Penn Medicine Biobank Informatics

OBI Influenced Software Design

Heather Williams, David Birtwell
Penn Medicine BioBank
University of Pennsylvania
Philadelphia, PA USA
hwilli@upenn.edu, birtwell@upenn.edu

***Abstract*— We present a use case of the Ontology for Biomedical Investigations [1] (OBI) informing the software design of a suite of biobanking applications. We describe how OBI has influenced the design of the Penn Medicine BioBank applications that support the collection, processing, and storage of biobank specimens and our work in creating a robust search system over data produced by BioBank applications and other sources. We show that applications that have been designed with the tenets of OBI in mind, particularly those of being reality based and modeling events as OBI style processes, have proven to effectively express richly interconnected data and be easily extendable.**

Keywords*— BFO; Biobanking; OBI; Ontology; Process; Search; Software Design

I. Introduction

Bio-specimens and the data gained by their analysis are valuable resources for bio-medical investigators. Biobanks, collections of bio-specimens (specimens) made available for research, are of extreme importance to investigators, because they can provide a large enough sample size to perform robust statistical analysis and can be used to find specimens with rare genotypes or phenotypes of interest. Information associated with specimens in biobanks and the subjects from whom the specimens were collected is frequently as important to research as the information gleaned from specimen analysis. Information technology such as databases and web application frameworks provide basic support for the storage and retrieval of biobank information. However, these technologies do not provide models for complex bio-medical data. Modeling such rich interconnected data remains a challenge for bio-medical investigators and informaticians, one that must be overcome for specimen based research to reach its full potential.

The Penn Medicine BioBank (PMBB) enables biomedical research by providing centralized access to a large number of annotated blood and tissue specimens. The Penn Medicine BioBank Informatics Team has been tasked with supporting this initiative by creating the informatics infrastructure to enable the collection, processing, and storage of specimens and associated subject data, and making the biomedical and demographic information associated with its subjects and specimens readily available to the research community. The information must be easily accessible, discoverable, and query-able, and data provenance must be maintained.

Since 2013, the PMBB informatics infrastructure has been implemented by a suite of biobanking applications collectively called Squash that are founded on OBI concepts with the aim of presenting and interacting with biobank information in a semantically rich ontology adherent manner. We designed our data model to follow patterns and conventions established by OBI, its higher order ontology Basic Formal Ontology [8] (BFO), and the OBO Relation Ontology [9]. BFO is a theory of the basic structures of reality currently being developed at the Institute for Formal Ontology and Medical Information Science (IFOMIS) at the University of Leipzig [11]. The OBO Relation Ontology provides guidelines for creating ontologies with consistent relational assertions.

To date, we have implemented a web based specimen collection and processing application named Pumpkin that makes heavy use of the concept of a process [2] and have prototyped a query system that searches over OBI annotated data. We have found that modeling events such as pre-storage specimen processes like aliquoting, centrifugation, and freezing as processes with specific end-points, inputs, and outputs, has led to a powerful application with an expressive data model that reflects reality and is easily transformable to an ontology friendly format. We also found that keeping our data model reality based following the example set by OBI has resulted in a data model over which it is easy to reason and that facilitates organic extension.

II. Methods

A. OBI Driven Software Design

Pumpkin was developed using the web application framework Grails [3] and is written in Groovy [4] and Java [5] using MySql [6] as the relational database backend. Pumpkin supports the specimen collection process from the initial creation of specimen collection packets, through the processing and ultimate storage of specimens.

Grails incorporates an object-relational mapping (ORM) powered by Hibernate [7] that provides an abstraction layer over relational databases. Instead of creating tables with fields and foreign keys, one creates inter-related domain classes that specify the database schema. Data are written and read from the database at the Groovy object level rather than via SQL. Using the rich Grails ORM, we were able to model our persistent data in a manner very similar to the way classes are defined in an ontology like OBI -- reality based with class inheritance.

When designing our data model we considered the concepts represented in OBI and the relationships between them and theorized how new concepts would be represented as a guide to designing persistent domain objects. In this way, OBI informs both the software architecture and the structure of data that is created by the application. Some examples of OBI terms that were modeled as domain objects are the concepts of protocol, specimen collection, containers, and specimens.

