Designing a Global Information Resource for Molecular Biology (Short Paper)[1].

Ulf Leser

Technische Universität Berlin,
Fachbereich 13 - CIS
Einsteinufer 17, D- 10587 Berlin.
Email: leser@cs.tu-berlin.de.

Abstract. Research in molecular biology is continuously producing an immense amount of data, but this information is spread over numerous heterogeneous data repositories. Their integration into a federated information system would drastically reduce the time a biologist has to spend browsing different WWW sites or databases in search for a particular piece of information.

In this study we point out the specific problems that molecular biology is posing to data integration. We present our approach to cope with these problems. It is based on a mediator architecture and uses query correspondence assertions (QCA) to describe sources in a flexible yet expressive manner. QCAs both capture content and query capabilities of arbitrary data sources with respect to a federated schema. Based on such QCAs a mediator can answer queries against the federated schema by constructing semantically equivalent combinations of source queries.

1. Introduction

Since the start of the Human Genome Project in the mid 80s, the production of data in the area of molecular biology has grown exponentially and still continues to do so. Hundreds of laboratories world-wide contribute with genomic maps, nucleotide sequences, protein structures etc. Laboratories make their raw experimental data publicly available on WWW sites. Additionally, processed data is often submitted to public repositories.

Molecular biology is a highly competitive area. Researches rely on having access to the most actual information. Therefore, biologists regularly check many different sources to get an exhaustive and up-to-date set of data for their field of interest. Furthermore, many interesting questions require that data from different sources is integrated and analysed together. Both tasks, i.e. searching many sources and combining data from different sources, currently have to be carried out manually.

---

[1] This research was supported by the German Research Society, Berlin-Brandenburg Graduate School in Distributed Information Systems (DFG grant no. GRK 316).

In this study we consider the problem of designing a data integration system for molecular biology. We start with an analysis of requirements for and obstacles against such a system (section 2). In section 3 we sketch the general architecture of our approach which implements a virtual integration of heterogeneous data sources. The central part of this architecture is a powerful query planning mechanism. It is based on query correspondence assertions (QCAs), a declarative formalism for the specification of the content and query capabilities of data sources. In section 4 we define the notion of QCAs and describe how they are used to find query plans. Section 5 discusses related work and section 6 concludes.

2. Data Integration in Molecular Biology: Requirements and Obstacles

In this section we give requirements for a federated information system (FIS) for molecular biology and point out particular difficulties in this domain. As a prerequisite we postulate that all integrated sources will preserve their full autonomy, which means that the developer of a system for data integration must not expect special facilities at the sources that are not present anyway. We only consider data retrieval.

Requirements

We assume a typical user of our FIS to be a researcher in molecular biology who requires an integrated access to data about biological objects such as genes, chromosome maps, clones etc. This data needs to be as *up-to-date* as possible. The result of a query should be *comprehensive* in the sense that as many as possible relevant data sources are included in the search. Users do not want to bother with different interfaces, different schemas, different query mechanisms or different formats of query results.

We conclude that the DIS must offer transparency on many levels. First, it should offer a uniform interface for user queries, based on a single, federated schema and logically accessed with a single query language. To offer maximum actuality, a virtual approach is necessary. Given a query against the federated schema, the FIS needs to determine the set of data sources that can contribute to answer the query, generate the appropriate source-specific queries and execute them. The results need to be combined, which comprises the computation of inter-source joins and building the union about semantically equivalent results from different plans.

Obstacles

Data sources in molecular biology are heterogeneous in content, access method, and structure ([7]). The most popular access technique today is the WWW, where interfaces can vary greatly: from simple HTML or ASCII pages and tables, through different levels of WWW fill-out forms to full query language access provided through a text-entry field.

Conventional approaches to database integration, such as federated databases ([12]), expect the sources to be full-fledged database management systems. They assume the existence of an export schema in a standard data model and a conventional query language. These assumptions do not hold in our setting. Access is in general not granted through a query language. For instance, most form-based interfaces only allow canned queries build from conjunctive connections of conditions on some predefined attributes, and many sites provide their data only in static HTML tables. Furthermore, the result of a query is not a set of tuples, but a HTML document with predefined format.

