¡Bienvenido a VIVO!

Moving from a custom application to one that uses VIVO-compatible Linked Open Data

Paul Friedman; Warren A Kibbe, PhD; Violeta Ilik, MLIS; Kristi Holmes, PhD; Justin B Starren, MD/PhD

**Migration**

**Objective**
- Map existing data to the VIVO-ISF ontology
  - Faculty data (appointments, research interests, clinical narrative)
  - InfoEd (grand and contract data)
  - PubMed (biomedical publications with collaborators)
  - Northwestern Active Directory (name, title, contact info)
- Ensure functionality remains in new system
  - Manage publication lists for reporting and metrics
  - Identify researchers involved in a specific area
  - Provide tools to visualize collaborations
  - Highlight faculty publications

**Process**
- Extract raw data into .csv files
- Publication export files match Symplectic Elements format
- Create local ontology extensions
  - https://github.com/viol/ontology_extensions
  - Ontology extensions to match NLM publication types
  - Additional data types created to accommodate unique identifiers and robust LatticeGrid data set
- Map data using Karma
  - Person, organization, positions, and publications data sets
  - http://usc-isi-i2.github.io/karma/
  - https://github.com/NUBIC/R2RML-Karma
- Import RDF via VIVO Site Admin

**Outcome**
- VIVO production instance with:
  - Faculty Member data
  - Organizational appointments and affiliations
  - Publication and co-author data

---

**Modularization**

**Objective**
- Extract functional components from the existing application
  - Visualizations:
    - Co-author network graphs
    - Organizational network graphs
    - Word clouds
    - Reporting
    - Institutional membership lists
    - MeSH similarity
  - Repurpose components for any VIVO-compatible Linked Open Data instance
  - Keep fully open-source

**Process**
- Note the functionality that does not exist in the VIVO software
- Talk to users to see which functions are most used and which are not
- De-couple components from the data schema and software stack
- Determine the libraries and data used by those components
- Extract as much code as possible:
  - UI (html, css) pulled out with very small changes
  - JavaScript left completely intact
- Create automated tasks to build data used by visualization libraries
- Leverage VIVO SPARQL Query API to extract data from triple-store

**Outcome**
- Open source project on github
  - https://github.com/NUBIC/vivo-vl
- Visualizations woven into VIVO instance

---

This project was supported from funding by the NIH CTSA grant UL1RR025741 to the Northwestern Clinical and Translational Sciences (NUCATS) Institute from the National Center for Research Resources