BIOINFORMATICS

Vol. 19 no. 0 2003, pages 1–8
DOI: 10.1093/bioinformatics/btg340

SEMEDA: ontology based semantic integration of biological databases

Jacob Köhler[1,*], Stephan Philippi[1] and Matthias Lange[2]

[1]AG Bioinformatics, Technical Faculty, Bielefeld University, Germany and
[2]AG Cytogenetics/Transcriptome Analysis, IPK-Gatersleben, Germany

Received on December 9, 2002; revised on April 7, 2003; accepted on June 9, 2003

ABSTRACT
Motivation: Many molecular biological databases are implemented on relational Database Management Systems, which provide standard interfaces like JDBC and ODBC for data and metadata exchange. By using these interfaces, many technical problems of database integration vanish and issues related to semantics remain, e.g. the use of different terms for the same things, different names for equivalent database attributes and missing links between relevant entries in different databases.
Results: In this publication, principles and methods that were used to implement SEMEDA (Semantic Meta Database) are described. Database owners can use SEMEDA to provide semantically integrated access to their databases as well as to collaboratively edit and maintain ontologies and controlled vocabularies. Biologists can use SEMEDA to query the integrated databases in real time without having to know the structure or any technical details of the underlying databases.
Availability: SEMEDA is available at http://www-bm.ipk-gatersleben.de/semeda/. Database providers who intend to grant access to their databases via SEMEDA are encouraged to contact the authors.
Contact: jacob.koehler@uni-bielefeld.de

INTRODUCTION

Searching and integrating data from various sources is often a prerequisite for research in the field of Molecular Biology and Bioinformatics. Due to the obvious importance of data integration for the life science community (Stevens *et al.*, 2001), a plethora of approaches for the integration of molecular biological databases exists (Davidson *et al.*, 1995; Freier *et al.*, 2002; Jakobovits, 1997; Karp, 1995). Due to the fact that most 'databases' were started as flatfiles, the most common approach to database integration is based on indexed flatfiles, e.g. DBGET/LinkDB (Fujibuchi *et al.*, 1998), SRS (Etzold *et al.*, 1996) and SIR (Ramu, 2001). At present, most molecular biological databases are implemented on relational database management systems (RDBMS) that provide standard interfaces like JDBC and ODBC for data and metadata exchange. By using these interfaces, many technical problems of database integration can be overcome, leaving mainly semantic issues as described in Karp (1995); Kim and Seo (1991); Williams (1997). As these problems still challenge current approaches to database integration, they are briefly summarized in the following:

(1) The fact that different databases often use different words for the same things results in problems that can be overcome by using either controlled vocabularies or ontologies like the Gene Ontology (Gene-Ontology-Consortium, 2001), EC numbers (NC-IUBMB, 1992), CAS Registry (Buntrock, 2001) etc. However, there is no systematic method to define which database uses which controlled vocabulary. Therefore, often 'uncontrolled vocabularies' and different controlled vocabularies are used across databases.
(2) Attribute names are often not self-explanatory or misleading and equivalent attributes have different names in different databases. Whereas one database might for example use the attribute name 'ec_nr', another database might use 'id' for an attribute, which also contains EC numbers. Out of this, attributes cannot be easily mapped between different databases.
(3) Querying databases often requires knowledge about the content of its tables, e.g. if a table only contains data about one species or one enzyme group. About which mouse species does the mouse genome database 'http://www.informatics.jax.org/' contain data? The database schema does not contain an attribute 'organism'. Unless the user is a biologist who knows that mouse experiments are generally done with special strains of *Mus Musculus*, it is impossible to find species information via the database query forms on the web page. This is no problem, as long as the interface of the source database is used. As systems for database integration have their own query interface, tables as described above have to be semantically refined with additional information.
(4) Due to the lack of a systematic linking mechanism between databases, even up to date integration systems

*To whom correspondence should be addressed.

such as SRS and KEGG (Kanehisa *et al.*, 2002) only link the 'most important' attributes. This is due to the fact that the number of existing molecular biological databases is too high to survey. Therefore, compared to the fact that at present more than 400 molecular biological databases exist (Baxevanis, 2003), the degree of interlinking is low (Williams, 1997).

Starting from this perspective, the next section of this article gives some basic definitions as a prerequisite for further discussions. Afterwards, concepts suitable to solve most of the aforementioned problems (2–4) of semantic database heterogeneity are introduced. With regard to (1), concepts are presented that can be applied when different databases use different controlled vocabularies. To demonstrate the practical use of the presented ideas, the implementation of these concepts in Semantic Meta Database (SEMEDA) is described. SEMEDA is then compared to existing systems for the integration of biological databases. Finally, a discussion of the results concludes the presentation.

DATABASE METADATA, CONTROLLED VOCABULARIES AND ONTOLOGIES

Database metadata is data about a database, which describes the logical structure and other relevant information about a data source. The term metadata will be used here to mean database schema information, data about the DBMS, and relevant technical data required to access a data source. Database metadata does not include data entries of the source databases. The schema of relational databases consists of datatypes (domains) and tables (relations). Tables consist of attributes (fields) with an associated datatype as domain, and may contain data within the limits of these domains (Date, 1982).

