Linked data experience at Macmillan: Building discovery services for scientific and scholarly content on top of a semantic data model

Tony Hammond and Michele Pasin

Macmillan Science and Education, The Macmillan Campus,
4 Crinan Street, London, N1 9XW, UK
{tony.hammond,michele.pasin}@macmillan.com
http://se.macmillan.com

Abstract. This paper presents recent work carried out at Macmillan Science and Education in evolving a traditional XML-based, document-centric enterprise publishing platform into a scalable, thing-centric and RDF-based semantic architecture. Performance and robustness guarantees required by our online products on the one hand, and the need to support legacy architectures on the other, led us to develop a hybrid infrastructure in which the data is modelled throughout in RDF but is replicated and distributed between RDF and XML data stores for efficient retrieval. A recently launched product – dynamic pages for scientific subject terms – is briefly introduced as a result of this semantic publishing architecture.

Keywords: ontology, OWL, RDF, science, semantic publishing, XML

1 Background

Macmillan Science and Education is a publisher of high impact scientific and scholarly information and publishes journals, books, databases and services across the sciences and humanities. Publications include the multidisciplinary journal *Nature*, the popular magazine *Scientific American*, domain specific titles and society owned journals under the Nature Publishing Group and Palgrave Macmillan Journals imprints, as well as ebooks on the Palgrave Connect portal.

Traditionally we have operated an XML publishing workflow with a document archive of over 1m articles and an averaged daily publication rate in the 100s of articles. As a prelude to moving towards a richer discovery environment in 2012 we began to experiment with linked data technologies and set up a public query service at data.nature.com[1] with RDF metadata describing a simple graph-based model.

In 2013 we embarked on a major new initiative to develop a new publishing platform for nature.com. We both extended and refined the linked data model

[1] We have since retired this service but will continue to make data snapshots available.

to manage our content and the relations between content items. Recently released discovery products in 2014 for this platform are based on this linked data foundation and include Subject Pages as a new navigational section, as well as bidirectional linking between articles and related articles.

2 Infrastructure

We have established a linked data architecture at the core of our publishing workflow and build on a common metadata model defined by an OWL ontology (see Fig. 1). This data model provides a number of significant benefits over traditional approaches to managing data: it encourages adoption of a *standard naming convention* by enforcing a global naming policy; it provides a *higher-level semantic plane* for data integration operations; it allows for *flexible schema management* consistent with an agile approach to software development; and finally it facilitates a simple means of maintaining *rich dataset descriptions* by allowing us to partition the data space using named graphs.

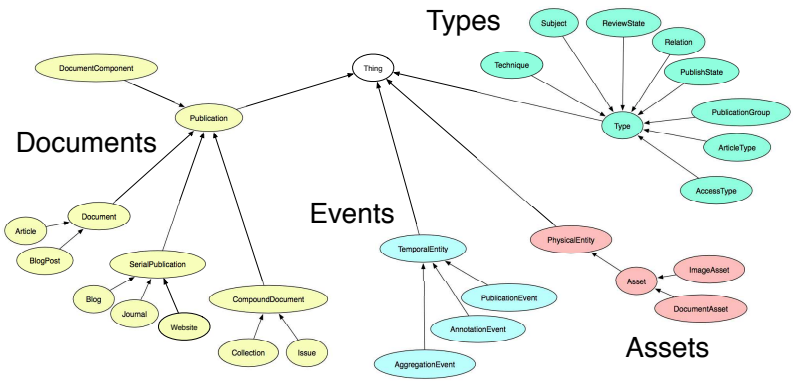

Fig. 1. Overview of main classes in the ontology.

To realize this common data model our Digital Systems, Science and Scholarly division has developed the Content Hub as part of the ongoing new publishing platform programme. All our publishable content is aggregated within the Hub and is presented as a simple logical repository. In practice, the data is distributed across multiple physical repositories. The ontology organizes the conceptual data model as well as managing the physical distribution of content within the Hub using XMP packets[2] for asset descriptions.

Our two core capabilities in managing the Hub are *content storage* and *content discovery*. Structured content is maintained as XML document sets within a MarkLogic repository which provides us with powerful text search facilities.

[2] http://www.adobe.com/products/xmp.html

By contrast, discovery metadata is modelled in RDF (and constrained using an OWL ontology). The discovery metadata is further enhanced by using RDF rule sets: object-oriented contracts for generating knowledge bases, and SPIN rules for inferencing and data validation.

