Chapter 1

SEMANTIC WEB APPROACH TO DATABASE INTEGRATION IN THE LIFE SCIENCES

Kei-Hoi Cheung[1,2,3,4], Andrew K. Smith[4], Kevin Y.L. Yip[4], Christopher J.O. Baker[6,7] and Mark B. Gerstein[4,5]

[1]Yale Center for Medical Informatics, [2]Anesthesiology, [3]Genetics, [4]Computer Science, [5] [5]Molecular Biophysics and Biochemistry, Yale University, USA, [6]Computer Science and Software Engineering, Concordia University, Canada, [7]Institute for Infocomm Research, Singapore.

Abstract: This chapter describes the challenges involved in the integration of databases storing diverse but related types of life sciences data. A major challenge in this regard is the syntactic and semantic heterogeneity of life sciences databases. There is a strong need for standardizing the syntactic and semantic data representations. We discuss how to address this by using the emerging Semantic Web technologies based on the Resource Description Framework (RDF) standard. This chapter presents two use cases, namely ***YeastHub*** and ***LinkHub***, which demonstrate how to use the latest RDF database technology to build data warehouses that facilitate integration of genomic/proteomic data and identifiers.

Key words: RDF database, integration, Semantic Web, molecular biology.

1. INTRODUCTION

The success of the Human Genome Project (HGP) [1] together with the popularity of the Web (or World Wide Web) [2] has made a large quantity of biological data available to the scientific community through the Internet. Since the inception of HGP, a multitude of Web accessible biological databases have emerged. These databases differ in the types of biological data they provide, ranging from sequence databases (e.g., NCBI's GenBank [3]), microarray gene expression databases (e.g., SMD [4] and GEO [5]),

pathway databases (e.g., BIND [6], HPRD [7], and Reactome [8]), and proteomic databases (e.g., UPD [9] and PeptideAtlas [10]). While some of these databases are organism-specific (e.g., SGD [11] and MGD [12]), others like (e.g., Gene Ontology [13] and UniProt [14]) are relevant, irrespective of taxonomic origin. In addition to data diversity, databases vary in scale ranging from large global databases (e.g., UniProt [14]), medium boutique databases (e.g., Pfam [15]) to small local databases (e.g., PhenoDB [16]). Some of these databases (especially the local databases) may be network-inaccessible and may involve proprietary data formats.

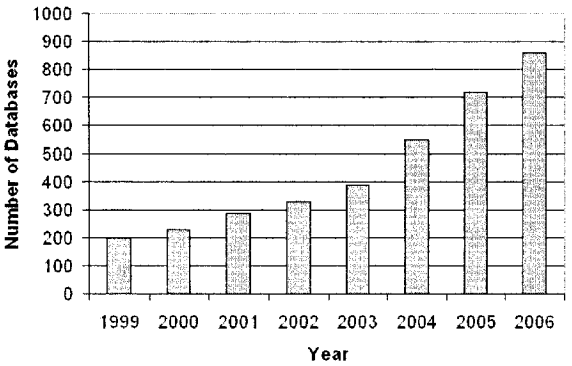

Figure 1-1. Number of databases published in the NAR Database Issues between 1999 and 2006.

Figure 1-1 indicates the rate of growth in the number of Web-accessible molecular biology databases, which were published in the annual Database Issue of Nucleic Acids Research (NAR) between 1999 and 2006. These databases only represent a small portion of all biological databases in existence today. With the sustained increase in the number of biological databases, the desire for integrating and querying combined databases grows. Information needed for analysis and interpretation of experimental results is frequently scattered over multiple databases. For example, some microarray gene expression studies may require integrating different databases to biologically validate or interpret gene clusters generated by cluster analysis [17].

For validation, the gene identifiers within a cluster may be used to retrieve sequence information (e.g., from GenBank) and functional information (e.g., from Gene Ontology) to determine whether the clustered genes share the same motif patterns or biological functions. For interpretation, such gene expression data may be integrated with pathway data provided by different pathway databases to elucidate relationships between gene expression and pathway control or regulation.

Database integration is of the key problems that Semantic Web aims to address. As stated in the introduction of World Wide Web Consortium's Semantic Web page (http://www.w3.org/2001/sw/): *"The Semantic Web is about two things. It is about common formats for interchange of data, where on the original Web we only had interchange of documents. Also it is about language for recording how the data relates to real world objects. That allows a person, or a machine, to start off in one database, and then move through an unending set of databases which are connected not by wires but by being about the same thing."*

Below we review the challenges faced when integrating information from multiple databases.

- **Locating Resources.** Automated identification of Websites that contain relevant and interoperable data poses a challenge. There is a lack of widely-accepted standards for describing Websites and their contents. Although the HTML meta tag (http://www.htmlhelp.com/reference/html40/head/meta.html) can be used to annotate a Web page through the use of keywords, such tags are problematic in terms of sensitivity and specificity. Furthermore, these approaches are neither supported nor used widely by existing Web search engines. Most Web search engines rely on using their own algorithms to index individual Websites based on their contents.
- **Data Formats.** Different Web resources provide their data in heterogeneous formats. For example, while some data are represented in the HTML format, interpretable by the Web browser, other data formats including the text format (e.g., delimited text files) and binary format (e.g., images) are commonplace. Such heterogeneity in data formats makes universal interoperability difficult if not impossible.
- **Synonyms.** There are many synonyms for the same underlying biological entity as a consequence of researchers independently naming entities for use in their own datasets or because of legacy common names (such as the famous "sonic hedgehog" gene name) arbitrarily given to biological entities before large-scale databases were created. Some such names have managed to remain in common use by researchers. An example of this problem is the many synonymous protein identifiers, assigned by laboratories to match their own lab-specific protein identifiers. There can also be lexical variants of the same underlying identifier (e.g., GO:0008150 vs. GO0008150 vs. GO-8150).
- **Ambiguity.** Besides synonyms, the same term (e.g., insulin) can be used to represent different concepts (e.g., gene, protein, drug, etc). This problem can also occur at the level of data modeling. For example, the concept 'experiment' in one microarray database (e.g., SMD [4]) may refer to a series of samples (corresponding to different experimental

conditions) hybridized to different arrays. In another microarray database (e.g., RAD [18]), an experiment may refer to a single hybridization.