For the scope of this publication, *Controlled vocabularies* are named lists of terms that are well defined and may have an identifier. The elements of a controlled vocabulary are called concepts. Concepts can either be defined implicitly or by explicitly listing them. The terms or identifiers of controlled vocabularies are often used as database entries.

DEFINITION:

Controlled Vocabulary CV:= named set of concepts c, with c:= (term, definition, identifier, synonyms).

EXAMPLE:

An example for a controlled vocabulary is the Enzyme Nomenclature (NC-IUBMB, 1992). Each concept (enzyme) has a term (recommended name), an identifier (EC number) and synonyms (systematic name, other names). The definition of an enzyme is given by references to literature, which describe the enzyme in more detail.

As opposed to a controlled vocabulary, an *ontology* consists of concepts which are linked by directed edges, thus forming a graph. The edges of an ontology specify in which way, e.g. 'is-a' or 'part of', concepts are related to each other.

According to Gruber 'an *ontology* is a specification of a conceptualization' (Gruber, 1993). Although most *ontologies* have a few core items in common (Stumme and Maedche, 2001), the notion about *ontologies*, their formal notation and how they should be implemented varies between people and research groups (Schulze-Kremer, 1997). Therefore, for the scope of this publication, precise definitions of these terms are provided in the following. Whereas *ontologies* based on description logics (Horrocks *et al.*, 2002) are more expressive, for the work presented in this publication a simple to use and understand *ontology* was sufficient and therefore preferred. This ontology definition is very similar to the prevalent RDF standard (http://www.w3.org/RDF/). Even though only a transitive 'is-a' hierarchy is required for querying databases, SEMEDAs *ontology* editor also supports editing of other relation types with other algebraic relational properties.

DEFINITION:

Ontology O := G(CV, E), with $E \subseteq CV \times CV$ and a totally defined function $t\colon E \rightarrow T$ which defines the types of the edges. T is the set of possible edge types, i.e. the semantics of an edge in natural language and its algebraic relational properties (transitivity, symmetry and reflexivity). All *ontologies* have an edge type 'is-a' $\in$ T. If two concepts c1, c2 $\in$ CV are connected by an edge of this type, the natural language meaning is 'c1 is a c2'.

EXAMPLE:

In Figure 1, the concepts vertebrate, animal and organism are connected by transitive 'is-a' relations, i.e. vertebrate 'is-a' animal and animal 'is-a' organism. The transitive 'is-a' relations can then be used to derive the fact that vertebrate 'is-a' organism. Examples for *ontologies* are the Gene Ontology, UMLS (Srinivasan, 1999), EcoCyc/MetaCyc (Karp *et al.*, 2002) and WordNet (Fellbaum, 1998).

As can be seen from the definition of *ontologies* and *controlled vocabularies*, *ontologies* can be reduced to controlled vocabularies simply by dropping information. This 'reducibility' can be used when ontology editors and browsers are developed: a database schema, which can store *ontologies*, can also store *controlled vocabularies*.

SEMANTIC DATABASE DEFINITIONS

How can databases be semantically defined using ontologies and controlled vocabularies? The main idea is to map tables and attributes of a database to a given ontology. This ontology should be formal with respect to the implementation of a transitive 'is-a' hierarchy, which connects all concepts. Although other hierarchies can be defined, only the 'is-a' hierarchy is used for query processing. In the following, principles of semantic database integration are defined and examples are

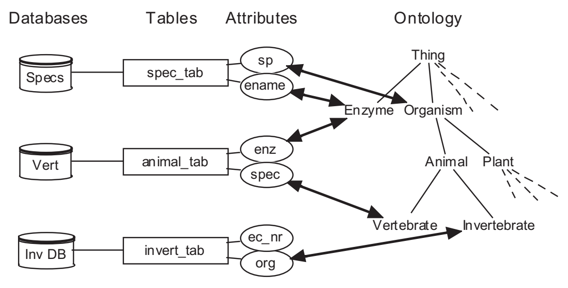


Fig. 1. Database attributes can be defined by linking them to ontology concepts (thick arrows). 'ename' and 'enz' are defined in this example as the same concept 'Enzyme', i.e. they both contain enzyme names. 'org' contains only invertebrates, whereas 'sp' contains both animal and plant.

given to explain how semantic definitions can be used to overcome semantic heterogeneity. The example queries use the syntax 'a:t' which has the meaning to search in the database attribute 'a' for the term 't'. Such queries are sufficient to illustrate the principles of semantic database queries, although the query interface of SEMEDA is implemented in a more user friendly and powerful way.

It is important to notice, that in the following, the terms 'attribute' and 'table' refer to database attributes and tables of the databases that are integrated, and not of any other elements that are superimposed on the original database schema.

Attribute semantics

Database attributes can be semantically defined by linking them to one or several concepts of an ontology. In consequence, attributes cannot only be addressed by a mapped concept *c*, but also by the sub-and superconcepts of *c*.