Specific Problems in Molecular Biology

The present status of existing data sources in molecular biology adds further problems on top of those described above. For example, there are no standards in terminology, neither on the schema nor on the instance level. To alleviate this problem, databases typically keep synonym lists of object names. Another problem is that experiments often have only a limited degree of accuracy. Results can vary or contradict each other. For instance, two laboratories can produce different values for the length of a clone. As it is beyond the scope of the FIS to decide which is the true value, all different values have to be considered equally. Last but not least, molecular biology is rapidly evolving. This is reflected both in a high rate of data production as in frequent changes in the structure of sources. Keeping a stable federated schema in the presence of frequent schema evolution is difficult. We conclude that the mapping mechanism between the federated schema and the sources must be flexible enough to bridge many different types of heterogeneity.

3. A Wrapper - Mediator Architecture

Our architecture (Fig. 1) follows a wrapper-mediator paradigm ([13]). A user accesses data sources through mediators. A mediator combines the data relevant for its domain from different sources which it accesses using source-specific wrappers.

The task of a wrapper is to leverage the different interfaces of data sources[2]. Each wrapper offers a uniform interface for mediators and translates requests into source specific queries, method calls, HTTP requests etc. In our system the interface between a wrapper and the mediator is a, though possibly very restricted, relational query interface. Each wrapper has an export schema; however, in contrast to federated databases ([12]), the wrapper can answer only certain, pre-defined queries.

A mediator holds a federated schema which models all concepts of the intended domain, which does not necessarily contain each source schema completely. Given a user query the mediator tries to find a sequence of queries against wrappers that yield a correct answer to the query. To carry out this task, the mediator holds a description of each wrapper in form of a set of query correspondence assertions. Mediators can also use other mediators which then take the role of a wrapper (see Fig. 1 and [6]). In the following we assume the existence of only one mediator for simplicity.

---

[2] We do not treat wrappers in detail in this paper.

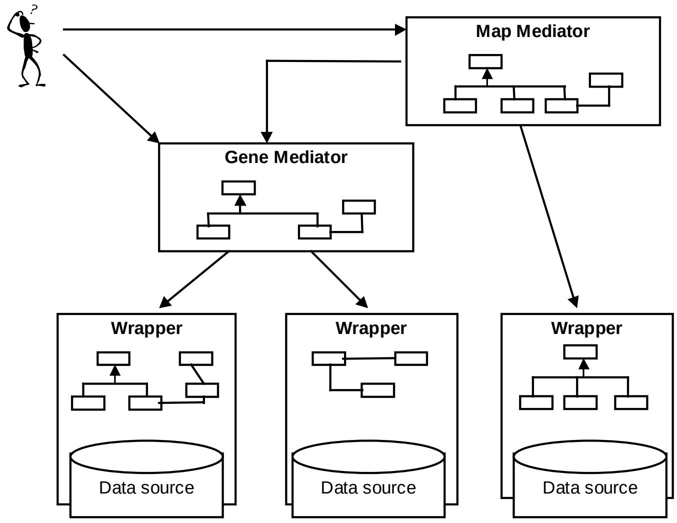

Fig. 1. A mediator-wrapper architecture for structured wrappers. Each mediator has its own federated schema, and mediators can use other mediators. Wrapper hide technical and data model heterogeneity; how they access their data sources is transparent for a mediator.

4. Query Answering based on Query Correspondence Assertions

Each mediator needs detailed knowledge about the content of each source, the relationships between their export and the federated schema and the possible queries. This is specified using QCAs.

4.1 Representing Data Sources through QCAs

A QCA has the following form:

`mediator query :=mediator view ⊇ source view := source query`

where the *mediator query* (MQ) is a conjunctive query against the mediator schema and a *source query* (SQ) is a query against the export schema of a wrapper which can be computed by the wrapper. All variables in the mediator view MV (source view SV) must also appear in MQ (SQ) for safety. SV and MV must have the same arity.
MQ and SQ can contain conditions in the form of equalities or arithmetic comparisons between variables and constants. A QCA is a rule about the relationship between the *extensions* of two views. It defines that the tuples resulting from executing SQ are also semantically meaningful tuples for MQ. Values are propagated from SQ to SV and MV into the mediator relations as defined in MQ.

