TOWARDS CONCEPTUAL MEDIATION*

A semantic architecture for dynamic integration of heterogeneous databases

Navas D., Ismael, Aldana M., José F.
Dpto. Lenguajes y Ciencias de la Computación, E.T.S. de Ingeniería Informática, Universidad de Málaga, Spain

Keywords: Integrating Heterogeneous Data Sources, Dynamic Mediation, Semantics, Ontology, Web Services

Abstract: Mediators are usually developed as monolithic systems which envelope the data source's semantics as well as its location. Furthermore, its architecture based on wrappers involves a high coupling degree among the mediator's components. This coupling does not allow sharing services with other organizations or the dynamic integration of new data sources. Therefore, wrappers must be re-designed and manually added for each mediation system. We propose an architecture for conceptual mediation in which the sources' query capabilities are published as web services. These services can be registered in one or more resource directories (Semantic Directories), which are the core of this architecture because they provide the needed flexibility and scalability for dynamic integration. Finally, we show an application in a bioinformatics context to validate our approach.

1 INTRODUCTION

In the last years, the Web has become a great information repository that is manually accessed in the majority of cases. The amount of information and the complexity of a reasonable treatment to take advantage of all this information have led to a lot of research concerning database integration. The main goal of these systems is to allow users to make complex queries over heterogeneous databases, as if it was a single one, using an integration schema. Mediators offer user interfaces for querying the system, based on the integration schema. These mediators transform user queries into a set of sub-queries that other software components (the wrappers), which encapsulate data sources' capabilities, will solve. Usually, sub-query results are unstructured documents that are translated to structured documents.

These mediation systems have evolved through several improvements over traditional mediation architecture (of systems such as Manifold (Levy, 1996) and TSIMMIS (Papakonstantinou, 1995)). There have been a lot of improvements over the traditional mediation context. Thus, the Carnot project (Collet, 1991) introduces the use of ontologies for modeling integration schemas. This system makes use of a Cyc global ontology to describe the information system. Taking advantage of the information stored in a global dictionary, a graph is generated for each SQL query. A semantic component uses this graph and extends it with relevant resource information, and then an optimal query plan is generated for the extended graph. In OBSERVER (Mena, 1996) the information loss in database integration is studied. This system allows users to establish an information loss maximum value. The user can thus choose between search with or without information loss. In (Sahuguet, 2000) and (Ashish, 1997) automatic wrapper generation is studied. These works are based on mediator feature abstraction. Starting from this approach, we can apply code reuse in wrapper design and implementation processes, decreasing development cost and time. The majority of proposed tools focus their efforts on document transformation that gives

* This work has been supported by the Spanish MCyT Grant (TIC2002-04186-C04-04)

Navas D. I. and Aldana M. J. (2004).
TOWARDS CONCEPTUAL MEDIATION - A semantic architecture for dynamic integration of heterogeneous databases.
In *Proceedings of the Sixth International Conference on Enterprise Information Systems*, pages 169-176
DOI: 10.5220/0002606501690176

