Semantic Model to Integrate Biological Resources

Zoé Lacroix *, Louiqa Raschid † and María Esther Vidal ‡
* Arizona State University
PO Box 875706, Tempe Arizona 85287-5706, USA
zoe.lacroix@asu.edu
† University of Maryland
College Park Maryland 20742, USA
louiqa@umiacs.umd.edu
‡ Universidad Simón Bolívar, Caracas, Venezuela 1080
mvidal@ldc.usb.ve

Abstract—We present a framework that uses semantic modeling to represent biological data sources, the multiple links that capture relationships among them, as well as the various applications that transform or analyze biological data. We introduce a data model that encompass three layers: the ontological layer composed of an ontology to represent the scientific concepts and their relationships, the physical layer of the physical resources made available to the scientists, and the data layer composed of the entries accessible through the different resources.

Keywords: Ontology; Semantic Web; Integration; Mediation; Bioinformatics.

I. INTRODUCTION

To exploit biological resources and to execute scientific protocols requires that scientists must understand thousands of available resources, including data sources (databases and files), links (hyperlinks and indices), and applications that transform scientific objects in a meaningful way. Mastering each of the resources made available to the scientists involve significant effort. For each resource, the scientist has to understand the capabilities provided, their interface, the format of their input and output, etc. Although data providers are willing to support the scientists and help them to exploit the resources in a meaningful way, scientists typically have great difficulty to keep up with the overwhelming number of resources, as they evolve over time. As a result, scientists express their scientific protocols with the resources they know best, not the resources best suitable to their scientific aim. It has been shown [3] that the selection of resources to evaluate a scientific protocol has a significant impact on the result. Indeed, the evaluation of a scientific protocol when there is a choice of resources, is not deterministic. While two resources may offer information on a scientific object, say a gene, the output of a query may differ on the number of entries retrieved, the characterization of these entries (number of attributes), the values of the properties, etc. For these reasons, it is critical that the scientists are provided with the support needed to select the resources best suitable for each scientific protocol.

The following challenges must be addressed:

1) To allow scientists to map all resources to a meaningful model composed of scientific concepts and their relationships.
2) To provide scientists with a semantic map to explore the properties of all available resources.
3) To express scientifically meaningful queries regardless of their implementation.
4) To offer scientists a selection of resources (source paths) to evaluate their queries, and to advice them to explore and compare alternate paths.
5) To allow scientists to select the resources best meeting their protocol’s aim.
6) To execute their queries.

In this paper we present a framework that will address these challenges. Our framework will represent a Web accessible network of data sources and the applications that link them. We define a source graph to model the graph of existing resources and an ontology graph to capture the scientific concepts used to classify the resources. There will be a mapping between the two graphs. A mapping from a resource to a concept or a relationship has a score vector that represents the various metrics that characterize the way the physical resource represents the concept. A data graph represents the entries in the sources, as well as links between entries; this includes hyperlinks, indices or applications. The data graph is mapped to the source graph using a belong to relationship between an entry (and physical links associated with that entry) and the data source that contains the entry. Finally, a query language to express queries against the ontology graph is included in the framework.

The paper is organized as follows: in Section II we define our data model and query language. Section III describes the problem of ranking source paths and a strategy to rank the source paths of a given query. The paper concludes with Section IV.

II. LIFE SCIENCE DATA MODEL AND QUERY LANGUAGE

In this section we define a Life Science data model and a query language.

A. Life science data model

The life science data model illustrated in Figure 1 is composed of a meta-schema (left) that structures the three
modeled levels: conceptual, physical and entry, and the meta-data (right) that represents the life science instance, including the ontology selected to organize the physical resources, the physical resources integrated, and the entries accessible by the system.

The meta-schema is itself composed of an ontology graph, a source graph, and a data graph respectively defined as follows. For all definition, \( N \) is a set of names (or labels).

**Definition 1:** An ontology graph is a pair \( OG = (C, L_C) \), where \( C \) is a set of classes \( \{c_1, c_2, \ldots \} \) and \( L_C \subseteq C \times N \times C \) is a set of labeled binary associations between classes of \( C \).

