Semantic web data warehousing for caGrid

James P McCusker[1], Joshua A Phillips[2] , Alejandra González Beltrán[3] , Anthony Finkelstein[3] Michael Krauthammer*[1]

[1]Department of Pathology, Yale University, New Haven, CT, US
[2]Semantic Bits, LLC, Reston, VA, US
[3]Department of Computer Science, University College London, London, UK

Email: James P McCusker - james.mccusker@yale.edu; Joshua A Phillips - joshua.phillips@semanticbits.com; Alejandra González Beltrán - a.gonzalezbeltran@cs.ucl.ac.uk; Anthony Finkelstein - a.finkelstein@cs.ucl.ac.uk; Michael Krauthammer*- michael.krauthammer@yale.edu;

*Corresponding author

Abstract

The National Cancer Institute (NCI) is developing caGrid as a means for sharing cancer-related data and services. As more data sets become available on caGrid, we need effective ways of accessing and integrating this information. Although the data models exposed on caGrid are semantically well-annotated, it is currently up to the caGrid client to infer relationships between the different models and their classes. In this paper, we present a Semantic Web-based data warehouse (Corvus) for creating relationships among caGrid models. This is accomplished through the transformation of semantically-annotated caBIG Unified Modeling Language (UML) information models into Web Ontology Language (OWL) ontologies that preserve those semantics. We demonstrate the validity of the approach by Semantic Extraction, Transformation and Loading (SETL) of data from two caGrid data sources, caTissue and caArray, as well as alignment and query of those sources in Corvus. We argue that semantic integration is necessary for integration of data from distributed web services and that Corvus is a useful way of accomplishing this. Our approach is generalizable and of broad utility to researchers facing similar integration challenges.

Introduction and Background

caGrid, a core technology of caBIG (Cancer Biomedical Informatics Grid) [1–5], is a semantically annotated grid sponsored by the National Cancer Institute that provides a consistent framework for grid web service annotation. There are three annotation sources. First, UML information models are registered to a central metadata repository, known as the Cancer Data Standards Repository (caDSR) [6], and harmonized with other registered models. Harmonization is achieved through data element reuse, which also enhances syntactic interoperability, where field names are compatible between data models. Data elements that are reused across information models are known as Common Data Elements (CDEs). Second, semantic interoperability is achieved by annotating CDEs and the UML classes that use them with concepts from the NCI Thesaurus (NCIt) [7–10], which has become a rich, cancer-focused ontology. Finally, W3C Schema definitions that define the XML encodings of the UML information models are registered to the Global Model Exchange (GME), and UML-to-XML mappings are registered to the the caDSR. Web Services Resource Framework (WSRF) grid services then publish metadata on their models, drawing from all three sources [2, 3, 11, 12].

Clients access caGrid to retrieve data from diverse services such as *omic* [13] stores (caArray) or tissue repositories (caTissue). From the client perspective, there is no transparent mapping of semantics onto data from grid services. When a caGrid client wants to join data from one service to another, or attempt to make claims about a particular datum being equivalent to another, it must inspect the metadata to determine if, and how, data from two services are interoperable. A naive client that is unaware of the service metadata will be unable to make that mapping. In other words, semantic interoperability is the job of the client and requires the ability to reason over (or interpret) the metadata: class hierarchies, attributes, associations and their corresponding annotations to establish equivalencies.

Fortunately, there is already a technology that is available that can perform exactly those tasks. The Web Ontology Language (OWL) [14] is a formal way of describing relationships among concepts and any data defined in the Resource Description Framework (RDF) [15, 16] that adheres to those standards. OWL is relevant here because it provides for class hierarchies, properties, and equivalencies. It also provides a means for multiple ontologies to coexist and for mappings to be defined between them. A client that can take advantage of these capabilities would have the ability to automatically map between data models on the grid. A client would import data from multiple grid services through queries and map that data onto ontologies derived from the published service metadata. It could then be joined within the Semantic Web environment to allow a much larger set of queries to be realized.

We propose a Semantic Web data warehouse system that enables users to map data from multiple caGrid data sources into an ontologically-driven data store, or knowledge base (KB). We are currently developing a Semantic Web data warehouse framework, which we call Corvus. Semantic Web data warehousing allows users to define which data sources they are interested in and automates the extraction, transformation, and loading process (ETL) through Semantic ETL (SETL) [17] across entire classes of data sources, such as sources published on caGrid. Semantic Web data warehouses are dynamic data stores which, as we will show, can model and store data from diverse grid services on the fly. Users will be able to query the grid in novel ways using the data warehouse as a proxy and will be able to dynamically integrate new data sources as needed.

Transforming UML to OWL

In order to perform semantic-web-based SETL on caGrid services, it is imperative to understand how to map UML constructs and their NCIt annotations from caGrid onto semantic-web constructs in OWL.
UML is the *de facto* standard for object-oriented visual modeling and has no formal semantics. Its main constructs are classes, attributes, associations, and generalizations. A UML class denotes a type for a set of objects with common characteristics, indicated as attributes. An association is a relation between classes and a generalization relates a parent class with a child class.
On the other hand, OWL is a knowledge modeling language with a formal semantics based on Description Logics (DLs) [18]. Its main constructs are classes, datatype properties, and object properties. An OWL class denotes a set of individuals (or instances). Properties are standalone entities, establishing relationships between individuals (object properties) or between individuals and data values (datatype properties) [14].
Although UML and OWL have similar constructs, they have significant differences. Mainly, UML follows a Closed World Assumption (CWA) but OWL an Open World Assumption (OWA). In CWA, lack of information means negative information. In OWA, lack of information means lack of knowledge.
Previous work has compared and contrasted UML and OWL and provided transformations between the two [19–25]. These transformations were motivated by different applications and specified in varying levels of detail. For example, Berardi *et al* [19] provided an incomplete transformation from UML class diagrams to DLs and analyzed the complexity of the reasoning to detect inconsistencies in the model. Evermann [23] described an exhaustive conversion to make a well-known ontology, specified in natural language, available in more formal representations.

