Building the Hydra Together: Enhancing Repository Provision through Multi-Institution Collaboration

Chris Awre, Tom Cramer

Abstract

In 2008 the University of Hull, Stanford University and University of Virginia agreed to collaborate with Fedora Commons (now DuraSpace) on the Hydra project. This project has sought to define and develop repository-enabled solutions for the management of multiple digital content management needs that are multi-purpose and multi-functional in such a way as to allow their use across multiple institutions. This article describes the evolution of Hydra as a project, but most importantly as a community that can sustain the outcomes from Hydra and develop them further. The data modelling and technical implementation are touched on in this context, and examples of the Hydra heads in development or production are highlighted. Finally, the benefits of working together, and having worked together, are explored as a key element in establishing a sustainable open source solution.

Introduction

The Fedora digital repository system[1] has been widely adopted around the world for a broad variety of uses (e.g., Jantz and Giarlo. 2006, Aschenbrenner et al. 2011, Tanifuji et al. 2009, Stracchino et al. 2009, Marill 2009). Those adopting the system have been able to take advantage of the elements contained within the name – a Flexible, Extensible, Digital Object Repository Architecture – and, for the most part, Fedora implementations are thus distinct. Whilst this is clearly a strength when having to manage different digital content types and collections, it has also led to solutions being created that have duplicated effort towards the same end goal. It also poses a dilemma for institutions wishing to cover the range of digital materials and different access requirements a truly institutional repository needs to deal with without re-inventing the wheel each time. The quest for a sustainable solution to this dilemma is a key goal in the embedding of Fedora repositories.

The dilemma was laid out by Fedora Commons (now DuraSpace[2]) in conversation at the Open Repositories 2008 conference in Southampton, and it became apparent that this was not an isolated problem. The Universities at Hull, Virginia and Stanford were each seeking a way to address this dilemma in a cost-effective way whilst still being able to take advantage of all the attributes Fedora provides. In other words, there was clear interest in how a reusable framework could be

[1] Fedora Commons, http://fedora-commons.org
[2] DuraSpace, http://duraspace.org/

established for use with Fedora to enable multipurpose, multifunctional repository-enabled solutions for use across multiple institutions.

The three institutions met with Fedora Commons/DuraSpace initially in September 2008, and at that meeting what would become the Hydra Project[3] was born. The issue they sought to address was encapsulated in two statements and assumptions:

- No single institution can resource the development of a full range of digital content management solutions on its own,
    - ...yet each needs the flexibility to tailor solutions to local demands and workflows.

- No single system can provide the full range of repository-based solutions for a given institution's needs,
    - ...yet sustainable solutions require a common repository infrastructure

From the very start, that concept of sustainability has underpinned discussions and developments. Whilst using Fedora for one-off collection management requirements was, and is, perfectly feasible, it was recognised that the three, now partner, institutions required a technical means to allow the repository to adapt to changing needs and content without having to implement another one-off solution: use had to be long-term. Nevertheless, it was also recognised early on that technology doesn't stand still, and all developments needed to allow for the day that Fedora itself may be replaced: sustainability of the content is required for the even longer-term. Whatever solution emerged would most valuably be one that could be applied, in principle at least, to other repository software systems as well.

Initial discussions highlighted that in establishing the framework solution outlined there was no wish to manage multiple repositories for different types of content, but that there was an equal desire to enable different views onto the repository to meet the needs of those different content types and the audiences accessing them. The use of the term 'Hydra' was, thus, deliberate – one body, many heads. Providing multiple points of access onto a common repository would enable more people to interact with the repository in different ways to meet their digital content management needs. This approach would support the development of internal communities of use as well as contributing to the wider Hydra community: placing the repository at the centre of digital content management infrastructure would also provide the impetus for sustaining its use.

The Quest for the Hydra had begun.

The early journey

Having established a common aim, it was agreed that it would be important to work together on that aim rather than go our separate ways and work on it

[3] Hydra Project, http://projecthydra.org

independently. The African proverb, *If you want to go fast, go alone, if you want to go far, go together* neatly summed up the intention. When setting out on this quest, though, it was recognised that we were not the first ones to do so. Interest in adaptable repository-enabled solutions for different content management needs has been reflected in other initiatives described at Open Repositories conferences and elsewhere, both for Fedora and other repository systems (e.g., Razum and Schwichtenberg 2009, Nguyen and Dalziel 2008, Phillips et al. 2007, Silva and Meece 2010). An important success factor around many of these was the presence or absence of a community around the technical development. Each partner would be able to bring their own insights and capabilities, and for sustainability, there is clearly safety in numbers. From its inception, Hydra has been designed to be an open, distributed project; an explicit aim was to develop not only a successful technology that would benefit each participating institution, but also to develop a thriving community that could maintain and advance the technology over time. A key aim has thus been to enable others to join the open source Hydra project as and when they wished, and to establish the mechanisms for sustaining the community and collaboration as much as any technical outputs that may emerge.