- **Relations.** There are many kinds of relationships between database entries including one-to-one and one-to-many relationships. For example, a single Gene Ontology identifier can be related with many UniProt identifiers (i.e. they all share the same functional annotation). An important structuring principle for genes and proteins, which leads to one-to-many relationships, is the notion of families based on evolutionary origin. A given protein or gene can be composed of one or more family specific units, called domains. For example, a UniProt entity may be composed of two different Pfam domains. In general a given Pfam domain [15] will be related to many UniProt proteins by this family association, and the UniProt proteins can in turn be related to other entities through various kinds of relationships (and similarly for GO). A transitive closure in such a relationship graph, even a few levels deep, can identify relationships with a great number of other entities. It is important to note, however, that there are certain relationship types for which following them in the wrong way can lead to incorrect inferences, with the family relationship being a key one.
- **Granularity.** Different biological databases may provide information at different levels of granularity. For example, information about the human brain can be modeled at different granular levels. In one database, the human brain may be divided into different anatomical regions (e.g., hippocampus and neocortex), another database may store information about the different types of neurons (e.g., Purkinje cells) at different brain regions (e.g., ventral paraflocculus). For an even finer level of granularity, some neuroscience databases store information about the membrane properties at different compartments of the neuron.

2. APPROACHES TO DATABASE INTEGRATION

There are two general approaches to database integration, namely, the data warehouse approach and the federated database approach. The data warehouse approach emphasizes data translation, whereas the federated approach emphasizes query translation [19]. The warehouse approach involves translating data from different sources into a local data warehouse, and executing all queries on the warehouse rather than on the distributed sources of that data. This approach eliminates various problems including network bottlenecks, slow response times, and the occasional unavailability of sources. In addition, creating a warehouse allows for an improved query

efficiency or optimization since it can be performed locally [20]. Another benefit in this approach is that it allows values (e.g., filtering, validation, correction, and annotation) to be added to the data collected from individual sources. This is a desirable feature in the domain of biosciences. The approach, however, suffers from the maintenance problem in light of evolution of the source database (both in structure and content). The warehouse needs to be periodically updated to reflect the modifications of the source databases. Some representative examples of biological data warehouse include BioWarehouse [21], Biozon [22], and DataFoundry [23].

The federated database approach concentrates on query translation [24]. It involves a mediator, which is a middleware responsible for translating, at runtime, a query composed by a user on a single federated schema into queries on the local schemas of the underlying data sources. A mapping is required between the federated schema and the source schemas to allow query translation between the federated schema and the source schemas. While the federated database approach ensures data is concurrent / synchronized and is easier to maintain (when new databases are added), it generally has a poorer query performance than the warehouse integration approach. Some representative examples of the federated database include BioKleisli [25], Discoverylink [26], and QIS [27].

2.1 Semantic Web Approach to Data Integration

Traditional approaches (including the data warehouse and federated database) to data integration involve mapping the component data models (e.g., relational data model) to a common data model (e.g., object-oriented data model). To go beyond a data model, the Semantic Web approach [28] relies on using a standard ontology to integrate different databases. Unlike data models, the fundamental asset of ontologies is their relative independence of particular applications. That is, an ontology consists of relatively generic knowledge that can be reused by different kinds of applications. In the Semantic Web, several ontological languages (implemented based on the eXtensible Markup Language or XML) have been proposed to encode ontologies.

2.1.1 RDF vs. XML

While the HyperText Markup Language (HTML) is used for providing a human-friendly data display, it is not machine-friendly. In other words, computer applications do not know the meaning of the data when parsing the HTML tags, since they only indicate how data should be displayed. To address this problem, the eXtensible Markup Language (XML) was

introduced, to associate meaningful tags with data values. In addition, a hierarchical (element/sub-element) structure can be created using these tags. With such descriptive and hierarchically-structured labels, computer applications are given better semantic information to parse data in a meaningful way.

Despite its machine readability, as indicated by Wang et al. [29], the nature of XML is syntactic and document-centric. This limits its ability to achieve the level of semantic interoperability required by the highly dynamic and integrated bioinformatics applications. In addition, there is a problem with both the proliferation and redundancy of XML formats in the life science domain. Overlapping XML formats (e.g., SBML [30] and PSI MI [31]) have been developed to represent the same type of biological data (e.g., pathway data).

The introduction of the Semantic Web [28] has taken the usage of XML to a new level of ontology-based standardization. In the Semantic Web realm, XML is used as an ontological language to implement machine-readable ontologies built upon standard knowledge representation techniques. The Resource Description Framework (RDF) (http://www.w3.org/RDF/) is an important first step in this direction. It offers a simple but useful semantic model based on the directed acyclic graph structure. In essence, RDF is a modeling language for defining statements about resources and relationships among them. Such resources and relationships are identified using the system of Uniform Resource Identifiers (URIs). Each RDF statement is a triplet with a **subject**, **property** (or **predicate**), and **property value** (or **object**). For example, *<"http://en.wikipedia.org/wiki/Protein#", "http://en.wikipedia.org/wiki/Name#", "http://en.wikipedia.org/wiki/P53#">* is a triple statement expressing that the subject *Protein* has *P53* as the value of its *Name* property. The objects appearing in triples may comprehend pointers to other objects in such a way as to create a nested structure. RDF also provides a means of defining classes of resources and properties. These classes are used to build statements that assert facts about resources. RDF uses its own syntax (RDF Schema or RDFS) for writing a schema for a resource. RDFS is more expressive than RDF and it includes subclass/superclass relationships as well as the option to impose constraints on the statements that can be made in a document conforming to the schema.

Some biomedical datasets such as the Gene Ontology [13], UniProt (http://expasy3.isb-sib.ch/~ejain//rdf/), and the NCI thesaurus [32] have been made available in RDF format. In addition, applications that demonstrate how to make use of such datasets have been developed (e.g.,[33, 34]).

