Repurposing VIVO Content with Drupal

MOTIVATION

VIVO consolidates content Historically it has been difficult for academic institutions to share Web content and keep it up-to-date without significant effort. Because the VIVO system aggregates content from many types of institutional data sources and makes it available in a standard, accessible format, VIVO can become an incredibly valuable resource for those working with institutional websites.

CMS + VIVO = less effort, more possibilities for websites

Web content management systems (CMS) include common tools for creating a site content, structure, functionality and appearance. By bringing VIVO content into a CMS, it becomes possible to restructure and compile VIVO content in a number of ways. VIVO content can be mixed with other content and, if configured to do so, can be kept up-to-date automatically or customize information within VIVO.

USE CASE: CALS Impact

Goal: Leverage VIVO’s web of relationships and various Drupal modules to create a dynamic, faceted browse and search interface that makes impact statements accessible and ultimately illustrates the scope of CALS research.

CHALLENGES

The vast majority of Drupal features rely on content being stored in the SQL database. To scale Drupal’s capabilities, it is necessary to:

- Define a VIVO RDF content type.
- Create a mapping.
- Persistently, the two systems align fairly well (see figures below).

VIVO content is to be added to the Drupal database (versus retrieving it as the file), this needs to be a mechanism for keeping up-to-date.

Existing RDF-related tools for Drupal focus on locally stored RDF and don’t offer results:

- Functional integration with VIVO.
- Some VIVO instances may offer a SPARQL endpoint, making it easier to retrieve content, but most won’t have content readily available for consumption proximately or Linked Data.

Drupal is a versatile CMS

As one of the largest open source CMSs, Drupal has evolved to become a very flexible framework for building both Web sites and Web applications. The availability of thousands of plugins modules) and a mature user-developer community make it a great candidate for integration with VIVO. While any development has been specific to the Drupal system, a similar approach could be used with other CMSs.

APPROACH

RDF data feeds

My approach to connecting Drupal with VIVO has been to develop a Drupal module that uses remote RDF content as data feeds.

The RDF Importer module allows a user to define a data feed by entering a simple list of Uniform Resource Identifiers (URIs). Advantaged users can generate the list of URIs dynamically with a SPARQL query.

The module is based on two powerful open source tools:

- RDF Importer is based on two powerful open source tools:
- http://drupal.org/project/feeds
- http://github.com/milesworthington/rdfimporter

Features needed for Drupal

A flexible import system that can be configured to handle any type of data and plug-in. Plug-ins can be written for each of the three import stages: fetching, parsing, and processing (saving) data.

Goals for RDF Importer module

- Make it as flexible as possible, including the integration with VIVO and RDF. To bring VIVO content into Drupal.
- Provide a means for keeping imported content up-to-date.
- Keep the process as generic as possible so it can be used with any VIVO instance or even remote RDF sources other than VIVO.
- Learn about how to organize content up to the site builder.

Difficulties with this approach

- Content stays only as fresh as the import frequency.
- With VIVO there is not yet a good way to check whether remote content has changed without retrieving all of it.
- Large tools require large amounts of bandwidth and server resources.

NOTES

RDF Importer module code

While the code for the RDF Importer module has not been posted online yet, it will soon be available on GitHub at:

http://github.com/milesworthington/rdfimporter

Questions and comments

Please feel free to contact me about anything regarding this work.

mwo@cornell.edu