To the best of our knowledge, semCDI [24, 25] is the only work providing an annotated-UML-to-OWL transformation based on the caGrid infrastructure. semCDI, as all the previous approaches, maps UML classes to OWL classes, UML attributes to datatype properties, and associations to object properties. "UMLClass", for this paper, refers to the OWL class that represents the original UML Class. semCDI extends this basic mapping with NCIt, as caGrid UML models are annotated with concepts from NCIt. In semCDI a NCIt concept is represented with an OWL class. A UMLClass is a subclass of its annotated concept. A weakness of semCDI is that it does not consider concepts associated with attributes nor other UML semantics such as multiplicities of associations or disjoint declarations. Additionally, concepts from NCIt are considered in isolation, without including all the relevant knowledge accessible from NCIt.
If UML attributes are modeled as datatype properties, as in semCDI, their annotated concepts can only be included as OWL annotation properties. OWL annotation properties are used to represent metadata on OWL constructs and are not considered for reasoning purposes. Moreover, using subsumption to represent the concept/UML class relationship results in an inconsistent ontology, which does not preserve NCIt semantics. This is due to UML classes annotations corresponding to concepts that are stated as disjoint in NCIt.
Considering the issues presented above, we have designed a different annotated-UML-to-OWL transformation that does not model attributes as datatype properties nor concept/UML class relationships as subsumption. Our transformation, described in detail under the Methods section, follows a general, modular approach for ontology development, can include annotations for all UML constructs, and preserves NCIt semantics.

Methods overview

In order to assess the feasibility of Semantic Web-based SETL on caGrid services, we first identified a real-world use case involving caGrid that included the need for semantically merging disparate information models. Specifically, we identified the need to join data from caTissue and caArray, two caGrid services exposing tissue and micro-array data, respectively. The use case involves the need to link a microarray experiment with clinical annotations linked to the specimen from which the experiment was derived. Imagine a situation where a specimen S is stored in caTissue and a microarray result M (derived from specimen S) is stored in caArray. In caGrid, it is possible to query a single service using the caGrid Query Language (CQL). Assume that we get results (data) from caTissue on S, which we call R_s. Equally, we get results from querying caArray, which we call R_m. As discussed above, the linking of R_s to R_m is not

trivial. There is a need to identify the classes and attributes in both the caTissue and caArray models that align and the constraints under which two instances from the two models can be linked together.
In this paper, we demonstrate that this can be elegantly accomplished using Semantic Web technology. We first set up an instance of caTissue and caArray on the caBIG training grid. We then loaded specimen information of a particular set of cell lines called NCI-60 into caTissue. NCI-60 is a collection of cancer cell strains for which there exists a multitude of micro-array experiments on those cell strains (gene expression or copy number experiments). We recorded the disease class of each of those specimens in the caTissue instance. We then loaded a NCI-60 gene expression set into our caArray instance. The quest was to link the caTissue and caArray datasets. We will present how we use SETL to perform this linking.

Results and Discussion

We were able to successfully load data from caTissue and caArray into a Semantic Web data repository (KB) and link caArray Sources with caTissue Specimens, allowing us to use clinical data from caTissue to enhance the analysis of gene expression data from caArray. Extraction was performed by querying caArray and caTissue caGrid services for all information regarding a specific Collection Protocol (caTissue) and Experiment (caArray), especially the Sources of the Experiment and the Cell Specimens of the Collection Protocol. The results of those queries were then transformed into OWL individuals that conformed to the corresponding OWL classes in OWL ontologies that were generated for caArray and caTissue. The data and ontologies were then loaded into a KB with a custom inferencing rule that inferred the link between Sources and Cell Specimens based on equivalence of the "CellLine" annotation from caArray and the "Label" field in caTissue. The significance of this join is important: caArray and caTissue, while based on the same framework (caGrid), share no application code and have been developed by completely separate teams led by different leadership groups. Also, while the data itself is related, in that it is about the same biological entities, it too was derived from completely separate sources at different institutions. Curation was limited to ensuring the use of the common designations for the cell lines in NCI-60 that are set forth by the NCI. The resulting ontologies, semantic data (in N-Triples format), and inferencing rules are available as additional files.
This result shows that it is possible to load related data from unrelated sources and link them in a generalizable fashion using rule-based logics. Performance of the repository is reasonable. Load times are shown in Table 1. The time taken to develop the integration rule was negligible: after learning the rule language, it took a knowledgeable software developer approximately 120 minutes to look up the needed

links, write, test, and validate the rule. Learning the rules language took an additional 60 minutes. The most time-consuming step was writing the rule, as extraction and transformation took far less time, and in the general case would be scheduled automatically. Figure 1 shows the Principal Components Analysis (PCA) Projection of 16 different clinical diagnoses that were extracted from the KB and joined to expression data of hybridizations from GEO GSE5949 [26], demonstrating the successful link between caTissue and caArray. The diagnoses that are shown are far more specific than the usual "cancer type" that is available in the GEO data set. Using this technique, other statistical analyses can be performed on annotations such as survival, gender, age at diagnosis, tissue site, or any number of clinical annotations that caTissue can be customized to contain. Please see the additional files for the PCA results and the diagnosis mapping.

SETL is a significant improvement over conventional ETL. An added benefit to this technique is that it is not limited to a single data source. These rules apply to any caArray and caTissue data source and with further refinement can be applied to more generic data sources that have been annotated in a similar way to these sources. Also, Semantic Web data warehousing allows a much cleaner separation of data into appropriate services. While caArray can store some of this information, it should not be considered a canonical source for information about biospecimens, only the information that is known at the time that the data is published and recorded, while caTissue may continue to include more information (such as more specific diagnoses) about the biospecimens and sources as they are included. This is an important point, as it allows researchers to learn new information about existing experiments as that information becomes available, rather than relying on what may be uncorrected or incomplete information.

Future Work

Corvus is, at this time, still a prototype system with components that serve as proof-of-concept. We plan on expanding its ability in the future to allow for automated SETL and linkage of scientific data. Work also needs to be done on providing visualizations and other meaningful user interfaces into the data that is accumulated. Currently, the generated ontologies do not consider the caDSR semantics: they only include the relationship to NCIt concepts without distinguishing between primary concept and qualifiers. This will be incorporated into the ontology generator. Also, the queries are complicated by the fact that an extra level of indirection is currently used to represent UML attributes: all attributes map onto a separate individual, rather than directly onto a datatype property. While this allows inferencing over these attribute types, it makes the final data more complicated to understand. This may be addressed through a simplified

interface over the raw KB. Also, many web services that contain useful biomedical data are not semantically annotated. NCBI's Entrez suite of databases is a prominent example. A method of generating ontologies and transformation services for these services will allow an informaticist to include such web services in a Semantic Web data warehouse. Some web service standards have already been implemented, such as Semantic Annotations for WSDL (SAWSDL) [27] to allow more generic semantic annotation of web services, and services that provide those annotations can be integrated through additional integration rules.