Traditionally, an ontology is defined to capture the richness of representation of a scientific concept. For example, ontologies have been developed to represent a gene \([1, 2]\), a protein \([6]\), etc., but these scientific concepts are related. In the life science data model, an instance of the ontology graph, is a set of labeled binary associations between classes of \( C \).

The source graph models the sources and links that implement the different concepts and relationships specified in the ontology graph. An instance of the source graph (source instance) consists of a list of existing resources and their links. For example, the set \( \{OMIM, PubMed, GeneCards\} \) is an instance of \( \{s_1, s_2, s_3\} \), and the PubMed Links provided by NCBI to retrieve all publications from PubMed related to an OMIM entry is an instance of \( L_S \). A hyperlink that from a GeneCards entry links to a publication on PubMed is also an instance of \( L_S \).

Each physical resource is mapped to the ontology in the following way. A data source is mapped to the classes of scientific concepts that it provides information about. For example, the sources GeneCards, Geneev, and NCBI Gene will be mapped to the class Gene. Each link or application, will be mapped to the edge between the classes representing respectively its input and output, and that is best capturing its scientific meaning.

**Definition 3:** A data graph is a pair \( DG = (O, L_O) \), where \( O \) is a set of objects \( \{o_1, o_2, \ldots \} \) and \( L_O \subseteq O \times N \times O \) is a set of labeled links between objects of \( O \).

Formally a Life Science data model is defined as follows:

**Definition 4:** A life science data model is a 7-tuple \( LS = (OG, SG, DG, m_S, m_{LS}, m_O, m_{LO}) \) where:

- \( OG = (C, L_C) \) is an ontology graph;
- \( SG = (S, L_S) \) is a source graph;
- \( DG = (O, L_O) \) is a data graph;
- \( m_S \) is a mapping from \( S \) to \( C \times [0,1] \) where for each \( s \in S \), there exists \( c \in C \) such that \( m_S(s) \) represents the class \( c \) that \( s \) implements, and \( m_S(s) \) represents a score \( v \) that measures a property of the implementation of \( c \) by \( s \);
- \( m_{LS} \) is a mapping from \( L_S \) to \( L_C \times [0,1] \) where for each \( L_{i,j} \in L_S \), there exists \( r \in L_C \) such that \( m_{LS}(L_{i,j}) \) represents the binary association \( r \) that \( L_{i,j} \) implements, and \( m_{LS}(L_{i,j}) \) represents a score \( v \) that measures a property of the implementation of \( r \) by \( L_{i,j} \);
- \( m_O \) is a mapping from \( O \) to \( S \) where for each \( o \in O \), \( o \)
is an entry of a data source \( s \in S \), i.e., \( m_O(o) = s \);
- \( m_{L_O} \) is a mapping from \( L_O \) to \( L_S \) such that for each \( o_i \) and \( o_j \in O \), and \( l_{i,j} \in L_O \), there exists \( l \in L_S \) such that \( l = (m_O(o_i), l_{i,j}, N, m_O(o_j))^1 \) and \( m_{L_O}(l_{i,j}) = l \).

The mapping between the logical level and the physical level defined by \( m_S \) and \( m_{L_S} \) may be characterized by:

- the number of attributes describing the entries corresponding to the class with respect to the description of the class by its internal ontology;
- the quality of the data source (reliability of the authorized institution, status of the curation, or a user preference);
- the number of entries capturing the coverage of the data source.

Similarly, the mapping \( m_{L_S} \) between a physical link or application and a relationship of the ontology may be measured by its domain, image, or cardinality. The model may include several of these semantics in a score-vector, but for sake of simplicity, we only include one score in the definition.

**B. Motivating Example**

A model of six biological resources OMIM, PubMed, Swiss-Prot, NCBI Protein, NCBI Gene, and GeneCards can be built as follows. For each data source, the classes best characterizing their entries are selected: OMIM is characterized by Gene and Disease, PubMed by Publication, Swiss-Prot and NCBI Protein by Protein, and NCBI Gene and GeneCards by Gene. This mapping defines \( m_S \). Then the links between the physical sources are analyzed so that their semantics is understood. There exist multiple links between these resources, and we will describe how to build the mapping \( m_{L_S} \) with a few. OMIM is linked to PubMed with an index provided by NCBI called PubMed Links. The links express that the publications from PubMed are related to the genetic disease of OMIM they are linked to. The conceptual link from the class Disease to the class Publication labeled is-related-to is mapped to the physical link PubMed Links. Similarly, GeneCards links a gene to the protein it codes for with a hyperlink defined in the entry. This physical hyperlink is mapped to the conceptual link codes-for between the classes Gene and Protein. Once the links between physical links are defined and mapped, the mapping \( m_{L_S} \) is completed. The data contained in each of the resources are virtually represented in the model as objects of \( O \). Figure 3 illustrates this model.