Initially, it was agreed to work together over a three-year period. Whilst project and institutional funding to support the effort would be sought, the Hydra project was initiated without specific funding, the common aim providing the impetus to undertaken the work. As the work has progressed support has been identified through related projects, and these have greatly assisted progress in different areas:

- Hydrangea: let the repository flower[4] – JISC, February-September 2011
- AIMS – An Inter-institutional Model for Stewardship of born digital archives[5] – Andrew W. Mellon, October 2009-September 2011
- EEMs (Everyday Electronic Materials)[6] – Andrew W. Mellon, September 2009-October 2010

Notwithstanding these distinct areas of activity, and the institutional priorities that have acted as key drivers in progressing the Hydra project, the heart of the work has been based around having a common repository infrastructure. However, this common infrastructure should not be rigid and presented as a total solution, but one that could be adapted and built on as required. Building on Fedora, the infrastructure should be about not re-creating Fedora's flexibility, but providing a structure that laid out basic ground rules for how that flexibility could be used effectively. Establishing that baseline would also allow the community to develop around it as different adaptations and uses emerged.

Hydra philosophy and responsibilities

The founding institutions have been equal partners throughout this development, and as Hydra Partners, now, have formed the initial basis of the governance of the

[4] Hydrangea: let the repository flower project, http://hydrangeainhull.wordpress.com
[5] AIMS project, http://www2.lib.virginia.edu/aims/
[6] EEMs project, http://lib.stanford.edu/eems

Hydra project. An important step through this process has been to exercise the original intention and include additional partners as appropriate. MediaShelf LLC came on board in 2010 as a technology partner, and has provided the impetus for putting so many of the Hydra design principles into practice. Notre Dame University, Northwestern University and the Rock and Roll Hall of Fame have sought to become Hydra Partners, and there have been expressions of interest from a number of others. This gradual expansion of the community has required the set of roles and responsibilities, the governance, to be established, to sustain ongoing development and that all Hydra Partners can, and will be expected to, contribute through. More formally,

"Hydra Partners are individuals, institutions, corporations or other groups that have committed to contributing to the Hydra community; they not only use the Hydra technical framework, but also add to it in at least one of many ways: code, analysis, design, support, funding, or other resources. Hydra Partners collectively advance the project and the community for the benefit of all participants."

There are four roles that can be undertaken as part of the Hydra project, some dependent on being a formal Hydra Partner. The first three are small, coordinating bodies whilst the fourth is open to anyone with an interest in using outputs from the Hydra project:

- Hydra Steering – This role is carried out by a core of the Hydra Partners, and has the responsibility for collaborative roadmapping and resource coordination. It is also responsible for governance of the technical core, evangelism, project infrastructure, and the organisation of meetings.

- Hydra Design – This role, currently encompassing input from all current Partners, has a focus on the functionality of Hydra and how this is defined and supported. It incorporates the definition of conceptual models, design patterns, data and content models (though see below), UI design, and appropriate documentation for all. A goal for those undertaking this role is to be able to share what is produced and identify funding opportunities to support the work.

- Hydra Developer – This role is open to all with an interest in the technical approach Hydra is taking, whether Partner or not. Hydra Developers will define the technical architecture, implement the Hydra Design requirements, coordinate development through community principles, manage release cycles, and document all work undertaken for future reference. There is a weekly Hydra Committers call, and regular mailings to the hydra-tech Google Group mailing list.

- Hydra Adopter – Anyone can download and run software from the Hydra project to develop their own Hydra head. Anyone can extend and modify software provided by Hydra. If you use the software, you're an adopter.

> Known adopters include Indiana University, University of Illinois at Urbana Champaign, Glasgow Caledonian University, as well as the Hydra Partners.

Inevitably as more contributions are made to the project by more people and institutions it is important that these contributions are both recognised and protected, particularly where software is being shared. A memorandum of understanding and partnership agreement have been drawn up in consultation with counsels at the Steering Group organisations, and separate code licensing agreements for both organisations and individuals have been established so it is very clear where code originates from and that it is being incorporated in Hydra overall. The Apache model has provided clear and valuable guidance and shape for these developments, as well as providing a trusted backstop reference point for those wishing to be involved in Hydra.

