Underpinnings of the Social Edition Appendix 3*

Ray Siemens

This work is produced by The Connexions Project and licensed under the
Creative Commons Attribution License †

External Image
Please see:
http://rup.rice.edu/image/cnx_images/shapes-buybutton.jpg

1 Appendix 3: Prototype Development Platform

1.1 Ruby on Rails

Ruby on Rails is a development framework that was created for use with the Ruby language. Its code generation tools and scaffolding make it ideal for fast and flexible prototype development. Developing simple features for a web application follows a very specific process in Ruby on Rails, and, since Rails handles a lot of the steps along the way for you, you save time. For example, forms that allow you to modify data in a database can be auto-generated by Rails. In PHP, by contrast, there are similar tools and frameworks that aid in rapid development, but there are too many to choose from and often time is cannibalized by the hunt for the most appropriate tools.

Ruby on Rails was chosen as our development platform because it offered a rapid prototyping environment, with a structured framework for documentation. This would save us time and provide us with a solid structure for our code base. In addition, because Ruby is a very object-orientated language, it would encourage the use of good software design principles. Ruby on Rails also provides an excellent basis for a Representational State Transfer (REST) API. REST API calls are special cases of regular HTTP URL requests, with an XML data payload defining the contents of the request. By documenting and publishing our REST API any partners that we wished to integrate and collaborate with would be able to make direct use of our database, using tools able to understand RESTful communications protocols.

1.2 Zotero

Zotero is an open source tool that automatically extracts bibliographic information from websites and organizes this data with the click of a button. As a Mozilla Firefox plug-in, Zotero is limited to users of that browser that have installed the plug-in. Even so, Zotero offers a quick and easy way to demonstrate some flashy reading tools for the next phase of development. For example, if PReE formatted bibliographic data for documents in a way that was easy for Zotero to parse, we could demonstrate this functionality.

*Version 1.2: May 14, 2010 6:06 am -0500
†http://creativecommons.org/licenses/by/3.0/

1.3 eXist

eXist is an open source database management system built on Java and XML technology. It stores XML data according to the XML data model and features efficient, index-based XQuery processing. Initially, an eXist database seemed like the best choice for storing and indexing XML data for PReE, so we developed an API to assist with interaction of the eXist database with Ruby and released it as open source (eXist XML-RPC API). Since then, we decided to move towards Fedora Commons and Fedora GSearch for storing and indexing our data. The advantages of using Fedora Commons (outlined below) seem to outweigh the advantages of using eXist.

1.4 Solr

Solr is an open source enterprise search server based on the Lucene Java search library, with XML/HTTP and JSON APIs, hit highlighting, faceted search, caching, replication, and a web administration interface. It runs in a Java servlet container such as Tomcat. Because of its search capabilities, Solr was integrated into the Ruby on Rails development of PReE. In the current version of PReE, metadata from the Shakespeare SonnetsREKn subset is injected into Solr's index. When a user searches for a document, Solr looks through the metadata for each document in PReE and identifies documents that seem to match the users query. Solr does a very fast job of this, as it is based on the quick Lucene Java search library. Although we were keen to investigate using Fedora GSearch for PReE, this tool has not yet been easily incorporated into the Ruby on Rails development framework.

1.5 Fedora Commons, Fedora GSearch, and RubyFedora

Fedora Commons provides sustainable technologies to create, manage, publish, share and preserve digital content as a basis for intellectual, organizational, scientific, and cultural heritage by bringing two communities together. Fedora Commons has number of advantages over using a Solr server for REKn/PReE. Essentially, documents in Fedora Commons are represented by objects in the system, and these objects have many unique abilities that will be of significant benefit to PReE:

a. Fedora objects have a special data stream reserved for Dublin Core metadata. When full-text documents from REKn/PReE are fed into Fedora Commons, the document metadata can be attached to the same objects as the full-text.
b. A Fedora object can have any number of data streams of many different types. If we wished to attach a series of images to a documents object, for example, this is easily accomplished in Fedora. We could also, for example, attach the original RTF file of a sonnet to a documents object. The possibilities here are endless.
c. Fedora objects data streams can be versioned. This ability would allow us to display different versions of the same document should we wish to implement this feature.

Fedora GSearch allows for searching these objects using Solr. We made contact with the developers of RubyFedora, a code library to make integration between Ruby on Rails and Fedora Commons much easier. RubyFedora would assist in incorporating Fedora Commons into PReE, but progress was cut short when it was realized that the library had not yet included the Fedora GSearch tool. Including the feature to search for documents inside a Fedora Commons repository would take a lot of time if we were forced to code the interaction with Fedora Commons and Fedora GSearch ourselves. Until such times as the Fedora GSearch tool has been built into the RubyFedora library, Fedora Commons would not be used as a replacement for Solr as a document index for PReE.