**C. Query language**

Queries may be expressed at any of the three layers of the model. Conceptual queries are queries expressed solely against the logical layer of the model composed of classes \( C \) and their relationships \( L_C \). Conceptual queries aim at expressing the scientific aim of a query. They are defined as paths over the conceptual graph composed of \( C \) and \( L_C \). For example, “retrieve publications relevant to a genetic disorder” is a conceptual query. A list of physical paths that may be used to evaluate the query is returned; this is implemented in the BioNavigation [4] prototype. The language also allows a query to specify physical resources that may or may not occur in the result path. A query expressed only over the physical resources explicitly specifies a scientific protocol as a succession of traversing physical resources. The language also allows queries to be expressed at the object level (specific data entries). We formally define the query language as follows:

A query is expressed as a path regular expression over the sets \( C \) and \( L_C \), where each class in \( C \) can optionally be annotated with a Boolean expression. The simplex expression is the comparison of a Field to a particular value. In this paper, a field can be either source or Object Content, and the relational operators can be “=” or “contains”. A condition over source and the relational operator “=” (source = “name-of-source”), restricts the query to some specific sources that implement the class. A condition on Object Content and the relational operator “contains”, (Object content contains “set-of-keywords”), specifies the keywords that must occur within objects in the object graph. The symbol \( \epsilon_C \) is a wildcard matching any class. On the other hand, the symbol \( \epsilon_L \) is a wild-card that represents any link. We use the symbol \( \epsilon \) to denote \( \epsilon_C \cdot \epsilon_L \). The BNF specification of the language syntax is given in Figure 4.

**Example 1:** The query: “Retrieve publications about genes that encode for proteins that are related to ‘lymphocytes and apoptosis’” is expressed as follows:

\[
\text{Q}_1 = \text{Protein} \\
\text{[Object Content} \text{"contains"} \{\text{lymphocytes, apoptosis}\}] \\
\text{is_coded_by.Gene} \\
\text{is_related_to.Publication}
\]

The result of the evaluation of the query \( Q_1 \) is two sets of paths \( S_p \) and \( S_o \). Paths in \( S_p \) are source paths, i.e., paths in the Source Graph composed of a succession of resources that implement \( Q_1 \). For each path \( P \in S_p \), there exists an object path in \( S_o \), i.e., path of the Data Graph, composed of a succession of subsets of resources that correspond to the datasets obtained by the execution of \( Q_1 \) on the source path.
III. RANKING RELEVANT SOURCE PATHS

In the life science model, queries are expressed in terms of scientific concepts and the answers correspond to the objects reached by traversing the different resources that implement them. Because a scientific concept may be implemented by different data sources, a navigational query can be answered by a choice of alternate paths between an origin source and a target source. Each source path may reach a different dataset. Example 2: Consider the query: “Retrieve the publications related to the disease ‘diabetes’ traversing any number of intermediate resources,” using the proposed query language, this query can be expressed as follows:

\[ Q_2 = \text{Disease}[\text{Object}_\text{content} \text{contains} \{ \text{diabetes} \}] \text{.is_related.to.Publication} \]

Note that the wildcard \( \epsilon \) denotes any source and link between sources that implement the scientific concepts Disease and Publication. The Kleene’s closure denoted by the star (“*”), expresses that any number of resources may be considered between these two scientific concepts.