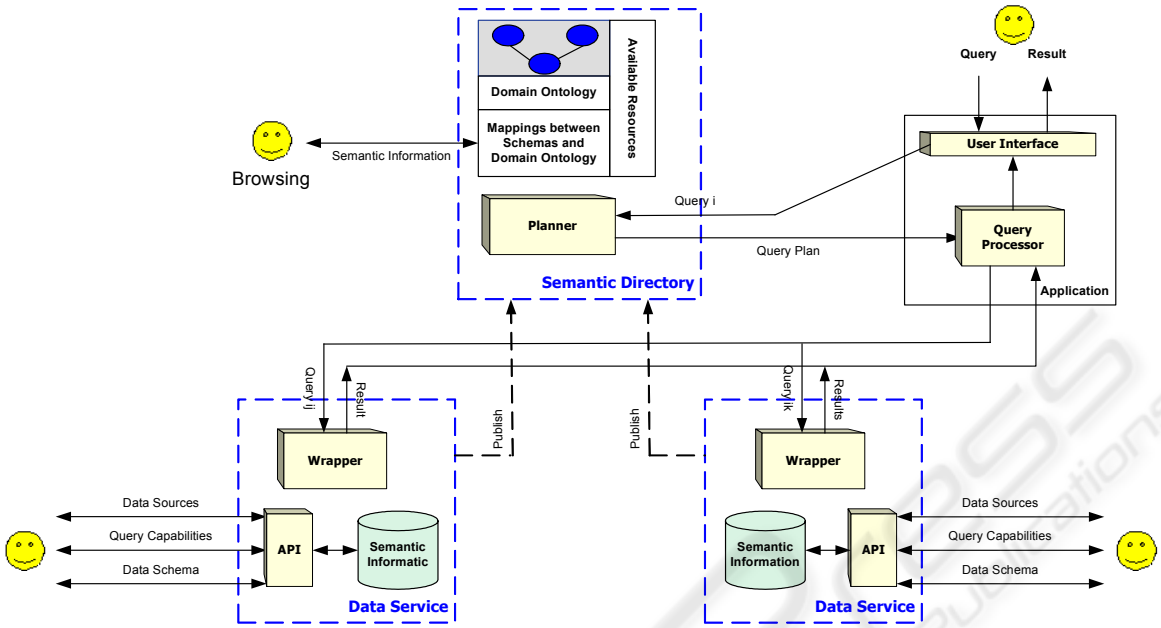

Figure 1: A Conceptual Mediation Architecture

them structure. However a few solutions, such as (Ashish, 1997), support full automatic wrapper generation. In any case, they do not include semantic information about the structured document generated or information about query capabilities. MIX (Baru, 1999) presents the use of XML for data representation and for modeling query interfaces. In this way, interoperability among mediators is assured. Finally, (Arjona, 2002) proposes the use of wrappers that return semantically annotated information. This paper introduces the semantic wrapper concept: "a semantic wrapper is a wrapper that can manage web knowledge". A Semantic Wrapper can be seen as the natural extension of current structural wrappers with the aim of adding semantics to the extracted information. However, this kind of wrapper is not used for exporting its semantics as an available service, so it is difficult for other mediators to reuse its knowledge. Mediators must thus analyze query result semantics to extract this knowledge.

Nonetheless, there are still several unsolved problems in heterogeneous data integration, including the following : (1) there is no reuse of components among mediators; (2) mediators and wrappers are strongly coupled; (3) query capabilities and semantics are not published as part of the (web) services; (4) software agents can not find wrappers, as they are "hidden" behind of mediators.

Our proposal is an architecture for conceptual mediation (see Section 2). This architecture includes directories in which an ontology and semantic information of resources are published. We propose to improve the wrapper implementation process by publishing them as web services, making their semantics accessible. This evolution from traditional wrappers to data services is motivated by several factors:

- Data services can be reused by other mediators or any other data accessing application.
- The semantics of data services is published on the web so that the services are readily available to other applications.
- Wrappers' query capabilities can be enveloped into one or more services.

This paper is organised as follows. Section 2 presents a novel architecture for conceptual mediation, which makes use of web services to allow dynamic integration. In section 3, we briefly describe a biological use case to validate our proposal. Finally, discussion and future works sections conclude this paper.

2 AN ARCHITECTURE FOR CONCEPTUAL MEDIATION

Our architecture seeks to make wrappers independent entities and to eliminate their ties with the mediator, thus increasing their reusability in different applications. We emulate P2P hybrid

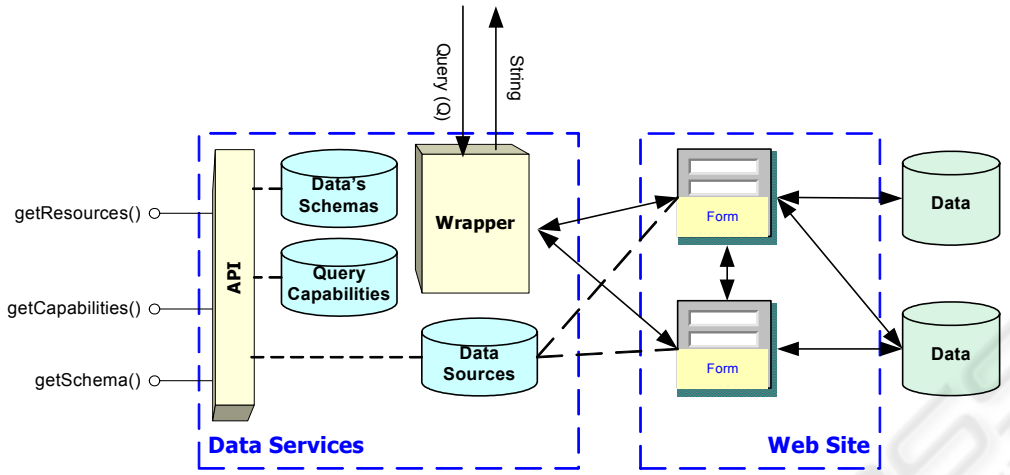

Figure 2: A Data Service Architecture

systems, which implement a directory with location information of available resources. In these systems the applications access directly to the resources by means of connections point to point that the directory has provided. Therefore, the flow of information is greatly reduced w.r.t. the one generated in the traditional client-server architectures.