2.1.2 OWL vs. RDF

While RDF and RDFS are commonly-used Semantic Web standards, neither is expressive enough to support formal knowledge representation that is intended for processing by computers. Such a representation consists of explicit objects (e.g., the class of all proteins, or P53 a certain individual), and of assertions or claims about them (e.g., "EGFR is an enzyme", or "all enzymes are proteins"). Representing knowledge in such explicit form enables computers to draw conclusions from knowledge already encoded in the machine-readable form. More sophisticated XML-based knowledge representation languages such as the Web Ontology Language [35] have been developed. OWL is based on description logics (DL) [36], which are a family of class-based (concept-based) knowledge representation formalisms [36]. They are characterized by the use of various constructors to build complex classes from simpler ones, an emphasis on the decidability of key reasoning problems, and by the provision of sound, complete and (empirically) tractable reasoning services. Description Logics, and insights from DL research, had a strong influence on the design of OWL, particularly on the formalization of the semantics, the choice of language constructors, and the integration of data types and data values. For an in-depth overview of OWL, the reader can refer to the chapter entitled: "OWL for the Novice: A Logical Perspective".

In the life science domain, the pathway exchange standard called BioPAX (http://www.biopax.org/) has been deployed in OWL to standardize the ontological representation of pathway data [37]. Increasingly, pathway databases including HumanCyc [38] and Reactome [8] have exported data in the OWL-based BioPAX format. As another example, the FungalWeb Project [39] has integrated a variety of distributed resources in the domain of fungal enzymology into a single OWL DL ontology which serves as an instantiated knowledgebase allowing complex domain specific A-box queries using DL based reasoning tools. In contrast [40] have translated a single large scale taxonomy of human anatomy from a frame-based format into OWL which supports reasoning tasks.

3. USE CASES

This section presents two use cases, namely **YeastHub** [33] and **LinkHub** (http://hub.gersteinlab.org/), which demonstrate how to use the RDF approach to integrate heterogeneous genomic data. Both of these use cases involve using a native RDF database system called Sesame (http://www.openrdf.org) to implement a warehouse or hub for integrating or

interlinking diverse types of genomic/proteomic data. Sesame allows a RDF repository to be created on top of main memory, relational database (e.g., MySQL and Oracle), and native RDF files. For small or moderate size datasets, the main memory approach provides the fastest query speed. For large amounts of data, Sesame utilizes the efficient data storage and indexing facilities provided by the relational database engine (e.g., MySQL and Oracle). Finally, the native file-based approach eliminates the need of using a database and its associated overhead at the cost of performance if the data files involved are large.

3.1 YeastHub

YeastHub features the construction of a RDF-based data warehouse (implemented using Sesame) for integrating a variety of yeast genome data. This allows yeast researchers to seamlessly access and query multiple related data sources to perform integrative data analysis in a much broader context. The system consists of the following components: registration, data conversion, and data integration.

3.1.1 Registration

This component allows the user to register a Web-accessible dataset so that it can be used by YeastHub. During the registration process, the user needs to enter information (metadata) describing the dataset (e.g., location (URL), owner, and data type). Such description is structured based on the Dublin Core metadata standard (http://dublincore.org/). To encode the metadata in a standard format, the Rich Site Summary (RSS) format was used. RSS is an appropriate lightweight application of RDF, since the amount of metadata involved is typically small or moderate. The RSS-encoded description of an individual dataset is called an "RSS feed". Many RSS-aware tools (e.g., RSS readers and aggregators) are available in the public domain, which allow automatic processing of RSS feeds. Among the different versions of RSS, RSS 1.0 was chosen because it supports RDF Schema. This allows ontologies to be incorporated into the modeling and representation of metadata. Another advantage of using RSS 1.0 is it that allows reuse of standard/existing modules as well as the creation of new custom modules. The custom modules can be used to expand the RSS metadata structure and contents to meet specific user needs.

3.1.2 Data conversion

Registered datasets often originate from different resources in different formats, making it necessary to convert these formats into the RDF format. A variety of technologies can be used to perform this data conversion. For example, we can use XSLT to convert XML datasets into the RDF format. For data stored in relational datasets, we can use D2RQ (http://www.wiwiss.fu-berlin.de/suhl/bizer/D2RQ/), for example, to map the source relational structure and the target RDF structure. In addition, YeastHub provides a converter for translating tabular datasets into the RDF format. The translation operates on the assumption that each dataset belongs to a particular data type or class (e.g., gene, protein, or pathway). One of the data columns/fields is chosen by the user to be the unique identifier. Each identifier identifies an RDF subject. The rest of the data columns or fields represent RDF properties of the subject. The user can choose to use the default column/field names as the property names or enter his/her own property names. Each data value in the data table corresponds to a property value. The system allows some basic filtering or transformation of string values (e.g., string substitution) when generating the property values. Once a dataset is converted into the RDF format, it can be loaded into the RDF repository for storage and queries. Additionally it can be accessed by other applications through API.

3.1.3 Data integration

Once multiple datasets have been registered and loaded into YeastHub's RDF repository, integrated RDF queries can be composed to retrieve related data across multiple datasets. YeastHub offers two kinds of query interface, allowing command line or form based query.

1. **Ad hoc queries.** Users are permitted to compose RDF-based query statements and issue them directly to the data repository. Currently the user can build queries in the following query languages: RQL, SeRQL, and RDQL. The user must be familiar with at least one of these query syntaxes as well as the structure of the RDF datasets to be queried. SQL users typically find it easy to learn RDF query languages.
2. **Form-based queries**. While ad hoc RDF queries are flexible, users who do not know RDF query languages often prefer to use supervised method to pose queries YeastHub allows users to query the repository through Web query forms (although they are not as flexible as the ad hoc query approach). To create a query form, YeastHub provides a query template generator. First of all, the user selects the datasets and the properties of interest. Secondly, the user needs to indicate which

properties are to be used for the query output (select clause), search Boolean criteria (where clause), and join criteria (property values that can be linked between datasets). In addition, the user is given the option to create a text field, pull down menu, or select list (in which multiple items can be selected) for each search property. Once all the information has been entered, the user can go ahead to generate the query form by saving it with a name. The user can then use the generated query form to perform Boolean queries on the datasets associated with the form.

