0 was set up based on physical inventory and could have only one physical asset per record. For example, an institution holds a copy (an instantiation) of a programme (the asset) and it is being digitised. All digital items (more instantiations) are nested under the asset record. This brought preservation, mezzanine and proxy instantiations stored within the asset record, and there was a user-friendly layout for those instantiations nested inside the asset. As the AAPB evolved, however, organising the collection switched from being very inventory based, to a union catalogue more like WorldCat.

There are some titles, such as programmes distributed through National Educational Television (NET), where copies exist in many different repositories. Each title, duplicated for every organisation that had a copy, of the same content, was redundant. Instead, one asset record for that title would suffice, with children for each organisation that holds a copy; for example, Library of Congress has this, while WGBH, WNET and IU have that. At the beginning, it was hard to imagine switching from the real parent being the organisation owning the record to the asset or title itself as parent and the organisation owning their own physical item of that title. It would have been a very big change to the structure of AMS1.0.

### **Actively maintained plugins**

The second change was to adopt a system that used more actively maintained plugins. The original AMS was built in 2013–2014. Lots of the plugins used were no longer supported and there were no good options for updating them. MINT, a data mapper, was the largest dependency, and MINT 1 was no longer supported, working with the Chrome browser only. As plugins die away, there were no good options. This provided a good reason to move to a Samvera solution where the community actively maintains the core components of the solution bundles.

### **A system MLA can continue to develop as needs change**

WGBH MLA wanted a system to evolve as needs change. Since the AAPB started with the inventory and first massive digitisation project, use of the AMS has changed. The original AMS was built by outside contractors, not WGBH or Library of Congress staff. It was built in PHP for which none of the WGBH MLA developers have actual knowledge or experience. The new AMS, built in Samvera, is a more mature project and with a future roadmap and a

robust community of expertise, both at other institutions and within WGBH.

## DECIDING ON AVALON VERSUS HYRAX

The initial plan was to build the AMS on Avalon 6 as this aligned most closely with AMS needs. This would have given developers time to focus on building AAPB-specific features, such as representing PBCore and the more hierarchical nature of the collections, on top of Avalon's current features. There were some problems, however. By the time the project launched, the Avalon team had decided to move the Avalon system onto the latest version of Hyrax (Hyrax 7). This entailed moving Avalon from a fully integrated system to individual core components that could be plugged in within the Hyrax framework. This work would not be done by the time the WGBH project needed to begin.

WGBH MLA decided instead to build the AMS starting with Hyrax. Some Avalon components were used. Some custom components were built, and WGBH worked with the Avalon team to build some components that did not yet exist in Hyrax. Custom AAPB features were identified along with what could be contributed back to the general framework for others to use, such as a batch import functionality.

## EXTENDING HYRAX

One of the core enhancements developed was representing AAPB metadata in the Hyrax framework. The AAPB uses PBCore, a metadata schema for audiovisual media that was initially developed by WGBH for the Corporation for Public Broadcasting. When expressed as XML, PBCore has a root document type called a Description Document. This includes data such as Title, Description, Subject, Contributors, Rights, etc. Most people using PBCore as the main metadata standard use this as the basis for their catalogue. A Description Document (Figure 3) contains a section of data about

AMS 2.0

### PBCore Description Document

- Asset**  
Title, Description, Subject, Contributors, Rights, etc.
- Instantiations**  
Physical Tape Format or Digital File Format, Duration, Media Type, Generation, Color, etc.
- Essence Tracks**  
Data Rate, Encoding, Sampling Rate, Aspect Ratio, etc.

Figure 2: PBCore Description Document

AMS 2.0

### PBCore Work Types in Hyrax

1 Description Document  
=/  
1 Work

Each repeatable section gets its own work type related to the others.

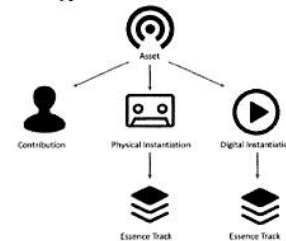


Figure 3: PBCore work types in Hyrax

the intellectual content and then repeatable sections of data about each copy of that content, called an instantiation. These can be physical or digital instantiations such as Physical Tape Format or Digital File Format, Duration, Media Type, Generation, Color, etc. Within the instantiation, there can be a section for essence tracks and specific things to record about each essence track such as Data Rate, Encoding, Sampling Rate, Aspect Ratio, etc.

