Submitted on: 14.04.2017

A Linked Data Approach to Digital Newspapers with Fedora and PCDM

David Wilcox
DuraSpace, Halifax, Canada
E-mail address: dwilcox@duraspace.org

Abstract:

Fedora is a flexible, extensible, open source repository platform for managing, preserving, and providing access to digital content, including newspapers. Fedora is used in a wide variety of institutions including libraries, museums, archives, and government organizations. Fedora aligns with modern web standards for modeling and exposing resources as linked data. The Portland Common Data Model (PCDM) is a flexible, extensible linked data domain model that is intended to underlie a wide array of repository and digital asset management system applications. PCDM provides a means to model data in an interoperable way, thus making it easier to share information across systems. Members of the PCDM community have undertaken an effort to model a variety of content using this domain model, including newspapers, which can then be implemented using Fedora.

This paper will provide an introduction to and overview of Fedora, with a particular focus on its rich linked data capabilities. PCDM will be used as an example to show how newspapers can be represented using a linked data model, and how this can be implemented in Fedora. Finally, the benefits of this approach will be discussed, which include but are not limited to: increased flexibility to support complex use cases, greater interoperability between systems and services, and opportunities to participate in the broader semantic web.

Keywords: fedora, open source, repository, linked data, data model.

Introduction

Fedora is a flexible, extensible, open source repository platform for managing and disseminating digital objects. Fedora is a software implementation - specifically a Java project - but it is based on a concept expressed in a paper published in 1998 by Sandy Payette and Carl Lagoze entitled, "Flexible and Extensible Digital Object and Repository Architecture (FEDORA)". In this paper, Payette and Lagoze detail a durable object model for digital repositories that responds to the fundamental requirement of an open architecture for digital libraries: "A fundamental requirement of an open architecture for digital libraries is a reliable and secure means to store and access digital content. FEDORA is a digital object and repository architecture designed to achieve these requirements, while at the same time providing extensibility and interoperability" (Payette & Lagoze, 1998, p.1). Extensibility and interoperability are key components of the original Fedora concept, and these ideas have been carried through the years as implementations have changed and adapted to new circumstances and requirements. Other key features of the Fedora architecture include support for diverse data types both currently and in the future, aggregation of these data types into potentially complex objects, the ability to disseminate these objects in different ways, and the capability of associating rights management schemes with these objects (Payette & Lagoze, 1998). These features, too, have stood the test of time, and can still be found in the most recent version of the Fedora software implementation twenty years later.

The most fundamental entity in the Fedora architecture is the digital object. This is a kind of structural container that includes any number of byte stream packages (PDF, JPG, XML, etc.), along with an interface layer that gives context to these byte streams. For example:

> *...a simple DigitalObject might have a structural kernel that contains a number of byte stream packages that are gif images and another byte stream containing Dublin Core metadata. On top of this structural layer there might be an interface layer that endows the DigitalObject with book-like behavior, allowing a client to access the table of contents or a specific page. The same DigitalObject might also have descriptive metadata behavior, allowing access to bibliographic fields such as the book's author or title* (Payette & Lagoze, 1998, p. 2).

Fedora is fundamentally an object repository; its purpose is to manage digital objects. This distinguishes Fedora from something like a triplestore, which contains many assertions (in the form of RDF triples) the subjects of which can be located at any URI anywhere in the world. By contrast, there is no data or metadata in a Fedora repository that is not associated with a digital object contained in that same repository.