DEFINITION:

A *semantic attribute definition* is a tuple (A, c) where A is an attribute and c ∈ O is a concept of the ontology O.

EXAMPLE QUERY:

'animal:mouse'—search in the attribute 'animal' for 'mouse'. In Figure 1, no database attribute is defined as animal. However, some sub- and superconcepts of animal are mapped to database attributes and should therefore be searched for 'mouse'. For this query, the fact cannot be derived that 'invertebrate' is unlikely to contain mouse data, thus both 'invertebrate' and 'vertebrate' should be searched. 'Plant' will not be searched because it is not a sub- or superconcept of animal, and also because no attribute is defined as 'Plant'. If the user query is more specific, for example 'vertebrate:mouse', the irrelevant database attribute 'invertebrate' would not need to be searched. Since database attributes should be semantically defined to be as specific as possible, no database attributes are semantically defined as general top-level concepts such as 'thing'.

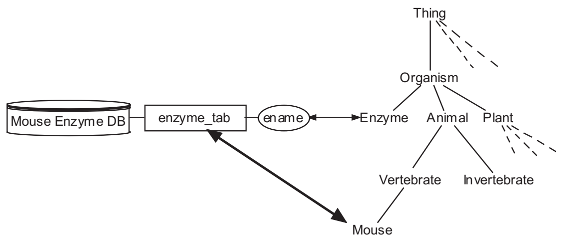


Fig. 2. The content of a database table can be refined by linking the table to a concept of the ontology. The database table 'enzyme tab' only contains mouse data in this example, but does not have an attribute 'species'. For database integration purposes the table has therefore to be refined in order to indicate that it contains mouse data.

Table semantics

The content of a database table can be *refined* by linking the table to a one or more concepts of the ontology. By r*efining* database tables, *all* entries of the database table are semantically annotated the same way. Principles for semantically annotating individual entries are purposely not suggested, since the manual annotation of thousands of individual database entries would make the integration process time consuming and inflexible.

DEFINITION:

A *semantic table definition* is a tuple (T, c) where T is a database table and c ∈ O is a concept of the ontology O.

EXAMPLE QUERY:

'animal:mouse' (Fig. 2). An imaginary 'mouse enzyme database' only contains information about mice; therefore no attribute for species exists. The database table 'enzyme_tab' is *refined* with a semantic table definition to contain mouse data. Refining a table this way is similar to adding an attribute to a table, which has the same value for each row.

Attribute value semantics

The semantics of attribute values (all entries of an attribute of a database), can be defined by using controlled vocabularies as datatypes for attributes, which is actually done in many existing databases.

DEFINITION:

If the datatype D(A) of an attribute A is a controlled vocabulary CV, i.e. D(A) = CV, the controlled vocabulary semantically defines the values of A.

EXAMPLE:

Searching a database attribute 'organism' with the controlled vocabulary 'systematic species name' as datatype for 'Mus Musculus' makes sense, since 'Mus Musculus' is a systematic

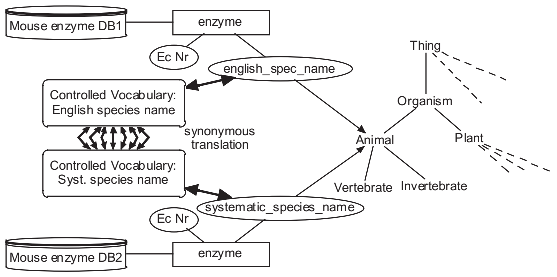


Fig. 3. By mapping synonymous concepts of controlled vocabularies, it is possible to relate database entries that use different terms for the same things.

species name. Searching this attribute for 'mouse' makes no sense, because 'mouse' is not a 'systematic species name'. By listing synonymous concepts between controlled vocabularies in a *translation list*, it is possible to relate entries between databases that use different terms for the same things.

DEFINITION:

Translation List := $\{(c1, c2) \mid c1 \in CV1, c2 \in CV2$ and $c1$ is a synonym of $c2\}$.

EXAMPLE:

Without a translation list, a search in both databases in Figure 3 would only find 'mouse' in the attribute 'english_spec_name' of DB1. By using the translation list it is possible to translate 'mouse' to 'Mus musculus', and in consequence also search the attribute 'systematic_spec_name'.

With this approach, controlled vocabularies do not need a synonym counterpart in the underlying ontology.

Two possibilities to set-up translation lists exist: (i) translation of all synonymous controlled vocabularies via one central controlled vocabulary, (ii) pairwise translation between all synonymous controlled vocabularies for which translation lists exist. Whereas the first approach is conceptually better, it has the disadvantage that it requires that the central controlled vocabulary contains a synonymous concept for each concept of all other controlled vocabularies. At present no such comprehensive centralized controlled vocabulary exist, and to build and maintain such a centralized terminology service out of several pre-existing controlled vocabularies is a very demanding time-consuming task by itself (Ingenerf *et al.*, 2001). The second alternative is more pragmatic, since it allows the reuse of pre-existing translation lists which are often provided and maintained by database providers or other authorities, e.g. CAS Registry numbers versus EC numbers, the Gene Ontology versus SWISS-PROT, EC numbers versus InterPro, etc.