One of the ways WGBH MLA enhanced Hyrax was to build PBCore work types into the Hyrax data model. In vanilla Hyrax, there is a work, which is essentially the asset. WGBH MLA added the ability to assign contribution works to an asset, as well as physical instantiations and associated essence tracks and digital instantiations and their associated essence tracks.

Because the AMS2.0 system is very Asset-centric, it has configured the search results to only return Asset records. All child records can be navigated to through the parent Asset. By doing this, the system stays

AMS 2.0

## Asset-centric Search

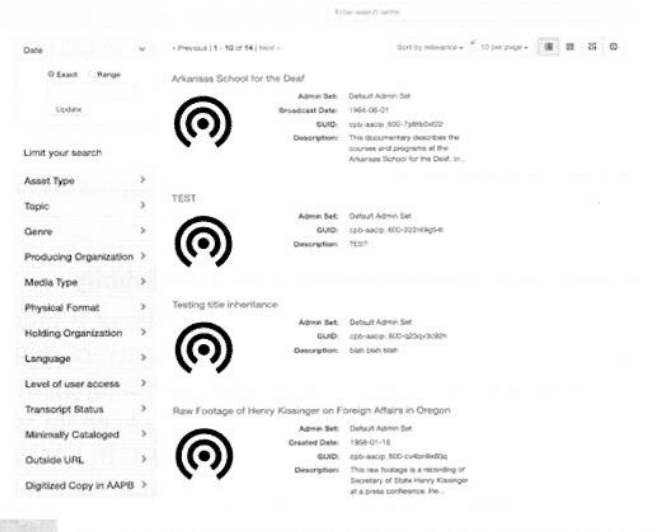


Figure 4: Asset-centric search

AMS 2.0

## Simple Cataloging Workflow

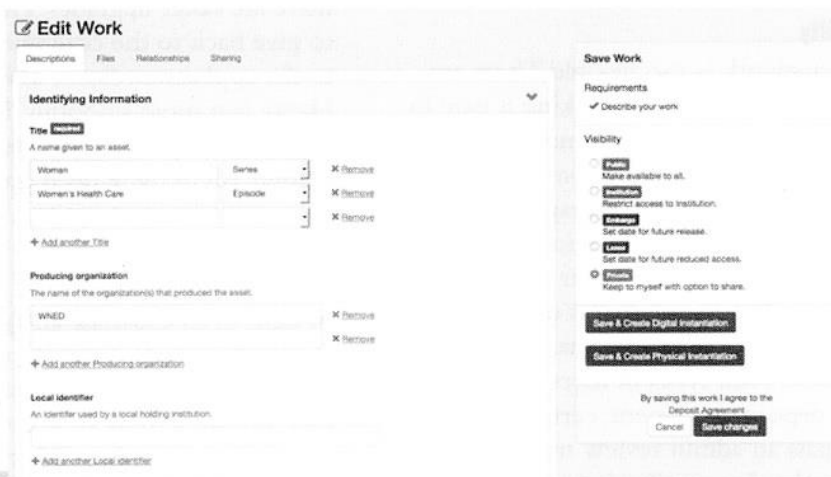


Figure 5: Simple cataloging workflow

analogous to how PBCore catalogues are expected to operate. It, however, is also hard to search for any data at the instantiation level. For example, a user cannot facet search if an item was moving image or recorded sound, or filter for assets that known copies exist on 0.75-inch videotape. Those types of searches are necessary for AAPB staff and contributors. The system had to index data from the child onto the parent work in Solr.

### Future data modelling ideas: broadcast series

Currently, series-level information must be repeated on each asset in the series. One future development is to model broadcast series within Hyrax's Custom Collection Types so that series-level information can be input once and then an asset may be assigned to belong to that series. Assets can, but do not have to, belong to a series. Assets can also belong to multiple series. When new Assets are imported,



an interface can help importers match them with series they should belong to.

### **ADVANTAGES OF HYRAX: WHAT YOU GET OUT OF THE BOX**

One advantage of Hyrax is it is a base platform to work from. There is automatic code generation and an existing framework. It is simple to generate a model, view, controller, actor stacks and tests. In addition, the styling looks pretty good. It is bootstrap and simple, but it can be customised if a more fancy or unique design is desired. It looks nice and clean with minimal CSS additions required. Another advantage is the flip flop functionality in Hyrax. It is easy to turn certain features on and off as required.