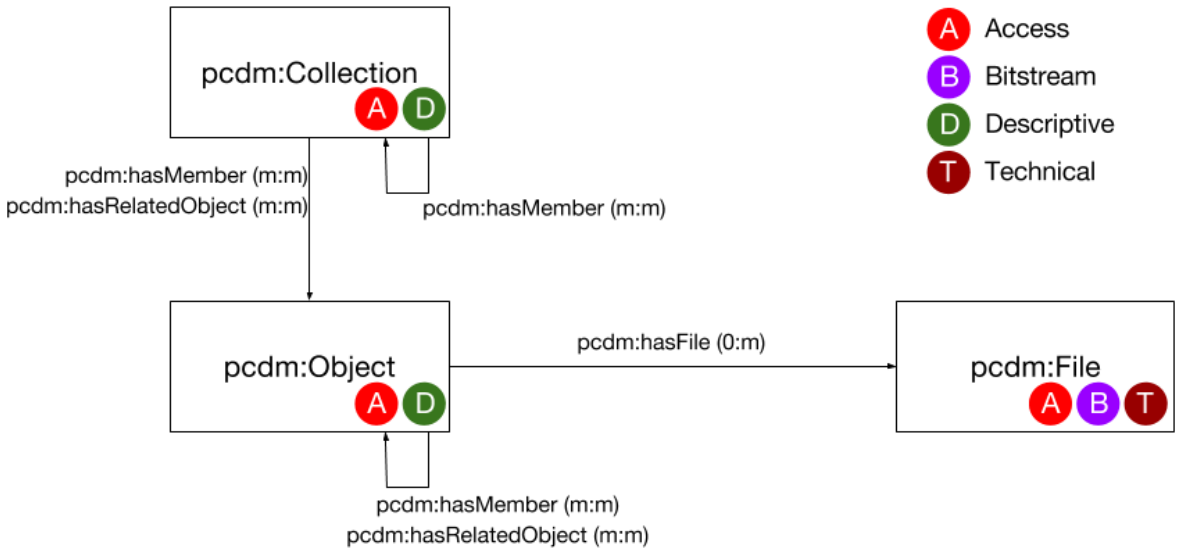
In addition to being an object repository, Fedora features native support for linked data. By leveraging the power of RDF to link together objects both within and outside the repository, Fedora provides enormous data modeling flexibility. Repository administrators are not limited by hierarchical data structures, which all too often fail to accurately represent the complexities of real world data. Instead, Fedora objects can be as atomic as desired, and can support whatever semantic relationships are needed to satisfy the needs of a conceptual data model, be it simple or complex. This foundation can be used to model objects, including digital newspapers, using the Portland Common Data Model, paving the way for greater interoperability and participation on the semantic web.

Portland Common Data Model

The Portland Common Data Model is a flexible, structural data model that is meant to underlie a wide variety of repository and digital asset management systems. It was originally developed within the Hydra ("Hydra Project," n.d.) community as a means to foster interoperability between different Hydra implementations, but it was quickly discovered that such an approach would be useful for broader interoperability within the Fedora community; for example, between Hydra and Islandora ("Islandora Website," n.d.) implementations. But even this scope was ultimately too narrow, as this model is not specific even to Fedora; rather, it can be used by any number of similar systems.

The purpose of PCDM is "to establish a framework that developers of tools (e.g., Hydra-based engines, such as Sufia, Curate, Worthwhile, Avalon; Islandora; custom Fedora sites) can use for working with models in a general way, allowing adopters to easily use custom models with any tool" ("Portland Common Data Model," 2016). PCDM takes a lowest common denominator approach, where it attempts to define a very limited set of general entities and relationships that can realistically be used to represent any number of different data structures. The model's developers and maintainers recognize that, in order to encourage adoption, "this model must support the most complex use cases, which include rich hierarchies of inter-related collections and works, but also elegantly support the simplest use cases, such as a single user-contributed file with a few fields of metadata" ("Portland Common Data Model," 2016). The model has already been applied to a variety of data types, including news media.