Conclusions

SETL is a valid technique for gathering information from semantically annotated grid services and provides opportunities for integration of data that was not necessarily designed to be integrated. This allows for dynamic analysis of many different data types and makes it possible for informaticists to continually integrate relevant new data sources as they become available with far less effort than would be needed in a traditional data warehousing environment. In turn, Corvus, along with the caGrid security and semantic annotation infrastructure, allows for integration of data across institutions as well as across applications as long as those institutions use the same semantic metadata. This has large implications for the possibility of increased collaboration in biomedical research.

Methods

At the core of Corvus is a Semantic Web-based data warehouse based on BigOWLIM, a state-of-the-art semantic store technology, within which we assemble our KB by integrating various caBIG data sets. A key feature of our approach involves using OWL ontologies that have been generated from semantically annotated caGrid UML information models. Components of the Corvus framework support a SETL workflow that pulls data from public caGrid data services and translates that data into RDF/OWL that conforms to the OWL ontologies generated, and then stores that information, along with the related ontologies, in a Semantic Web KB. Because of this, it is possible to dynamically combine both caBIG and non-caBIG data sets while preserving semantic annotation of the caBIG information models and simultaneously enable the use of Semantic Web technologies such as SPARQL (SPARQL Protocol and RDF Query Language), Semantic Web Rule Language (SWRL), and DL reasoning services.
SETL in Corvus consists of the following steps: generation of OWL ontologies from caGrid information models for caTissue and caArray and loading them into the KB; submitting one or more queries to caGrid data services to extract the clinical data about NCI-60 from caTissue and gene expression data based on

GEO GSE-5949 from caArray; and transforming that data into RDF triples, which are then loaded into the KB. As the data are loaded into the KB, custom rules are used to infer relationships between the data from the two sources. Finally, the data is queried to produce a set of classes for the hybridizations of GSE5949, which are used to label a plot of the first two principal components of the expression data as an example visualization of the combined datasets.

Data Preparation

We use two caBIG database applications, caTissue and caArray, to demonstrate the ability to link related information from independent databases. caTissue is a biospecimen banking and management tool developed through the NCI for use in research tissue banks. It is able to store information about biospecimens and the individuals they originated from, especially clinical information. caArray is a microarray management tool developed through the NCI and is an Micrarray Gene Expression Database (MGED)-compliant array repository. Both caTissue and caArray can publish caGrid services that expose their data.

caTissue and caArray instances were deployed with caGrid services that published to the caGrid training grid. Expression data was curated from GEO GSE5949 [26] by downloading the data and converting them into the MicroArray and Gene Expression (MAGE)-TAB format using the GEOImport and TabConverter tools from the tab2mage project. Additional curation was needed to fix some references to array designs and to ensure that all *Characteristics[CellLine]* entries were valid and entered. The data were then uploaded to caArray [28]. Data on the cell lines, such as specific clinical diagnosis, were collected from the NCI SKY/M-FISH & CGH Database [29] and curated into a caTissue instance [30]. Every effort was made to model the understood relationships among the cell lines, including discoveries of derivation or sourcing from the same patient. The clinical diagnosis is from the SKY/M-FISH & CHG Database. The clinical diagnosis in caTissue is the most specific diagnosis in caTissue that matches the diagnosis from the source database.

SETL Process

The Corvus SETL process (Figure 2) is designed to enable new models and data to be dynamically integrated into the data warehouse. It is composed of the Ontology Generator and Data Extractor (Extraction), Transformer (Transformation), and finally a KB loader that loads the data into the data warehouse (Loading).

The Ontology Generator generates OWL ontologies from published caGrid data service UML information models. These ontologies express the UML information model, semantic annotations on those models, and the relevant parts of the NCIt. We generated ontologies from the caArray 2.1 and caTissue Suite 1.1 models. These ontologies are then loaded into the Corvus data warehouse. In this case we extract the UML models for caTissue Suite and caArray.

The Data Extractor handles CQL queries of objects and the relationships between those objects. For example, we query caTissue for a CollectionProtocol object. Here, the path information indicates how the associated CellSpecimen objects should be included in the resulting object graph. The DataExtractor uses the CQL and path information to pull XML data from caGrid data services.

The ETL Process then passes the XML data to a Transformer Service instance that provides an XML to OWL transformation. The resulting OWL instance data are then loaded into the Corvus data warehouse.

Ontology Generator

We implemented an OWL generation component that extracts metadata from caDSR projects and caGrid data services. It implements the automatic transformation process from grid services metadata (annotated UML models) into OWL ontologies as described below. We take a modular ontology development approach, as depicted in Figure 3.

It is necessary to model two aspects of caGrid metadata to be able to, in turn, model caGrid data: the domain model, such as relationships among entities, their attributes, and other entities; and the relationships between entities and their semantic meaning as registered in the NCIt.

The Semantic Metadata ontology models the NCIt annotations of UML classes and attributes using the property *semanticMetadataCollection*. Every class and attribute in a caGrid UML model must be annotated with concepts in the NCIt. However, NCIt is extremely large and covers the entire biomedical domain. We use the methodology in [31] to extract relevant subsets from NCIt. This methodology has the following properties [31]: a) it preserves NCIt semantics; b) it includes everything that is relevant to the particular information model ontology; and c) it imports only what is relevant. To extract the NCIt subset for a particular data model, we build a signature including all the annotations used in the model.