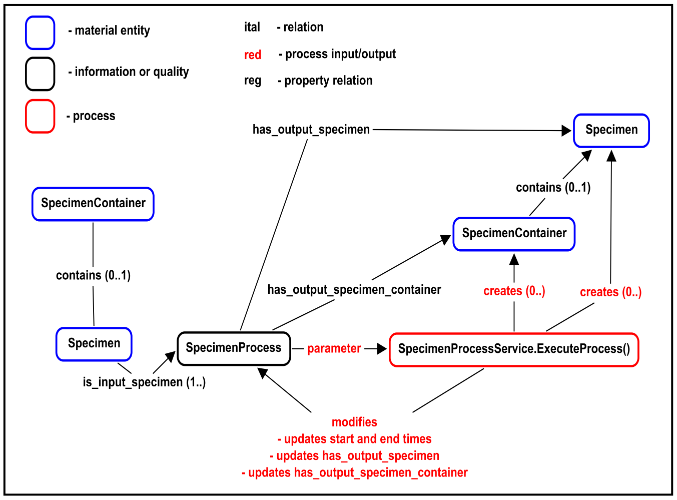

Fig. 1. The specimen process architecture expressed here in an informal graph exemplifies how OBI concepts influenced the design of Pumpkin. Specimens, SpecimenContainers, and SpecimenProcesses are all modeled as persistent domain objects.

The concept of a process heavily influenced our design. From the BFO concept of a process, we included both start and end times in our process domain classes. The OBI relationships has_specified_input and has_specified_output are implemented as well. For example, we modeled a domain super-class SpecimenProcess that includes input specimens, output specimens, start and end times, and a user (participant). Subclasses of SpecimenProcess include common specimen processes like aliquot, spin (centrifugation), dilute, and trash. Given specimen processes modeled in this way, each specimen is part of a directed specimen process graph that starts with a specimen extraction process and terminates with processes that have output specimens bound for storage.

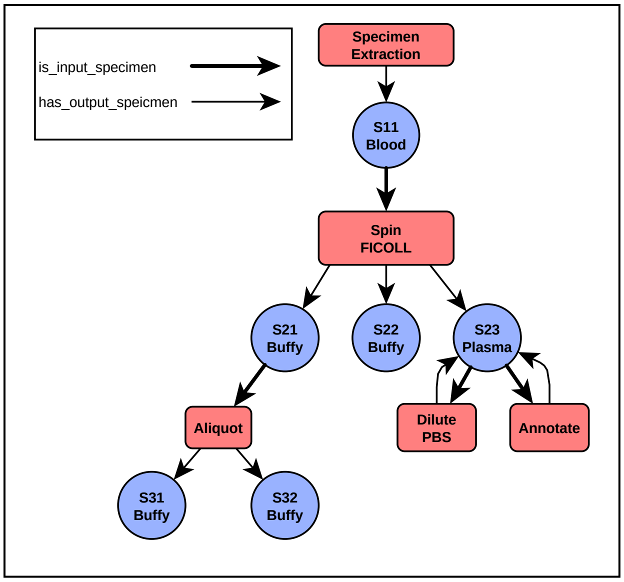

Fig. 2. An example of a typical specimen workflow showing specimen processes and their input and output specimens as modeled in Pumpkin following OBI guidelines. S11 is the primogenitor specimen. Because specimen processes are explicitly modeled, information can be directly associated with processes and specimens or inferred via the graph. For example, information pertaining to the specimen extraction of S11, like the study subject, is directly associated only with S11 and discoverable for derivative specimens by graph traversal.

B. *OBI Annotated Data Search*

We have developed a prototype search system that implements a natural language query interface over OBI annotated data. Ontology experts analyzed several small existing biomedical data sets and created a mapping between the data and concepts in the OBI ontology. D2RQ [10] was used to present these annotated data as a SPARQL endpoint.

To enable natural-language-like queries (NLQ), a pipeline following the standard programming language compilation process was created. An NLQ is first parsed as per a fully specified context free grammar. The resulting parse tree is fed to an interpreter that creates a logical query representation. A query generator takes as input this logical query representation and generates a SPARQL query that is run against the NLQ query endpoint.

III. Results

Pumpkin has been in production since June 2013 and to date has stored over 90,000 specimens from over 8,000 collections. Its design has proven to be adequate to handle our initial collection specifications and be easily extendable to additional processes and concepts, such as new specimen and collection attributes. Since the data model is reality-based and expressive as it is in graph form, it provides a common representation for all biobank related data, independent of individual lab nomenclature and idiosyncrasies. Because the data are stored in a harmonized data model, no transformation is required to query across these data.

IV. Discussion

Early in the requirements gathering and design process, it became clear that one of the primary difficulties of biobanking informatics is the heterogeneity and interconnectedness of the information involved. Application developers are mostly unaccustomed to modeling entities and processes as diverse and complex as those found in biology and biobanking. While the volume of data is small in modern terms, the complexity and fragility is great. In order to remain useful, each bit of information concerning a biological process must be richly explained, which often means complex links to other bits of information and semantic definitions. Our development team, staffed with computer science and math majors, found itself ill equipped to meet the challenge of modeling the information of a robust biobanking informatics landscape. Traditional data modeling techniques as they apply to relational and document databases fall short. It was only after several months acquainting ourselves with OBI and ontology concepts in general that we were able to see a path to an informatics system that would provide data expressivity equal to the task. What ensued was the implementation of a web based biobanking application designed from the ground-up to be OBI compliant.