3.1.3.1 Example query to correlate essentiality with connectivity

```
SELECT DISTINCT ns0orf,ns0connectivity,ns4accession,ns4name,ns5growth_condition,
ns5clone_id, ns5expression_level
FROM
{source58640} ns1:orf {ns1orf},
{source58639} ns2:orf {ns2orf},
{source58638} ns3:DB_Object_Synonym {ns3DB_Object_Synonym},
{source58638} ns3:GO_ID {ns3GO_ID},
{source58636} ns4:name {ns4name},
{source58636} ns4:accession {ns4accession},
{source55396} ns5:orf {ns5orf},
{source55396} ns5:growth_condition {ns5growth_condition},
{source55396} ns5:expression_level {ns5expression_level},
{source55396} ns5:clone_id {ns5clone_id},
{source58642} ns0:connectivity {ns0connectivity},
{source58642} ns0:orf {ns0orf}
WHERE
ns0connectivity="80"
AND ns5expression_level="1"^^<http://www.w3.org/2001/XMLSchema#longInteger>
AND ns5clone_id="V182B10"^^<http://www.w3.org/2001/XMLSchema#string>
AND ns5growth_condition="vegetative"^^<http://www.w3.org/2001/XMLSchema#string>
AND ns0orf=ns1orf
AND ns1orf=ns2orf
AND ns2orf=ns3DB_Object_Synonym
AND ns3DB_Object_Synonym=ns5orf
AND ns3GO_ID=ns4accession
USING NAMESPACE
ns2=<http://mcdb750.med.yale.edu/yeasthub/schema/schema58639.rdf> ,
ns3=<http://mcdb750.med.yale.edu/yeasthub/schema/schema58638.rdf> ,
ns1=<http://mcdb750.med.yale.edu/yeasthub/schema/schema58640.rdf> ,
ns0=<http://mcdb750.med.yale.edu/yeasthub/schema/schema58642.rdf> ,
ns5=<http://mcdb750.med.yale.edu/yeasthub/schema/schema_triples.rdf#> ,
ns4=<http://139.91.183.30:9090/RDF/VRP/Examples/schema_go.rdf>
```

ns0orf	ns0connectivity	ns4accession	ns4name	ns5growth_condition	ns5clone_id	ns5expression_level
YBL092W	80	GO:0005842	cytosolic large ribosomal subunit (sensu Eukaryota)	vegetative	V182B10	1
YBL092W	80	GO:0003735	structural constituent of ribosome	vegetative	V182B10	1
YBL092W	80	GO:0006412	protein biosynthesis	vegetative	V182B10	1

Figure 1-2. SeRQL query statement correlating between gene essentiality and connectivity.

Figure 1-2 shows a RDF query statement written in SeRQL (Sesame implementation of RQL), which simultaneously queries the following yeast resources: a) essential gene list obtained from MIPS, b) essential gene list obtained from YGDP, c) protein-protein interaction data (Yu et al. 2004), d) gene and GO ID association obtained from SGD, e) GO annotation and, f) gene expression data obtained from TRIPLES [41]. Datasets (a)- (d) are distributed in tab-delimited format. They were converted into our RDF format. The GO dataset is in an RDF-like XML format (we made some slight modification to it to make it RDF-compliant). TRIPLES is an Oracle

database. We used D2RQ to dynamically map a subset of the gene expression data stored in TRIPLES to RDF format.

The example query demonstrates how to correlate between gene essentiality and connectivity, based on the interaction data. The hypothesis is that the higher its connectivity, the more likely that the gene is essential. The example query includes the following Boolean condition: *connectivity = 80*, *expression_level = 1*, *growth_condition = vegetative*, and *clone_id = V182B10*. Such Boolean query joins across six resources based on common gene names and GO IDs. Figure 1-2 (at the bottom) shows the query output, which indicates that the essential gene (YBL092W) has a connectivity equal to 80. This gene is found in both the MIPS and YGDP essential gene lists. This confirms the gene's essentiality as the two resources might have used different methods and sources to identify their essential genes. The query output displays the corresponding GO annotation (molecular function, biological process, and cellular component) and TRIPLES gene expression data.

3.2 LinkHub

LinkHub can be seen as a hybrid approach between a data warehouse and a federated database. Individual LinkHub instantiations are a kind of mini, local data warehouse of commonly grouped data, which can be connected to larger major hubs in a federated fashion. Such a connection is established through the semantic relationship among biological identifiers provided by different databases.

A key abstraction in representing biological data is the notion of unique identifiers for biological entities and relationships (and relationship types) among them. For example, each protein sequence in the UniProt database is given a unique accession by the UniProt curators (e.g., Q60996). This accession uniquely identifies its associated protein sequence and can be used as a key to access its sequence record in UniProt. UniProt sequence records contain cross-references to related information in other genomics databases. For example, Q60996 is cross-linked in UniProt to Gene Ontology identifier GO:0005634 and Pfam identifier PF01603, although the kinds of relationships, which would here be "functional annotation" and "family membership" respectively, are not specified in UniProt. Two identifiers such as Q60996 and GO:0005634 and the cross-reference between them can be viewed as a single edge between two nodes in a graph, and conceptually then an important, large part of biological knowledge can be viewed as a massive graph whose nodes are biological entities such as proteins, genes, etc. represented by identifiers and the links in the graph are typed and are the specific relationships among the biological entities. The problem is that this

graph of biological knowledge does not explicitly exist. Parts of it are in existence piecemeal (e.g., UniProt's cross references to other databases), while other parts do not exist, i.e. the connections between structural genomics targets and UniProt identifiers. Figure 1-3 is a conceptual illustration of the graph of relationships among biological identifiers, with the boxes representing biological identifiers (originating database names given inside) and different edge types representing different kinds of relationships. For reasons of efficiency, we have implemented this relationship graph using MySQL. However, we have converted this relational database into its RDF counterpart for exploring the RDF modeling and querying capabilities.