3 Challenges

Initially we attempted to query the triplestore and deliver data through a generic, linked data API but increasingly were frustrated in meeting delivery expectations especially as query complexity mounted with multiple includes, specific orderings, facetting, and text searching requirements. We found that our implementation failed in two critical dimensions: performance and robustness. Typical result sets were being delivered in seconds or tens of seconds, whereas we were being tasked to achieve ˜20 ms, some 2-3 orders of magnitude faster. Additionally we faced system challenges based around enterprise features such as security, transactions, and updates.

It soon became apparent that in order to better support our online products we needed an application-oriented API that more directly reflected the page data model. This led to our developing a hybrid system for storage and query of the data model.

The main principles we used for the API were that data should be represented as consumed, rather than as stored; it should be provided in a single call and support common use cases in simple, obvious ways; that it ensure a consistent speed of response for more complex queries; and that it build on a foundation of standard, pragmatic REST using collections and items.

The data organization within the Hub is shown in Fig. 2. The data is modelled throughout in RDF but is now replicated and distributed between RDF and XML data stores. We have added semantic sections as RDF/XML includes within our XML documents. Retrievals are realized with XQuery, and augmented by in-memory key/value lookups, yielding acceptable API response times typically in the 10–100 ms range depending on complexity. RDF queries are currently restricted to the build time phase of data assembly with data enrichment and integration managed at the ETL layer using both SPARQL query/update together with SPIN rules. The API delivers JSON to a front end for rendering as HTML by querying with XQuery over RDF/XML includes in XML documents. These RDF/XML sections are subsequently exported into a triplestore for offline model validation and reporting.

In sum we use RDF/OWL as an organizing principle for our discovery data but we have preconditioned the data storage and index layouts in XML for simple and efficient retrieval access.

4 Products

The new publishing platform is increasingly being used to deliver products that benefit from a large-scale dynamic integration of related content. The first of

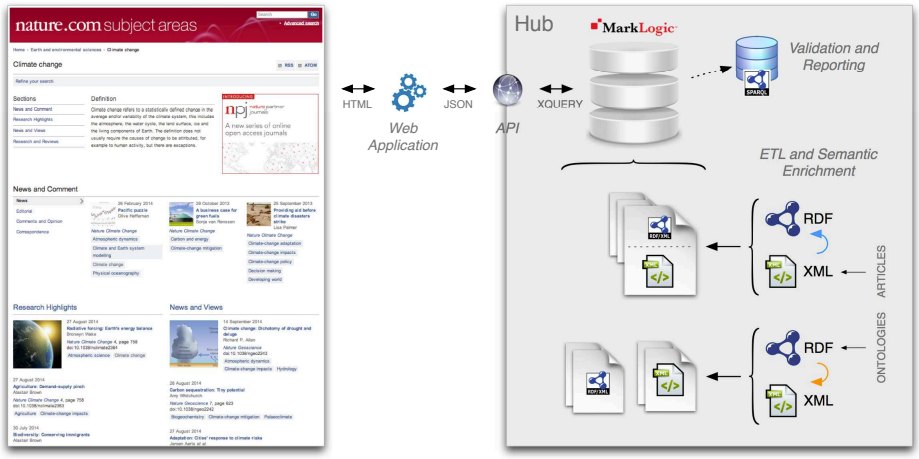

Fig. 2. Data organization in generic query architecture.

these products is Subject Pages[3], a new section on the nature.com site that allows users to browse content thematically, rather than via the more usual journal paradigm. This product was launched in early June 2014, and generated more than 200,000 views in its first month.

Subject Pages automatically aggregates content from across the site based on the annotation of that content (by authors and editors) using a taxonomy of scientific subject terms developed in-house. The taxonomy includes subject terms of varying levels of specificity such as *Biological sciences* (top level), *Cancer* (level 2), or *B 2 cells* (level 7). In total there are more than 2750 subject terms, organised into a polyhierarchical tree using the SKOS vocabulary. Subject Pages provides mechanisms to find items of interest, either by searching or navigating the hierarchy, and organizing the articles displayed on the page based on their predefined article-type (e.g. research or news). Moreover, subject terms are used to drive custom ads, jobs posts and events information which match the main topic of the page.

5 Future Work

Future aims are threefold: 1) to grow the data model with additional things and relations as new product requirements arise; 2) to open up the user query palette to more fully exploit the graph structure while maintaining an acceptable API responsiveness; and 3) to create an extended mindshare and understanding throughout the company in the value of building and maintaining the discovery graph as a core enterprise asset.

[3] http://www.nature.com/subjects