

VIVO – Development Road Map

Development Areas

The following is a list of areas (in alphabetical order) we propose the development team address over the course of this project. These areas describe high-level sections of work for the development team and are not an attempt to identify specific features or tasks, which will come at a later stage, particularly in response to the prioritized user scenarios due at the end of May 2010.

Instead, the goal is to get an idea of what might be involved at a high-level to accomplish our goals for the end of the project in August 2011. Once we have a sense of the landscape at this larger scale, we can immediately zero in on the here and now and begin to align features with areas so that we can identify specific tasks and determine what can be delivered when throughout the course of the project.

Please see the accompanying roadmap diagram to view how these areas relate to each other on a timeline.

Aggregator Service

Developing system that collects data from multiple VIVO installations and offers a SPARQL endpoint on that aggregated data

- Explore existing tools available

depended on by: National Network Search

Application Architecture

Improving application architecture to support modular configuration, extension, and page generation

- Evaluate potential configuration and extension methods
- Evaluate potential template engines
- Evaluate development of request routing
- Refactor the way pages are built/rendered
- Separate business logic and presentation logic
- Define internal template variables
- Evaluate potential of simple customization (via browser) of default theme

depended on by: Data Ingest, Data Review by User, Editing Improvements, File Management, Ontology Graph Management, Website Content Framework, Website Search

Audience Analysis

Defining our users as segments of audiences grouped together based on similarities in behavior

- Identify tasks
- Group tasks based on behavior affinity
- Name task groups (audience segments)

depended on by: everything else, to some extent

Data Ingest

Developing tools and methods for automated integration of non-semantic data into VIVO

- Integrate external tools
- Provide examples, documentation, advice
- Produce tools and documentation for data source ontologies and intermediate ontologies
- Improve support for data integration using reasoning

Data Review by User

Providing the ability for user to confirm the validity of imported data

- Design strategy to handle rejections utilizing graph management
- Use publication acceptance/rejection as first test case

depends on: Application Architecture, Editing Improvements

Editing and Display Configuration

Preventing distortion of the ontologies to control editing or display behavior

- Develop configuration capabilities for controlling editing and display behavior
- Incorporate configuration changes into edit and display code

depends on: Editing Improvements, Website Content Framework

External SPARQL Endpoint

Provide ready access to VIVO data for external applications

- Data visibility control through graph management
- Address security and scalability implications of open access

depends on: Ontology Graph Management

Editing Improvements

Improving and documenting the features for editing in the Vitro system

- Define markup standards for form elements
- Identify and develop form submission workflow
- Explore form builder option
- Explore and analyze rich text-editor (WYSIWYG) options
- Ensure separation of semantics and presentation
- Enable bulk editing of individuals

depends on: Application Architecture, Editing and Display Configuration, Linking to External Individuals, Reasoning Scalability
depended on by: Ontology Editor, Self Editing

File Management

Providing the ability to upload and manage files

- Explore available file upload tools for integration vs. in-house development
- Represent files in the RDF model
- Develop method for managing files on disk

depends on: Application Architecture, Editing Improvements

Inter-ontology Mapping

Providing the ability to query for data using terms from other ontologies

- Hide mappings from the default ontology backend display
- Evaluate the use of reasoning to serve these queries on the fly

depends on: Local Ontology Support, Ontology Graph Management, Reasoning Scalability

Linking to External Individuals

Enabling the linking of individuals across VIVO installations

- Develop user interface that queries for external individuals
- Ensure consistent use of URIs across installations
- Explore potential trackback mechanism/notification (semantic pingback?)

depends on: Editing Improvements
may depend on: Aggregator Service, National Network Search

Local Ontology Support

Improving the ontology editor to better support extending existing ontologies and creating inter-ontology mappings

- Prevent editing of core ontology
- Visually distinguish classes and properties from different ontologies
- Provide control over which graph is being written to
- Allow editing of single ontology at a time

depends on: Ontology Change Management, Ontology Editor

National Network Search

Developing web application to search over the aggregated national data

- Explore existing tools available

depends on: Aggregator Service

Ontology Change Management

Continuously improving support for migrating data and aligning local extensions with changes to the VIVO core ontology

- Maintain mappings between local extensions and VIVO core ontology
- Evaluate need to support previous ontology versions

depends on: RDF and OWL Representation

depended on by: Local Ontology Support

Ontology Editor

Improving the ontology editor both to streamline the editing process and fully support all OWL language features

- Create forms based on established markup standards
- Improve class/property hierarchy displays
- Develop interface for editing complex class expressions

depends on: Editing Improvements

Ontology Graph Management

Supporting the use of multiple graphs to improve performance, ontology editing, and data maintenance

- Split ontologies into their own graphs
- Separate TBox and ABox
- Develop multiple graph architecture for data ingest

depends on: Application Architecture, RDF and OWL Representation

Provenance

Planning architecture to support recording information on the source of ingested and manually edited data

depends on: Ontology Graph Management

Public/Private Display

Providing users with the ability to selectively display or hide data about themselves

- Consider graph based vs. annotation approaches
- Ensure that data isn't unexpectedly exposed via inference
- Develop user interface

RDF and OWL Representation

Improving Java objects to optimize use of RDF and OWL features

- Evaluate Sesame and OWL API
- Permit use of complex class expressions throughout the application
- Evaluate potential support for blank nodes

depends on by: Ontology Editor, Ontology Change Management, Ontology Graph Management, Reasoning Scalability

Reasoning Scalability

Improving ability to reason on large knowledge bases and non-memory models

- Explore available tools
- Explore possibilities for custom reasoning implementations

Self Editing

Allowing users to login and edit information pertaining to themselves

depends on: Editing Improvements

Taxonomy Functionality

Providing ability to use existing taxonomies for tagging of instance data and for search enhancements

- Decide on local graph storage of taxonomies
- Provide access to taxonomies over the Web

depends on: Ontology Graph Management
depends on by: Website Search

Semantic Search

Improving search to take advantage of the semantic nature of the data in the model

- Explore options for results faceting
- Explore query rewriting and expansion
- Integrate taxonomies

depends on by: National Network Search, Website Search

Triple Store Scalability

Optimizing internal data access by taking advantage of RDF models that are not in-memory and taking advantage of remote SPARQL end points in an effort to reducing the number of discrete API calls

- Update DAO methods to use SPARQL where possible
- Remove reliance on lazy loading behavior

depends on: Ontology Graph Management

User Roles & Authorization

Gather requirements, design and implement support for user roles & authorization

- Explore options for user roles & authorization related to site management
- Explore options related to content and ontology editing

Website Content Framework

Providing the ability to create and customize the information architecture for a local installation of the VIVO application

- Create new browse and navigation capabilities
- Identify replacement for "tab"

- Provide support for clean URLs
- Integrate IU development for network visualization

depends on: Application Architecture
depended on by: Editing and Display Configuration

Website Search

Improving local search to incorporate semantic search features and updated interface

- Explore display of result elements, layout, and styles
- Explore faceting methods
- Explore options to export search results in standard formats (csv, xml?)

depends on: Application Architecture, Semantic Search, Taxonomy Functionality