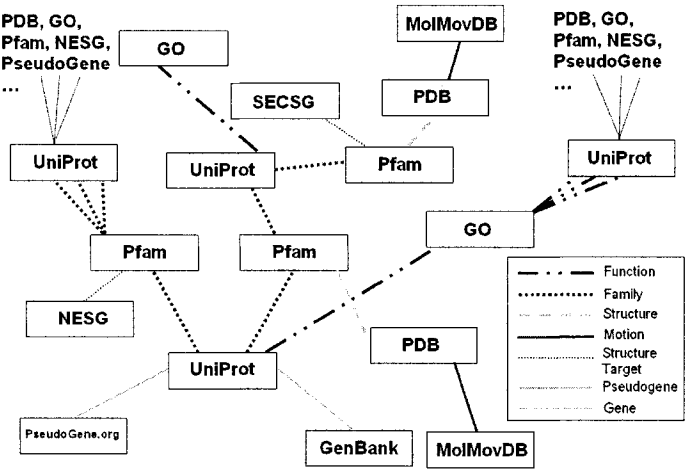

Figure 1-3. An example relationship graph among biological identifiers.

3.2.1 LinkHub Web interface

The primary interactive interface to the MySQL LinkHub database (MySQL) is a Web-based interface (implemented using the so-called AJAX technologies [13], i.e. DHTML, JavaScript, DOM, CSS, etc.) which presents subsets of the graph of relationships in a dynamic expandable / collapsible list view. This interface allows viewing and exploring of the transitive closure of the relationships stemming from a given identifier interactively one layer at a time: direct edges from the given identifier are initially shown and the user may then selectively expand fringe nodes an additional layer at a time to explore further relationships (computing the full transitive closure is prohibitive, and could also cause the user to "drown" in the data, and we thus limit it initially, and in each subsequent expansion, to anything one edge away, with the user then guiding further extensions based on the

relationships chosen for further exploration). Figure 1-4 is a screenshot of the interface and provides more detail. It also allows users to query and view particular types of path in the graph.

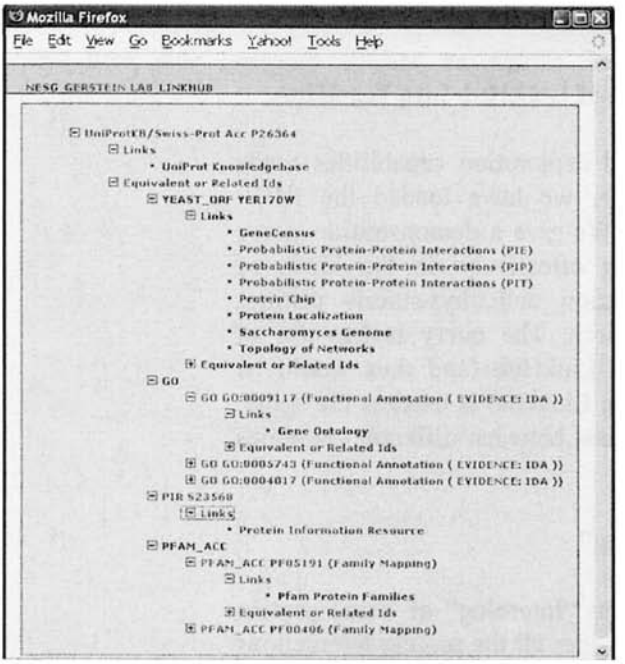

Figure 1-4. LinkHub Web Interface

For example, one might want to view all proteins in some database sharing the same Pfam family as a given protein. In LinkHub, Pfam relationships are stored for UniProt proteins, so one could view the sibling family members of the given protein by specifying to view all proteins, which can be reached by following a path of types like the following:

Given protein in database → equivalent UniProt protein → Pfam family → UniProt proteins → other equivalent proteins in database.

An important use of this "paths query" interface is as a secondary, orthogonal interface to other biological databases in order to provide different views of their underlying data. For example, the molecular motions database MolMovDB [14] provides movie clips of likely 3D motions of proteins, and one can access it by PDB [15] identifiers. However, an useful alternative would be a "family view" interface where the user queries with a PDB identifier and requests to see all available motions for proteins that are

in the same family as the query PDB identifier. LinkHub provides this interface for MolMovDB (we also provide a similar "family view" interface to structural genomics data, e.g. see the NESG's SPINE [16, 17] target pages such as http://spine.nesg.org/target.pl?id=WR4 for the "NESG Family Viewer" links).

3.2.2 RDF queries through integration of LinkHub into YeastHub

To demonstrate the data interaction and exploration capabilities made possible by the RDF version of LinkHub, we have loaded the RDF-formatted LinkHub dataset into YeastHub. We give a demonstration query written in SeRQL to show how one can effectively do the kinds of interesting exploratory scientific investigation and 'hypothesis testing' commonly done at the beginning of research. The query makes use of information present in both YeastHub and LinkHub (and thus would be impossible without joining the two systems). LinkHub is used as the 'glue' to provide both direct and indirect connections between different genomics identifiers.

3.2.2.1 Example query to find "interolog"

The example query here is to find Worm "Interolog" of Yeast protein interactions. With this query we want to consider all the protein interactions in yeast (S. cervisiae) and see how many and which of them are possibly present between their homologs in worm (C. elegans), i.e. as interologs [20] in worm. We thus start with a dataset containing known and predicted yeast protein interactions which is already loaded into YeastHub [21]; here the interactions are expressed between yeast gene names. For each Yeast gene name in the matched interaction set, we can use LinkHub's data as 'glue' to determine its homologs (via Pfam) in worm by traversing paths in the LinkHub relationship graph of type:

Yeast gene name → UniProt Accession → Pfam accession → UniProt Accession → WormBase ID .