Two tenets of OBI stand out as particularly significant. The first is the dedication to remaining reality based. It is often more convenient to model data for a given requirement in a way that satisfies that requirement only, usually following the path of least resistance of the implementation technology, than it is to adhere to a reality based model. From the outset, we committed ourselves to a reality based data model following the example set by OBI. While this commitment did prove difficult and seemed unnecessarily so at times, inevitably it led to an understandable and often surprisingly easily extendable data model. The second is our choice to model events as BFO style processes, occurrents with temporal boundaries, following the OBI convention of including process inputs and outputs. It was unclear at the outset that this approach would lead to an improved data model. We found however that much like our commitment to remaining reality based, modeling our processes in this way resulted in an understandable and easily extendable data model.

This approach has not been without challenges. One notable difficulty arose around efficient information retrieval from the database. To get the full data for a particular specimen, the specimen process graph must be generated, which in our initial implementation required recursive domain class traversals resulting in an explosion of computationally expensive database calls. We addressed this issue via shortcut pointers in the database. In most instances the data needed for a particular specimen are associated with either the specimen itself or its primogenitor specimen. To alleviate the computational load of traversing the process graph for common tasks, each specimen was assigned a direct pointer to its primogenitor specimen allowing single database queries rather than recursive searches.

In the hopes of finding a more general solution to efficient data retrieval, we are experimenting with mirroring our data in a graph database. Graph databases are designed to store and operate efficiently over data in graph format and may provide a mechanism to perform efficient reads of our data.

In addition to specimen processes, we have loosely modeled the concept of a ‘task’ to follow the OBI methodology as a time-based process with inputs and outputs. Currently, we model specimen intake as a task. In the future, other tasks will be included. As with specimen processes, task data will be expressed in graph form and so the same efficiency considerations will exist and methods used.

Still to be developed is Carnival, the system that will tie together the subject and specimen data generated by Squash applications with data from other sources and present them in a discoverable and query-able format. This will be an expansion of the prototype natural language query tool. The data generated by and stored in Squash applications exist at rest in a form that is compatible with OBI. We plan to annotate any additional data from sources outside Squash with OBI terms and provenance information in order to create a unified search endpoint.

Through our experiences attempting to create ontology adherent database applications, we have gained an appreciation for the valuable work that has been and continues to be done in ontology development. We suspect that the perceived value of ontologies within the biomedical research community will increase over time as those outside the immediate ontology community learn the contributions that ontologies like OBI and BFO can make towards their efforts. It remains to be seen whether ontology influenced software design will be adopted by the broader software development community, but if there is continued success of the Penn Medicine Biobank, it will be due in large part to the influence ontologies have had on our software development team.

REFERENCES

[1] Brinkman RR, Courtot M, Derom D, Fostel JM, He Y, Lord P, Malone J, Parkinson H, Peters B, Rocca-Serra P, Ruttenberg A, Sansone SA, Soldatova LN, Stoeckert CJ Jr, Turner JA, Zheng J; OBI consortium. (2010) Modeling biomedical experimental processes with OBI. J Biomed Semantics. 2010 Jun 22;1 Suppl 1:S7.PMID: 20626927

[2] Whetzel PL, Noy NF, Shah NH, Alexander PR, Nyulas C, Tudorache T, Musen MA. BioPortal: enhanced functionality via new Web services from the National Center for Biomedical Ontology to access and use ontologies in software applications. Nucleic Acids Res. 2011 Jul;39(Web Server issue):W541-5. Epub 2011 Jun 14.

[3] Grails. [https://grails.org].

[4] Groovy. [http://groovy.codehaus.org].

[5] Java. [http://www.java.com].

[6] MySQL. [http://www.mysql.com].

[7] Hibernate. [http://hibernate.org].

[8] Grenon P, Smith B, Goldberg L. Ontologies in Medicine. IOS Press; 2004. Biodynamic Ontology: Applying BFO in the Biomedical Domain. pp.20–32.

[9] Smith B, Ceusters W, Klagges B. Relations in Biomedical Ontologies. Genome Biology. 2005;6:R46. doi: 10.1186/gb-2005-6-5-r46.

[10] D2RQ [http://d2rq.org/].

[11] Grenon, P. and Smith, B. (2004) “SNAP and SPAN: Towards Dynamic Spatial Ontology”, Spatial Cognition and Computation, 4:1, 69-103.