

Produce reports in VIVO

Produce reports in VIVO

back up to [User Stories: Defining features and functionality VIVO needs - September 2011](#)

User types involved

Administrators would tend to be heavy users of this feature.

Narrative User Story (for sharing/review/voting)

The Assistant Dean of Research Development at Weill Cornell shared the following as an example of the type of report he would hope to have from a system like VIVO.

Example Need

The Prostate Cancer Foundation (PCF) would like to know the institutional commitment by WCMC to prostate cancer research as they consider WCMC as an invitee to their newly initiated program project and infrastructure grant mechanism that funds 3 million over 5 years. (say similar to their [challenge awards](#))

Elements of report

- Within the last fiscal year, July 1, 2010- June 30, 2011, research in prostate cancer at WCMC has been funded by total \$ received, # of individual funded grants, from X # of sources, sortable by PI
 - Note "prostate cancer" is not contained within 1 department (medicine, pathology, pharmacology), so VIVO's ability to scour different data sets would be required.
 - Eventually research support could be benchmarked vs. other VIVO user institutions
- Export this data year by year for last 10 years- How was funding for prostate cancer research effected by the late 2000's recession? Which PIs received new (not renewals) during this time period of stringent funding- these may be our most impressive and innovative PIs?
- Report- In the last 1, 5 and 10 years- How many independent PIs at WCMC work on prostate cancer? (grants - lead PI, co-PI, and investigator names on grants, WCMC faculty author/co-author on papers, profiles, details, etc).
- Which PIs have the most collaborations based on grant support? (compare lead PI, co-PI, and investigator names on grants). For example, Mark Rubin is listed as a collaborator on grants or publications with 55 different investigators.
 - Note: this type of information would also be useful to PIs submitting grants as there are always a section on institutional resources and other places to demonstrate a strength in a particular area (by highlighting infrastructure, other PIs and PI expertise, etc).

Conclusion

With this report, he would be able to conclude the following:

Weill Medical College of Cornell University has 38 individual investigators conducting research on prostate cancer. Collectively, these investigators currently hold 22 independent grants totaling 17.8 million dollars in funding. Within the past 5 years, the efforts of WCMC faculty has resulted in 236 publications. Indicative of our expanding focus on elucidating the causes of and potential cures to prostate cancer, in each of the previous 5 years, the number of grants, total grant dollars and individual publications has increased. The research conducted by our faculty has also been highly collaborative in nature. Publications and funded grant proposals within the past year alone are highlighted by collaborations with over 220 faculty members at 65 domestic and 14 international institutions. In addition to our productivity in research, our bench to bedside approach is facilitated by the 18 Weill Cornell clinical faculty members at who currently see patients for prostate cancer and its related conditions at the NewYork-Presbyterian Hospital. Furthermore, over 50% of grants received in the past 5 years have included both a basic scientist and a clinician.

Background

The ability to run reports, in essence to do sophisticated searching, is something that could blow VIVO users away. I know it could be awesome, because there is already tool that does this, and it's name is the SPARQL Query Builder.

Wish list for improvement

The charge for this task is to make a more user-friendly version of the much ballyhooed SPARQL Query Builder. Here are some improvements:

- Don't call it a SPARQL Query Builder. Call it "Advanced Search." The text above the search might say "I want a list of..."
- Don't list namespaces in the dropdown menu.
- Hide subjects that are used sparsely or not at all in a given instance of VIVO. No dead ends! One shouldn't have to have a working knowledge of the ontology to know what kind of reporting is available – *this was much of the focus of Michael Grobe's work on the HUBzero mini-grant in 2011*
- Automatically show the (once again, used) corresponding predicates for a given subject. This would eliminate the need for the "add property" button. Numerical properties should give one the ability to limit by greater than, less than, and according to a range using, perhaps a Kayak-like slider.
- Try hard to eliminate buttons. For example, a search for faculty member who are PIs should have three buttons – thing (faculty member), property (principal investigator), and a little X to delete that criteria.
- Naming alternatives – "subject" should be "thing"; "predicate" should be "property"; "generate query" should be "Search"... if this last button exists at all. You could just run the search each time obviating the need for such a button.