Then, for each pair in the yeast protein interaction dataset, we determine if both of its yeast gene names lead to WormBase IDs [22] in this way and identify those WormBase IDs as possible protein interactions. The SeRQL query statement together with a portion of its corresponding output is shown in Figures 1-5 (a) and (b).

```
SELECT DISTINCT Yeast_Protein_1, Yeast_Protein_2, Worm_Protein_1, Worm_Protein_2
FROM
{ppi}    it:Protein1              {Yeast_Protein_1},
{lhYO1}  lh:identifiers_id        {Yeast_Protein_1},
{lhYO1}  lh:identifiers_type      {lhYOType},
{lhYO1}  lh:mappings_type_synonym {lhUP1a},
{lhUP1a} lh:identifiers_type      {lhUPType},
{lhUP1a} lh:mappings_type_Family_Mapping {lhPFAM1},
{lhPFAM1} lh:identifiers_type     {lhPFType},
{lhPFAM1} lh:mappings_type_Family_Mapping {lhUP1b},
...
WHERE
Yeast_Protein_1 = "YAL005C" AND
Yeast_Protein_2 = "YLR310C" AND
YEAST_ORF  = "YEAST_ORF" AND
(UNIPROT_KB = "UniProtKB/Swiss-Prot Acc" OR
 UNIPROT_KB = "UniProtKB/TrEMBL Acc") AND
PFAM_ACC   = "PFAM_ACC" AND
WORMBASE   = "WORMBASE"
USING NAMESPACE
it=<http://yeasthub2.gersteinlab.org/yeasthub/schema/the_platinum_standard_for_ppi20060224234451_schema.rdf>,
lh=<http://yeasthub2.gersteinlab.org/yeasthub/datasets/manual_upload/linkhub_schema.rdf#>
```

(a)

Yeast_Protein_1	Yeast_Protein_2	Worm_Protein_1	Worm_Protein_2
YAL005C	YLR310C	CE00103	CE01784
YAL005C	YLR310C	CE00103	CE16278
YAL005C	YLR310C	CE00103	CE19874
YAL005C	YLR310C	CE00103	CE28290
YAL005C	YLR310C	CE00103	CE31570
YAL005C	YLR310C	CE00103	CE31571

(b)

Figure 1-5. (a) SeRQL query statement for retrieving Worm "Interolog" of Yeast protein interactions. (b) Query output.

4. CONCLUSION AND FUTURE DIRECTIONS

Semantic Web (RDF) database technologies have been maturing over the past several years. The two use cases (LinkHub and YeastHub) presented in this chapter show that RDF data warehouses can be built to serve some practical data integration needs in the life science domain. While the relational database is the predominant form of database in use in life sciences today, it has the following limitations that can be addressed by the RDF database technology.

- While a relational schema can be exposed to local applications, it is not directly visible to Web agents. RDF or RDF schema can act as a gateway to allow relational databases to expose their data semantics to the World Wide Web.
- In relational databases, data links are implemented as primary-foreign key relationship. The meaning of this link relationship is implicit, and the semantics of the relationship cannot be specified as in RDF. Furthermore the primary-foreign key relationship cannot be applied to linking data items between separate relational databases. In RDF databases, link semantics are captured explicitly (through named RDF properties). These property-based links can be used to link data components between separate RDF graphs.

- The relational data model is not the natural approach to modeling hierarchical data that is hierarchical in nature. Such a parent-child relationship is usually captured in a relational table by adding a parent id column. Navigating or retrieving data based on such a hierarchical structure is typically done using self-join in a relational query statement (SQL). The main limitation of such an approach is that we need one self-join for every level in the hierarchy, and performance will degrade with each level added as the joining grows in complexity. RDF schema supports the subclass/superclass relationship and RDF databases are more optimized to support this type of parent-child data inference.

As the number of databases continues to grow, it is also important to explore how to build a federated database system based on Semantic Web technologies, which allows queries to be mediated across multiple Semantic Web databases. Such efforts have begun in the Computer Science research community (e.g., [42]). In the life science domain, Stephens et al. have demonstrated how to build a federated database using Cerebra (http://www.cerebra.com/) for integrating drug safety data [43]. Cerebra makes it possible to mediate queries against multiple RDF databases. In addition to supporting RDF query, it operates with OWL ontologies and OWL-based inferencing rules. However, it does not support standard OWL query languages (e.g., OWL-QL). Instead it uses XQuery to process the OWL ontologies and their associated data. XQuery is a standard query language for XML-structured data, yet it does not take advantage of the rich expressiveness provided by OWL.

To explore the full potential of the Semantic Web in data integration, we need to address the following areas.

- **Conversion.** There is a wealth of biological data that exists in other structured formats (e.g., relational format and XML format). We need to provide methods to convert these formats into a Semantic Web format (e.g., RDF or OWL). Such a conversion can be divided into syntactic and semantic parts. While both are important, semantic conversion usually takes a longer time to accomplish, since more effort is required to decide on the proper ontological conceptualization. This may be overcome in part by the ongoing development and improvement of bio-ontologies carried out by the biomedical ontology community including the National Center for Biomedical Ontology[x]. From a practical viewpoint, it might be easier to do the syntactic conversion first and followed by a gradual semantic conversion process. Based on the common syntax, data integration and semantic conversion can proceed in parallel. In addition to converting structured data into Semantic Web format, efforts are underway to extract data from the biomedical

literature (unstructured text) and structure the extracted results into Semantic Web formats.

