Roadmap Features

Please Comment

ⓘ

Please share your comments regarding this document. Use the comments area below, share your thoughts in listservs and in other email. Please let your voice be heard. The ideas below were collected from many emails, documents, meeting notes, presentations, and personal conversations. Please share comments, concerns, suggestions, improvements, questions. This document will be revised throughout the roadmap process. A draft roadmap should be in place in time for the conference.

Background

This document lists proposed features for inclusion in the VIVO roadmap process. Significant refinement of the items is expected by the Steering Group before discussion in the community and by the community prior to prioritization. See the Proposed Roadmap Process document for a description of the steps for transforming proposed features into an approved roadmap.

In this document, “End User,” means a person who is using VIVO for their work. This would include faculty, students, librarians, support staff, administrators, executives, agency personnel and others. It does not include developers, system administrators or data managers.

By “Stewards” we mean the people providing application support for VIVO at local sites. This includes system administrators, data managers, user support staff, trainers, and local developers. It does not include community members working on the VIVO software and its related materials.

By “Technical” we mean architects, developers, and ontologists contributing to the technical effort required to improve VIVO.

Many members of the community are end users, stewards and technical.

The remainder of this document names “features” — high level descriptions of capabilities that could enhance VIVO. Each feature might have one or more use cases. The refinement of features is part of the software design process. The roadmap process attempts to identify features which should be pursued in subsequent design and development. Clarification of features is needed in order to roughly estimate the amount of effort needed to developer the feature, and to prioritize features based on their descriptions and estimated effort.

Proposed End User Features

1. Improve the interface, particularly regarding person profiles — more attractive, more focused on scholarship
2. Provide biosketch and CV output. Extensible standard formats such as NIH, NSF, CVN, Acumen, CASRAI
3. Ensure that VIVO profiles appear on screen in under 2 seconds
4. Improve the interface for non-profile items — fewer lists, more visualization, more data summary, drill down
5. Provide one button bi-directional profile update with other VIVO systems. That is, if a profile exists in two VIVO systems representing the same person, provide a mechanism for synchronizing the two profiles
6. Provide one button bi-directional profile update with ORCID
7. Provide one button bi-directional profile update with Fedora
8. Provide one button bi-directional profile update with SciENCV
9. Provide one button upload/download of a person’s works to/from BibTex
10. Provide one button upload/download of a person’s works to/from EndNote
11. Provide one button load of meta data from Figshare
12. Improve the manual editing interfaces — drag and drop a presentation to a profile, a PDF to profile, a photo to profile, draw connections between things (people to people, people to concepts, people to works)
13. Support personal annotation of anything in VIVO
14. Support grouping of anything in VIVO — define a group, add/sub from group, show group, email group.
15. Provide mobile interface — VIVO should have optimized presentation on phone and tablet
16. Provide the Duke embeddable widgets for non VIVO websites, bring VIVO content to other university sites
17. Support social network analysis — exports to SNA tools, simple SNA visualizations and metrics
18. Support for research impact analysis. Ontology extensions, and outputs for gathering and using research impact data
19. Repair/replace/augment all visualizations — put anything with a location on a map. Put anything with dates on a timeline. Put anything with connections in a dimension free network diagram. Use standard iconography to orient user
20. Reporting improvements — provide a suite of 50-100 queries which are presented as finished reports in CSV and PDF formats. Publications by department over time, grants over time. Course, grants, papers, by faculty per unit.
21. Provide cross site search capability
22. Provide a VIVO Searchlight application
23. Provide expert finding capability, including “people like me”
24. Provide alt metrics for scholarly works in VIVO
25. Provide single sign on using a user’s ORCID and their ORCID password. Provides opportunity to host profiles for end users across institutions.

