Leveraging existing Web frameworks for a SIOC explorer to browse online social communities

Benjamin Heitmann and Eyal Oren

Digital Enterprise Research Institute
National University of Ireland, Galway
Galway, Ireland

Abstract. Since online Semantic Web applications are based on existing Web infrastructure, developing these applications could leverage experiences with and infrastructure of existing frameworks. These frameworks need to be extended to deal with the different nature of Semantic Web data. We introduce several extensions to the Ruby on Rails Web development framework to support Semantic Web application development, and demonstrate them by developing a SIOC explorer. This online application integrates information from heterogeneous communities, allowing users to explore this information and find relevant posts and topics across these sites without the need to manually visit the different sites.

1 Introduction

Today's World Wide Web is undergoing a transition towards tomorrow's Semantic Web. Applications on the Semantic Web can provide benefits to their users by integrating data from many different sources. One challenge that developers encounter is the lack of support for developing such Semantic Web applications, compared to the many frameworks that exist for creating "ordinary" Web applications.

One option for developers, is to implement the Semantic Web elements of their application manually and to use an existing Web application framework for creating the Web interface. But as we will explain, this option is not optimal given the fundamental differences between Semantic Web data and traditional relational data, restricting the parts of the framework that can be reused.

Instead, this paper reports on the extension and modification of an existing Web development framework with components for consuming and processing Semantic Web data. This approach leverages the existing Web development infrastructure while also catering for the requirements of the Semantic Web.

We demonstrate this framework through the development of a browser for online social communities using the SIOC vocabulary. The browser allows the end-user to explore integrated information from various community sites (forums, mailing lists, weblogs) which would otherwise need to be visited separately. The aggregated view can be filtered using the rich post meta-data and allows users to discover people's contributions or active topics across sites.

1.1 Contribution

Our framework consists of several components that extend the Ruby on Rails Web application framework. Rails is a mature framework for developing Web applications, with many focused libraries and plugins and a large and lively developer's community. The first component is our ActiveRDF library, a high-level RDF API, that uses Ruby's dynamic meta-programming features to overcome several fundamental mismatches between object-oriented data and the Semantic Web (1). The second component is our BrowseRDF library for faceted navigation of large datasets. The third component is a SIOC crawler which integrates several command-line tools to form a "Semantic Web enabled processing pipeline" for fetching SIOC data. We combine these three components into our SIOC explorer prototype.

1.2 Outline

The rest of the paper is structured as follows: Section 2 introduces our motivating scenario: integrating data from various online communities. Section 3 discusses the relation between developing "traditional" Web applications and Semantic Web applications, and describes how to leverage existing development infrastructure. Section 4 then introduces our approach to Semantic Web development, consisting of three extensions to the Ruby on Rails framework, and demonstrates this approach in our SIOC browser prototype.

2 Scenario: integrating online social communities

As an example, we consider the development of an application for collecting information from online social communities. "Online communities" is a generic term for community sites such as forums, weblogs, mailing lists or IRC channels. Some of these channels (such as forums or bulletin boards) are more centralised, others (weblogs, IRC channels) are more decentralised and disparate. But from an abstract perspective all such communities are relatively similar: they allow users to group themselves online and exchange and discuss about their particular topics of interest.

Often, discussions range over several of these communication channels. For example, to solve an installation problem of a wireless card in the Ubuntu Linux distribution, a user should search the Ubuntu community forums for some helpful advice but also look on weblogs and the ubuntu-users mailing list. Currently, users have to browse these communication channels manually and repeat their query in various different systems: the forum software, the mailing list online archives, a weblog search engine, etc. For the end-user, it would be convenient if all these community sites were aggregated in a single place, allowing him to search for solutions to his problem in only one system.

Integrating such data currently involves: (i) collecting the heterogeneous data, i.e. crawling or "screen-scraping" and then parsing various formats and

websites such as RSS, Atom, (X)HTML websites, e-mail archives, IRC chatlogs; (ii) integrating these sources into a unified structure and format; (iii) consolidation of resources and concepts such as users and topics mentioned on different sites but with different identifiers (e.g. usernames or email addresses).

3 Application development from the Web to the Semantic Web

The World Wide Web is described by (2), as a combination of Hypertext and the Internet. Documents can link to other documents or parts of them, independent of the document location. This enabled the creation of interlinked webs of documents without central control or a central repository.