As an example, consider the following federated schema: a 1:n relationship connects genes (G) and exons (E), where an exon has a sequence and a position (S,P). This is modeled by two relations: 'gene(G,E)' and 'exon(E,S,P)'. Now imagine a source which stores a report of all exon and intron sequences (distinguished by an attribute T) of genes. The wrapper exports this as a non-normalised table ‘genes(G,T,S)’. This is described through the following QCA:

```
gene(G,E),exon(E,S,P) =: v1(G,S) ⊇
                               s1.v1(G,S) := genes(G,T,S),T=’exon’;
```

This example shows some features of QCAs. Attribute positions and relation names can be different in the export and in the mediator schema. Not all attributes in the mediator schema get values through the SQ. In this case, attributes that do not form joins (like P) are padded with NULL, while for join-attributes (like E) an artificial value is created (to retain the link between the exon and gene). The combination of ‘gene’ and ‘exon’ in the mediator schema is semantically not identical to ‘genes’ in S1: only those tuples of ‘genes’ which fulfil the condition “T=’exon’” are meaningful inside in the mediator’s world. Such conditions can also appear in the MQ, indicating global properties of an export schema, and are used to prune irrelevant sources. We use the ‘⊇’ symbol to emphasise the fact that the (virtual) tuples in the mediator relations are a superset of the tuples of one source as there can be many sources for each value.

QCAs can compensate for a large number of conflicts, such as structural conflicts, attribute units and granularity, different names, synonyms and homonyms in schema etc. Schematic conflicts ([10]) are not expressible.

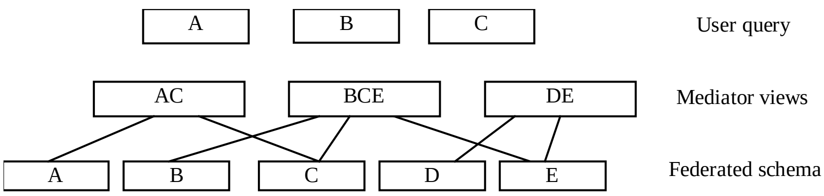

Fig. 2. A user query addresses the relation of a federated schema, although only views are accessible. The problem is therefore to find view combinations.

4.2 Using QCAs to generate Query Plans

The task of the mediator is to answer arbitrary user queries using a given set of QCAs. Queries which are, by incident, identical to a MQ of a QCA can be answered directly. But such a QCA will in general not exist. Hence, the mediator needs to find a combination of MQs (a plan) that is semantically equivalent to (or at least contained in) the user query. Once a plan is found, the answer can be computed easily by executing the corresponding SQs.

Note that every combination of MQs is a query against the federated schema. The solution to our problem is based on semantic query containment [1]. This technique allows to show whether the result of a query is contained in the result of another query for all possible instances of a database.

A feasible strategy is therefore to first generate combinations of MQs and then test for each combination if it is contained in the user query. We do not need to build all possible combinations. The problem of finding candidate combinations is illustrated in Fig.2, which draws on an analogy to the problem of answering queries using views ([8]). In our setting, the views are the MQs, as they describe the only queries a mediator can answer directly. Given a user query, the problem is to find a semantically equivalent combination of views. This is computable for conjunctive queries and conjunctive views, but NP-complete since view sequences can not be generated incrementally. Consider the following example:

```
User query (UQ): r1(X,Y),r2(Y,A),r3(Y,B);
r1(X,Y),r2(Y,A) =: v1(X,Y,A) ⊇ s1.r(X,Y,A) := r1(X,Y,A);
r1(X,Y),r3(Y,B) =: v2(X,Y,B) ⊇ s2.r(X,Y,B) := r2(X,Y,B);
```

It is easy to see that a greedy substitution algorithm fails: first using `v1` and thereby substituting `r1` and `r2` of the query makes `v2` inapplicable and vice versa. However, `<v1,v2>` is a semantic equivalent rewriting of UQ. This shows that in principle all possible combinations of MQs must be considered; fortunately, there is an upper bound on the length of these combinations ([8]).

This approach can be improved further. Query containment is shown by finding containment mappings between the variables and constants of the queries. We first compute all such mappings between each relation of the user query and each MQ which contains this relation. In a second step we construct all combinations of these mappings that cover the user query completely and test the compatibility of the mappings at the same time. If all mappings of a plan are compatible the plan is contained in the user query, and no additional test is necessary. A detailed description of the algorithm which also constructs plans of minimal length can be found in [5].