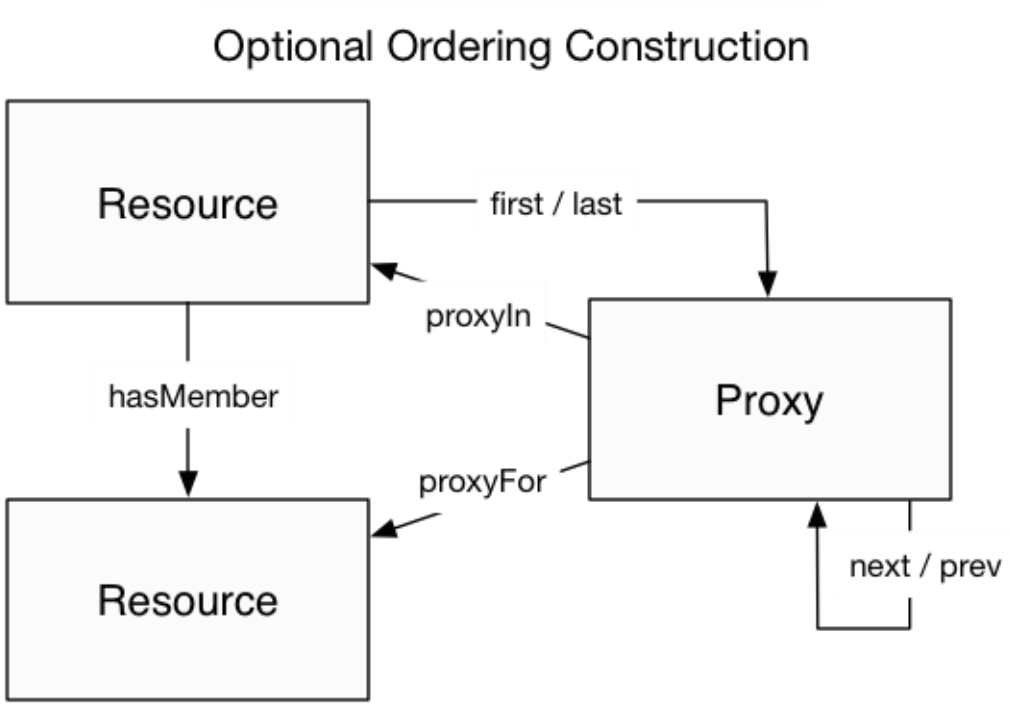
PCDM can be represented very simply using the following diagram:



PCDM specifies three primary entity types: collection, object, and file. Collections are, as the name implies, aggregations of similar items. Objects represent conceptual items, what we might think of as containers. Files are the byte streams or binaries themselves: PDFs, JPGs, TIFFs, etc. Conceptually, files are representations of the object they are associated with; for example, a newspaper page might be modelled as an object, and the representations of that page - a high resolution TIFF, a low-resolution JPG, an OCR text file etc. - would each be modeled as files associated with the page object.

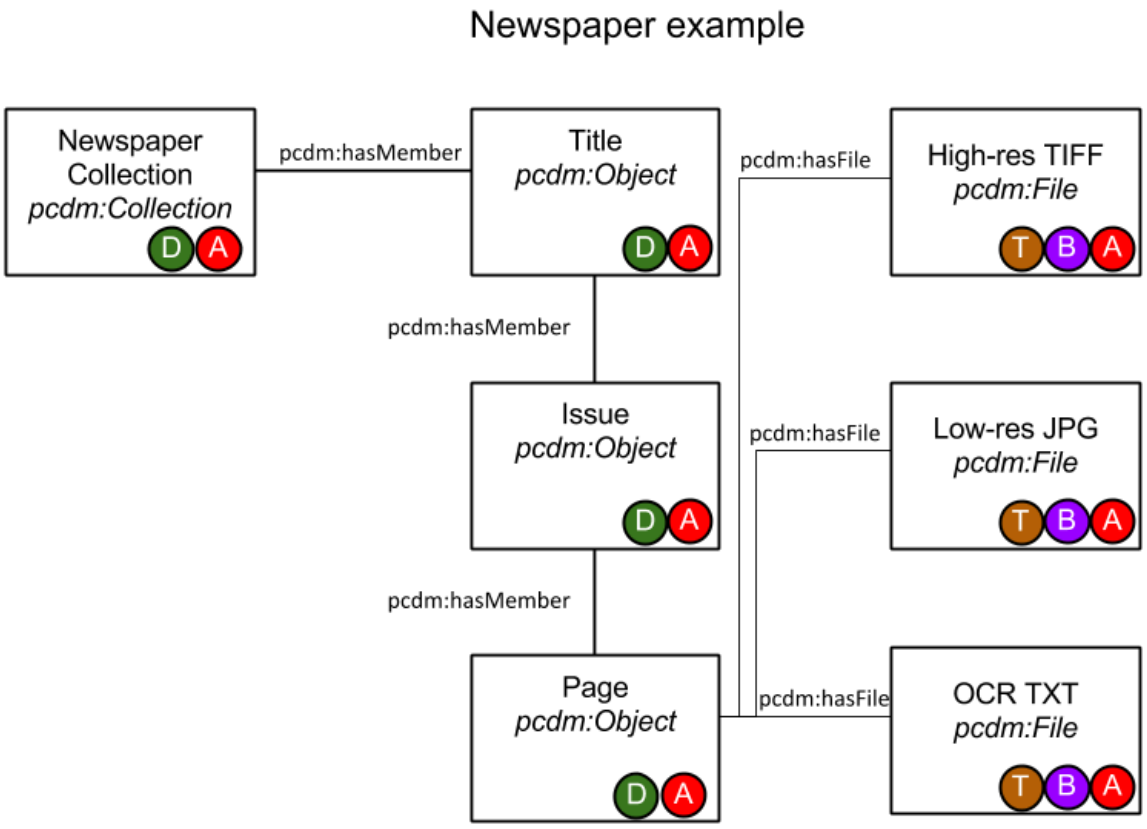
Collections and objects can be nested; collections can contain other collections, and objects can contain other objects. However, only objects can contain files, and nothing further can be nested under files. PCDM also dictates where metadata (in the form of RDF triples) may be stored: collections and objects can each contain descriptive and access metadata (i.e. metadata specifying what users and roles have access to a particular resource), but files may not contain descriptive metadata. This is because a file is seen as a representation of a particular object, so any descriptive metadata would exist at the object level. Likewise, only files may contain technical metadata; it would not make sense to store information related to filetype, file size, etc. at the object or collection level. Note that Fedora does not itself pose such restrictions - it is entirely possible to model resources however you wish in Fedora, whether in accordance with PCDM or not. But for the purposes of conforming to the PCDM domain model, these restrictions must be adhered to.