The evaluation of \( Q_2 \) may lead to significantly different datasets depending on the resources chosen to execute it. A source path that validate \( Q_2 \) is OMIM.PubMedLinks.PubMed, that is a path of length 1 from OMIM (mapped to Disease) to PubMed (mapped to Publication), via the NCBI link PubMedLinks (mapped to is_related.to). This path executed on keywords characterizing ‘diabetes’ retrieves 48,941 entries. On the other hand, there is an alternative path OMIM.NucleotideLinks.Nucleotide.PubMedLinks.PubMed of length 2 from OMIM to the NCBI Nucleotide (mapped to Sequence), via the NCBI link NucleotideLinks, then to PubMed via the NCBI link PubMedLinks. This path retrieves 64,534 entries. Additionally, there is another path of length

1. OMIM.Parse.PubMed, where Parse is an application that extracts PubMed identifiers from OMIM entries. This last path retrieves 50,843 entries.

Example 3: Consider the query “Retrieve the publications about proteins that are related to ‘lymphocytes and apoptosis’, and that are encoded by a gene”, which is expressed as follows:

\[ Q_3 = \text{Gene.codes.for.} \text{Protein[Object}_\text{content} \text{contains} \{ \text{lymphocytes,apoptosis} \} \text{.is_related.to.Publication} \]

Suppose that a scientist consider the source graph in Figure 3 to evaluate the query.

The scientist may select OMIM, which contains information related to genes as the origin source of the exploration. From OMIM there are two links L1 or the link L2, both implementing the logical relationship codes_for between Gen and Protein. L1 and L2 respectively reach Swiss-Prot and NCBI Protein. Both data sources contain information about proteins. Finally, the links L3 or L4 connect respectively Swiss-Prot and NCBI Protein to PubMed.

According to the different navigational choices, the scientist could traverse any the following source paths:

- \( SP_1 = \text{OMIM.L1.Swiss-Prot.L3.PubMed} \)
- \( SP_2 = \text{OMIM.L2.NCBIProtein.L4.PubMed} \)

As seen in Examples 2 and 3, implementations of a scientific concept may differ from source to source, and the properties of the retrieved object datasets may vary. The search space constituted of all source paths for a given query may be large, users may be interested in a recommendation of which paths may provide better answers to their query.

For example, consider source paths \( SP_1 \) and \( SP_2 \), and suppose that a user prefers the protein objects provided by Swiss-Prot instead of the protein objects provided by NCBI Protein. Then, based on this information, one could estimate the relevance of both paths, and rank them.

In this section, we describe the problem of estimating the relevance of a source path with respect to a query. A source path \( SP \) is a succession of one or more single paths. A single path is the succession of a source \( S_i \), a link \( L_{ij} \), and a source \( S_j \).

Definition 5: Given a life science model \( LS \), a query \( Q \), and a source path \( SP = S_1 L_{12} S_2 ... S_m \), the annotation of \( SP \) with respect to \( Q \) is a pair \((LC,R)\), where:

\[ LC = \bigcirc_{i=1}^{m-1} (m_S(S_i).1,m_{LS}(L_{i+1}).1,m_S(S_{i+1}).1) \]

where \( \bigcirc \) denotes the concatenation of the logical classes and relationships implemented by the sources and links in \( SP \).

\[ R = 1/n \times \sum_{i=1}^{n-1} (m_S(S_i).2,m_{LS}(L_{i+1}).2,m_S(S_{i+1}).2) \]

i.e., the sum of the scores of the logical classes and relationships implemented by the sources and links in \( SP \). If \( m_S(S_i).2 \) and \( m_{LS}(L_{i+1}).2 \) model a user preference for the data in source \( S_i \) and link \( L_{i+1} \), respectively, then, the expression \( R(S) \) models the relevance of the
answers provided by the evaluation of the path \( SP \). We call this metric relevance.

Using the expression \( LC \), a source path can be defined as follows:

**Definition 6:** Given a query \( Q \) represented by the regular expression \( RE \), a source path \( SP \) for \( Q \) is a finite sequence of a source \( S_i \), a link \( L_{i,j} \), and a source \( S_j \), such that: a) two sources are not connected by the same link twice in the path, and b) \( LC(SP) \) corresponds to an expression in the language induced by the regular expression \( RE \).

The metric relevance estimates the quality of the objects reached by the source paths that implement a particular query. We note that the quality of a query answer may be measured by many other metrics. For example, possible metrics include:

- **path cardinality:** estimates the number of objects paths are explored while the source path is traversed of the query.
- **target object cardinality:** estimates the number of target objects reached following the source path.
- **evaluation cost:** estimates the time needed to evaluate the source path.
- **attribute cardinality:** estimates the number of attributes or properties reached following the source path.