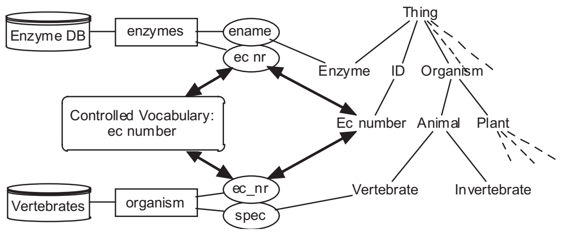


Fig. 4. Database attributes which are defined as the same concept and share the same controlled vocabulary as their domain can be used for cross-referencing between database attributes.

Database links and cross-references

The previous definitions can be used to derive *cross-references* for an attribute A, i.e. the set of all attributes which share the same *semantic attribute definition* and use the same controlled vocabulary (*attribute values definition*). It is important to note that these cross-references explicitly include attributes which are mapped to sub- and superconcepts of A. Cross-references can be used to automatically generate database links as well as to automatically derive which database tables can be joined.

EXAMPLE:

In Figure 4, the query 'animal:mouse' would find the EC number of mouse enzymes. By using semantic cross-references, a system can automatically generate links to other database tables that contain further information about EC numbers. In the example, additional information can therefore be found in the 'enzymes' database.

DESIGN AND IMPLEMENTATION

To demonstrate the potential of the principles of semantic database integration as described above, SEMEDA was developed. Well-established features, which already exist in other systems, like multi-database views, integration of bioinformatic analysis tools/applications and the download of results in different data formats, have not (yet) been integrated. Systems which already provide these functions are for example OPM/MQS (Topaloglou *et al.*, 1999), IBMs DiscoveryLink Haas *et al.* (2001) and SRS.

SEMEDA was developed as a 3-tiered system. It consists of a relational database backend (Oracle 8i or newer) to store ontologies, database metadata and semantic database definitions. Java Server Pages are used in the middle tier to dynamically generate the HTML frontend.

The databases to be integrated into SEMEDA are accessed directly via JDBC. For homogeneous SQL access to heterogeneous data sources which cannot be directly accessed via JDBC, like flatfiles and web pages, an external module, the BioDataServer (Freier *et al.*, 2002) is used. Alternatively, other systems for structured access to heterogeneous data sources could be used, for example IBMs DiscoveryLink and DiscoveryHub from geneticXchange.

The principles of semantic database integration as introduced in the preceding sections often require finding all sub- or superconcepts of a given concept. For this purpose, Oracles proprietary 'CONNECT BY PRIOR' SQL extension enables for efficient tree-processing in a declarative programming style. More details on SEMEDA's Architecture are given in Köhler *et al*. (2000).

SEMEDA's architecture allows handling of large controlled vocabularies and ontologies with a virtually unlimited number of concepts. The Gene Ontology was successfully imported to demonstrate SEMEDA's potential to collaboratively edit and browse large ontologies.

Despite of these capabilities, SEMEDA uses its own custom ontology for database integration. This ontology is a small top-level ontology, which defines databases at the schema level. In contrast to this approach, ontologies like the Gene Ontology generally serve as controlled vocabularies that are used to unify data entries between different databases. Out of this, existing ontologies and SEMEDA's ontology serve different purposes and therefore can complement one another.

The idea behind the top level structure of SEMEDA's ontology is that attributes in molecular biological databases generally store *names*, *identifiers*, *properties* and *free text descriptions* of real world objects. Names and identifiers serve to identify real world objects, whereas properties and descriptions store facts about those objects. Out of this, 'name', 'identifier', 'description' and 'property' are used as top-level concepts in SEMEDA's ontology.

Users can computationally access ontologies and database metadata of SEMEDA by directly connecting to SEMEDAs relational backend. Only confidential database metadata of the underlying data sources such as host, port, login etc. is hidden. The connection details are available on request from the authors.

To access the features of SEMEDA, three graphical user interfaces exist. These interfaces are described in more detail in the following with special emphasize on the query interface.

Meta DB: The meta database interface is used to collaboratively browse and edit database metadata, ontologies and semantic database definitions as introduced in the previous section. In order to make a database table accessible via SEMEDA, at minimum one attribute per database table has to be semantically defined. This reduces the work of adding or removing databases significantly.

To provide controlled access via the meta database interface of SEMEDA, different user groups with different permissions exist: 'Everybody' is allowed to browse all data, but confidential database metadata is hidden. 'Database providers' are allowed to semantically define databases and to suggest new concepts to available ontologies. 'Administrative accounts' have full permissions on all meta data aspects.

Admin tools: The administrative tools interface is used to perform various administrative tasks. In detail, database metadata can be imported, database schemata for the BioDataServer can be generated/submitted, and newly suggested ontology concepts can be released by administrative accounts.