The Domain Model ontology represents UML semantics. UML class attributes are attached to classes using the *umlAttributeCollection* property. Relationships among entities are modeled using sub-properties of *umlAssociation*. The extracted NCIt subset is imported to the standardized transformation of UML to OWL. As an example, Figure 4 depicts part of the caArray 2.1 ontology. UML classes are associated with

NCIt concepts via the property *semanticMetadataCollection*, such as *Hybridization* and its concept *nci:Nucleic_Acid_Hybridization* in the example. UML attributes are individuals that are linked via the property *umlAttributeCollection* and have NCIt annotations, but in turn have a special property *dm:datatype* that contains the value of that attribute for the individual it is linked to. UML classes and UML attributes are defined as subclasses of *dm:UMLClass* and *dm:UMLAttribute*, respectively.
UML class hierarchies are represented with the *rdfs:subClassOf* construct. UML associations (*a has_a b*) are modeled as *rdfs:subPropertyOf umlAssociation*. Cardinality restrictions represent multiplicities of associations. If an association is bidirectional, an inverse property is defined. UML classes and attributes are defined to be disjoint unless they have a relationship in the class or property hierarchy. Since the DL expressiveness of the resulting ontologies is $\mathcal{SHIQ}(D)$ [18], it is possible to use existing OWL-DL reasoners with them.
The generated ontologies provide an integrated view and formal representation of the caGrid data services' metadata. As shown below, these ontologies can be extended to consider instance data, providing the semantic framework for data integration.

Data Extractor

For this effort, we queried caArray and caTissue. The Data Extractor component works with most caGrid data services that have been generated from the caCORE Software Development Kit (SDK). It relies on knowledge of caCORE conventions for naming of object identifiers (i.e. primary keys) and XML-UML mapping rules. In the future, the Data Extractor will pull metadata about XML-UML mapping rules and identifiers directly from caDSR.
Since the current version of CQL does not support projections and the XML results returned by most caBIG data sources do not contain foreign key values, we cannot avoid what is know as the "$n + 1$ select" problem [32], in which we must execute one query to retrieve an initial data set and then n additional queries to retrieve information associated to each item in the initial data set, where n is the size of the initial data set. Some data services, such as caArray, partially address this problem by automatically including information about associated objects in the XML results document. For example, the result of a query for Experiments includes information about all Hybridization, Sample, Source, Extract, and other objects that are associated with each Experiment. The next version of CQL will allow the query to indicate what associated objects should be included in the results [33]. It took about 20 minutes to extract the data we need from a caArray experiment containing 300 samples. Figure 5 depicts the paths that were traversed.

Transformer Services

We expose an XML to RDF/OWL Transformation service to convert from the caGrid XML to RDF that conforms to the ontologies generated by the ontology generator. The Transformer Service provides a framework for exposing XML-to-XML transformations as stateful grid services. These services advertise what kinds of transformations they support and therefore enable clients to dynamically discover available transformations. This will be necessary in the future as we intend to dynamically integrate both caBIG and non-caBIG data sources. We have provided a general-purpose Transformer implementation that will transform XML from caCORE SDK generated data services by using caCORE SDK UML-to-XML conventions. In the future, we will enhance this implementation to pull UML-to-XML mapping metadata from caDSR.

Loading Data from caTissue and caArray

The output from the transformation service was then loaded into Corvus. Corvus supports a number of triple stores, but in this case we used BigOWLIM. We had two sets of transformed data: data from caTissue and data from caArray. To link the two data sets, we made use of the caTissue and caArray data models stored in Corvus and wrote a rule to link the Source (biological source) object in caArray to the CellSpecimen object in caTissue based on the Source's cell line name. Inferencing was done using custom rules implemented in the Ontotext's TRREE language, used by BigOWLIM. The rule added (Figure 6) finds all caArray Sources where they have a term-based characteristic (of which "CellLine" is one) that matches the Label of a caTissue CellSpecimen. If there is a match, a property *corvus:derived_from* is added to the Source and has the value of the *catissue:CellSpecimen*. The property *corvus:derived_by* is the inverse of *corvus:derived_from*. This pattern can be used to link data of different types across databases and can possibly be generalized using axioms of the classes and properties to be compared.

Data Queries and Analysis

Using the query in Figure 7, the caTissue clinical diagnosis is extracted from the database and paired with the name of the caArray Hybridization it corresponds to. This makes it possible to display those diagnoses in any analyses that are made. Also available, but not extracted, are: gender, age at diagnosis, ethnicity/race, or any other clinical annotations that are added to a caTissue Suite repository. caTissue Suite has the added capability of "Dynamic Extensions", which allow the application to be extended dynamically in a way that keeps it compliant with caGrid. A Principal Components Analysis is made of

the expression data and the projection is colored with the diagnoses extracted.

Competing Interests

The authors declare that they have no competing interests.

Authors' contributions

JPM curated the experimental data and developed the rules for integrating caArray and caTissue data. He also performed the PCA analysis and wrote the query to extract clinical diagnoses. He wrote the Abstract, Background (except for Transforming UML to OWL), Results and Discussion, Conclusions, Data Preparation, Loading Data from caTissue and caArray, and Data Queries and Analysis sections. JP developed the extraction and transformation components and wrote the relevant parts of the Methods section, including the overall process and the Data Extractor and Transformer Services sub sections. AGB analyzed previous UML-to-OWL transformations, designed a general approach to transform annotated-UML-to-OWL preserving the domain ontology semantics, developed the OWL generation component. AGB and AF wrote the sections Transforming UML to OWL and Ontology Generator. MK provided vision, scope, and needs analysis, and provided significant edits to most sections. All authors participated in revision and have read and approved the manuscript.

Acknowledgments

James McCusker's and Michael Krauthammer's work was funded in part by the Yale SPORE in Skin Cancer.

Joshua Phillips' work was funded in part by the caBIG Architecture Workspace.

Alejandra González Beltrán and Anthony Finkelstein are grateful to Cancer Research UK and the UK National Cancer Research Institute Informatics Initiative for support for their research.

References

1. Buetow KH: **Cyberinfrastructure: Empowering a"Third Way"in Biomedical Research**. *Science* 2005, **308**(5723):821–824.
2. Saltz J, Oster S, Hastings S, Langella S, Kurc T, Sanchez W, Kher M, Manisundaram A, Shanbhag K, Covitz P: **caGrid: design and implementation of the core architecture of the cancer biomedical informatics grid**. *Bioinformatics* 2006, **22**(15):1910.
3. Oster S, Langella S, Hastings S, Ervin D, Madduri R, Kurc T, Siebenlist F, Foster I, Shanbhag K, Covitz P: **caGrid 1.0: A grid enterprise architecture for cancer research**. In *AMIA Annual Symposium* 2007.