Hydra technical framework

Although not the initial starting point for the Hydra project, the ultimate aim is, of course, for there to be working Hydra solutions at the partners and other institutions. This aim has informed the development of the two components that make up the Hydra technical framework: data modelling and technical architecture design. The overriding emphasis in both cases has been to keep it simple. There is no value in creating a framework that is too complex for others to both use and build on. Hence, the ability to re-use what is developed in different ways, as in the use of the Lego™ bricks, has been an important undercurrent in the project's work. This has also allowed us to respect local needs and preferences. Indeed, this approach has informed an implementation model that has proved itself in practice, whereby Hydra Partners design for local need based on the framework and then feed back developments to the Hydra core.

Those wishing to implement their own Hydra solutions have two routes via which they can benefit from, and hopefully contribute back to, the Hydra community.

Adoption of the Hydra data model

As noted in an earlier section, adoption of Fedora brings with it a high degree of flexibility. A major part of this lies in how the content being managed and its associated metadata is structured within the repository. Fedora applies its digital object model as a container: however, defining a structure for content is a powerful tool that can be applied in other content management technologies as well. Hydra has thus sought to establish a set of baseline principles that aids the common structuring of content whilst still allowing for local adaptation and extension. The principle data model adopted by Hydra lays out the core metadata and structure that is required to build Hydra-compliant objects, leveraging the Fedora object model in the first instance but applicable elsewhere as well. Information on the models is available on the Hydra wiki[7]. Others are planned, though it is anticipated

[7] Hydra objects, content models and disseminators, https://wiki.duraspace.org/display/hydra/Hydra+objects%2C+content+models+%28cModels%29+and+disseminators

that a large proportion of content commonly held within repositories can be addressed using the initial models or combinations of these.

Key to the success of the way objects are stored is the metadata associated with them this that is required to shape a Hydra-compliant object. The following metadata datastreams are recognised as being of common value and use:

Compulsory

- Dublin Core (an internal datastream used by Fedora only)
- RELS-EXT (relationship metadata)
- rightsMetadata (metadata describing who can and can't view or carry out actions on an object – key for authorisation. Hydra is using its own rights metadata schema for simplicity, though others could be used)

Optional

- descMetadata (descriptive metadata, required where a splash page will be displayed, but not always required for all objects)
- contentMetadata (used to hold information about links from a splash page or object, for example an ORE resource map or METS StructMap)
- technicalMetadata (used as appropriate to store technical information about the object)
- provenanceMetadata (information about actions upon an object, e.g., as stored within premisEvents)
- sourceMetadata (information about where an object originates from, e.g., as held within METS sourceMD)

In all cases, these datastreams can be used to hold metadata in different formats, as suggested in the examples, albeit that for initial implementations simpler Hydra schemas have often been used. But the modelling has been carried to provide a baseline that others can make use of, without starting from scratch each time.

Fedora presents this structuring as a content model (or cModel, to use the Fedora terminology). In developing Hydra, different connotations for this term, though, were uncovered, resulting in an agreement not to call what we are doing a content model! Three meanings emerged, and the distinction is a useful clarification of elements of the Hydra architectural design:

- Atomistic vs. compound object modelling – Hydra defaults to atomistic for the majority of cases, though compound can be used if required
- Fedora cModels – Hydra makes use of these internal Fedora modelling constructs to structure the content in a Fedora repository
- Hydra Ruby models – The technical implementation of Hydra uses a models structure in Ruby on Rails to translate internal repository models to the user interface whilst retaining their richness

The Hydra data model, as it has become known, can be used as required to describe objects, and built out as required. Specific implementations can be shared with others across the community: hence, use of the data model for a collection of images at one institution can be described so that others might then benefit in their own management of images. There will also be different ways in which the data model is used for the same content type, and those coming new to the Hydra project in the future should be able to select the approach that suits them best.

Adoption of software from the Hydra project

The realisation of the modelling carried out in the Hydra project has been an implementation using Ruby on Rails. Initially produced as a beta reference implementation, the code is now presented as a collection of the different components involved, allowing adopters to use these as they require them (the Lego™ brick approach described in an earlier paragraph). A summary of these components and what they bring to Hydra is given in Figure 1. For those wishing to know more, a deep dive into the Hydra technology is available (Zumwalt 2011). All the code produced by the project is available through the project's github site[8].