The quick adoption rate of this new paradigm of information sharing resulted in a demand for making dynamic content available. New systems were created, which could generate documents on behalf of user interaction and which could constantly incorporate new information into the created documents.

Tim Berners-Lee(2) not only envisioned the World Wide Web, he also outlined the vision of the Semantic Web. The Semantic Web was intended as a combination of knowledge representation using semantic networks and the Internet. Instead of having isolated repositories of knowledge, each with their own concepts and semantics, on the Semantic Web common concepts and their semantics could be shared and linked. This would allow knowledge repositories to link to each other, forming an interlinked web of data without central control or a central repository.

As standards emerge for such a Semantic Web, static but linked knowledge repositories are now possible. But, similarly to the situation on the "ordinary" web, a demand will arise for combining and processing data based on user interaction and for dynamically incorporating new data into the knowledge repositories to reflect changes. The resulting knowledge repositories will be of a dynamic nature, continuously integrating new data from various heterogeneous data sources.

3.1 The relationship between Web and Semantic Web applications

Web applications are part of the World Wide Web if they can be accessed over the Web: (i) they expose their functionality through URLs which provide access to the web application, (ii) these URLs can be used to access the human usable interface, and (iii) these URLs can also be used to access the machine processable interface, allowing integration between web applications.

Semantic Web applications are part of the Semantic Web if they can (i) consume, (ii) process, and (iii) optionally publish semantic web data. Accessed data can be the output from another Semantic Web application. Published data can in turn be used as input for another Semantic Web application.

Semantic Web applications can, additionally, be part of the World Wide Web, if they expose themselves on the Web. This class of Semantic Web applications build on existing Web infrastructure. Developing these class of applications

therefore can leverage the existing frameworks for developing Web applications, as long as we can make the necessary adjustments to overcome the differences between traditional, centralised, relational data and the upcoming, decentralised, graph-based Semantic Web data.

3.2 Decentralised data on the Semantic Web

The SIOC[1] initiative aims to ease the integration of online social community information (3). SIOC provides on the one hand a unified ontology to describe such online communities, and secondly, several exporters which translate community information from weblogs, forums, and mailing lists into RDF using the SIOC vocabulary.

Semantic Web data is expressed using RDF[2], a graph-based representation language. Statements in RDF are triples consisting of a subject, a predicate and an object which assert that the subject has a property with some value. RDF Schema[3] is the schema language for the Semantic Web, allowing the description of vocabularies in terms of classes and properties.

While the Semantic Web is data-oriented and the World Wide Web is document-oriented, both are fundamentally decentralised, heterogeneous, and open: anyone can make any statement at any location, using any vocabulary or structure. In contrast, as shown in Table 1, traditional database-driven Web applications are typically centralised, with a fixed schema, a fixed vocabulary and a single data source.

Web applications	Semantic Web
centralised	decentralised
one fixed schema	semi-structured
one fixed vocabulary	arbitrary vocabulary
centralised publishing	publish anywhere
one datasource	many distributed datasources
closed world	open world

Table 1. Traditional and Semantic Web data

The conceptual and physical decentralisation of the Semantic Web can lead to (i) naming differences, since one person might describe the "author" of a book, a second the "writer", and a third the "creator"; to (ii) differences in data structures, since one person might describe his language skills in his personal profile, another person his pets, and a third his favourite colours; and to (iii) federated storage, since statements can simply be published to any Web location without central registration. In contrast to typical relational database applications, the

[1] `http://sioc-project.org`
[2] `http://w3.org/RDF/`
[3] `http://www.w3.org/TR/rdf-schema/`

Semantic Web has no central data repository, no central agreement on meaning, no central policy on terminology, and no necessary agreement on structure.

3.3 Extending existing Web application frameworks

Frameworks for building Web applications, such as Struts[4], Ruby on Rails[5] and Django[6] are a popular way to develop Web applications such as our example online communities application. These frameworks overcome the traditional problem of Web scripting languages, the intermixing of business logic, presentation templates markup and database operations, by using the model–view–controller (MVC) pattern (4).