4. Langella SA, Oster S, Hastings S, Siebenlist F, Phillips J, Ervin D, Permar J, Kurc T, Saltz J: **The Cancer Biomedical Informatics Grid (caBIG) Security Infrastructure.** In *AMIA Annu Symp Proc*, *Volume 433* 2007:7.
5. Langella S, Hastings S, Oster S, Pan T, Sharma A, Permar J, Ervin D, Cambazoglu BB, Kurc T, Saltz J: **Sharing data and analytical resources securely in a biomedical research grid environment**. *Journal of the American Medical Informatics Association* 2008, **15**(3):363–373.
6. Warzel DB, Andonyadis C, McCurry B, Chilukuri R, Ishmukhamedov S, Covitz P: **Common data element (CDE) management and deployment in clinical trials**. In *AMIA... Annual Symposium proceedings [electronic resource]*, *Volume 2003*, American Medical Informatics Association 2003:1048.
7. Hartel FW, de Coronado S, Dionne R, Fragoso G, Golbeck J: **Modeling a description logic vocabulary for cancer research**. *Journal of Biomedical Informatics* 2005, **38**(2):114–129.
8. Sioutos N, Coronado S, Haber MW, Hartel FW, Shaiu WL, Wright LW: **NCI Thesaurus: A semantic model integrating cancer-related clinical and molecular information**. *Journal of biomedical informatics* 2007, **40**:30–43.
9. de Coronado S, Haber MW, Sioutos N, Tuttle MS, Wright LW: **NCI Thesaurus: using science-based terminology to integrate cancer research results**. *Medinfo* 2004, **11**(Pt 1):33–37.
10. Fragoso G, de Coronado S, Haber M, Hartel F, Wright L: **Overview and utilization of the NCI Thesaurus**. *Comparative and Functional Genomics* 2004, **5**(8).
11. Covitz PA, Hartel F, Schaefer C, Coronado SD, Fragoso G, Sahni H, Gustafson S, Buetow KH: **caCORE: A common infrastructure for cancer informatics**. *Bioinformatics* 2003, **19**(18):2404–2412.
12. Komatsoulis GA, Warzel DB, Hartel FW, Shanbhag K, Chilukuri R, Fragoso G, Coronado S, Reeves DM, Hadfield JB, Ludet C: **caCORE version 3: Implementation of a model driven, service-oriented architecture for semantic interoperability**. *Journal of biomedical informatics* 2008, **41**:106–123.
13. Ge H, Walhout AJM, Vidal M: **Integrating 'omic' information: a bridge between genomics and systems biology**. *Trends in Genetics: TIG* 2003, **19**(10):551–60, [http://www.ncbi.nlm.nih.gov/pubmed/14550629]. [PMID: 14550629].
14. McGuinness DL, Harmelen FV: **OWL web ontology language overview**. *W3C recommendation* 2004, **10**:2004–03.
15. Miller EJ: **An introduction to the resource description framework**. *Journal of Library Administration* 2001, **34**(3):245–255.
16. Klyne G, Carroll JJ, McBride B: **Resource description framework (RDF): Concepts and abstract syntax**. *W3C recommendation* 2004, **10**.
17. Spies M: **An ontology modelling perspective on business reporting**. *Information Systems* 2009.
18. Baader F, Calvanese D, McGuinness DL, Nardi D, Patel-Schneider PF (Eds): *The Description Logic Handbook*. Cambridge University Press 2003.
19. Berardi D, Calvanese D, De Giacomo G: **Reasoning on UML Class Diagrams**. *Artificial Intelligence* 2005, **168**(1-2):70–118.
20. Gašević D, Djurić D, Deved V: **MDA-based Automatic OWL Ontology Development**. *International Journal on Software Tools for Technology Transfer (STTT)* 2007, **9**(2):103–117.
21. IBM: **Ontology Definition Metamodel - OMG Adopted Specification** 2007, [http://www.omg.org/cgi-bin/apps/doc?ptc/07-09-09.pdf]. [Accessed October 2008.].
22. Knublauch H: **UMLBackend: plug-in for Protégé** [http://protege.cim3.net/cgi-bin/wiki.pl?UMLBackend]. [Accessed April 2009.].
23. Evermann J: **A UML and OWL description of Bunge's upper-level ontology model**. *Software and Systems Modeling* 2008, :1619–1366.
24. Shironoshita EP, Jean-Mary YR, Bradley R, Kabuka MR: **semCDI: Semantic Query Formulation for caBIG**. *Journal of the American Medical Informatics Association (JAMIA)* 2008, **15**(4):559–568.

25. Shironoshita EP, Bradley RM, Jean-Mary YR, Taylor TJ, Ryan MT, Kabuka MR: **Semantic Representation and Querying of caBIG Data Services**. In *Proceedings of the 5th International Workshop on Data Integration in the Life Sciences (DILS'08)*, *Volume 5109 of* Lecture Notes in Bioinformatics. Edited by Bairoch A, Cohen-Boulakia S, Froidevaux C, Springer 2008:108–115.

26. Shankavaram U, Weinstein J, Kahn A: **Comparison between cell lines from 9 different cancer tissue (NCI-60) (U95 platform)** 2006, [http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE5949].

27. Kopecky J, Vitvar T, Bournez C, Farrell J: **SAWSDL: Semantic Annotations for WSDL and XML Schema**. *Internet Computing, IEEE* 2007, **11**(6):60–67.

28. **caArray - Experiment Details - E-GEOD-5949** [http:/espresso.med.yale.edu:38080/caarray/project/shank-00006].

29. **SKY/M-FISH/CGH Database** [http://www.ncbi.nlm.nih.gov/sky/skyweb.cgi?submitter=NCI60+cell+line+panel_Genetics+Branch_I.R.Kirsch&form_type=display_cases].

30. **caTissue Suite caGrid Service Endpoint** [http:/espresso.med.yale.edu:18080/wsrf/services/cagrid/CaTissueSuite].