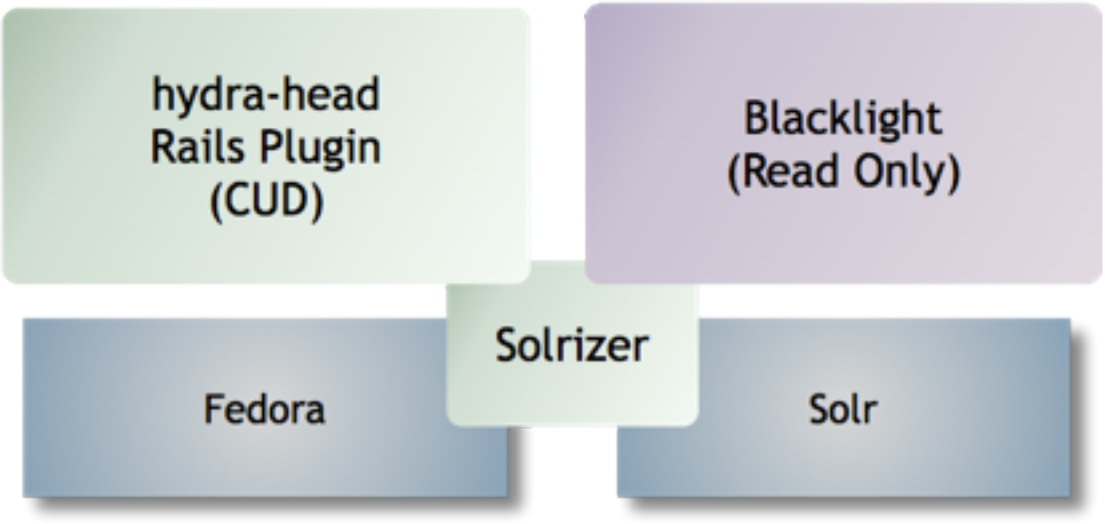

Figure 1: The Hydra technical framework implementation

The founding Hydra Partners have based their implementations of the Hydra head Rails plugin on the use of Ruby on Rails 3 applications, most of which are either gems or will be soon to facilitate cross-forking and merging of code. Why Ruby? This agile framework offers rapid development and deployment, a well-structured MVC approach that enables clean organisation of the code, and a solid testable environment. Hydra has a central continuous integration capability for all code submissions[9], to both encourage confidence in what is submitted from across the community and also to ensure proper process in managing code submissions from the wide number of committers.

[8] projecthydra github site, https://github.com/projecthydra

[9] Hydra continuous integration, http://hudson.projecthydra.org/

The health of a technology community such as Hydra can be measured through the number and regularity of commitments to a site such as github. An ongoing analysis of Hydra and its various components by Ohloh concluded that Hydra related open source development sits in the top 10% of open source projects in terms of effort expended and contributions made: this is the result of over 30 developers committing to the projects, far more than any one institution would be able to devote to this effort. The hydra-tech discussion list that is open to all has over 130 members and we anticipate interest developing further as implementations go live and the outputs from Hydra can be viewed more widely.

For those interested, a Getting Started guide is available via the projecthydra.org website[10]. Whether the routes described are followed as an Adopter or as a Partner is a local decision. Input from additional Partners is, of course, welcome to fully realise the aim of going far by travelling together.

Hydra heads

The implementations of Hydra so far have been wide and varied, and are described in Table 1.

Institution	Hydra head(s)/material type
Stanford University	SALT – a head to enable the management of digital archives and their presentation. This head is being used to inform development of Hypatia (see below). EEMs (Everyday Electronic Materials) – a head to assist in the capture and cataloguing of items sourced from the web for inclusion in targeted library collections. ETD – a head for the receipt and processing of electronic theses and dissertations
University of Virginia	Libra – a self-archiving head for the capture and open access for research outputs. Initially focused on journal articles, this can now be used for book, book chapters, and conference materials. Dataset management is also planned.

[10] Getting started with Hydra, http://hydraproject.org/technology/getting-started/

University of Hull	Hydra in Hull – a generic institutional repository allowing the management and dissemination of digital content as required by the University, including research outputs.
Notre Dame University	Atrium – a Hydra head facilitating the presentation of exhibitions based on repository collections. Videos – a prototype head for the management of video collections.
Northwestern University	Images – a Hydra head for the management of image collections, including the ability to crop/edit the images and store outputs from such actions.
Rock and Roll Hall of Fame	Video processing – a head to assist with the processing of a video digitisation process.
Joint initiative	Hypatia – a technical implementation of the AIMS model for the management of born digital archives.

