VIVO Glossary

Term	Meaning
Adoption (of VIVO)	The process by which institutions decide that they will have and support a VIVO. The process often onvolved multiple stakeholders which may include the library, the research office, the health center, the provost's office and others. The decision involves committing resources to a project to implement VIVO, as well decisions regarding the content of the VIVO, the sources of data for VIVO, and the office that will have responsibility for maintaining the content in VIVO.
Apache Tomcat	Apache Tomcat (or Jakarta Tomcat or simply Tomcat) is an open source servlet container developed by the Apache Software Foundation (ASF). Tomcat implements the Java Servlet and the JavaServer Pages (JSP) specifications from Sun Microsystems, and provides a "pure Java" HTTP web server environment for Java code to run. http://tomcat.apache.org/
Classes	Individuals in VIVO are typed as members of one or more classes organized and displayed as a hierarchy. Child or sub-classes inherit the properties of their parent or super-classes. A child class is a specialization of a parent class – if Person is the most general class for people, then Faculty Member and Student are more specialized child classes. The class hierarchy in an ontology acts much like a taxonomy, where everything about a parent is also true for its children. Once classes have been created, they can be populated with individuals – as data on an underlying framework. A class is somewhat comparable to a noun in grammar. That is, it is most often a person, a place or a thing.
Class groups	Class groups are a VIVO-specific extension to support using VIVO as a public website as well as an ontology and content editor. Class groups are what the name implies: informal groupings of classes to organize pick lists, search results, and the VIVO Index page. Most often they mirror the top levels of the class hierarchy, with notable exceptions: Sometimes it is important to highlight one or two classes that may logically fall one place in the class hierarchy but warrant an independent category. At Cornell, for instance, courses are separated from other activities into their own class group to reflect their prominence in the academic life of the institution. Some classes are not intended to serve as public, free-standing data elements such as people, organizations, grants, or publications. Examples include Educational Background, a class to hold information about one degree on a person's list of degrees in the educational background section of their VIVO profile. Information about a single degree earned would make little sense outside of the context of the person's profile and would not be added or modified out of that context. A class that is not in a class group will not appear on the pick list for adding new individuals on the Site Admin page or on the list of classes (by class group) on the Index page. To add an individual in a class not in a class group, navigate to the class in the class hierarchy and click the Add New Individual in this Class button.
Class hierarchy	Individuals in VIVO are, however, typed (using the rdf:type property) as members of one or more classes organized and displayed as a hierarchy. Child or sub-classes inherit the properties of their parent or super-classes. A child class is a specialization of a parent class – if Person is the most general class for people, then Faculty Member and Student are more specialized child classes. The class hierarchy in an ontology acts much like a taxonomy, where everything about a parent is also true for its children. The class hierarchy provides a framework to help identify the different types of individuals modeled in a VIVO application. Class definitions are combined with specifications of the relationships or properties that may be associated with members of the class. Classes and properties together determine what statements can be created to describe individuals (either as assertions by users or imported from other data sources, or as inferences through reasoning). Classes are much more evident when defining a VIVO ontology or adding content than they are from the public view of a VIVO installation. They do appear in search results and on the Index page as the second-level faceting (or sub-types) under the broad, top-level class groups such as people, activities, events, and organizations.
Faceted browsing	In VIVO, faceted browsing refers to the ability to restrict search results to "facets" – features of the data being returned by the search. For example, One might search for "cancer" and then choose "People" to restrict the search results to people whose work is cancer-related.
Implementation (of VIVO)	The process by which institutions take the VIVO software as develivered by the project, and create a production service for their institution. Implementation is best done as a defined project with scope, resource and timeline, managed by a project manager under the direction of stakeholders. Implementation of VIVO can take several weeks to more than a year depending on the complexity and size of the institution and the support of the institution for the implementation.
Individuals and Classes	All the statements referencing the same identifier as either a subject or objects collectively describe an individual in the VIVO system. You may encounter various alternative names for individual including resource, entity, item, node, or object. Note: Individuals in RDF have no fixed structure, unlike objects in object-oriented programming languages or records in relational databases. The collecton of triples with the individual as a subject or object of the triple constitute the information available regarding the individual.