Query DBs: The query interface is implemented in a way that the ontology does not hide the source databases, but rather guides the user to the relevant database tables/attributes and supports the construction of appropriate queries. If a user enters the query interface (see Fig. 5), a list of ontology concepts is displayed. On the right-hand side of each concept a set of coloured icons is shown. Each icon represents a database table with an attribute mapped to the concept. A mouse over ToolTip displays for each icon the names of the source database, the database table and the attribute. After selecting one of these icons, a query form for the respective database table may be used to query the database. This query form is automatically generated based on the database metadata stored in SEMEDA. The query form displays all attributes of the selected table and allows the user to choose the attributes, which have to be included in the result. The database attributes which are semantically defined (see section *Attribute Semantics*) are labelled using the names of ontology concepts rather than the often misleading attribute names of the source databases. Mouse over ToolTips for the attribute labels display further information about a database attribute, such as its datatype (see *Attribute Value Semantics*). If a semantic table definition exists for a table (see section *Table Semantics*), it is also displayed.

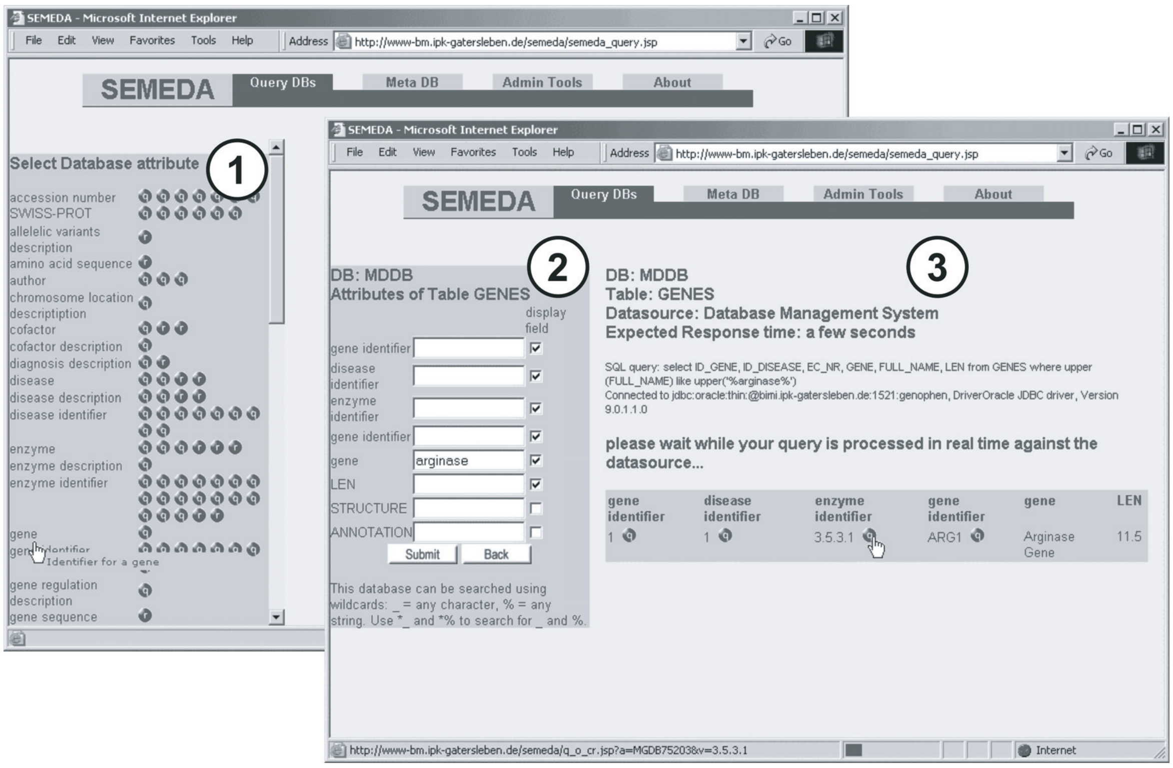
In those databases, which are accessed directly via JDBC, all attributes are searchable using pattern matching. Heterogeneous data sources are queried indirectly using the BioDataServer, which does not yet support pattern matching.

With this mechanism the user is able to browse across several databases without having to know the databases, their schemata or a database query language. The system is transparent in a way that it always displays from which source the data was retrieved. This is important, since most users would not trust a system, which retrieves data without reporting their particular origin.

COMPARISON WITH OTHER SYSTEMS

In the following, SEMEDA's methods for semantic database integration are compared to existing systems in this area. Since the motivation to develop SEMEDA was to demonstrate the practicality of the introduced principles for semantic database integration, features beyond this scope are not discussed.

In most database integration systems, attribute semantics can be defined by assigning the same names to equivalent database attributes. A drawback of this approach is that it does not systematically cover the situation where one database attribute is more general than another. An example would be an attribute 'organism' that stores species names


SEMEDA - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://www-bm.ipk-gatersleben.de/semeda/semeda_query.jsp
Go
SEMEDA
Query DBs
Meta DB
Admin Tools
About
1
2
3
Submit
Back
http://www-bm.ipk-gatersleben.de/semeda/q_o_cr.jsp?a=MGDB75203&v=3.5.3.1
Internet

Select Database attribute

accession number
SWISS-PROT
allelelic variants description
amino acid sequence
author
chromosome location descriptiption
cofactor
cofactor description
diagnosis description
disease
disease description
disease identifier
enzyme
enzyme description
enzyme identifier
gene
gene identifier
Identifier for a gene
gene regulation description
gene sequence

DB: MDDB
Attributes of Table GENES

		display field
gene identifier		✓
disease identifier		✓
enzyme identifier		✓
gene identifier		✓
gene	arginase	✓
LEN		✓
STRUCTURE		
ANNOTATION		

This database can be searched using wildcards: _ = any character, % = any string. Use *_ and *% to search for _ and %.