#### **Flexibility**

The framework is also flexible. There are configurable work types, making it easy to create multiple work types and define the possible relationships between them. The 'collection type' can be leveraged, and this is something that WGBH is looking at for a future iteration to solve other data modelling challenges. The requirements of file upload based on user type can be changed. For example, certain types of users are presented with a deposit agreement, certain users can bypass an admin review triage, and the visibility levels on collections and individual assets can be set.

#### **The Samvera and Hyrax community**

Another huge benefit is the Hyrax and Samvera community itself. It is rare to find a community that actually develops and is constantly available to answer questions when organisations want to implement. There is access to the original developers and it is easy to ping code authors with questions. People can easily go on Slack and ask questions like 'How are we supposed to be using this? Does it make sense to use this way? Do you have other ideas?' and there is always a response

and quick code resolution. The code WGBH MLA submitted back to Hyrax core has been merged within a day or so. Everyone is active and engaged and willing to help out. The community is a big advantage with real-time support and feedback.

#### **Maturity**

Hyrax is more mature than previous Samvera applications. WGBH MLA started development on Hyrax 2.0 and has been able to keep up with point releases since then. WGBH MLA has had a few commits to put code back into the community — features that were not just useful for WGBH but were useful for the Hyrax community as whole. Hyrax is also becoming a more stable product. It is becoming easier to use and there are faster upgrades. This makes it easy to give back to the community faster, back to the application faster, ultimately making Hyrax run more smoothly. No one in the community sprints ahead, leaving everyone behind. All of these been great advantages.

### **CHALLENGES SO FAR**

In light of all those advantages, there are always challenges. Everything is under development. This was known from the start, but something to keep in mind that complicates planning. Of course, no one wants a stagnant product, and new features will always be developed, but a decision needs to be made when to jump in and when to wait, and the answer will vary greatly depending on circumstances, and the value-add or gain for waiting.

The Avalon team was in the process of componentising features WGBH MLA wanted to use. WGBH MLA had to incorporate Avalon components such as the media player, but because of Avalon's coinciding development roadmap, WGBH MLA decided to push discussion of media display until later on. This informed the development roadmap, by looking at the features to work on early and those to

wait on. Some features depend on others, however, so there is an inherent hierarchy and order for feature development.

WGBH MLA worked on a generic batch import solution that was subsequently shared with the community. This was developed with the goal of building something that worked for WGBH MLA and other Samvera partners. Additional batch ingest solutions have been developed. WGBH MLA built a Hyrax-batch ingest gem with the Avalon team but it requires some heavy lifting to implement. It will require more work to make it easier to use. It was developed with the assumption that Hyrax would perform well with parallel ingest.

What should be contributed back to the community? What unique expertise does WGBH have that might be useful to others? Contributing code always takes more time than writing code. To participate and be a good community member, it is important to understand that contributing will take more time — not simply to make the code suitable for generic use, but there is a cycle, a process and a need to run tests, get feedback from community members, and sometimes change things based on feedback. It does lengthen the development cycle. There is a cost-benefit analysis to consider.

In an open source community, documentation tends to be either not so good or fantastic. The community itself is great source of institutional knowledge, but it needs to be documented with a referenceable tutorial. Lack of documentation has always been a challenge with scattered sources. There are a lot of places to look, even for just one section of the code. The community is aware of the issue and is trying to resolve it.

WGBH MLA was initially planning to launch AMS2.0 in June 2019, but concurrent processing during ingest posed a big challenge. This has been exceedingly puzzling and not performed very well in terms of either speed or success rate. WGBH MLA did not test large batches of data ingest until late in the project, so the lack of performance testing early on is a lesson

learned. The performance turned out to be prohibitively bad on the Amazon resources being used. WGBH MLA still does not know what the right combination is (but hopes to find out with some tests). WGBH MLA is currently in the process of ingesting the dataset from AMS1.0 to AMS2.0, the new system, and in the process of working out these issues with batch ingest.