Ingesting data	The process by which data in formats external to VIVO, such as spreadsheets, tables, databases or documents, become triples stored in the VIVO triple store.
Inverse Properties	VIVO was designed to create and show object properties as pairs in opposite directions – from the subject to the object and vice-versa. These bi-directional relationships allow users to navigate from a person to a related department or grant while also supporting lists of department members on department pages or investigators on grant pages. This feels natural, is less work to maintain, and helps assure that an end user arriving at a VIVO page from a Google search can see and navigate easily to contextual information. Paired directional properties are complementary in meaning – if the Biology Department is part of its parent College of Life Sciences, the College has part Biology Department. Object properties can be specified to be uni-directional, however, if the complement or inverse would make no sense or if creating and storing the inverse would provide no value to the application.
JFact	JFact is the Java version of the FaCT++ OWL DL reasoner. JFact substitutes the Pellet reasoner since VIVO 1.8.
JIRA	JIRA is an Atlassian tool for managing tickets related to issues with a software system. The VIVO project uses JIRA to manage issues.
Karma data integration tool	Karma is an open source tool (Apache 2 License) created by a team from Information Science Institute, University of Southern California. According to their website: "Karma is an information integration tool that enables users to quickly and easily integrate data from a variety of data sources including databases, spreadsheets, delimited text files, XML, JSON, KML and Web APIs. Users integrate information by modeling it according to an ontology of their choice using a graphical user interface that automates much of the process. Karma learns to recognize the mapping of data to ontology classes and then uses the ontology to propose a model that ties together these classes. Users then interact with the system to adjust the automatically generated model. During this process, users can transform the data as needed to normalize data expressed in different formats and to restructure it. Once the model is complete, users can published the integrated data as RDF or store it in a database."
Linked Data	"In computing, **linked data** (often capitalized as **Linked Data**) is a method of publishing structured data so that it can be interlinked and become more useful throughsemantic queries. It builds upon standard Web technologies such as HTTP, RDF and URIs, but rather than using them to serve web pages for human readers, it extends them to share information in a way that can be read automatically by computers. This enables data from different sources to be connected and queried.[1] Tim Berners-Lee, director of the World Wide Web Consortium (W3C), coined the term in a 2006 design note about the Semantic Web project.[2]" – Wikipedia
Lucene	Apache Lucene is a high-performance, full-featured text search engine library written entirely in Java. It is a technology suitable for nearly any application that requires full-text search, especially cross-platform. http://lucene.apache.org/java/docs/
MeSH	Medical Subject Headings - the controlled vocabulary used by PubMed and MEDLINE
National Network	A deprecated term. During the NIH grant period for VIVO (2009-12), the collection of VIVO systems was refered to as a "national network" and VIVO was described as "Enabling a National Network of Scientists" In 2012, VIVO changed its tag line to "Connect Share Discover" – much more inclusive and indicative of the true nature of VIVO. VIVO embraces all scholarship, across the world.
ontology	An ontology defines the common terms and concepts (meaning) used to describe and represent an area of knowledge. An ontology can range in expressivity from a Taxonomy (knowledge with minimal hierarchy or a parent/childstructure), to a Thesaurus (words and synonyms), to a Conceptual Model (with more complex knowledge), to a Logical Theory (with very rich, complex, consistent, and meaningful knowledge).