DB: MDDB
Table: GENES
Datasource: Database Management System
Expected Response time: a few seconds

SQL query: select ID_GENE, ID_DISEASE, EC_NR, GENE, FULL_NAME, LEN from GENES where upper (FULL_NAME) like upper('%arginase%')
Connected to jdbc:oracle:thin:@bimi.ipk-gatersleben.de:1521:genophen, DriverOracle JDBC driver, Version 9.0.1.1.0

please wait while your query is processed in real time against the datasource...

gene identifier	disease identifier	enzyme identifier	gene identifier	gene	LEN
1	1	3.5.3.1	ARG1	Arginase Gene	11.5

Fig. 5. SEMEDA's database query interface. (1) All concepts for which database attributes exist are listed. Each of the round icons represents a database table that contains an attribute for the concept. (2) After selecting one of the icons an appropriate form for the respective table may be used to query the database. (3) The result set is extended with icons that are used to cross-reference other databases.

of plants, animals and fungae versus an attribute 'animal' that stores only names of animals. Most systems also do not differentiate between different vocabularies used in equivalent attributes of different databases. One database might for example use English species names for an attribute 'organism', whereas another database might use systematic species names. In SEMEDA, the vocabularies used can be defined. The concepts described in section *Attribute Value Semantics* can be used to translate between different vocabularies. Furthermore, SEMEDA's link generation capability is unique: in some systems like SRS and OPM/MQS pairwise cross-references to all relevant databases have to be provided manually. In contrast, once a database is semantically defined in SEMEDA against the ontology, all relevant database links and cross-references can be derived automatically. If n is the number of database attributes to be used for linking between databases, the number of integration steps (i.e. database links to be defined) in SRS and OPM/MQS greatly increases with the number of attributes to be interlinked [integration steps $= n^*(n-1)/2$]. In SEMEDA it is sufficient to define each database attribute only once against the ontology (integration steps $= n$). A system that has similar link generation capabilities like SEMEDA is KEGG's DBGET/LinkDB (Fujibuchi *et al.*, 1998). DBGET/LinkDB automatically generates links between databases using EC numbers and EMBL/Genebank accession numbers, but unlike SEMEDA it cannot use other database attributes for link generation.

Besides SEMEDA, other ontology based systems for semantic database integration are TAMBIS (Stevens *et al.*, 2000) and the F-LOGIC based system described in (Ludäscher *et al.*, 2001). These three systems have in common that they use ontologies to support users at querying data sources, and that they use a mediator/wrapper architecture for accessing the data sources.

In the following, the capabilities of these three systems will be compared. Whereas TAMBIS and the F-LOGIC based system both use description logic to model the ontology and to query data sources, SEMEDA mainly makes use of the transitivity of the 'is-a' hierarchy to guide the user to appropriate database tables. From there on, SQL via JDBC is used. Therefore in the two other systems, database queries potentially have the expressiveness of the first order predicate calculus, although TAMBIS uses the less expressive GRAIL as a description logic language (Peim *et al.*, 2002).

In the two description logic based systems, the fact that the underlying data sources are hidden from the users has potential disadvantages which SEMEDA does not have: (i) in many application scenarios, users need to know the origin of the data retrieved by a database query, which is not the case in the description logic based systems. (ii) When a new database is connected to the system, all database attributes have to be semantically defined to be available via the integration system. (iii) Due to the declarative nature of logic based queries, users cannot know in advance against which data sources a query is processed. This may easily lead to queries that require set operations (such as union, intersect, disjunction) between several databases. This may not be a problem as long as the databases being integrated are reasonably small and the costs for network traffic are negligible. For real-time access to databases with large numbers of entries that are accessed via the Internet, such query costs are generally prohibitive (Eckman *et al.*, 2001). (iv) In TAMBIS only one data source for one type of data can be connected to the integration system, for example either SwissProt or PIR as a protein source (Peim *et al.*, 2002).

Also, the query frontends of the three systems vary largely. To query the system described in Ludäscher *et al.* (2001), F-LOGIC is used. F-LOGIC is very expressive, but requires programming language skills. Therefore, form based HTML frontends are used on top of predefined F-LOGIC queries. However, these HTML frontends are highly specific to queries and data sources. The TAMBIS system uses a graphical representation of the underlying ontology to interactively build database queries. Although this frontend is powerful, building queries is not trivial and may be too difficult for medical doctors, e.g. in the GALEN project (Rector *et al.*, 1998), it was learned that medical doctors had difficulties understanding the data structure of ontologies. Therefore, in SEMEDA the user is not confronted with the data structure of the underlying ontology, and generic frontends are automatically generated once a database has been semantically defined.