- **Standard identifiers.** The problem with URL's is that they always point to a particular Web server (which may not always be in service) and worse, that the contents referred to by a URL may change. For researchers, the requirement to be able to exactly reproduce any observations and experiments based on a data object means that it is essential that data be uniquely named and available from many cached sources. The Life Science IDentifier or LSID (http://lsid.sourceforge.net) is designed to fulfill this requirement. An LSID names and refers to one unchanging data object (version numbers can be attached to handle updates). Every LSID consists of up to five parts: the Network Identifier [44]; the root DNS name of the issuing authority; the namespace chosen by the issuing authority; the object id unique in that namespace; and finally an optional revision id for storing versioning information. Each part is separated by a colon to make LSIDs easy to parse. For example, "urn:lsid:ncbi.nlm.nig.gov:GenBank:T48601:2" is an LSID with "urn:lsid" being the NID, "ncbi.nlm.nig.gov" the issuing authority's DNS name, "GenBank" the database namespace, "T48601" the object id, and "2" the revision id. Unlike URLs, LSIDs are location independent. This means that a program or a user can be certain that what they are dealing with is exactly the same data if the LSID of any object is the same as the LSID of another copy of the object obtained elsewhere. As an example of LSID usage, the Entrez LSID Web service (http://lsid.biopathways.org/entrez/) uses NCBI's Entrez search interface to locate LSIDs within the biological databases hosted by the NCBI. The LSID system is in essence similar to the role of the Domain Name Service (DNS) for converting named Internet locations to IP numbers.
- **Standardization of RDF/OWL Query Languages.** One of the reasons for the wide acceptance of relational database technology is that it comes with a standard and expressive database query language – SQL. Current RDF databases provide their own versions of RDF query languages (e.g., SeRQL for Sesame, iTQL for Kowari, and Oracle RDF query language). These query language variants provide different features. To integrate/consolidate these features, SPARQL is an emerging standard RDF query language (http://www.w3.org/TR/2004/WD-rdf-sparql-query-20041012). Even though it is a moving target, SPARQL-compliant query engines such as ARQ (http://jena.sourceforge.net/ARQ) have recently been implemented. For OWL ontologies, more expressive query languages are required. OWL-aware query languages (e.g., RDQL and nRQL [45]) are supported by specific OWL reasoners including

Pellet and Racer [45]. OWL-QL is a candidate standard query language for OWL.

- **Support of OWL reasoning.** Current RDF databases do not support OWL, although they can act as OWL data repositories. It would be useful to extend these RDF databases to support OWL querying and reasoning. One way of doing this is to create a reasoning layer on top of the RDF database. To this end, reasoner plugins such as OWLIM (http://www.ontotext.com/owlim/) have recently been made available for some RDF databases such as Sesame. Also, more direct and native support of OWL by RDF databases is desirable.

ACKNOWLEDGMENTS

This work was supported in part by NIH grant NHGRI K25 HG02378 and NSF grant BDI-0135442.

REFERENCES

[1] Cantor C.R. Orchestrating the Human Genome Project. *Science*. 248: 49-51, 1990.

[2] Berners-Lee T., Cailliau R., Luotonen A., Nielsen H. F., and Secret A. The World-Wide Web. *ACM Communications*. 37(3): 76-82, 1994.

[3] Benson D. A., Boguski M. S., Lipman D. J., and Ostell J. GenBank. Nucleic Acids Research. 25(1): 1-6, 1997.

[4] Gollub J., Ball C., Binkley G., Demeter J., Finkelstein D., Hebert J., Hernandez-Boussard T., Jin H., Kaloper M., Matese J., et al. The Stanford Microarray Database: data access and quality assessment tools. *Nucleic Acids Research*. 31(1): 94-6, 2003.

[5] Edgar R., Domrachev M., and Lash A. Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Research*. 30(1): 207-10, 2002.

[6] Bader G. D., Betel D., and Hogue C.W.V. BIND: the Biomolecular Interaction Network Database. *Nucl. Acids Res.* 31(1): 248-250, 2003.

[7] Peri S., Navarro J., Kristiansen T., Amanchy R., Surendranath V., Muthusamy B., Gandhi T., Chandrika K., Deshpande N., Suresh S., et al. Human protein reference database as a discovery resource for proteomics. *Nucl. Acids. Res.* 32: D497-501, 2004.

[8] Joshi-Tope G., Gillespie M., Vastrik I., D'Eustachio P., Schmidt E., de Bono B., Jassal B., Gopinath G.R., Wu G.R., Matthews L., et al. Reactome: a knowledgebase of biological pathways. *Nucleic Acids Res.* 33(Database issue): D428-32, 2005.

[9] Hill A. and Kim H. The UAP Proteomics Database. *Bioinformatics*. 19(16): 2149-51, 2003.

[10] Desiere F., Deutsch E. W., King N. L., Nesvizhskii A. I., Mallick P., Eng J., Chen S., Eddes J., Loevenich S. N., and Aebersold R. The PeptideAtlas project. *Nucl. Acids. Res.* 34 (Database Issue): D655-8, 2006.

[11] Dwight S. S., Harris M. A., Dolinski K., Ball C. A., Binkley G., Christie K. R., Fisk D.G., Issel-Tarver L., Schroeder M., Sherlock G., et al. Saccharomyces Genome

Database (SGD) provides secondary gene annotation using the Gene Ontology (GO). *Nucl. Acids. Res.* 30(1): 69-72, 2002.

[12] Blake J. A., Eppig J. T., Bult C. J., Kadin J. A., and Richardson J. E. The Mouse Genome Database (MGD): updates and enhancements. *Nucl. Acids. Res.* 34(Database Issue): D562-7, 2006.

[13] Ashburner M., Ball C., Blake J., Botstein D., Butler H., Cherry M., Davis A., Dolinski K., Dwight S., Eppig J., et al. Gene ontology: tool for the unification of biology. *Nature Genetics.* 25: 25-29, 2000.

[14] Apweiler R., Bairoch A., Wu C. H., Barker W. C., Boeckmann B., Ferro S., Gasteiger E., Huang H., Lopez R., Magrane M., et al. UniProt: the Universal Protein knowledgebase. *Nucl. Acids Res.* 32(90001): D115-119, 2004.

[15] Bateman A., Birney E., Cerruti L., Durbin R., Etwiller L., Eddy S., Griffiths-Jones S., Howe K., Marshall M., and Sonnhammer E. The Pfam Protein Families Database. *Nucleic Acids Research.* 30(1), 2002.

[16] Cheung K., Nadkarni P., Silverstein S., Kidd J., Pakstis A., Miller P., and Kidd K. PhenoDB: an integrated client/server database for linkage and population genetics. *Comput Biomed Res.* 29(4): 327-37, 1996.

[17] Shannon W., Culverhouse R., and Duncan J. Analyzing microarray data using cluster analysis. *Pharmacogenomics.* 4(1): 41-51, 2003.