open source	Open-source software (OSS) is computer software for which the source code and certain other rights normally reserved for copyright holders are provided under a software license that meets the Open Source Definition or that is in the public domain.[citation needed] This permits users to use, change, and improve the software, and to redistribute it in modified or unmodified forms. It is very often developed in a public, collaborative manner -"definition from Wikipedia" http://www.opensource.org/
Open VIVO	OpenVIVO is an open hosted VIVO that anyone can join. See http://openvivo.org To have a profile in OpenVIVO you must have an ORCiD identifier. The VIVO Project operates OpenVIVO as a technology demonstration, allowing people to experience VIVO. The project also shows new features in OpenVIVO, using it as a public laboratory for VIVO. Finally, OpenVIVO serves as a demonstration of VIVO's data sharing capability. All OpenVIVO data is available for download and reuse at http://openvivo.org/data
OWL (Web Ontology Language)	The Web Ontology Language (OWL) is a semantic markup language for publishing and sharing ontologies on the World Wide Web. Where earlier knowledge representation languages have been used to develop tools and ontologies for specific user communities (particularly in the sciences and in company-specific e-commerce applications), they were not defined to be compatible with the architecture of the World Wide Web in general, and the Semantic Web in particular. http://www.w3.org/TR/owl-features/

Pellet Reasoner	Pellet is an OWL 2 reasoner. Pellet provides standard and cutting-edge reasoning services for OWL ontologies. For applications that need to represent and reason about information using OWL, Pellet is the leading choice for systems where sound-and-complete OWL DL reasoning is essential. Pellet includes support for OWL 2 profiles including OWL 2 EL. It incorporates optimizations for nominals, conjunctive query answering, and incremental reasoning. http://clarkparsia.com/pellet In release 1.8, the Pellet reasoner was removed from VIVO because it is incompatible with the Apache 2 license.
Properties	A thesaurus extends taxonomy by providing certain standard relationships among concepts or terms – not only the broader term and narrower term relationship of the hierarchy itself but also relationships such as related term or see also. In an ontology, the class hierarchy framework can be extended with more open-ended relationships, called object and data properties. Object properties connect two individuals (a subject and object) with a predicate. Data properties connect a single subject with some form of attribute data. Data properties have defined datatypes including string, integer, date, datetime, or Boolean. Both object and data properties are often but not always defined to have a domain class, such as Person or Event, specifying the class membership of the individuals serving as subjects for each object or data property statement. Object properties also may have a range class that specifies the class membership of the individuals serving as objects of the property predicate.
Property Editing	VIVO has two property editors – one for object properties and another for data properties. Data properties are simpler to create and edit because only a domain class and a datatype need to be selected. Object properties also have a range class and additional decisions must be made about whether or not to create an inverse property. VIVO as an application is designed to create and show object properties from the subject to the object and vice-versa. Bi-directional relationships allow users to navigate from a person to a related department or grant while also supporting lists of department members on department pages or investigators on grant pages. This feels natural, is less work to maintain, and helps assure that an end user arriving at a VIVO page from a Google search can see and navigate easily to contextual information. The properties are typically directional in nature and opposite in meaning – if the Biology Department is part of its parent College of Life Sciences, the College has part Biology Department. Object properties can be uni-directional, however, and they can also be defined as symmetric. A symmetric property asserts the same relationship in both directions, with the most common example being related terms.
Property Management	Object properties connect two individuals (a subject and object) via a predicate functioning as the verb phrase in a sentence. For example, the verb phrase "is teacher of" or "teaches" connects a subject Person with the object Course in an RDF statement in the same way as the verb connects the subject and the object in the sentence, "Professor Smith teaches Economics 101." Data properties connect a single subject with some form of attribute data. While in theory every bit of information in VIVO (or any RDF database) could be modeled as a full-fledged individual member of a class, it is more convenient to store some information in simpler form using defined datatypes including string, integer, date, datetime, or Boolean. The objects of data properties are not shared, nor can they be the subjects of other statements.
Protégé	Protégé is a free open source ontology editor developed by Stanford Center for Biomedical Informatics Research at the Stanford University School of Medicine. Protégé platform provides a growing user community with a suite of tools to construct domain models and knowledge-based applications with ontologies. Protégé 4.1 has a built-in reasoner - HermiT, which will automatically classify your ontology and after that process is complete shows the inferred class hierarchy under the Entities tab.