- No superlong dropdowns. Try collapsing different subtypes of classes into overarching types – people, grants/agreements, publications, organizations, events, etc. Same with properties. (Site administrators should be able to easily configure what shows up in this search based on what content they have. Or, as suggested above, this could be done automatically.) Once that is selected, the user interface offers a second option to drill down further.
- Display human-readable view of properties (no domesticGeographicFocus)
- Display the results as the user constructs the search, e.g.
 - People ("You have 10,000 results")
 - People with WCMC affiliation ("You have 3,000 results")
 - People with WCMC affiliation who are associate professor ("You have 500 results")
- Ability to easily delete a row such as is done on vivosearch.org
- Sample canned searches to show off what the tool can do

*Collapse duplicate records of the same person into one. For example, this (http://vivo.med.cornell.edu/admin/sparqlquery?query=PREFIX%3A%3Chttp%3A%2F%2Fwww.w3.org%2F1999%2F02%2F22-rdf-syntax-ns%23%3E%0D%0APREFIX%3A%3Chttp%3A%2F%2Fwww.w3.org%2F2000%2F01%2Frd-schema%23%3E%0D%0APREFIX%3A%3Chttp%3A%2F%2Fwww.w3.org%2F2001%2FXMLSchema%23%3E%0D%0APREFIX%3A%3Chttp%3A%2F%2Fwww.w3.org%2F2002%2F07%2Fowl%23%3E%0D%0APREFIX%3A%3Chttp%3A%2F%2Fwww.w3.org%2F2003%2F11%2Fswrl%23%3E%0D%0APREFIX%3A%3Chttp%3A%2F%2Fwww.w3.org%2F2003%2F11%2Fswrlb%23%3E%0D%0APREFIX%3A%3Chttp%3A%2F%2Fvivo.mannlib.cornell.edu%2Fns%2Fvivo%2F0.7%23%3E%0D%0APREFIX%3A%3Chttp%3A%2F%2Fpurl.org%2Ffontology%2Fbibo%2F%3E%0D%0APREFIX%3A%3Chttp%3A%2F%2Fpurl.org%2Fdc%2Fterms%2F%3E%0D%0APREFIX%3A%3Chttp%3A%2F%2Fpurl.org%2Fnet%2Ffc4dm%2Fevent.owl%23%3E%0D%0APREFIX%3A%3Chttp%3A%2F%2Fxmlns.com%2Ffoaf%2F0.1%2F%3E%0D%0APREFIX%3A%3Chttp%3A%2F%2Ffaims.fao.org%2Ffaos%2Fgeopolitical.owl%23%3E%0D%0APREFIX%3A%3Chttp%3A%2F%2Fvivoweb.org%2Ffontology%2Fprovenance-support%23%3E%0D%0APREFIX%3A%3Chttp%3A%2F%2Fpurl.obolibrary.org%2Fobo%2F%3E%0D%0APREFIX%3A%3Chttp%3A%2F%2Fvivoweb.org%2Ffontology%2Fscientific-research%23%3E%0D%0APREFIX%3A%3Chttp%3A%2F%2Fwww.w3.org%2F2004%2F02%2Fskos%2Fcore%23%3E%0D%0APREFIX%3A%3Chttp%3A%2F%2Fvivoweb.org%2Ffontology%2Fcore%23%3E%0D%0APREFIX%3A%3Chttp%3A%2F%2Fweill.cornell.edu%2Fvivo%2Ffontology%2Fwcmc%23%3E%0D%0A%0D%0ASELECT%3A%3Chttp%3A%2F%2Fpurl.org%2Fvivo%2Ffontology%2Fwcmc%23%3E%0D%0A%3FFacultyMember1+core%3AhasPrincipalInvestigatorRole+.%0D%0A%3FPrincipalInvestigatorRole1+.%0D%0A%3FPrincipalInvestigatorRole1+rdftype%3Atype+core%3APrincipalInvestigatorRole+.%0D%0A%7D%0D%0A&resultFormat=RS_TEXT&rdftype=RDF%2FXML-ABBREV) is no good.