The MVC pattern separates an application into three parts: the application *model* manages data representation and business logic, the *views* present the data and manage user interaction, and the *controller* handles control flow. The Web application frameworks provide, for each part of the MVC pattern, middleware libraries that support application development. Given the different nature of the Semantic Web, some of these libraries can be reused directly but additional provisions must also be made:

Models: from relational data to Semantic Web graphs Existing frameworks rely mostly on a direct object-relational mapping to automatically construct the models in the MVC pattern from the relational schema (5). But Semantic Web data does not follow one fixed schema and some data might not be described by any schema. For example, when integrating data from online communities we will encounter descriptions outside of the SIOC schema, such as concept hierarchies using SKOS[7], user profiles using FOAF[8], or project descriptions using DOAP[9]. The use of such additional information is explicitly advocated by SIOC, since SIOC itself only describes basic relations between online communities. SIOC exporters are encouraged to use additional terms to express further information about their users or their discussion topics.

Views: navigating large and arbitrary datasets In existing frameworks application developers construct the navigation interface manually, although aided by more abstract HTML template languages. Such navigation interfaces are almost always limited to the application's domain model and restricted to data that the developers initially anticipated. But on the Semantic Web we can encounter arbitrary data outside of our initially expected schema, so additional provisions are necessary to allow navigation based on that data. For example,

[4] `http://struts.apache.org`
[5] `http://rubyonrails.org`
[6] `http://www.djangoproject.com/`
[7] `http://www.w3.org/2004/02/skos/`
[8] `http://foaf-project.org/`
[9] `http://usefulinc.com/doap/`

we would like to browse our integrated community information chronologically (by time of posting) and by the author of the post. Such properties are part of SIOC and we can thus manually create a navigation interface for them. But if we encounter richer author profiles including the workplace of authors, their country of origin, or their field of expertise (none of which are in the SIOC schema) we would like to filter the posts based on these properties as well.

4 Developing the SIOC browser prototype

To develop an integrated solution for the online communities scenario described in Sect. 2, we extended the Ruby on Rails framework with components for consuming and processing Semantic Web data. One such component is ActiveRDF (1), which addresses the "model" mismatch and maps RDF data onto objects. The second component is BrowseRDF (6), a faceted browsing engine that enables navigation of large Semantic Web datasets without domain-specific navigation knowledge. The third component is a SIOC crawler which crawls, extracts, normalises, and integrates SIOC data from various community sites (which use different methods of exposing and linking to their SIOC data).

4.1 Augmenting Ruby on Rails

Ruby on Rails is an MVC-based rapid application development framework (7). Developers can generate models, views, and controllers matching their data, and can customise these to implement their business and domain logic. The model is typically provided by an automatic mapping from an existing database, the controller describes the control-flow in Ruby and the view is specified through HTML templates with embedded Ruby code.

Ruby on Rails has two main strengths: on the one hand it provides default application logic for the generic parts of web applications and several helper methods for data manipulation and JavaScript effects, alleviating developers from these tasks. On the other hand, since Ruby on Rails is targeted towards web applications that operate on relational databases, it integrates the business logic with the domain data using an object-relational mapping: database tables serve as domain models and database tuples become Ruby instances.

Each of our extension components is designed to augment and integrate with Ruby on Rails. ActiveRDF can serve as a data layer in Ruby on Rails, replacing or augmenting the default ActiveRecord layer. The BrowseRDF navigation algorithms are implemented as a library that provides generic navigation on top of ActiveRDF; the library also includes helpers that generate the appropriate HTML navigation code in any Ruby on Rails application.

The SIOC crawler uses several libraries and command-line tools; cURL[10] and Hpricot[11] to extract links to the SIOC RDF data from the community sites; the

[10] `http://curl.haxx.se/`
[11] `http://code.whytheluckystiff.net/hpricot/`

Redland (8) "rapper" utility to fetch the actual SIOC RDF and to normalise it into ntriples; the Linux "cron" daemon to schedule periodic updates of the data; and Berkeley DB[12] as a persistent hashtable of visited sites.

4.2 SIOC explorer

Our prototype "SIOC explorer" aggregates data from various online community sites and allows users to browse and explore all disparate information in an integrated manner. The prototype, online at `http://activerdf.org/sioc`, source code at `http://launchpad.net/sioc-ex`, can be used as a feed reader to explore and subscribe to SIOC-enabled community sites such as weblogs, mailing lists, forums and IRC chats. The SIOC-enabled sites export SIOC data in a similar manner as RSS feeds. When prompted by the user, our application "subscribes" to such a feed and regularly polls these sites for updated content.