Proposed Stewardship Features

1. Simplify theming. Provide a simple theming option — logos, color, welcome text. Provide a full theming option — CSS control. Support theming improvements without having to rebuild
2. Include ingest from DOI. From a spreadsheet and manually.
3. Include ingest from PubMed. Given one or more PubMed IDs, add publication data including abstract, MeSH terms and grants cited to VIVO
4. Include ingest from ISBN. Given an ISBN, or a list of ISBN, use a standard data source to get attributes of the book in real-time and add to VIVO
5. Ingest from NIH Reporter
6. Ingest from USPTO
7. Ingest from grants.gov included
8. Ingest from clinicaltrials.gov included
9. Enforce data integrity from ontology (cardinality, domain and range). Prevent data from entering VIVO that is contradictory to the ontology

10. Provide a collection of open social plug-ins that can be augmented locally, and selected by end users at run time
11. Provide Karma scripts for ingest with guide and training for local implementation and customization
12. Use standard URIs, such as those from registry services, when possible, for shared entities rather than creating redundant local URIs. Examples include DOI, ORCID, DBpedia
13. Provide data analyst interfaces for SAS, SPSS and R. Include batch processes for exporting large amounts of data in formats ready to be used by data analysts
14. Provide standard input data sets for journals, universities, publishers, cities, dates. Possibly from WikiData. Provide a simple process for updating these data sets
15. Provide dot releases of software that do not require an upgrade
16. Provide dot releases of the ontology that do not require an upgrade
17. Load ontology from files that are automatically generated from VIVO-ISF, replacing the manually generated files that are loaded now.
18. Support technical assessment of VIVO through one button install, sample data, sample outputs, and guided tour
19. Allow better configuration of the user interface based upon annotations on the ontology
20. Add common local extensions into VIVO-ISF, reducing need for local extensions
21. Provide easier mechanism for both central and local extensions of the ontology, and easy import of modules relevant for the local VIVO instance
22. Provide latitude and longitude in ontology
23. Provide attribution ontology for indicating the role an individual played in the development of a scholarly work
24. Provide data citation capability
25. Provide software citation capability
26. Provide ontology extensions for support of humanities scholarship
27. Provide ontology extensions for support of provenance (data lineage)
28. Provide a store (catalog) of selectable third party web apps for inclusion in the interface
29. Support third party triple stores
30. Provide an ingest from SHARE Notify Harvester, populating VIVO with elements from SHARE. Some sites might populate their entire VIVO solely from SHARE Notify

Proposed Technical Features

1. Move to Maven
2. Reorganize repos so that everything needed to develop and run VIVO is available from a single repo
3. Provide software from GitHub
4. Provide one button repo to IDE capability
5. Move to continuous integration
6. Move to Assembla
7. Move to Github issues
8. Include VIVO Vagrant with distribution
9. Continue modularization work
10. Continue separation of interface (view/controller) from model. Use API to put/get data from the VIVO backend.
11. Application plug-in capability — provide a simple mechanism for a web app to appear in VIVO.
12. Improve ontology extract and load
13. Integrate ontology and software development into a single open source process
14. Adopt a standard Javascript framework for developing future interfaces. New end user and steward features to be built with the new interface framework and reliance on the API.
15. Eliminate distinction between ontologies loaded at start-up and ontologies loaded at run-time. Support only run-time
16. Restructure code to use more information from the ontologies and store less business logic in the code
17. Provide a standard set of APIs that simplify and standardize the use of VIVO data in other applications
18. Provide a means for an API developer to register a new API
19. Provide a means for a web app developer to register a new web app
20. Remove features from the application that do not work — visualizations, CSV ingest, others.
21. Rewrite the custom editing forms to improve functionality and modularity
22. Provide a binary release for developers
23. Provide Docker as part of the standard distribution
24. Provide coding standards and code training

Checklist

The following references were used to gather proposed features

1. Leadership Group notes
2. IFest Notes
3. Steering Committee notes
4. VIVO Strategic Plan
5. Workgroup notes
6. Task Force notes
7. Additional wiki pages on use cases, proposed features, release planning for future releases, others
8. Personal conversations and emails

Next Steps

The community discusses process and features in preparation for preference activity the week of July 6. See Roadmap Process for the complete process