Table 1. Hydra head example implementations

These are the heads in development and/or production. There are others in planning, and known commercial implementations as well through MediaShelf. The heads developed thus far offer both a cross-section of repository needs, and variations around similar content to highlight the flexibility that can be introduced to meet local requirements. A further development is the integration of Hydra with other institutional systems as part of the implementation, e.g., student admin systems for ETD processing at Stanford, and library catalogue integration for combined access at the University of Hull. The repository is thus an embedded part of the institutional landscape.

Why work together?

There have been many digital repository developments in recent years, many showcased through Open Repositories. Some have been adopted widely, many have not. The Hydra project has sought to learn from these experiences in adopting its own path:

- Hydra provides a core basis upon which others can build (travel?), assured that they are developing in a way that others will find useful in their own environments.
- Hydra provides a data model that can be used by others, avoiding the need to establish individual models on each occasion.
- Hydra continues to provide software that others can, and have, used to address local needs, and allowing them to focus on these needs as a priority over the underlying infrastructure.
- The Partners are guided by the governance that will allow partners to further contribute at the appropriate level or area of interest.
- Hydra is informed by a community of users that can provide mutual support in both development and use of the repository solutions that emerge.

Starting a repository project from scratch can be a daunting 'hill' to climb. In bringing institutions and people together, Hydra is seeking to provide a tested path over that hill.

The Quest for the Hydra

So have we gone far by travelling together? Undoubtedly yes. We have caught many glimpses of what we thought Hydra was over the course of the collaborative discussions, many of which turned out to be mirages, and there has not always been agreement. But the implementations now in place are built on the solid foundation of a clear view of what Hydra is. Following the expiry of the initial three-year period over which the founding Partners agreed to work there is renewed commitment to continue Hydra's development and seek out how it can further support the management of digital content within our institutions. A pattern of quarterly face-to-face meetings is in place and a key aim for the near future is to more fully formalise and roll out the governance of Hydra to provide the proper underpinning for future development. Alongside this, additional Hydra heads will be developed across the Partners and the developer infrastructure will continue to be enhanced to provide a proper basis for all contributions.

Do come and join the quest, and build the Hydra together.

References

Aschenbrenner, A., Enke, H., Fischer, T., Ludwig, J. (2011) Diversity and interoperability of repositories in a grid curation environment. *JODI: Journal of Digital Information*, 12 (2): 1-10, http://journals.tdl.org/jodi/article/view/1896

Jantz, R., Giarlo, M. (2006) Digital archiving and preservation: technologies and processes for a trusted repository. *Journal of Archival Organization*, 4 (1/2): 193-213

Marill, J. (2009) NLM selects digital repository software. *NLM Technical Bulletin*, Issue 368, 13

Nguyen, C., Dalziel, J. (2008) Muradora: A Turnkey Fedora GUI Supporting Heterogeneous Metadata, Federated Identity, And Flexible Access Control. *Presentation at 3rd International Conference on Open Repositories 2008, Fedora User Group, Southampton, 4th April 2008*, http://pubs.or08.ecs.soton.ac.uk/111/

Phillips, S., Green, C., Maslov, A., Mikeal, A., Leggett, J. (2007) Manakin: a new face for DSpace. *D-Lib Magazine* 13 (11/12, November/December 2007), http://www.dlib.org/dlib/november07/phillips/11phillips.html

Razum, M., Schwichtenberg, F. (2009) eSciDoc infrastructure: a Fedora-based e-research framework. *Presentation at 4th International Conference on Open Repositories 2009, Fedora User Group, Atlanta, 20th May 2009*, http://smartech.gatech.edu/jspui/handle/1853/28474

Silva, C., Meece, S. (2010) Kultivating Kultur. *Presentation at 5th International Conference on Open Repositories 2010, EPrints User Group, Madrid, 8th July 2010*, http://biecoll.ub.uni-bielefeld.de/volltexte/2011/5143

Stracchino, P., Feng, Y. (2009) Learning to YODL: building York's Digital Library. *Ariadne*, 30 (Issue 61, October 2009), http://www.ariadne.ac.uk/issue61/stracchino-feng/

Tanifuji, M., Takaku, M., Otsuka, S., Todoroki, S. (2009) Implementation and outlook of a new repository system at the National Institute of Materials Science. *Joho Kanri*, 52 (12): 888-901

Zumwalt, M., Sadler, B., Meloni, J. (2011) Hydra framework and Hydra developer community: open source collaboration in action. *Presentation at 6th International Conference on Open Repositories 2011, DSpace and Fedora User Group, Austin, 10th June 2011*, http://prezi.com/1lmhfhcvjhmm/hydra-technical-framework/