DISCUSSION

SEMEDA supports querying databases via an easy to use, yet powerful interface. This interface enables users to query databases without requiring knowledge of the structure or any technical details about the data sources. In addition, it guides users to relevant databases, and provides cross-references from query results to other relevant databases. This is especially useful when many databases are integrated. At present more than 400 molecular biological databases exist, i.e. more than a human can survey.

As of now (March 2003), mainly six databases can be accessed via SEMEDA's query interface: SWISS-PROT (Bairoch and Apweiler, 2000), OMIM (Hamosh *et al.*, 2002), RZPD (www.rzpd.de), EMP (Selkov *et al.*, 1996), MDDB (Freier *et al.*, 2000) and KEGG (Kanehisa *et al.*, 2002). Adding a new database to SEMEDA requires typically less than one day, depending on the size of the database and how many new concepts have to be added to the ontology. Removing a database from the system requires a few mouse clicks. All tasks required to connect databases to SEMEDA can be done at runtime using SEMEDA's frontend. Thus databases can be added or removed from the system without programming language skills.

Database providers who intend to grant access to their databases via SEMEDA are encouraged to contact the authors. SEMEDA can also be used to grant semantically integrated access to relational databases that do not have an own user interface. This is possible since SEMEDA allows generating frontends automatically based on database metadata, which can be imported via JDBC. Therefore, in order to make databases accessible via SEMEDA, database providers only have to grant JDBC access.

At present, due to security concerns, many database owners do not grant JDBC access to their databases. However, all major DBMS have mechanisms to restrict user rights. Usually it is possible to restrict read access to a few tables and to hide table attributes and tablerows by using views. In addition, JDBC type 3 drivers, which usually support encryption, have recently become available for all major DBMSs http://industry.java.sun.com/products/jdbc/drivers

Thus, by using the methods introduced in this publication, database integration systems can be built with up to date data exchange methods instead of flatfiles. These methods do not require the maintenance of one complex integrated schema, hence avoiding problems related to database schema integration. Compared to database integration solutions like SRS, the work required to add a database to SEMEDA is lower than the work required to add a database to an indexing system: scripts for parsing flatfiles are not needed and the main work for adding a new database to the integration system consists in the semantic definition of database attributes, tables and the vocabularies used.

Besides adding more databases to the system, future developments of SEMEDA include the implementation of multi-database views, provision of result sets in different data formats, extension of SEMEDA's database search capabilities and translation list support for controlled vocabularies. For more efficient access to data sources, multi-database connection pooling as well as appropriate data caching mechanisms will be implemented.

REFERENCES

Bairoch,A. and Apweiler,R. (2000) The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. *Nucleic Acids Res.*, **28**, 45–48.

Baxevanis,A.D. (2003) The molecular biology database collection: 2003 update. *Nucleic Acids Res.*, **31**, 1–12.

Buntrock,R.E. (2001) Chemical registries—in the fourth decade of service. *J. Chem. Inf. Comput. Sci.*, **41**, 259–263.

Date,C.J. (1982) A formal definition of the relational model. *SIGMOD Record*, **13**, 18–29.

Davidson,S.B., Overton,C. and Buneman,P. (1995) Challenges in integrating biological data sources. *J. Comput. Biol.*, **2**, 557–572.

Eckman,B., Lacroix,Z. and Raschid,L. (2001) Optimized seamless integration of biomolecular data. In *IEEE International Conference on Bioinformatics and Biomedical Engineering (BIBE '01)*. Bethesda, Maryland, pp. 23–32.

Etzold,T., Ulyanov,A. and Argos,P. (1996) SRS: information retrieval system for molecular biology data banks. *Methods Enzymol.*, **266**, 114–128.

Fellbaum,C. (1998) WordNet: an electronic lexical database. *Language, Speech, and Communication*. MIT Press, Cambridge, MA, pp. xxii, 423.

Freier,A., Hofestädt,R., Lange,M., Scholz,U. and Stephanik,A. (2002) BioDataServer: an SQL-based service for the online integration of life science data. *In Silico Biol.*, **2**.

Freier,A., Hofestädt,R., Lange,M., Scholz,U. and Töpel,T. (2000) MD-CAVE—the metabolic diseases database—a system for storing information about human inborn errors. In *Second International Conference on Bioinformatics of Genome Regulation and Structure (BRGS2000)*, vol. 1, pp. 66ff. ICG Novisibirsk.

Fujibuchi,W., Goto,S., Migimatsu,H., Uchiyama,I., Ogiwara,A., Akiyama,Y. and Kanehisa,M. (1998) DBGET/LinkDB: an integrated database retrieval system. *Pac. Symp. Biocomput.*, 683–694.

Gene-Ontology-Consortium (2001) Creating the gene ontology resource: design and implementation. *Genome Res.*, **11**, 1425–1433.

Gruber,T.R. (1993) A translation approach to portable ontology specifications. *Knowledge Acquisition*, **5**, 199–220.