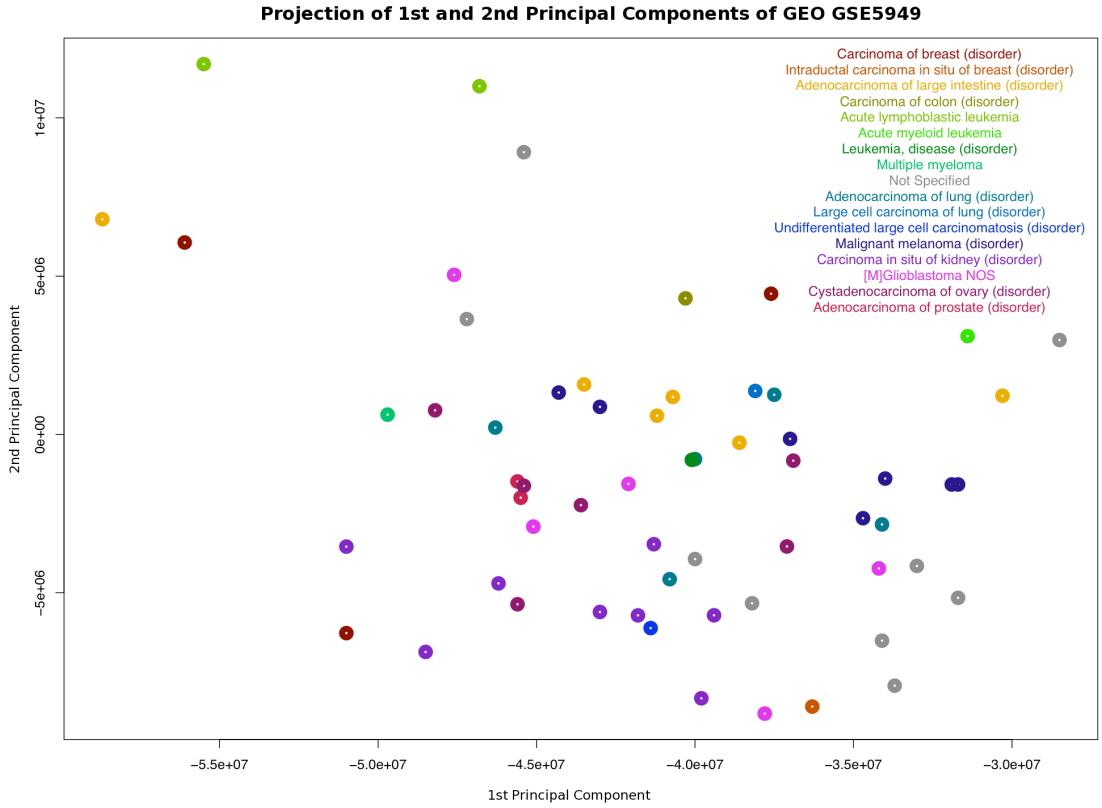
31. Jiménez-Ruiz E, Grau BC, Sattler U, Schneider T, Llavori RB: **Safe and Economic Re-Use of Ontologies: A Logic-Based Methodology and Tool Support**. In *Proceedings of the European Semantic Web Conference*, *Volume 5021 of* LNCS. Edited by Bechhofer S 2008:185–199, [\url{http://dx.doi.org/10.1007/978-3-540-68234-9_16}].

32. **SQL n + 1 Selects Explained - Pramatr Blog** [http://pramatr.com/2009/02/05/sql-n-1-selects-explained].

33. **CQL 2 - Data Services - cagrid.org** [http://carid.org/display/dataservices/CQL+2].