[18] Manduchi E., Grant G.R., He H., Liu J., Mailman M. D., Pizarro A. D., Whetzel P. L., and Stoeckert C. J. RAD and the RAD Study-Annotator: an approach to collection, organization and exchange of all relevant information for high-throughput gene expression studies. *Bioinformatics.* 20(4): 452-9, 2004.

[19] Sujansky W. Heterogeneous database integration in biomedicine. *Journal of Biomedical Informatics.* 34: 285-98, 2001.

[20] Buneman P., Davidson S., Hart K., Overton C., and Wong L., A Data Transformation System for Biological Data Sources. in *Proc. 21st Int. Conf. VLDB.* 158-169, 1995.

[21] Lee T.J., Pouliot Y., Wagner V., Gupta P., Stringer-Calvert D.W., Tenenbaum J.D., and Karp P.D. BioWarehouse: a bioinformatics database warehouse toolkit. *Bioinformatics.* 7: 170, 2006.

[22] Birkland A. and Yona G. BIOZON: a hub of heterogeneous biological data. *Nucl. Acids. Res.* 34 (Database Issue): D235-42, 2006.

[23] Critchlow T., Fidelis K., Ganesh M., Musick R., and Slezak T. DataFoundry: information management for scientific data. *IEEE Trans Inf Technol Biomed.* 4(1): 52-7, 2000.

[24] Sheth A. and Larson J. Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases. *ACM Comput. Surveys.* 22(3): 183-236, 1990.

[25] Kolatkar P.R., Sakharkar M.K., Tse C. R., Kiong B. K., Wong L., Tan T.W., and Subbiah S. Development of software tools at BioInformatics Centre (BIC) at the National University of Singapore (NUS). in *Pac. Symp. Biocomputing.* Honolulu, Haiwaii 735-46, 1998.

[26] Haas L. M., Schwarz P. M., Kodali P., Kotlar E., Rice J.E., and Swope W.C. DiscoveryLink: A system for integrated access to life sciences data sources. *IBM Systems Journal.* 40(2): 489-511, 2001.

[27] Marenco L., Wang T.Y., Shepherd G., Miller P.L., and Nadkarni P. QIS: A framework for biomedical database federation. *J Am Med Inform Assoc.* 11(6): 523-34, 2004.

[28] Berners-Lee T., Hendler J., and Lassila O. The Semantic Web. Scientific American. 284(5): 34-43, 2001.

[29] Wang X., Gorlitsky R., and Almeida, J. S. From XML to RDF: how Semantic Web technologies will change the design of 'omic' standards. *Nat Biotechnol.* 23(9): 1099-103, 2005.

[30] Hucka M., Finney A., Sauro H., Bolouri H., Doyle J., Kitano H., Arkin A., Bornstein B., Bray D., Cornish-Bowden A., et al. The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics.* 19(4): 524-31, 2005.

[31] Hermjakob H., Montecchi-Palazzi L., Bader G., Wojcik J., Salwinski L., Ceol A., Moore S., Orchard S., Sarkans U., Mering C. V., et al. The HUPO PSI's Molecular Interaction format—a community standard for the representation of protein interaction data. *Nature Biotechnology.* 22: 177-83, 2004.

[32] Goldbeck J., Fragoso G., Hartel F., Hendler J., Parsia B., and Oberthaler J. The National Cancer Institute's Thesaurus and Ontology. *Journal of Web Semantics.* 1(1), 2003.

[33] Cheung K.-H., Yip K.Y., Smith A., deKnikker R., Masiar A., and Gerstein M. YeastHub: a Semantic Web use case for integrating data in the life sciences domain. *Bioinformatics.* 21(suppl_1): i85-96, 2005.

[34] Neumann E.K. and Quan D. Biodash: A Semantic Web Dashboard for Drug Development. in *Pacific Symposium on Biocomputing.* 176-87, 2006.

[35] Donis-Keller H., Green P., Helms C., Cartinhour S., Weiffenbach B., Stephens K., Keith T., Bowden D., Smith D., Lander E., et al. A Genetic Linkage Map of the Human Genome. *Cell.* 51: 319-337, 1987.

[36] Baader F., Calvanese D., McGuinness D., Nardi D., and Patel-Schneider P. The Description Logic Handbook.Cambridge University Press, 2002.

[37] Luciano J. S. PAX of mind for pathway researchers. *Drug Discov Today.* 10(13): 937-42, 2005.

[38] Romero P., Wagg J., Green M., Kaiser D., Krummenacker M., and Karp P. Computational prediction of human metabolic pathways from the complete human genome. *Genome Biol.* 6(1): R2, 2004.

[39] Baker C.J.O., Shaban-Nejad A., Su X., Haarslev V., and Butler G. Infrastructure for Fungal Enzyme Biotechnologists. *Journal of Web Semantics.* 4(3), 2006.

[40] Golbreich C., Zhang S., Bodenreider O. The Foundational Model of Anatomy in OWL. Journal of Web Semantics. 4(3), 2006.

[41] Kumar A., Cheung K.-H., Tosches N., Masiar P., Liu Y., Miller P., and Snyder M. The TRIPLES database: A Community Resource for Yeast Molecular Biology. Nucl. Acids. Res. 30(1): 73-75, 2002.

[42] Chen H., Wu Z., Wang H., and Mao Y. RDF/RDFS-based Relational Database Integration. in *ICDE*, Atlanta, Georgia, in press, 2006.

[43] Stephens S., Morales A., and Quinian M. Applying Semantic Web Technologies to Drug Safety Determination. *IEEE Intelligent Systems.* 21(1): 82-6, 2006.

[44] Miller R., Ioannidis Y., and Ramakrishnan R. Schema Equivalence in Heterogeneous Systems: Bridging Theory and Practice. *Inf. Sys.* 19(1): 3-31, 1994.

[45] Haarslev V., Moeller R., and Wessel M. Querying the Semantic Web with Racer + nRQL. in *Proceedings of the KI-04 Workshop on Applications of Description Logics.* Ulm, Germany: Deutsche Bibliothek, 2004.