Fig. 1. Overview page of the SIOC explorer

All SIOC content is integrated into a local RDF store and then displayed in various ways. Figure 1 shows the overview page of the current prototype, with a list of several weblogs and forums. Users can decide to browse a particular forum, or see all posts aggregated from all sites. Not only posts from weblogs are shown, but also posts from from online community forums, IRC chats, mailing lists, etc. which are all described using the same SIOC RDF vocabulary.

After selecting a particular forum, the user is presented with the list of posts in that forum in the reverse chronological order, as shown in Fig. 2. As usual in feed readers, each post is summarised and can be expanded to read the full content. Also, "lateral" browsing is supported: clicking on the creator of a post shows all posts (including replies) written by this person, across all forums; clicking on a topic shows all posts tagged with this topic, again across all forums. In contrast to ordinary readers, our lateral browsing works across all types of

[12] `http://www.oracle.com/database/berkeley-db.html`

community forums: clicking on the user "Cloud" will not only show all his weblog posts, but also his emails from him and his contributions to IRC. The SIOC ontology enables this integrated browsing by providing the conceptual framework for unifying the content from the various community sites.

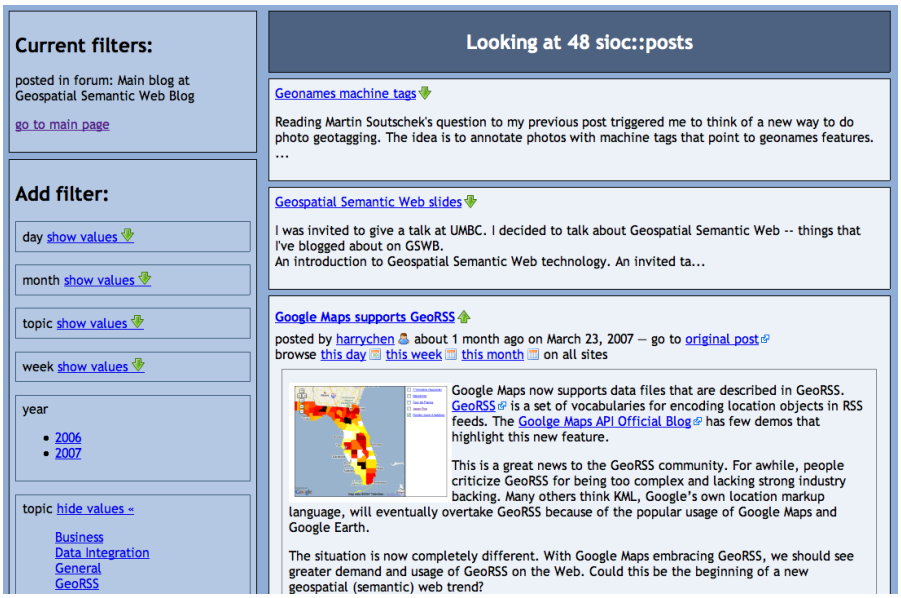

Fig. 2. Reading posts in the SIOC explorer

Finally, a generic faceted navigation interface is offered on the left-hand side, displaying relevant facets that are not part of the default interface. For example, since browsing posts by creation date, user, or topic is already supported through the "lateral" browsing discussed previously, those facets are not available on the navigation bar. But if the imported SIOC data (in this particular screenshot) has some unexpected facets, we can browse the posts based on e.g. creator, modification date, detailed user profiles (more than a simple username) and topic description (more than a simple topic tag).

Some facets (like the year) contain only "simple" values while others, such as maker or topic, can be further expanded to see subsequent subfacets. Fig. 3 shows the values for the subfacets of the facet "maker" for two different persons with posts about the topic "semantic web" in the database.

Application developers can customise the facet navigation to their needs and for example choose to exclude or include certain facets, or choose to exclude certain advanced operators such as the inverse join or the existential join.