In this paper, we only illustrate the metric relevance (see [5], for more details on other metrics).

**Example 4:** Suppose the following user preferences for the data published by sources and links, is provided to our data model:

<table>
<thead>
<tr>
<th>Concept</th>
<th>Resource</th>
<th>Preference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gene</td>
<td>OMIM</td>
<td>1.0</td>
</tr>
<tr>
<td>Protein</td>
<td>NCBI GeneCards</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>NCBI Gene</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>NCBI Protein</td>
<td>0.7</td>
</tr>
<tr>
<td></td>
<td>Swiss-Prot</td>
<td>0.9</td>
</tr>
<tr>
<td>codes_for</td>
<td>( L_1 )</td>
<td>0.9</td>
</tr>
<tr>
<td></td>
<td>( L_2 )</td>
<td>0.7</td>
</tr>
<tr>
<td>is_related_to</td>
<td>( L_3 )</td>
<td>0.8</td>
</tr>
<tr>
<td></td>
<td>( L_4 )</td>
<td>0.8</td>
</tr>
<tr>
<td>Disease</td>
<td>OMIM</td>
<td>1.0</td>
</tr>
<tr>
<td>Publication</td>
<td>NCBI PubMed</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Then, scores that measure relevance of source paths \( SP_1 \) and \( SP_2 \) could be as follows:

- \( R(SP_1) = (1.0 + 0.9 + 0.9 + 0.8 + 1.0)/3 = 1.53 \)
- \( R(SP_2) = (1.0 + 0.7 + 0.7 + 0.8 + 1.0)/3 = 1.4 \)

**The Source Path Annotation problem:** Given a life science data model \( LS \) and a query \( Q \) over classes in \( C \) and links in \( L_C \), the source path annotation problem is finding the set of annotated source paths \( SP \) that implement \( Q \) in \( LS \). Each path in \( SP \) is annotated with a pair \( (LC,R) \), where \( LC \) corresponds to a sequence of logical classes and relationships, and \( R \) is a value that measures how relevant is the source path \( SP \) for \( Q \).

**A Naive solution for the Source Path Annotation problem:** In previous research[4, 5], we presented the algorithm ESearch that performs a breath first strategy, and traverses the space of paths in a source graph enumerating the execution paths of a query. Using ESearch, a naive solution for the source path annotation problem can be performed in two steps: a) produce all the source paths that implement a query, and b) compute the pair \( (LC,R) \) that annotates each source path. This naive approach requires an exhaustive search of the source paths in the source graph, a sequential scan of the complete set of paths to compute the metric values, and the ranking of the complete set of paths.

The ESearch algorithm runs in polynomial time in the size of the graph, if the graph is cycle-free and all paths are cycle-free. If \( d \) is the maximum number of sources that can precede a source in the \( SG \), and \( b \) is the maximum length of (cycle free) paths satisfying the regular expression, then \( O(d^b) \) is an upper bound for ESearch. For general graphs, the number of source paths can be exponential on the size of the graph, and a complete probing at query time will be need to rank the paths.

Because the search space of source paths may be large, an exhaustive enumeration of the whole set of paths may be very unefficient, thus unacceptable. In the future, we plan to implement search techniques that allow to identify source paths with highest relevance scores, without having to traverse the whole of space of possibilities.

**IV. Conclusion and Future Work**

In this paper we have defined a life science data model comprised of three levels: the ontology level, the physical level and the data level. Each level is represented by an ontology that reflects the semantics of the concepts modeled. Mappings allow to represent implementations of scientific concepts and relationships among them. Scores are attached to these mappings. They are used to specify the quality of the implementation, and to compute metrics’ values that measure the characteristics of the objects that can be retrieved during query evaluation. The current formalization does not involve reasoning services yet. In the future, we plan to integrate inference engines to our current query engine, to infer new concepts, sources or objects, in terms of the properties specified in the different ontologies that compose our data model.

**Acknowledgements** The research that motivated this paper was partially supported by NSF grants IIS0222847, IIS0223042, IIS0431174, and IIS0430915, and by the grant 1R03 LM008046 from the NIH National Library of Medicine.

**References**
