20110216 Developer Integration Meeting Notes

Agenda

2 /16	8 am	Breakfast (Einsteins delivered to room @ CTRIP)
	9 - 10: 00	

- Release planning as an integrated project
  - Branch structure (dev/staging/trunk?)
  - Discussing release schedules
  - Planning releases |

	10:00 - 10: 15	Brea k
	10:30 - 11: 15	

- Visualization
  - Caching
  - Getting Flash generated from source by Ant script
  - Geomap and Scimap integration |

	11:15 - 12: 00	Harvester and ingest
	12-1:30	Lunch Break (eat in)
	1:30 - 3	

- National Search
  - What will the national search be?
  - User facing features
  - Infrastructure to support these features
  - How it relates to search within VIVO
  - Semantic vs./plus text indexing
  - Supporting data coming from outside of VIVO
  - What could we deliver in the next release
  - How confine ourselves to what can be accomplished
  - Defining independent work components to proceed in parallel |

	3:00-3: 15	Break
	3:15-4	High-level road map review for the next release
	4 - 5	National Conference Call
	6pm	Dinner (Stubbies & Steins - need to check menu first)

What to Use as Our Process for Release Planning (9:07am - 9:44)

Jon and Chris B talked last night and they are interested in gathering input on features, bug fixes, and priorities from the implementation sites.

We have only six months left. That is a significant constraint.

Institutional memory is important, we may want to focus on preserving that.

Need to focus on features desired by NIH (most notably "national search").

We need to put effort into finalizing documentation and other end of project tasks.

Cannot pretend that architectural changes are zero time and zero cost. If we decide to do these we need to plan them, schedule for them, and promote them as completed work. We should not pursue them for their own sake but for as part of a larger goal.

Should we have an open call for feature request process to do planning?

We could solicit input on the release planning by asking, "We plan to do this, do you have a suggestions about this or how we could do it?"

Other projects do wish lists or adding issues as feature requests. Brainstorming sites (e.g. http://brainstorm.ubuntu.com/).

Maybe we should use Idea Torrent on Sourceforge. Can we hook this up with the feedback form? (Nick S. says we can just turn this on in Sourceforge as a hosted app)

We may want to make a road map that extends after the grant funding period. If we have the current devs do the design for large features then they will be documented for other developers who are less familiar with the system. Chris B. suggested a RFC sort of format.

Decision:
Over the next six months we will focus on features desired by NIH, preserving institutional memory and end of grant tasks. This will be informed by feedback from mini-grants recipients, implementation sites and user testing (not in priority order).

We will start exploring Idea Torrent as a tool to gather ideas on development directions. We will turn on idea torrent in the summer and promote it at the conference as part of this exploration.

What is the release schedule over the next 6 months? (9:45am - 10:16am)

Need to do national search, vivo will need a change to its search to accommodate this.

For the last conference we worked hard to provide a release beforehand but then most sites did not install the new release for the conference. We should avoid deadlines that aren't aligned with sites' needs.

We are likely to want an incremental development process for the national search – and not just introduce it as a finished product.

The harvester translations need to be kept in snyc with the ontology. The harvester translation files are not really considered part of the harvester code release. FL continually releases new versions of these files so they might not need to be synchronized with the ontology.

If we have more releases then we will have to devote more resources to testing.

Goals:

Allow people evaluating the system to use the latest code, ideally with a body of sample data.

Allow implementation sites to use the latest code, without waiting for a release.

We should not plan a release for right before the conference.

Visualization (10:33 am -)

Caching (10:33 - 11:14)

Background. The temporal graph vis needs counts of all the publications for a organization and its sub-organizations. This is recalculated every time the temporal vis is requested. We would like to cache this. Other people will want to cache things too. Can we implement a general solution?
Options:

1. cache the result of the construct query for each organization
2. cache the results of a construct query that generates a single model of just orgs, pubs and people and run the vis against this model
3. cache the JSON that is generated for each organization
4. store more granular results at the level of each organization that can be summed up into the count for a parent org. This might be a more general solution, but only for data that is meaningful to sum in that way.
5. use memcached (http://memcached.org/)
6. use an HTTP page cache like Squid (http://www.squid-cache.org/Intro/)

Is this a general problem of where do we want to store stuff that we do not want to show as part of the ontology or our primary triple store.

One solution:

Allow graphs of vis data to be added and saved and regenerated every night.

Have a class with the following methods:

```
Model generate( ModelSelector modelsFromTheSystem);
String getUri();
```

Then a running vivo system could generate this model and store it as TDB. We might need to regenerate this every night.

Do we need to return a dataset instead of a Model?

Do we want a way for these to end up getting displayed in the public front end?

Why not put the TDB in the SDB? Because then it would be in the set of public graphs.

Can flash be added to the build process? (11:14 - 11:15)

Right now the source code for the flash is not in VIVO SVN. We should move the code to the SVN repository and add ant tasks to do the compile.

Sci Maps (11:15 - 11:22)

UCSD developed SciMap but it is not open source. Can we include this without causing library problems? We think that we can.

Do we want to put effort into developing features against a library that we might not be able to use in the future?

Harvester and Ingest (11:34am -)

Ensure that all functionality that exists in the command line interface exists in the VIVO ingest tools. Most of the ingest tools are in JenaUtils or JenaIngestUtils.