Our proposal for conceptual mediation arises from two main considerations on the basic architecture of mediation: (1) on the one hand the isolation of wrappers, which are encapsulated as web services (Data Services for us); and (2) on the other, the added directory (Semantic Directory) with information about these Data Services (See Figure 1). Although the basic configuration has been developed with only one directory, nothing prevents us from having distributed configurations and/or from talking about more complex architectures with several (maybe replicated) semantic directories (for reasons of failure tolerance, availability or scalability). This architecture allows wrappers to contribute data, schemas of information and query capabilities in a decentralized and easily extensible manner.

A data service needs to be registered in one or more semantic directories in order to be used by a mediator or any other software agent. In other words, data services, like P2P hybrid systems, must know the location of semantic directories that are in the same application domain. Finally, public interfaces of data services and semantic directories will allow applications that share the communication protocol with it to take advantage of knowledge about available directory resources. Next we present the components of the proposed architecture as well as their functionality.

2.1 Semantic Data Services

Semantic directories offer essential services for obtaining the stored information and semantics through several web methods, such as getting and/or navigating over the domain ontology, the resources' address, and the mappings. Besides, each directory can offer a set of additional services. Figure 1 shows a service for query processing which returns query plans (Planner). This plan can be used and processed by a specific type of application: the mediators. These applications need to have a minimal query processor, which must understand the plans, send sub-queries to data resources and compose the partial results. Note that the planners return sub-queries in a language that data services can evaluate. However, semantic directories could be increased with other services in order to support other types of applications/agents.

On the other hand, data services provide the minimal elements for solving queries. We will define this type of service and then describe how it can solve access to a resource that has been included in a query plan.

Definition 1: A wrapper is a function W: (Q , R) $\rightarrow$ D. Given a query Q and a resource R, a wrapper returns a structured document D with the result of evaluating query Q in resource R.

Definition 2: A data service DS is a web service that offers several web methods for obtaining semantic and other information for querying a wrapper (it is not the wrapper itself).

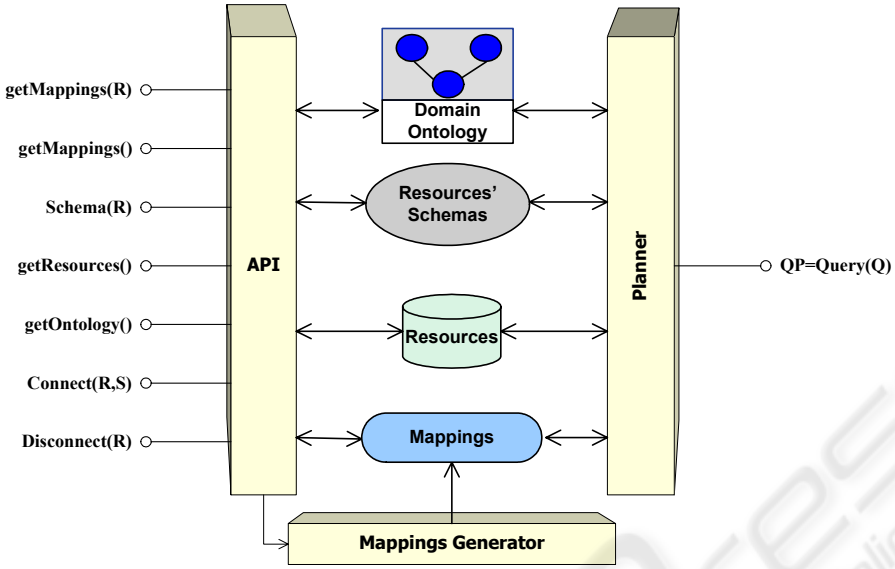

Figure 3: Internal Architecture of a Semantic Directory

The goal of this type of component is to allow applications to access wrapper functionalities. In this way, we have designed an extensible and adaptive architecture in which we can define a data service as "a service which offers wrapper query capabilities using web protocols".