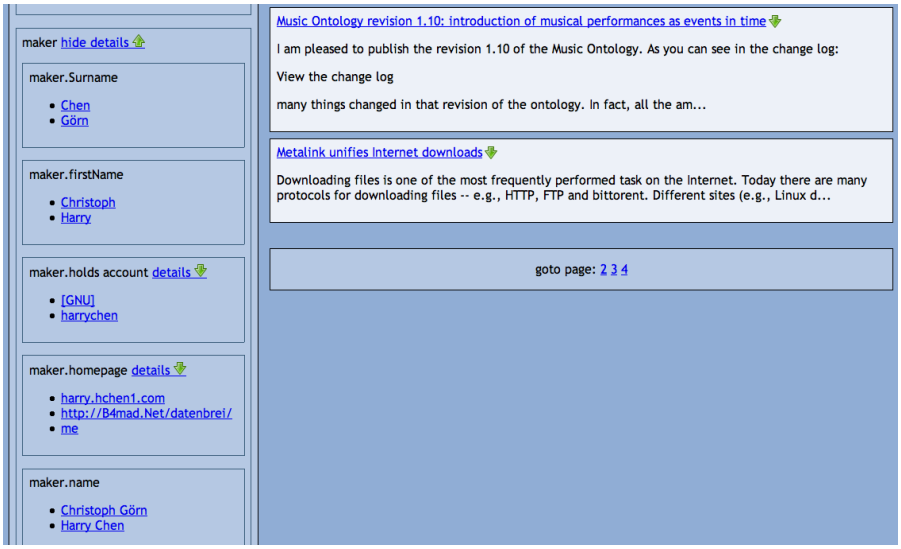

Fig. 3. Faceted browsing in the SIOC explorer

4.3 Development effort

Using ActiveRDF and our other extensions, the integration of Rails with RDF data was straightforward and the development effort was quite low. The models itself are automatically provided as virtual models, the controller (with all application logic) contains around 95 lines of code and the views contain around 100 lines of abstract HTML. The SIOC crawler consists of around 150 lines of code. Most control-flow handling (such as routing HTTP requests) and interface code (such as Javascript generation) is provided by the Rails libraries.

5 Conclusion

Since online Semantic Web applications are based on existing Web infrastructure, developing these applications could leverage experiences with and infrastructure of existing frameworks. Using existing frameworks abstracts typical implementation patterns and shifts implementation effort from the developer to the framework.

As we have discussed, existing frameworks, many of which are based on the model–view–controller paradigm, cannot be reused completely for Semantic Web applications without resolving additional requirements. On the "model" part, the impedance mismatch between Semantic Web data and object-oriented programming needs to be resolved. On the "view" part, next to the domain-specific navigation, an automatic domain-independent navigation interface is needed that is not restricted to a specific schema.

We have introduced our extensions to the Ruby on Rails framework and discussed how they cater for these requirements. As a practical scenario, we have shown how this extended frameworks supports developing a SIOC browser. This SIOC browser integrates information from heterogeneous communities, allowing users to explore this information and find relevant posts and topics across these sites without the need to manually visit the different sites. The application consists of around 350 lines of (manually written) code, apart from the generated Rails code, which could be considered quite little for its functionality.

Acknowledgements This material is based upon works supported by the Science Foundation Ireland under Grants No. SFI/02/CE1/I131 and SFI/04/BR/CS0694.

References

[1] Oren, E., Delbru, R., Gerke, S., Haller, A., Decker, S.: ActiveRDF: Object-oriented semantic web programming. In: Proceedings of the International World-Wide Web Conference. (2007)
[2] Berners-Lee, T.: Weaving the Web. Collins (2000)
[3] Breslin, J., Harth, A., Bojars, U., Decker, S.: Towards semantically-interlinked online communities. In: Proceedings of the 2nd European Semantic Web Conference. (2005)
[4] Reenskaug, T.: Thing-Model-View-Editor, an example from a planning-system. Technical report, Xerox PARC (1979)
[5] Fowler, M.: Patterns of Enterprise Application Architecture. Addison-Wesley (2002)
[6] Oren, E., Delbru, R., Decker, S.: Extending faceted navigation for RDF data. In: Proceedings of the International Semantic Web Conference. (2006)
[7] Thomas, D., Hansson, D.H.: Agile Web Development with Rails. 2nd edn. Pragmatic Programmers (2007)
[8] Beckett, D.: The design and implementation of the Redland RDF application framework. Computer Networks **39**(5) (2002) 577–588