Resource Description Framework (RDF)	The Resource Description Framework (RDF) is a framework for representing information in the Web. RDF has an abstract syntax that reflects a simple graph-based data model, and formal semantics with a rigorously defined notion of entailment providing a basis for well founded deductions in RDF data. The vocabulary is fully extensible, being based on URIs with optional fragment identifiers (URI references, or URI refs).
semantic web	The Semantic Web is an evolving development of the World Wide Web in which the meaning (semantics) of information and services on the web is defined, making it possible for the web to "understand" and satisfy the requests of people and machines to use the web content. It derives from World Wide Web Consortium director Sir Tim Berners-Lee's vision of the Web as a universal medium for data, information, and knowledge exchange. At its core, the semantic web comprises a set of design principles, collaborative working groups, and a variety of enabling technologies. Some elements of the semantic web are expressed as prospective future possibilities that are yet to be implemented or realized. Other elements of the semantic web are expressed in formal specifications. Some of these include Resource Description Framework (RDF), a variety of data interchange formats (e.g. RDF/XML, N3, Turtle, N-Triples), and notations such as RDF Schema (RDFS) and the Web Ontology Language (OWL), all of which are intended to provide a formal description of concepts, terms, and relationships within a given knowledge domain. -"definition from Wikipedia" http://semanticweb.org/wiki/Main_Page
Smushing data	Aggregating the properties of multiple individuals that are inferred to be owl:sameAs one another because they share a common value for a certain property. VIVO's smush utility rewrites all the properties of these individuals using a single URI.

SPARQL	"**SPARQL** (pronounced "sparkle", a recursive acronym for **SPARQL** Protocol and RDF Query Language) is an RDF query language, that is, a semantic query language for databases, able to retrieve and manipulate data stored in Resource Description Framework (RDF) format." – Wikipedia. SPARQL is used by VIVO to add and retrieve data.
Symmetric Properties	Object properties may also be defined as symmetric. A symmetric property asserts the same relationship in both directions, without any notion of complementary or inverse meaning. When two terms are defined as "related" there is no implication that one has precedence over the other in the relationship, whereas if one term is "derived from" another, the second term would have the different and complementary property called "has derivation."
Tomcat	(See Apache Tomcat above). Software which VIVO generally uses to serve web pages. Tomcat is part of the Apache foundation and is therefore is freely available. It most commonly uses port 8080 but can be configured for others. For example a generic VIVO installation is visible by pointing a browser to http://myvivo.edu:8080. This must be configured in the build.properties file when first installing VIVO. A VIVO site can be configured to eliminate this port suffix by adding a Context element in the Tomcat/conf/server.xml file.
Triples	VIVO is built using "triples" consisting of a subject (an individual in ontology terms, also sometimes referred to informally as an item or entity), a predicate (an object property or a data property) and an object (once again, any individual in VIVO). These "triples" are also called statements and reflect the structure of a sentence in ordinary language. Subject-predicate-object triples express the relationships among the individuals in VIVO using object properties and support attributes of individuals (e.g., first name, start date) using data properties.
Triplestore	The location for storing VIVO data. The default VIVO installation calls for a MySQL database to hold the information in VIVO but there are other alternative storage options both established and under exploration.
URI/URL	Uniform Resource Identifier. In VIVO this has the appearance of a URL since it generally begins with the domain name of the VIVO server (e.g. www.mysite.edu). Sometimes these serve as actionable links but it is more important to think of them as unique identifiers that represent something in a particular VIVO instance (a person or a department, for instance). The Cornell VIVO URI for the Laboratory of Ornithology for example is: vivo.cornell.edu/individual/individual5548 while the URL for the Lab web page is something different.
Virtual Appliance	A virtual appliance is a pre-packaged computing environment often containing an application and all its dependent parts. The approach simplifies installation at the price of performance and flexibility. VIVO virtual appliances are useful for development and for evaluation of VIVO.
Visualization	General term for techniques for presenting data to humans using graphics or other visual methods.
Vitro	Vitro is a semantic web engine and application that provides a web interface for managing ontologies and using them to represent data in any domain. VIVO uses Vitro to provide an application for representing scholarship.