The publication of these online web services using UDDI (Universal Description Discovery Integration) could allow other applications to dynamically discover wrappers by means of an easy interface. However, our data services have been devised for publication in a specialized type of directory: the semantic directory.

Figure 2 shows the internal architecture of a data service (for a web resource) which accesses several available data resources through web forms. This type of service uses two step processing: (1) a first step for accessing data sources using a wrapper, which solves a query and returns an XML document; and (2) the second one for exporting the wrapper's query capabilities and its semantics as a web service. The web service's semantics includes information about query capabilities, data schemas and data provenance. The latter is necessary, for example in the context of bioinformatics where it is important to know which is the information resource that is being used (due to different data quality and reliability). The data services need to describe their source capabilities for translating user queries into sub-queries that are understood by the concrete source interface.

In a first approach we used the type of query capabilities described in (Yemeni, 1999), and we had defined the annotations for each attribute or element. These query capabilities were expressive enough to represent the capabilities of web resources. However, we wanted to provide a solution for heterogeneous data sources, because the biological resources could be relational databases, web resources or plain text. In our present approach we use p-Datalog (Vassalos, 1997) to describe query capabilities. It is a Datalog variant, which copes with the need of a more powerful description language. The p-Datalog source description language allows defining capabilities for conjunctive queries. A result of the cited paper is that p-Datalog can not describe capabilities of certain powerful sources.

This type of service not only encapsulates a wrapper's query service, but also provides access to the schema of the information they store. For this purpose the getSchema(), getCapabilities() and getSourcesInformation() methods are published as an API, and they return respectively the data's schema, the query capabilities, and information related to data sources used by the data service.

2.2 Semantic (Resource) Directories

Semantic directories are the core of this architecture, because they provide essential services for application domain users. Next we will define several necessary terms for the semantic directory definition and describe this type of resource directory.

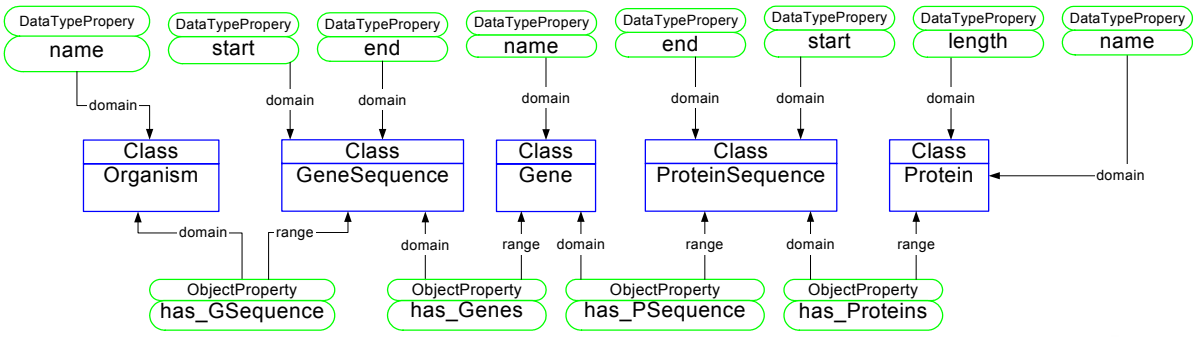

Figure 4: Domain Ontology

Definition 3: A query plan QP is a tuple <Qs , Rs , Cp> where Qs is a vector of queries; Rs is a vector that contains the resource address where these queries must be evaluated and CP is a composition plan that includes a method to compose the result of evaluating each query Q in the correct resource R.

Definition 4: A planner P is a function P: Q → QP. Q is a query and QP is a query plan.

Definition 5 : A Semantic Directory SD is a server that stores a domain ontology DO, mappings between data resource schemas (M) and DO, and information about available resources. Besides, it can offer a set of services, such as a planner P. That is, we can define a semantic directory as "a server that offers information about available web resources (Data Services), a domain ontology, mappings between resource schemas and this ontology, and provides services (for example, a query planner) to application domain users".

