2014 March Hackathon at Duke

This page was for the 2014 Hackathon held at Duke University on March 18, 2014

A Hackathon will be held Tuesday March 18 prior to the 2014 VIVO Implementation Fest (iFest). The hackathon will be a chance to meet and work with other people working to develop and implement VIVO.

Possible groups or topics

Please add your project idea or name to an existing idea.

Faceted Search

Ideas: ???

Interested in teaming:

CV & Biosketch Generation to Google Docs, DropBox, other

Ideas:

> Work with and possibly extend or repurpose Mike Conlon's biosketch generator on Github. It's written in Python and builds data by making linked data requests to a VIVO.
>
> Time could also be spent investigating how a web service could get initial input for a user (e.g. build my biosketch) and then write the created document to a file sharing service like Dropbox or Google Drive.

Interested in teaming: Chris Barnes, Florida (maybe make it a mobile app, easy button "make me a CV")

Open Social Gadgets / ORNG

Ideas: Working with Eric Meeks on improving ORNG integration with VIVO 1.6 or new gadgets?

Interested in teaming:

Schema.org for VIVO profiles

Ideas: Use the schema.org Person type to add markup to VIVO's foaf-person template. Use Google's Rich Snippet tool to verify the markup.

Interested in teaming: Steve McCauley, Brown

Visualizations

Ideas: Integrating Simon's Capability Map functionality with Solr? Other visualizations?

Interested in Teaming:

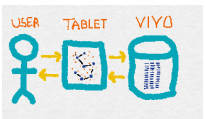
VIVO Dashboards

Ideas: For example, extended dev dashboard versus administrator dashboards (like Paul's idea at Weill Cornell)?

Interested in teaming:

VIVO Expert Finder

Idea: Create an expert finder mobile app so that VIVO data can be used by faculty and others to find experts at sites running VIVO. The user would indicate a concept of interest. The app would generate an interactive, bi-modal, force-directed graph of the concept and concepts associated with the concept, and people associated with the concept and people associated with the related concepts. User controls would support degree of association with concept for inclusion on the graph and other filters, recentering the graph on person or concept, hover for extra info about concept or person, click through to VIVO for concept and person. Click on the network thumbnail for a prototype interactive, bi-modal, force-directed graph. Click on the child-like art for tongue-in-cheek architecture diagram.



Possible technologies: Angular.js for mobile app framework, D3.js for graph layouts, Twitter Boostrap for CSS, SPARQL to VIVO for data. Javascript controller.

Interested in teaming: Mike Conlon, UF

VIVO Mobile

Ideas: For example, support for responsive web design like Bootstrap? PhoneGap or Native apps? Also consider these tools

- https://www.easel.io/ -- build Bootstrap-based web sites quickly, can import existing sites to get started, and then export HTML/CSS (FreeMarker?)
- http://easelinc.github.io/tourist/ -- a simple Javascript library for creating guided tours through your web app -- also see http://blog.easel.io/blog/2013/05/23/tourist-js/ – ALEX: Imagine Tourist.js in-browser tours for new VIVO users and admins ("how to extend ontology", "how to import CSV", "how to import existing ontology")

Interested in teaming:

VIVO use cases

Ideas: Gather to write use cases and requirements for VIVO at various institutions.

Jon Corson-Rikert has offered to lead this session.

Generic VIVO-Independent Relationship Creation Tool

Idea: Share and potentially extend the code used at LASP to build the skill relationship creation tool into a more general 'relationship mapping' tool. A generic relationship mapper could be used by any institution that wishes to have a webapp other than the VIVO UI that would allow users to add relationships to the VIVO database. Additional institution (or relationship)-specific information could also be added to the mapping tool, such as the 'skill level' dropdown menu associated with the skill being mapped in LASP's tool. This would allow users with domain-specific knowledge (but not necessarily any familiarity with semantic technologies or VIVO specifically) to add semantically linked metadata about their domain.

Possible technologies: Angular.js for application's MVC framework , SPARQL to VIVO for data (including SPARQL update API), generic Javascript.

Interested in teaming: Michael Cox, LASP

VIVO SPARQL update clients

VIVO 1.6 includes a SPARQL update API. This is a much anticipated feature because it will allow sites to automate data loading and processing and still take advantage of the VIVO reasoning and Solr indexing. During the Hackathon, a team can work together to build sample client libraries for a variety of scripting languages (Perl, PHP, Python, Ruby, others?) that can add, delete and manipulate VIVO data using the SPARQL update API. This will help team members to leverage the new API in their workflows and also serve as a reference for other people working with VIVO. Since the API is standards based, most existing SPARQL libraries can be used to work with the API. See an example in Python.

Interested in teaming:

Alternative RDF triple stores for VIVO

Ideas:

Study the published data on benchmarks and analyses of commonly used triple stores. Focus on the ones that show the fastest response times on complex SPARQL queries. The best candidate(s) for VIVO will be a triple store that is easy to set up and does not require a lot of code changes on VIVO. Note that the VIVO development team and other institutions have investigated the usage of triple stores other than Jena in the past. They are invited to share their knowledge and experience on this topic. For more information on the literature, Paul Albert from Weill Cornell Medical College has created a Google Doc for your reference.

Interested in teaming: Eliza Chan, Weill Cornell Medical College

Other?

Please add your project idea to the list.

Hacker Setup

If you would like to have a running VIVO instance, it's recommend to set it up in advance of the Hackathon. You can email Ted Lawless <tlawless@brown.edu> or Chris Barnes with questions on getting setup with the tools below.

The VIVO Apps and Tools working group will hold an hour long call on Tuesday March 4 at 1pm EST to assist with setting up VIVO software and tools for the Hackathon.

Resources

- VIVO Vagrant
    - A pre-configured VIVO instance. This is a quick way to get a VIVO instance up and running. If you want to use this Vagrant image during the Hackathon, please download and install prior to arriving since it takes several minutes and lots of bandwidth.
- Fuseki
    - Fuseki allows you to setup a SPARQL endpoint for VIVO that can be queried by remote scripts, services, software.
- VIVO tools and maintainers
    - Tools identified or created by the VIVO community for working with VIVO data.
- Version control - team members might want to collaborate using version control and posting their work to Github. If you are unfamiliar with Git, clients are available that makes getting started easier.
    - Github for Mac or Windows. This works well for collaborating via Github but is limited to that; it won't work with Git repositories stored on other servers.
    - Sourcetree from Atlassian. This will work with any Git host, not just Github.
- S2S Framework
    - From RPI, a user interface framework that can query SPARQL endpoints and build dynamic, faceted browse interfaces.
- http://jsfiddle.net/ -- Duke uses this and has some ideas for creating shared value for VIVO community

FAQ

Do I have to be a programmer or experienced with VIVO to attend?

- No. All are welcome. Those with ideas for reusing VIVO data or matching it with external sources can contribute a great deal.

I'm a programmer, but don't have a current Java development environment. Will I be able to participate?

- Yes. Many of the projects above will reuse VIVO data inside of other tools and laguages, like Javscript or Python. You could also use the Vagrant image listed about to get a working Ubuntu server with a JDK and VIVO installed. The VIVO project also maintains documentation on developing VIVO with Eclipse.