Steven W. would like a user interface to create the harvester script as part of the administrator tools.

Take the features that already exist and integrate the harvester code, then develop a list of additional features.

Would like to allow the upload of MODS formatted citation references.

Brian L. does not plan to add features to the old workflow tools. There is a wiki page: http://confluence.cornell.edu/display/ennsrd/Ingest+Workflow+Language

look into

http://www.topquadrant.com/products/TB_Composer.html

JonCR lower hanging fruit might be documentation, sample data, and SPARQL query examples (constructs and selects).

Lunch (noon-1:30pm)

National Search

What are the user-facing features of the national component? (1:30pm - 2:33pm)

Google like?

Facebook like?

Inter-vivo linking? (currently implemented by defining an external namespace)

Info-graphic collaboration reports

Indexing Harvard profiles and Collexis RDF (in VIVO ontology)

Take advantage of semantic features

Dynamically faceted search results based on result set and ontology

Have the search result hit set as an output. (Micah's idea)

DBpedia like search

Amazon suggestions

CTSA related tasks like collaborator searches.

WCMC might want to be able to have preconditions for search to create configured filters on a search (CTSC members across its affiliates, Cornell affiliates, Tri-Institutional affiliates)

CTSA might want to find all grants in a subject area and get the sum of dollars of those grants.

Do we want to serve the CTSAs or do we want to focus on general search?

Maybe we should dodge this question by making a general system that other people can create a domain specific version of.

Question: are we bound to deliver a specific set of features for the NIH? We don't think that we need deliver a specific architecture.

MeSH terms, recognize when someone is searching for that (not time to do natural language processing of search terms).

ChrisB wants ability to do search term expansion (a search for pancreatitis looks for data associated with more general controlled vocab terms and finds the Dept of Nephrology)

Example of dynamically faceted search results based on result set and ontology:

Post search filter: search for "david harris", select people type, show stuff like "papers by David Harris" and "Coauthors with David Harris" (Micah and NickC)

http://impact.cals.cornell.edu an example of post search filter using VIVO data in Drupal

Nick Shows Mockups (2:34pm - 3:00pm)
Nick C and Miles W have prepared mockups and Nick showed them and explained the details. They would like to avoid leaving the local search behind.
file name is vivo_search_ui_wireframes_v0.1.pdf

The first page is a result set with left navigation (starting by classgroup, institutions, and then additional faceting below such as type, distance to, subject). There is spelling correction in the search.

Second page shown is a local search with info in the faceting area about the results on the national network so that a person can make a local search and then move to the national search.

Conclusions (3:00pm - 3:10pm)

We should work on implementing Nick and Mile's mock ups. Also maybe a reporting/info-graphic generation system.
Adding visualizations to the search results might be interesting. Take the set of individuals and pass that to a visualization?

Architecture (3:10 - 3: pm)

Consider a n-tier architecture. Head nodes with local second tier nodes could be considered an aggregated system. A head node with non-local second tier nodes could be considered a federated system.

Release planning (3:38pm - 4:30pm)

Areas from the Road Map (May, 2010):

- aggregator service
- application architecture
    - freemarker and getting rid of jsp
    - un-JSPing n3 editor
- audience analysis
- data ingest
- data review by user
- editing and display configuration
- external SPARQL endpoint
- editing improvements
    - custom editing
    - ontology editor
- file management
- inter-ontology mapping
- linking to external individuals
    - making a link to an external individual
    - big pick-list problem
- local ontology support
- national network search
- ontology change management
- ontology migration and versioning
- ontology editor
- ontology graph management
- provenance
- public/private display
- RDF and OWL representation
- reasoning scalability
- self editing
- taxonomy functionality
- website content framework
- website search
- triple store scalability

That's a lot of work items. What features do we need to do to clean up for end of project? What features do we need to do to support adoption? There is no documentation for the whole application architecture. Jim brought up the example from the mail list where someone wanted to make a new page and the long chain of tasks required to do that in VIVO right now.

What are the priorities for the Indiana people? They are interested in caching, compiling the flash source, a google map representation of the VIVO sites, integrating SciMaps into the VIVO visualizations, organizational hierarchy vis. Also top collaborators.

What to do tomorrow morning? (4:30pm -)
We should not do the SVN migration tomorrow since the release is still in progress.

We are not sure what to do about Vitro and if it should go in its own sourceforge project.

Pros to keeping vivo and vitro in the same sourceforge project:
keeps the community communication channel simpler

Cons to keeping vivo and vitro in the same sourceforge project:
if a significant community wants to build something else on vitro (not evident yet)

Plan the release features and release schedule.

Jon CR asked "What if we were to have a release in 3 months?"

Micah & visualization team: caching, bug fixes, stopping sparkline drop off, would like to do these things by the end of March, that may not be realistic.

Brian L. & ontology team: Inter-ontology mapping, taxonomy features, SDB related work like testing postgres, Maybe testing TDB for the main models.

Nick C. & UI team: not worried about the next release but would like to think further out in the project. Would like to schedule time to get feedback from users and to make changes based on that. Leaving the application in a state that could be used by adopter. Interested in proxy editing.

Brian C. & App team: Un-JSPing the n3 editing, aggregator node, RDF to XML for Solr document builder node.

Chris B. Interested in groups

Should work on a list of things to work on and priorities. This is for the end of the Aug. and for the end of the extension.