A semantic directory stores an ontology (described with OWL), which must be generic for the application domain (a Domain Ontology). This ontology describes the core knowledge that is shared by a set of applications or a user community. For our purpose, the Domain Ontology can be seen as an abstraction of the knowledge of the resource's schemas. Thus each schema could be considered as a refinement of the domain ontology. The main functionality of the directory is to provide access to semantic information stored in it. However, it can offer several value added services, for example for returning an execution plan. This type of service increases its functionality whenever similarity between both models (the Domain Ontology and the resources' one) also increases. Information about data services will be added to a semantic directory when services register in it. So the data service owner must use the connect method (offered by the directory, see Figure 3) in order to publish his/her service. This method saves information about the service and calculates mappings between the directory's domain ontology and the service's schema. Semantic directories periodically ask registered data services for updated information. This allows maintaining schemas, query capabilities and availability updated at all times and consistent with data services. If there are changes, then the directory recalculates mappings. Besides, the owner of a service can disconnect it, making it unavailable to be included in query plans.

The storage of mappings between service schemas and the domain ontology is very important, due to their use in query planning. In the basic case, applications send queries (Q_i) in terms of the ontology to the semantic directory, which returns a query plan (QP) for this query. Each plan includes links to services, a set of sub-queries for each service (Q_{i1} ... Q_{in}) and a composition strategy. The plans are based on mappings stored in the semantic directory. Note that this architecture does not implement a query processor, so the integration applications can evaluate the QPs, releasing the semantic directory of this task. Note that a mediator is just one of the possible applications which has a query processor to perform the query plan.

Besides, information stored in directories could be used by other applications or agents that search for information about: ontologies, data services whose schema accomplish a condition, etc. In a special context agents can implement their own query plan generation algorithms. For this reason, the directory's API provides several methods to retrieve information stored in it, which are: getOntology(), getResources(), getSchema(R), getMappings() and getMappings(R).
The getOntology() method returns an OWL document, which includes the description of the domain ontology stored in the directory. A list of

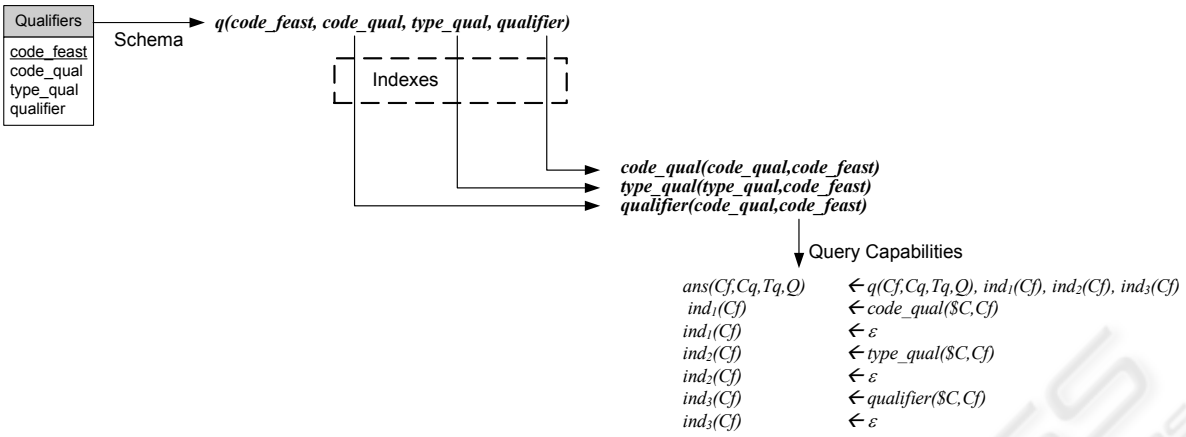

Figure 5: Micado's Query Capabilities. We have a qualifiers table in the Micado database. If we want to provide access to query this using a data service, we need to define the table with the *q* predicate. In order to allow users to query with any combination of the last three fields, we need to define three indexes. Finally, we describe the query capabilities needed.

available resources can be obtained with the getResources() method. For each resource R, we can ask for its schema making use of the getSchema(R) method. Finally, mappings between a schema or overall schemas and the domain ontology are returned respectively by the getMappings(R) and getMappings() methods.

3 CONCEPTUAL MEDIATORS

The accumulated biological knowledge needed to produce a more complete view of any biological process (e.g. sequences, structures, gene-expression data, pathways) is disseminated around the world in the form of biological sequences and structure databases, frequently as flat files, as well as image and scheme-based libraries, web-based information, particular and specific query systems, etc. Although it is a common place statement that the volume of data in biology is growing at exponential rates, nonetheless, the key characteristic of biological data is not so much its volume, but its diversity, heterogeneity and dispersion (Rechemann, 2000).

