VIVO v1.8 Release Announcement

For more detailed information on the features described here, please see the VIVO 1.8 Release Notes

More modular architecture

The VIVO application is now partitioned into a main body and seven modules focusing on specific application-related functions including content persistence, application configuation, the search engine, the search indexer, reasoning functions related to the ontologies in VIVO, the system to store files such as uploaded images, and the tool to process uploaded images. The modules provide well-defined interfaces, encouraging alternative implementations from the VIVO community.

Performance improvements

The time required for a full re-inferencing of VIVO's content has been dramatically reduced and rebuilding the search index is also faster in v1.8, due largely to reduced memory usage. VIVO v1.8 also starts up in about half the time required by VIVO v1.7.

Interactive management of "faux" properties

VIVO v1.6 introduced the idea of faux properties, which provide improved specialization of property labeling and display on profile pages without adding new OWL properties to the VIVO-ISF. VIVO v1.8 includes new pages for viewing and managing faux properties interactively.

Individuals represented by VCards

VIVO v1.8 supports the practice of representing otherwise unidentified co-authors and other people external to the host institution using VCard information, rather than as FOAF persons with VIVO pages in their own right until further information about their identity is known. In this way, the data model is not overwhelmed by co-authors, co-investigators, and other persons who are incidental to the institution.

Triple store flexibility

VIVO can be configured to use several different types of triple stores, but as of this release has only been extensively tested in production environments with Jena SDB, which remains the default configuration. VIVO has an interface to a generic triple store that supports standard SPARQL query and update protocols, such as Jena TDB running either within the application or through an Apache Fuseki SPARQL endpoint. VIVO also has recently been confirmed to run on Virtuoso via a SPARQL endpoint, but again has not been extensively tested with that triplestore.

Added schema.org markup

The HTML markup in profile pages now includes "microdata" tags that are recognized by most popular search engines. These tags allow the search engine to easily extract structured data from VIVO, improving the appearance of search results.

A new customization guide

The VIVO documentation now includes a guide entitled ***Changing the appearance of VIVO***. Much of this information was already available in the VIVO wiki. It has been curated, updated, and augmented with new customization techniques. This guide is available in the VIVO wiki, as a tree of 22 wiki pages. It is also included in the VIVO distribution, as a 118-page PDF document.

A new search index builder

The search indexer code has been re-written and re-designed, to make it more efficient, more maintainable, and more configurable. Most changes to the search index configuration can now be accomplished by changes to a runtime file.

Additional improvements

In addition to numerous bug fixes, changes were made to reduce duplication of CSS and JavaScript files, to improve display features and performance, to make VIVO compatible with Java 1.8, and to improve protection against cross-site scripting attacks and click-jacking.

GitHub committers for this release

- Nate Prewitt (University of Colorado)
- Graham Triggs, Alex Viggio (Symplectic)
- Ted Lawless (Brown University)
- Patrick McElwee (Duke University)
- Brian Lowe (Ontocale)
- Jim Blake, Huda Khan, Tim Worrall, Jon Corson-Rikert, Holly Mistlebauer (Cornell University)