PCDM also provides a means of ordering resources; something that is certainly relevant to digital newspapers. Ordering is accomplished via an extension to the core domain model:



As we can see, ordering is achieved via proxy resources that stand in for the resource they are representing. In this way, a given resource (e.g. a book) could support multiple different page orders without creating duplicates of the original page resources for each order. The utility of this flexibility becomes obvious when we consider anthologies and compendiums; imagine, for example, a map collection. You might have several volumes containing maps of specific regions, each with their own page order, but you may also want to create an exhibit collecting maps from several different volumes into a single collection. Using the PCDM ordering extension, you could create two proxies for each map: one for its order in the original volume, and another for its order in the new exhibit collection. Each proxy refers back to the original map so there is no need to duplicate content, and new proxies and page orders can be introduced or modified easily.

With the basics of PCDM and page ordering in mind, how might we model a digital newspaper? As with any flexible data model, there is more than one way to do this; however, the purpose of PCDM is to provide a means of modeling data in an interoperable way. Therefore, it makes sense to standardize as much as possible on similar models within the community. Here is an example of how a newspaper might be modeled using PCDM:



As we can see, there are several items that must be mapped to PCDM entities: collection, title, issue, page, and the associated page derivatives. We can easily map the newspaper collection to pcdm:collection, and the title, issue, and page would each be mapped to pcdm:object (which can be nested as deeply as required). Each page derivative would be represented as a pcdm:file associated with the corresponding page object.

Once this initial basic mapping is complete we could layer on the ordering extension to account for page order within issues, and issue order within titles. While this introduces some complexity in terms of data modeling and maintenance, it is much more flexible than maintaining order statically with each page and issue. The result is an atomic, linked data representation of a newspaper than can be indexed and shared as widely as desired.

Benefits of Fedora and PCDM

One might rightly ask: why should I go through all this trouble to model and represent newspapers using Fedora and PCDM? This is a critical question, and one that we can answer in several ways.

One of the principal benefits of this approach is flexibility: simply put, it is difficult to anticipate future uses of the materials we steward. Who knows what future users and researchers will find useful? Who knows what novel discoveries someone might make using

our digital collections? With this uncertainty in mind, it makes sense to adopt a flexible, extensible data model that represents newspapers as atomically as possible. Using PCDM, we can use separate entities to represent titles, issues, and pages, and we can even go further to represent individual articles as separate objects. Of course, greater flexibility also means greater maintenance overhead (e.g. it is easier to maintain a smaller number of objects and RDF relationships) so resource constraints should be taken into consideration when developing a data modeling strategy.