BioBroker (Aldana, 2004) is an XML mediator which supports biologist working in the field of proteomics and genomics. This integrates heterogeneous data sources: the nucleotide (EMBL) and protein (SWISS-PROT) sequence information (Bairoch, 2000), the worldwide repository of three-dimensional structures of biological macromolecules (PDB) (Berman, 2000) and the MICADO relational database devoted to microbial genomes (Biaudet, 1997). These databases have been selected on the basis of the difference in content, format, access mechanism, and geographical location. We are currently developing a mediator for integrating biological resources, which will interact with a semantic directory and several data services. This will be the natural evolution of the previous XML mediator (BioBroker). So next we present a sample use of the proposed architecture for integrating gene expression databases.

The semantic directory for BioBroker resources includes an ontology which covers the concepts of genomic and proteomic (see Figure 4). This figure shows only some properties due to space limitations. This simple ontology, which treats knowledge about organisms, genes and proteins, is related with the data sources through mappings between the data schemas and the domain ontology.

We are developing the data services for each of the BioBroker's wrappers. In each one we must define its semantics (query capabilities, data provenance, etc.). Figure 5 shows the query capabilities for the MICADO data service.

Let us suppose that we need to find the *code_feat* of the genes of the *Bacillus Subtilis* that contains the *LKKQFVEAFG* protein sequence (note that the field qualifier has mapped to the field name of the gene class). This query will be sent to the semantic directory and it will return a query plan, which includes a set of sub-queries, the data service in which to evaluate each of them and a data composition algorithm. In our example, this query will produce the next sub-queries (which must be evaluated in the SwissProt, EMBL and Micado databases):

(Swiss-Prot and EMBL)
ans(Gn) ← *g(Gn,Sq), sequence('*LKKQFVEAFG*',Gn)*
(Micado)
ans(Cf) ← *q(Cf,Cq,Tq,Q), qualifier(Gn,Cf)*

Note that the second sub-query needs the Gene Name (*Gn*) from the first one. Thus, the evaluation must be sequential. Once results of first sub-query are received, the system eliminates possible inconsistencies and duplicates, and makes use of these results into the second sub-query. Finally the system translates results to ontology instances and returns them to user application.

4 DISCUSSION

In this paper we describe an architecture for conceptual mediation based-on a P2P and Web Services architecture that presents advantages with respect to traditional mediator system approaches. The semantics introduced in the Semantic Directories allows users to make more expressive queries, viz. semantic queries that other mediators cannot solve. They greatly increase the query capabilities of this type of mediator. Besides, the mappings between schemas and the domain ontology of a semantic directory provide support for solving these queries over the available resources.

The use of an architecture like P2P introduces a high level of uncoupling between wrappers and mediators or any other application, which could involve wrappers. The directories supply an easy way to integrate data sources and open new directions in dynamic integration. Besides, our proposal entails additional profits, such as the reuse of wrapper components, access to data services for other applications, use flexibility, etc. It provides elements to obtain major interoperability among integration systems that cooperate in the same application domain or that belong to other domain with which they have certain relationships.

In several domains in which there are no "technological" users, such as biologists, dynamic integration is a very important issue. In this context, it is necessary to give users a simple environment for integrating data information without the modification of the mediator's code or of the integration schema. Using a domain ontology, users can design queries starting from specific knowledge that belongs to their field of research. However, the proposed architecture requires that somebody implement wrappers and publish them in the semantic directories.

From our point of view, CBSD technology (Szyperski, 1998) can help automatic generation of wrappers, allowing us to configure data source accesses as well as to choose appropriated algorithms for each task. By applying this technology users can generate wrappers just by knowing the resources in which the information can be found. Note that these resources are well known by users, so they make use of them in their daily work.

The case for use in biology is evidence of the suitability of this kind of architecture for the bioinformatics domain. In particular, the usefulness of a mediator system is demonstrated by a diverse set of applications aimed at combining expression data with genomic, sequence-based and structural information, so as to provide a general, transparent and powerful solution that goes beyond traditional gene expression data clustering.

Our architecture opens new ways to address interesting issues, such as query decomposition algorithms, result integration, data service location, searches in data directories, etc.