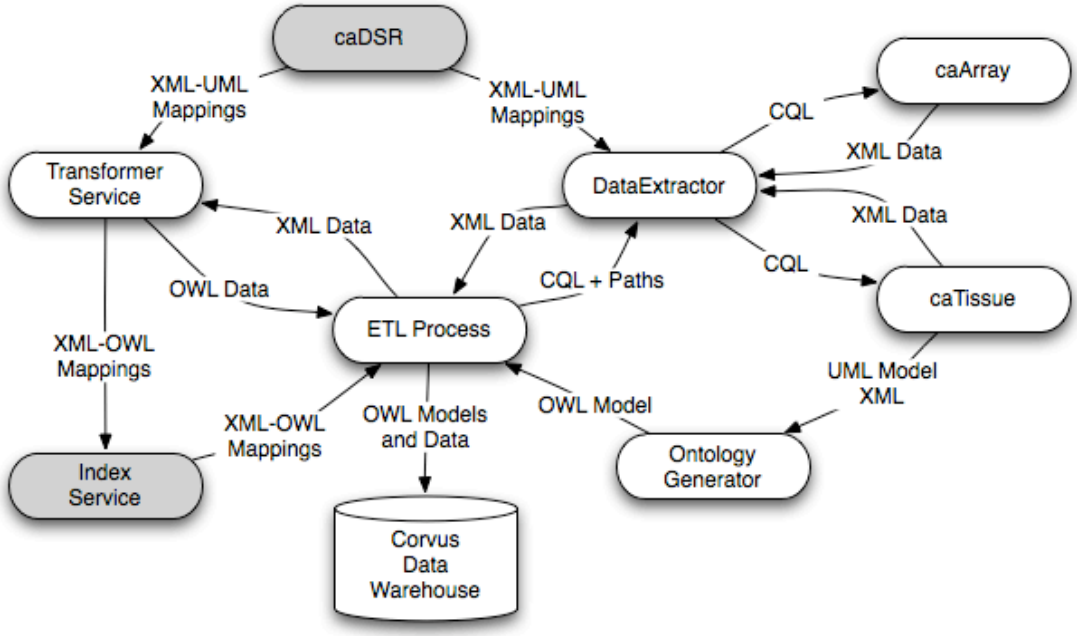
Figures

Figure 1 - Principal Components Analysis



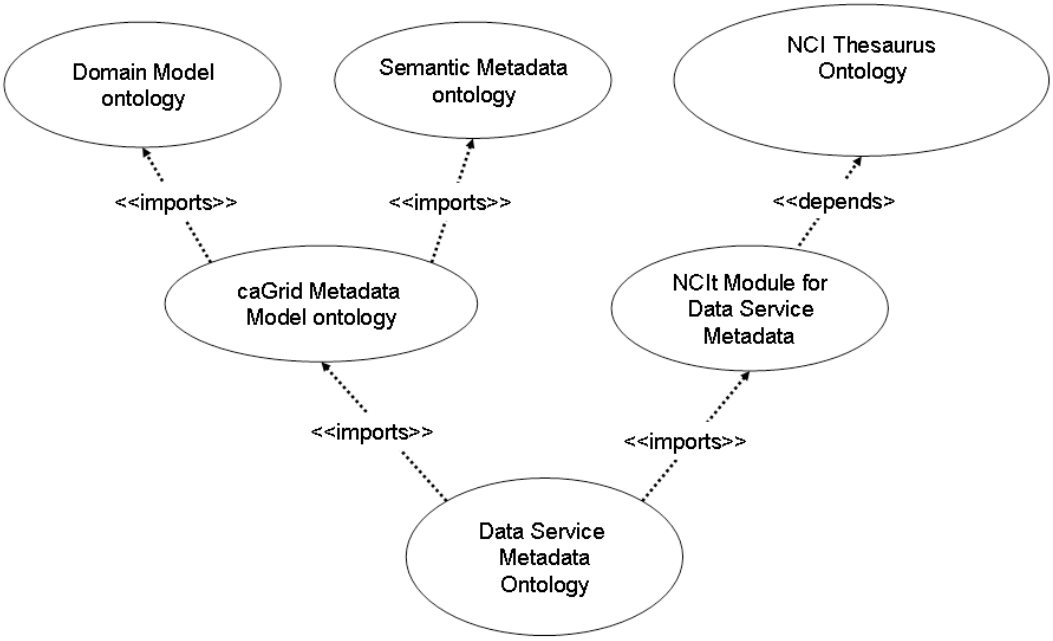
Projection of the first two principal components of gene expression microarray experiment GSE5949 from GEO. The clinical diagnoses for the biological source cell line were extracted from caTissue and joined using Corvus.

Figure 2 - Corvus ETL Process



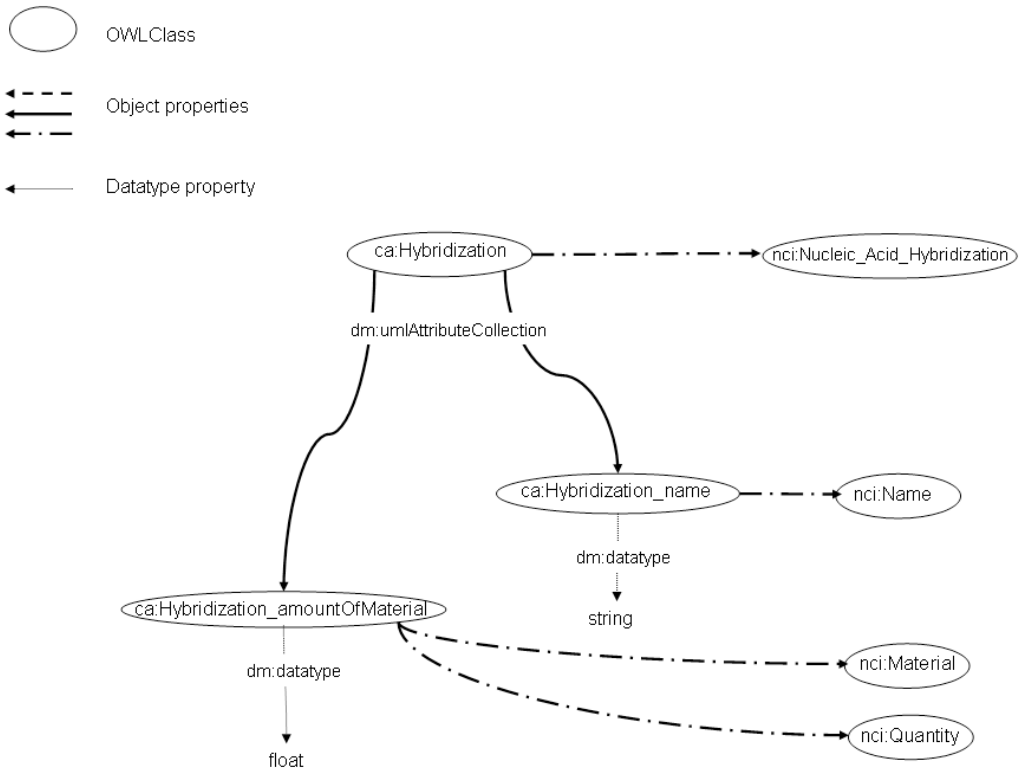
The Corvus ETL Process

Figure 3 - Modules Diagram



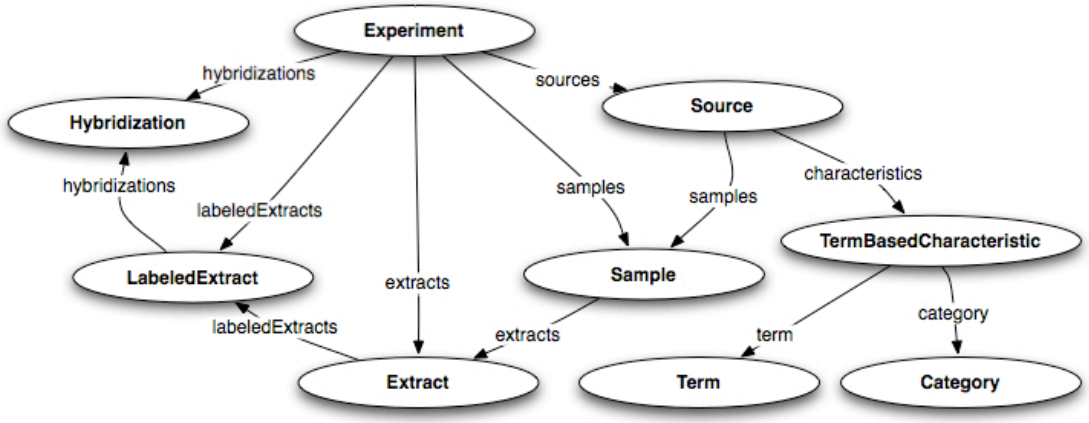
Package diagram showing the import and dependency relationships between ontology modules.