Haas,L.M., Schwarz,P.M., Kodali,P., Kotlar,E., Rice,J.E. and Swope,W.C. (2001) DiscoveryLink: a system for integrated access to life sciences data sources. *IBM Syst. J.*, **40**, 489–511.

Hamosh,A., Scott,A.F., Amberger,J., Bocchini,C., Valle,D. and McKusick,V.A. (2002) Online Mendelian inheritance in man (OMIM), a knowledgebase of human genes and genetic disorders. *Nucleic Acids Res.*, **30**, 52–55.

Horrocks,I., Patel-Schneider,P.F. and van Harmelen,F. (2002) Reviewing the Design of DAML + OIL: an ontology language for the semantic web. In Dechter,R., Kearns,M. and Sutton,R. (eds) *Eighteenth National Conference on Artificial Intelligence, AAAI-2002*. Edmonton, Alberta, Canada, pp. 792–797.

Ingenerf,J., Reiner,J. and Seik,B. (2001) Standardized terminological services enabling semantic interoperability between distributed and heterogeneous systems. *Int. J. Med. Inf.*, **64**, 223–240.

Jakobovits,R. (1997) Integrating heterogeneous autonomous information sources, Univ. of Washington, p. 26.

Kanehisa,M., Goto,S., Kawashima,S. and Nakaya,A. (2002) The KEGG databases at GenomeNet. *Nucleic Acids Res.*, **30**, 42–46.

Karp,P.D. (1995) A strategy for database interoperation. *J. Comput. Biol.*, **2**, 573–586.

Karp,P.D., Riley,M., Paley,S.M. and Pellegrini-Toole,A. (2002) The metaCyc database. *Nucleic Acids Res.*, **30**, 59–61.

Kim,W. and Seo,J. (1991) Classifying schematic and data heterogeneity in multidatabase systems. *IEEE Computer*, **24**, 12–18.

Köhler,J., Lange,M., Hofestädt,R. and Schulze-Kremer,S. (2000) Logical and semantic database integration. In Young,D.C. (ed) *Bioinformatics and Biomedical Engineering*. Arlington, Virginia, USA, pp. 77–80.

Köhler,J. and Schulze-Kremer,S. (2002) The semantic metadatabase (SEMEDA): ontology based integration of federated molecular biological data sources. *In Silico Biol.*, **2**, 0021.

Ludäscher,B., Gupta,A. and Martone,M.E. (2001) Model-based mediation with domain maps. In *17th Intl. Conference on Data Engineering (ICDE)*. IEEE Computer Society, Heidelberg, Germany.

NC-IUBMB (1992) *Enzyme Nomenclature 1992 : Recommendations of the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology on the Nomenclature and Classification of Enzymes*. Published for the International Union of Biochemistry and Molecular Biology by Academic Press, San Diego, pp. xiii, 862.

Peim,M., Franconi,E., Paton,N.W. and Goble,C.A. (2002) Query processing with description logic ontologies over object-wrapped databases. In *14th International Conference on Scientific and Statistical Database Management (SSDBM'02)*. Edinburgh, Scotland, pp. 27–36.

Ramu,C. (2001) SIR: a simple indexing and retrieval system for biological flat file databases. *Bioinformatics*, **17**, 756–758.

Rector,A.L., Zanstra,P.E., Solomon,W.D., Rogers,J.E., Baud,R., Ceusters,W., Claassen,W., Kirby,J., Rodrigues,J.M., Mori,A.R. *et al.* (1998) Reconciling users' needs and formal requirements: issues in developing a reusable ontology for medicine. *IEEE Trans. Inf. Technol. Biomed.*, **2**, 229–242.

Schulze-Kremer,S. (1997) Adding semantics to genome databases: towards an ontology for molecular biology. *Ismb*, **5**, 272–275.

Selkov,E., Basmanova,S., Gaasterland,T., Goryanin,I., Gretchkin,Y., Maltsev,N., Nenashev,V., Overbeek,R., Panyushkina,E., Pronevitch,L. *et al.* (1996) The metabolic pathway collection from EMP: the enzymes and metabolic pathways database. *Nucleic Acids Res.*, **24**, 26–28.

Srinivasan,P. (1999) Exploring the UMLS: a rough sets based theoretical framework. *Proc. AMIA Symp.*, 156–160.

Stevens,R., Baker,P., Bechhofer,S., Ng,G., Jacoby,A., Paton,N.W., Goble,C.A. and Brass,A. (2000) TAMBIS: transparent access to multiple bioinformatics information sources. *Bioinformatics*, **16**, 184–185.

Stevens,R., Goble,C., Baker,P. and Brass,A. (2001) A classification of tasks in bioinformatics. *Bioinformatics*, **17**, 180–188.

Stumme,G. and Maedche,A. (2001) FCA-merge: a bottom-up approach for merging ontologies. In *International Joint Conference on Artificial Intelligence*. Seattle, Washington, USA, pp. 225–234.

Topaloglou,T., Kosky,A. and Markowitz,V. (1999) Seamless integration of biological applications within a database framework. *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, 272–281.

Williams,N. (1997) Bioinformatics: how to get databases talking the same language. *Science*, **275**, 301–302.