In light of all the challenges, once the AMS 2.0 is launched, it is hoped that its benefits will outweigh its challenges. WGBH MLA knows already that the ingest of new collections will take longer in the new AMS than in AMS1.0, and there are plans to monitor whether it will pose risks to the efficiencies that are already pretty well established.

## SUMMARY OF THE AMS2.0 PROJECT

The AMS2.0 data model is based on PBCore and allows tracking of instantiations, which is important for managing information about multiple copies, and potentially different formats, of the same item/title. PBCore is a cataloguing standard for the description of audiovisual content, a data-sharing tool, and much more. Since its development in the early 2000s, dozens of organisations have been using PBCore's comprehensive and flexible features for their media archiving needs.

The original goal was to build off an instance of Avalon, using most of the basic structure and code in Avalon for the solution. As a result, the data model is different from the IU version of Avalon; although the technology stack is similar, it veers a bit from the overall Avalon solution bundle. The biggest problem with this solution is ingest speed. The data migration is in process and it is a slow migration, not managing bulk ingest very well. Bulk ingest may just be slow, and WGBH's case an anomaly given the number of files being ingested into a rather complicated data model. The Samvera community has been developing the ability to use the Samvera stack with alternative repositories to Fedora. A project called

Valkyrie is undertaking that work and it may prove to be the required solution. For systems managing media, this is a bit of a relief, because the media files were never stored in Fedora, but rather in external file servers with a pointer stored in Fedora.

### Implementing Avalon for WGBH

At the end of the AMS2.0 project, WGBH MLA was awarded an NEH Challenge grant to build capacity and infrastructure at WGBH MLA including the digitisation and preservation of about one-third of the collection that is on obsolete and aging formats — close to 83,000 items. Part of the infrastructure build was to develop a system to manage the created digital files for internal access and use.

WGBH MLA decided to use the Samvera solution bundle Avalon, as is, over Hyrax for internal WGBH content. There are a good number of examples of people successfully using Avalon in production, and for WGBH it did not need to be as strict about keeping the hierarchical data model, so going with the less configurable data model of Avalon was less of a hindrance for this project. There was very little development required, so it would be easy implement. WGBH MLA successfully mapped WGBH's data model (based on PBCore descriptive fields) onto the Avalon data model, and successfully tested the Avalon Ingest application programming interface, which ingests records and points to media stored in a separate system.

### FINAL THOUGHTS

Samvera solutions have a set code base and technology stack, but as with all systems, niche needs mean creating customisations. Working in tandem with an open source community brings challenges and benefits. The software and system may not require a licence fee, but staff are needed to maintain code for updates, versioning and customisations. In addition, being part of a community means contributing back, which is fulfilling but time-consuming.

Nevertheless, it is rewarding and beneficial to contribute to more than a single instance of a system and thereby help other colleagues. The community makes decisions about the direction of the technology, and those decisions are not dictated based on market revenue potential. Different components and software dependencies, however, are updated at different times, and the changes need to be tracked. In addition, the needs of an organisation, and the timeline to fulfil those needs, are not always in sync with others in the open community. An update or new version may take longer than expected to be released and customisations may cause problems with updates if too different from the basic code.

For Samvera, support is abundant with a community of developers, managers and users that can help answer questions. There is a pool of people to tap for staffing needs, and service providers to hire to help with a project. The Samvera community has service vendors available for hire to help build to specific needs, and those contributions remain part of the open community.

Currently, WGBH is installing an instance of Avalon to manage access to the growing digital archive. Once installed, and Avalon is being used in its current form, WGBH will begin offering contributions to the community for further enhancements and upgrades. WGBH may even opt for a hosted version of Avalon.

### ACKNOWLEDGMENT

This paper was based on a paper written by Richard Green and a presentation created by Sadie Roosa and Casey Davis-Kaufman.

### Reference

1. Davis Kaufman, C., Cariani, K., Roosa, S., Corum, J. and Myers, A. (2019) 'Implementing Samvera Open Source Technology at WGBH and the American Archive of Public Broadcasting Association of Moving Image Archivists', available at: [https://www.slideshare.net/WGBH\\_Archives/implementing-samvera-open-source-technology-at-wgbh-and-the-american-archive-of-public-broadcasting](https://www.slideshare.net/WGBH_Archives/implementing-samvera-open-source-technology-at-wgbh-and-the-american-archive-of-public-broadcasting) (accessed 10th April, 2020).