5 FUTURE WORK

As future works, we propose several lines of mediator development. Increasing the automation level in wrapper creation and adding to this process data service generation previously described, which will reduce even more their cost of development. Another interesting line stems from studying the possibility of giving more semantics to these data services, taking into account service quality, relations with other domain ontologies, etc. Using this additional information, we could generate alternative query execution plans, allowing applications to choose the one which is more suitable or generating them based on certain features (using local resources for the application location). Besides, the scalability of this architecture will provide the possibility of integrating not only data services but also semantic directories, making possible a full semantic integration of resources and the interoperability between applications. Thus, we will provide elements to achieve interoperability between semantic integration systems that cooperate in the same application domain or have certain relations (Semantic Fields). Furthermore, we will introduce a solution to integrate semantic fields and obtain better query capabilities.

We plan to study automatic mapping between schemas and ontologies taking into account a previous experience (Madhavan, 2002). It can be applied to establish correspondences between retrieved document schemas and directory ontologies in those new systems developed using our architecture. Finally, we are interested in establishing a semantic model to define the data service's query capabilities, which improves query planning by adding inferences about query

capabilities to the reasoning between schemas and the domain ontology.

REFERENCES

Aldana, J.F., Roldán, M., Navas, I., Pérez, A.J. and Trelles, O., 2004. Integrating Biological Data Sources and Data Analysis Tools through Mediators. ACM Symposium on Applied Computing. Nicosia, Cyprus.

Arjona, J.L., Corchuelo, R.. Ruiz-Cortés, A. and Toro, M., 2002. Extraction of Semantically-Meaningful Information from the Web. Second International Conference on Adaptive Hypermedia and Adaptive Web Based Systems. Springer–Verlag. Málaga, Spain. pp. 24–35.

Ashish, N. and Knoblock, C., 1997. Semi-automatic Wrapper Generation for Internet Information Sources. In conference on Cooperative Information Systems.

Bairoch, A. and Apweiler R., 2000. The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. In Nucleic Acids Res 28.

Baru, C., Gupta, A., Ludäscher, B., Marciano, R., Papakonstantinou, Y., Velikhov, P. and Chu, V., 1999. XML-based Information Mediation with MIX. In Demostrations, ACM/SIGMOD, pages 597-599.

Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N. and Bourne, P.E., 2000. The Protein Data Bank. Nucleic Acids Research, Vol. 28, No. 1 235-242.

Biaudet V, Samson F and Bessieres P., 1997. Micado a network-oriented database for microbial genomes. In Bioinformatics, Vol 13, 431-438. Oxford University Press.

Collet, C., Huhns, M. and Shem, W., 1991. Resource Integration Using a Large Knowledge Base in Carnot. IEEE Computer, pages 55–62.

Levy, A., Rajaraman, A. and Ordille, J., 1996. Querying Heterogeneous Information Sources Using Source Descriptions. In Proc. VLDB, pages 251-262, Mumbai (Bombay), India.

Papakonstantinou, Y., Garcia-Molina, H. and Widom, J., 1995. Object Exchange Across Heterogeneous Information Sources. In Proc. ICDE, pages 251-260, Taipei, Taiwan, March.

Madhavan, J., Bernstein, P., Domingos, P. and Halevy A., 2002. Representing and Reasoning about Mappings Between Domain Models. In proceedings of the AAAI Eighteenth National Conference on Artificial Intelligence.

Mena, E., Kashyap, V., Sheth, A. and Illarramendi, A., 1996. OBSERVER: An Approach for Query Processing in Global Information Systems based on Interoperation across Pre-existing Ontologies. In conference on Cooperative Information Systems.

Sahuguet, A. and Azavant, F., 2000. Building Intelligent Web Applications Using Lightweight Wrappers. In Data Knowledge Engineering.

Szyperski, C., 1998. Component Software. Beyond Object-Oriented Programming. Addison-Wesley.

Vassalos, V. and Papakonstantinou, Y., 1997. Describing and Using Query Capabilities of Heterogeneous Sources. In *VLDB'97, 23rd International Conference on Very Large Data Bases*.

Web Services. XML-RPC,SOAP, sobre PHP, Perl, y otros conceptos. http://web-services.bankhacker.com/

Yerneni, R., Li, C., Garcia-Molina, H. and Ullman, J.D., 1999. Computing capabilities of mediators. In SIGMOD, pages 443-454.