Utilizing Drupal for the Implementation of a Dublin Core-Based Data Catalog

Lisa Federer, MLIS, MA, AHIP | NIH Library | Office of Research Services | Office of the Director | NIH | Bethesda, MD

For more information, contact Lisa Federer at lisa.federer@nih.gov

NIH Library | http://nihlibrary.nih.gov

NC-2014 – Austin, Texas

Background
As biomedical researchers are increasingly asked to share their data, solutions are needed to help them easily describe their data. The ideal solution would support multiple metadata schemas and subject-appropriate controlled vocabularies, as well as provide search and browse capabilities for end users to enable data discovery and facilitate re-use.

Why Dublin Core?
Biomedical research covers a broad range of topics and comprises many different types of datasets, including both clinical and bench laboratory research. Given the heterogeneous nature of biomedical research, a broad, generic metadata schema like Dublin Core is more suitable than a subject- or discipline-specific metadata schema.

What is Drupal?
Drupal is open source content management software designed for organizing and publishing digital content. Because it is highly customizable, Drupal is useful in a variety of contexts. Drupal is well-suited for implementing a data catalog because it allows administrators to create structures and controlled vocabularies to make it easy for users to input and describe content.

Incorporating Controlled Vocabularies Using the “Taxonomy” Function

The administrator can create multiple taxonomies and controlled vocabularies. The Drupal user community has also created taxonomies that can be imported. Taxonomies support multiple levels of parent/child relationships. The terms become part of the autocomplete widget function mentioned above.

Future Challenges and Opportunities
Data sharing is a developing practice that will continue to evolve as more funders and journals require researchers to share data. Librarians and data managers should assess researchers’ workflows for describing and discovering data to ensure that data catalogs enable data discovery without being cumbersome to use.

Building the Catalog with the “Structure” Function

For each content type, the administrator can create a set of fields, including help text, and can define parent/child relationships between fields. The “term reference” field type allows for the use of an autocomplete term widget that draws from predefined controlled vocabularies.