- Export results to CSV/Excel.
- Limiting vs. displaying by field – Some things the end user is searching for are limits (only show me this), and some things a person wants displayed but not excluded on the basis of. For example, looking back at the example above, if the user wants to see the funding for grants but the funding hasn't officially been entered into the system, s/he would still want it to be displayed. Therefore, it might make sense to tell the system to also show certain data of properties corresponding to what the main thing being searched for. Another example, if you search for a person, you might also want to see departmental affiliation, number of collaborators, email, etc. However you do it, the user interface should make a clear distinction between limiting fields and display fields, and there be the option to activate and deactivate the display fields that correspond to your search criteria.
- Sort by display field, A-Z or Z-A for text, numerically for numbered fields. Fields available for sorting should be defined either developers or the administrator. I'm not sure, for example, there is a pointing in sorting journals by page number.
- Users should be able to easily produce graph results by selected display fields. For example, say a person wants a count of the number of publications each research affiliated with the department of Anesthesiology contributed to in the year 2011. Here is how this should work with the list of results appearing throughout:
 - User selects journal articles.
 - Dropdown menu populates with options including one "published in the year"
 - When user selects that option, a slider appears, with user pinching slider to 2011
 - User clicks on plus button for new row
 - Available and relevant properties appear, one of which is "has author"
 - When that is selected, all the properties for people appear including one "has appointment"
 - When that is selected, a list of departments and other appointment-conferring organizations appear
 - User selects anesthesiology
 - User clicks on "visualize" and is given the option to choose from the display fields the x-axis (number of publications) and the y-axis (person)
- Ability to be alerted to new results from a given search via email or RSS feed
- Create a report from a user-defined group of people objects. For example:
 - If I work at a CTSC, I want to keep track of new publications by any of the different people that I did consults for.
 - If I am a librarian or statistician and I assisted in uncredited fashion on a systematic review, I also want to pay attention to people for whom I did consults.
- Canned reports - while these will not suit everyone, not having to re-invent the wheel will be helpful and it also is often easier to generate a new query based off a form.
 - VIVO should ship with developer-defined custom reports, but also...
 - Ability for system administrators to create and save institution-specific reports
 - Canned reports could be specific to certain user types such as a:
 - Faculty member
 - active grants in their department(s) or field
 - Departmental chair
 - grant activity by person in their department
 - University administrator
 - publications produced by a certain group of people in the past X years of a certain impact factor (see page 7 of [this manually generally report](#))
 - most frequent or biggest funding organizations
 - Colleague
 - System administrator
 - a group of people in VIVO who have a certain type of data missing (e.g., no email)
 - which custom classes/properties are unused
 - table of classes or properties, in order by frequency of use

- It should be possible to copy, email, and run reports over different VIVO installs. The problem here is that we don't know really know exactly what would be useful, but the ability to easily share reports queries among institutions would be a good thing.

This is a very tall order – and depends first on high-quality, consistently-defined data – for example, there are several different sets of dates on grants, and one reason traditional reporting systems tend to use data warehouses of flattened, pre-analysed data rather than more dynamic queries is that doing annualized reports on multi-year grants requires some judgement, and NSF and NIH treat grant renewals differently.

Technical considerations

- This is really hard. Best of luck.

Priority or staging considerations

For VIVO 1.5 we are focusing on more limited goals such as being able to define a menu page in VIVO that is a standing report specified by a save SPARQL query. This is a small step toward the goals described above, but it may be possible to accelerate this work by working to fold Michael Grobe's code back into the VIVO core code.

See also the issue [NIHVIVO-2397](#), "Integrate Joomla mini-grant work on SPARQL query generator," part of the "[Report generation](#)" component for VIVO 1.6