Figure 4 - Annotated UML Class OWL Representation



Representation of the Hybridization class from caArray in OWL form. Hybridization is annotated with the NCIt term "Nucleic Acid Hybridization" and has two attributes: "name" and "amountOfMaterial", which in turn have their NCIt annotations. The values of these attributes are maintained via the *dm:datatype* property.

Figure 5 - caArray Query Paths



caArray query paths – a superset of this graph is needed to extract all information about an experiment.

Figure 6 - Join Cell Lines By Labels

```
Id: joinCellLinesByLabels


     // a is a Source
     a <rdf:type>                                        <caarray:Source>
     // b is a TermCharacteristic of a
     a <caarray:Source_characteristics_TermBasedCharacteristic> b
     b <rdf:type>                                        <caarray:TermBasedCharacteristic>


     // i is a cellSpecimen with a label which resolves to the same as h.
     i <rdf:type>                    <catissuecore:CellSpecimen>
     i <dm:umlAttributeCollection>   j
     j <rdf:type>                    k
     k <rdfs:subClassOf>             l
     l <rdf:type>                    <owl:Restriction>
     l <owl:onProperty>              <sm:semanticMetadataCollection>
     l <owl:someValuesFrom>          <nci:Label>


     j <dm:datatype>                 h
```

```
    -------------------------------

     // Therefore, the Source a is derived from the CellSpecimen i,
     // and i is derived by a.
     a <corvus:derived_from> i
     i <corvus:derived_by>   a

}
```

A TRREE inferencing rule that describes an inferred relationship between cell specimens in caTissue and biological sources in caArray. The relationship here is derivation, as described by *derived_from* and *derived_by* properties.

Figure 7 - Hybridization Diagnosis SPARQL

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX corvus: <http://krauthammerlab.med.yale.edu/ontologies/corvus-mapping.owl#>
PREFIX dm: <http://www.cs.ucl.ac.uk/staff/a.gonzalezbeltran/owl/domainmodel.owl#>
PREFIX nci: <http://www.cs.ucl.ac.uk/staff/a.gonzalezbeltran/owl/EVS/Thesaurus.owl#>
PREFIX sm: <http://www.cs.ucl.ac.uk/staff/a.gonzalezbeltran/owl/semanticmetadata.owl#>
PREFIX caarray: <http://www.cs.ucl.ac.uk/staff/a.gonzalezbeltran/owl/caarray2_1.owl#>
PREFIX catissuesuite: <http://www.cs.ucl.ac.uk/staff/a.gonzalezbeltran/owl/catissuesuite1.owl#>
PREFIX catissuecore: <http://www.cs.ucl.ac.uk/staff/a.gonzalezbeltran/owl/catissuecore1_2.owl#>

SELECT DISTINCT ?hybridizationName ?diagnosis
WHERE
{

    # Find all Sources that are derived from a Specimen.
    ?source
```

```
    corvus:derived_from ?specimen.

# Traverse from Source to Hybridization.
?source
    caarray:Source_samples_Sample ?sample.
?sample
    caarray:Sample_extracts_Extract ?extract.
?extract
    caarray:Extract_labeledExtracts_LabeledExtract ?labeledExtract.
?labeledExtract
    caarray:LabeledExtract_hybridizations_Hybridization ?hybridization.
?hybridization
    dm:umlAttributeCollection ?hybridizationNameAtt.

# Extract the name of the Hybridization
?hybridizationNameAtt
    rdf:type ?hybridizationNameAttType .
?hybridizationNameAttType
    rdfs:subClassOf ?hnaSmr .
?hnaSmr
    rdf:type owl:Restriction;
    owl:onProperty sm:semanticMetadataCollection;
    owl:someValuesFrom nci:Name.

?hybridizationNameAtt
    dm:datatype ?hybridizationName.

# Find all Specimen Collection Groups that are linked to our Specimens.
?scg
    catissuecore:SpecimenCollectionGroup_specimenCollection_CellSpecimen ?specimen.
```

```
    # Extract the Clinical Diagnosis at the time the
    # Specimen Collection Group was collected.
    ?scg
        dm:umlAttributeCollection ?scgDiagnosisAtt.
    ?scgDiagnosisAtt
        rdf:type ?scgDiagnosisAttType .
    ?scgDiagnosisAttType
        rdfs:subClassOf ?scgdaSmr .
    ?scgdaSmr
        rdf:type owl:Restriction;
        owl:onProperty sm:semanticMetadataCollection;
        owl:someValuesFrom nci:Clinical_Diagnosis.
    ?scgDiagnosisAtt
        dm:datatype ?diagnosis.
}
```

SPARQL query used to retrieve the clinical diagnosis for the cell line used in every hybridization of GEO GSE5949. The query takes advantage of the inferred relationship of *derived_from* that is described in Figure 6.

Tables

Table 1 - Load and Query Performance

Load and query times for the operations used. Computer used has an Intel Core 2 Quad @ 2.40GHz and 4 GB of memory. The repository was single-threaded.

Stage	Data Size (Entities)	Data Size (Statements)	Processing Time (s)
Loading Ontologies	57,526	88,654	473
Loading Data	14.003	607,532	910
Query	-	-	3.24

Additional Files

Please see the paper web site at http://krauthammerlab.med.yale.edu/wiki/Semantic_web_data_warehousing_for_caGrid for updated files and further information.

Additional file 1 — corvus-mapping.owl

The OWL definition of the derived_by property used to link Cell Specimens and Sources.

Additional file 2 — catissueCaarray.list

List of external ontology URLs to load into Corvus.

Additional file 3 — catissueData.ntp

N-Triple data for caTissue collection protocol, participant, and specimen individuals.

Additional file 4 — annotationsLinkage.ntp

caArray Annotation Linkages in N-Triple format.

Additional file 5 — experimentWAannotations.ntp

caArray Experiment Data and Annotations in N-Triple format.

Additional file 6 — sourceCharacteristics.ntp

caArray Source Characteristics in N-Triple format.

Additional file 7 — corvus.pie

TRREE inferencing rule configuration, including rule to link caTissue Cell Specimens with caArray Sources.

Additional file 8 — diagnosis.txt

Diagnoses for hybridizations, tab separated.

Additional file 9 — pca.csv

Principal Components Analysis of gene expression data, tab separated.