5. Related Work

There are a number of projects that aim at data integration in molecular biology. None of them uses a mediator approach as described in this work, and the majority is based on materialisation with the known problems of data actuality. SRS ([3]) offers comfortable keyword search mechanisms in a set of locally installed flat files but does not support structured queries or distributed sources. OPM ([2]) is a multidatabase query language without a uniform federated schema.

On the other hand, there are numerous research projects in computer science that are based on query mediation (see [4] for a survey). Our system was mainly influenced by the Information Manifold ([9]) which introduced the approach of 'answering queries using views' as a method for data integration. However, our planning algorithm is more advanced and can omit the expensive test for semantic equivalence. Furthermore QCAs are an extension of their capability records.

6. Conclusions

We have presented the design of a data integration system for molecular biology. Considering the requirements given in section 2, we believe that our approach fulfils most of them. It offers full schema, location and query language transparency, i.e. the users sees only one uniform interface to a globally distributed and heterogeneous information space. The data is as actual as the sources are. Query planning considers all possible sources for each type of information and hence obtains the most comprehensive answer possible. QCAs can be used to model a wide range of source interfaces, including WWW forms, native SQL access and CORBA object requests. An additional advantage of the design is good maintainability ([6]). Reacting on changes in source schemas can for instance in many cases be treated by changing one or more QCAs.

There are also drawbacks. The correctness and completeness of answers depends on the administrator who specifies QCAs; wrong or missing QCAs result in wrong or missing answers. Whether or not a query against the federated schema is answerable at all also depends on the available sources respectively QCAs. Administration of large sets of QCAs is an open problem. The computational complexity of the plan generation can not be ignored. Potentially, there are exponentially many plans which are at least partly enumerated and tested. We plan to apply heuristic criteria for selecting plans with higher priority, based on the completeness, actuality and update frequency of sources ([11]).

References

[1] Aho, A. V., Y. Sagiv, et al. (1979). “Equivalence among Relational Expressions.” SIAM Journal of Computing **8**(2): 218-246.

[2] Chen, I. A. and V. M. Markowitz (1995). “An Overview of the Object-Protocol Model (OPM) and OPM Data Management Tools.” Information Systems **20**(5): 393-418.

[3] Etzold, T., A. Ulyanov, et al. (1996). “SRS: Information Retrieval System for Molecular Biology Data Banks.” Methods in Enzymology **266**: 114-128.

[4] Hull, R. (1997). Managing Semantic Heterogeneity in Databases: A Theoretical Perspective. 16th ACM PODS.

[5] Leser, U. (1998). Combining Heterogeneous Data Sources through Query Correspondence Assertions. 1st Workshop on Web Information and Data Management, Washington, D.C.

[6] Leser, U. (1998). Maintenance and Mediation in Federated Databases. 8th WITS, Helsinki, Finland, to appear.

[7] Leser, U., H. Lehrach, et al. (1998). “Issues in Developing Integrated Genomic Databases and Application to the Human X Chromosome.” Bioinformatics **14**(7): 583-690.

[8] Levy, A. Y., A. O. Mendelzon, et al. (1995). Answering Queries using Views. 14th ACM PODS, San Jose, CA pp. 95-104.

[9] Levy, A. Y., A. Rajaraman, et al. (1996). Querying Heterogeneous Information Sources Using Source Descriptions. 22th VLDB, Bombay, India pp. 251-262.

[10] Miller, R. J. (1998). Using Schematically Heterogenous Structures. ACM SIGMOD, Seattle, Washington pp. 189-200.

[11] Naumann, F., J. C. Freytag, et al. (1998). Quality driven Source Selection using Data Envelopment Analysis. Int. Conf. on Information Quality, MIT, Cambridge.

[12] Sheth, A. and J. A. Larson (1990). “Federated Database Systems for Managing Distributed, Heterogeneous and Autonomous Databases.” ACM Computing Survey **22**(3).

[13] Wiederhold, G. (1992). “Mediators in the Architecture of Future Information Systems.” IEEE Computer **25**(3): 38-49.