Another clear benefit to adopting Fedora and PCDM is interoperability; both in terms of sharing data between Fedora instances and with the broader linked data web. Fedora is inherently flexible and agnostic with regard to how you model and store your content, which is why PCDM adoption is so important. Previously, different Fedora instances represented data in different ways, making it difficult if not impossible to share data between Fedora deployments. PCDM provides an opportunity for Fedora implementers to align their data models so data stored in one Fedora instance can be understood by a different Fedora instance. This is particularly important in terms of data migrations; Fedora can be used with different frameworks (e.g. Hydra and Islandora) and implementers may want to migrate from one framework to another over time. Implementers may also want to have a single Fedora repository and expose the same data through different frameworks to achieve different results. Without a common data model, this task is extremely difficult to realize. But by adopting PCDM, we can more easily share and migrate data across different Fedora implementations.

But what about the world beyond Fedora? As previously mentioned, Fedora implements the Linked Data Platform (LDP) recommendation from the W3C ("Linked Data Platform 1.0," 2015, February 26). This recommendation provides a standard for client/server interoperability on the web using linked data. As an LDP server, Fedora provides linked data in accordance with this recommendation. This means that clients implementing the LDP recommendation can interact with Fedora via its REST API and understand something about the content of the repository without knowing anything about the underlying technology. But this basic understanding only extends to the general structure of the repository; LDP is mostly concerned with how to navigate the repository using a concept called containment. It doesn't have anything to say about types of content (beyond a distinction between RDF and non-RDF sources) but PCDM provides another layer of description. A client that implements both LDP and PCDM could interact with a Fedora repository and understand not only the basic containment tree, but also how collections are organized into objects and their associated files. This level of interoperability can promote greater discovery of the contents of repositories on the web, thereby broadening the global linked web of knowledge.

Greater discovery brings its own set of benefits, including driving more traffic to your online collections. One of the primary principles of linked data is to link to canonical resources (via persistent URIs) where they exist rather than storing a local copy (Berners-Lee, 2006, July 27). This guards against unnecessary duplication, confusion about which resource is the main source of record, and the risk of incorrect or out-of-date information. For example, if your library holds a particular newspaper, you can provide it on the web via persistent URIs that other sites can link to. This not only establishes a primary source for the newspaper, it drives traffic back to your collections when people find the links elsewhere and follow them. Once they arrive on your site, they may continue to explore and discover other resources of interest. Thus, you can provide a useful service to the public while also increasing the relevance and usefulness of your collections.

Conclusion

As a flexible, extensible, open source repository platform, Fedora provides a means to manage preserve, and provide access to many different types of digital content, including newspapers. Fedora is focused on aligning with modern web technologies and standards, most notably the Linked Data Platform. This recommendation allows Fedora to structure resources as linked data and provide them to clients and applications in a standardized way. The Portland Common Data Model builds on the Linked Data Platform to provide an interoperable data model for Fedora resources, including newspapers. By applying this model to collections of newspapers in Fedora, we can structure our resources to be interoperable, not just with other Fedora implementations, but with any application or service that leverages LDP and PCDM. This standardization and interoperability opens the door to a number of great benefits, including the flexibility to adapt to changing needs over time, easier data sharing, and greater discoverability of your unique collections and resources. Over time these and other benefits will only increase as more applications and services adopt modern linked data standards and expose their resources on the linked data web. The future of libraries and other cultural heritage and scientific research organizations lies in the world of linked data and the many exciting opportunities it presents.

Acknowledgments

This paper would not have been possible without the support of the Fedora community, particularly those institutions that have joined DuraSpace as members in support of the Fedora project. The work of the Portland Common Data Model community has also been instrumental in providing a solid background for this work.

References

Berners-Lee, Tim. (2006, July 27). *Linked Data*. Retrieved from https://www.w3.org/DesignIssues/LinkedData.html

Hydra Project. (n.d.). Retrieved from https://projecthydra.org

Islandora Website. (n.d.). Retrieved from http://islandora.ca

Linked Data Platform 1.0. (2015, February 26). Retrieved from https://www.w3.org/TR/ldp/

Payette, S., & Lagoze, C. (1998). Flexible and Extensible Digital Object and Repository Architecture (FEDORA). Lecture Notes in Computer Science, 1513, 41-59. doi:10.1007/3-540-49653-x_4

Portland Common Data Model. (2016, August 9). Retrieved March 29, 2017, from https://github.com/duraspace/pcdm/wiki