KASBi: Knowledge-Based Analysis in Systems Biology

Maria del Mar Roldán-García , Ismael Navas-Delgado, Amine Kerzazi, Othmane Chniber, Joaquín Molina and José F. Aldana-Montes

Computer Languages and Computing Science Department, University of Málaga, Málaga, Spain
{mmar, ismael, kerzazi, chniber, jmolina, jfam}@lcc.uma.es

Abstract. The analysis of information in the biological domain is usually focused on the analysis of data from single on-line data sources. Unfortunately, studying a biological process requires having access to disperse, heterogeneous, autonomous data sources. In this context, an analysis of the information is not possible without the integration of such data. This paper describes how KOMF, the Khaos Ontology-based Mediator Framework, is used to retrieve information and crystallize it in a (persistent) Knowledgebase. This information could be further analyzed later (by means of querying and reasoning). These kinds of systems (based on KOMF) will provide users with very large amounts of information (interpreted as ontology instances once retrieved), which cannot be managed using traditional main memory-based reasoners. We propose a methodology for creating persistent and scalable knowledgebases from sets of OWL instances.

Keywords: Life Science, Information Integration, Semantic Web, Reasoning.

1 Introduction

The need for data integration started when the number of applications and data repositories began to grow rapidly. The first approaches appeared in the 80s, and formed the basis for the research in this area. The evolution continued over mediator based systems, such as AMOS II [1], DISCO [2], TSIMMIS [3] and Garlic [4]. Then, agent technology was used in some systems like InfoSleuth [5] and MOMIS [6]. Finally, the new technologies appearing have been used in data integration: XML (MIX [7]), ontologies (OBSERVER [8]).

The rapid growth of the Internet has given us access to an unprecedented number of heterogeneous information sources. This huge amount of information and the complexities of handling it have given rise to a lot of research concerning practical approaches to the Semantic Web.

Semantic Web searches have been based on already existing systems, and the proposed approaches offer a limited amount of information for agents. Search engines cannot interpret all the information available because documents have not yet been semantically annotated. We propose the use of an ontology-based mediator framework (the Khaos Ontology-based Mediator Framework, KOMF) to access varied information from diverse biological databases [9]. KOMF has been

successfully instantiated in the context of Molecular Biology for integrating data sources [10].

This application can be used to extract integrated information from the set of databases included in the system. This information is retrieved as a set of ontology instances. However, the analysis of these instances is limited in KOMF. In order to apply analysis tools it is necessary to store the instances appropriately to facilitate their access. However, the sheer number of instances that must be retrieved make the use of a traditional reasoner unfeasible [11,12]. Thus, we propose the use of DB-OWL [13], a persistent and scalable reasoner that is able to deal with this large amount of information. It stores the ontologies in a relational database, using a description logic reasoner to pre-compute the class and property hierarchies, and to obtain all the ontology information (i.e. properties domain and range) which is also stored in the database. Furthermore, a simple but expressive query language has been implemented, which allows us to query and reason on these ontologies. This reasoner implements both Tbox (ontology structure) queries and Abox (ontology instances) inferences. Tbox queries can be evaluated directly using the query language. On the other hand, Abox inferences are evaluated when a query is sent to the system to obtain complete results. Both, Tbox queries and Abox inferences are implemented using only the information stored in the database.

In summary, the goal of this paper is to introduce a tool to generate Knowledgebases to boost the analysis capabilities of knowledge obtained from user queries. The combination of a data integration system with a knowledgebase is a novel approach that opens new ways of analyzing the information based on the knowledge.

This paper is organized as follows: Section 2 describes previous works, focused in the main characteristics of KOMF and the architecture of DBOWL and how it stores the ontologies and how reasoning and queries are implemented. Section 3 focuses on the tool developed to create queries and build knowledgebases based on KOMF results. Finally, we conclude with some remarks on the work presented.

2 Previous Works

The main goal of KOMF is to integrate data sources which are accessible via the internet or can be downloaded for local use. In this context, wrappers are an important part of the internal elements of Data Services. A wrapper is an interface that translates data into a common data model used by a mediator.

Data sources in some domains such as Molecular Biology are usually public and downloadable. For these cases we have designed patterns to retrieve data sources stored as flat files for later storage in an XML database. Data Services, independently of the development process, are distributed software applications that receive queries in XQuery and return XML documents.

The KOMF architecture is composed of three main components: *The Controller (*receives user requests and coordinates the mediator components), *The Query Planner (*elaborates one or several query plans to compute the user query from different data

sources) and *The Evaluator/Integrator (*analyzes the query plan, and performs the corresponding call to the data services involved in the sub-queries of the query plan).

The BioDataServer [14] is a database integration system, which implements a mediator-wrapper architecture. This tool uses a SQL-based query language and an XML data format to allow easy use of the resulting data sets of the integration process. Data integration is based on user defined integrated schema and adapters that wrap any kind of data source. The main advantage of using ontologies is the possibility of using reasoning to find new knowledge (infer new instances or assertions that are not present in the individual databases). However, it is also relevant because this kind of proposal takes advantage of semantics commonly accepted by a community.

DBOWL stores the OWL-DL ontologies in a relational database, and supports Tbox queries (queries on the ontology structure), Abox inferences (reasoning on the ontology instances) and ECQ (Extended Conjunctive Queries) queries [15]. In order to create the relational database for ontology storage, a Description Logic Reasoner is used. Thus, the consistency of the ontology as well as the inferences about the ontology structure are delegated to this reasoner and DBOWL focuses on reasoning on instances (large numbers of them). Both, Tbox queries and ECQ queries are implemented by translation to SQL. Abox inferences are implemented by java functions and SQL views [12].

DBOWL consists of two services, an OWL storage system and an OWL querying system. The OWL storage system stores the OWL ontology in the database. Starting from an OWL file, the class/subclass hierarchy (the concepts taxonomy), the property/subproperty hierarchy (the properties taxonomy), the ontology structure information and the ontology instances are computed using a description logic reasoner. Subsequently, a relational schema is created in order to store all this information. Finally, Abox inferences are evaluated and the views are created. The DBOWL querying system performs both Tbox queries and ECQ queries over the ontology stored in the relational database.

Several alternative approaches for dealing with OWL ontologies have been presented. Some of them are very popular in the biological field. Protégé [16] is an ontology editor. It can be used combined with a description logic reasoner in order to make queries and inferences on the ontology. Due to a description logic reasoner being used, reasoning is in main memory. Thus, Protégé is not able to deal with large ontologies. Some approaches using relational technology have also been presented. Instance Store [12] uses a DL reasoner for inferring Tbox information and storing it in a relational database. However, the ontology definition language does not allow the definition of binary relationships. From our point of view, this is an important expressiveness limitation. Moreover, Instance Store only evaluates a few Abox reasoning, namely subsumption of concepts and equivalent classes. It implements them by reducing them to terminological reasonings and evaluates them using a DL reasoner. The main feature of DBOWL is that it can deal with a large number of instances evaluating most of the OWL-DL reasoning.

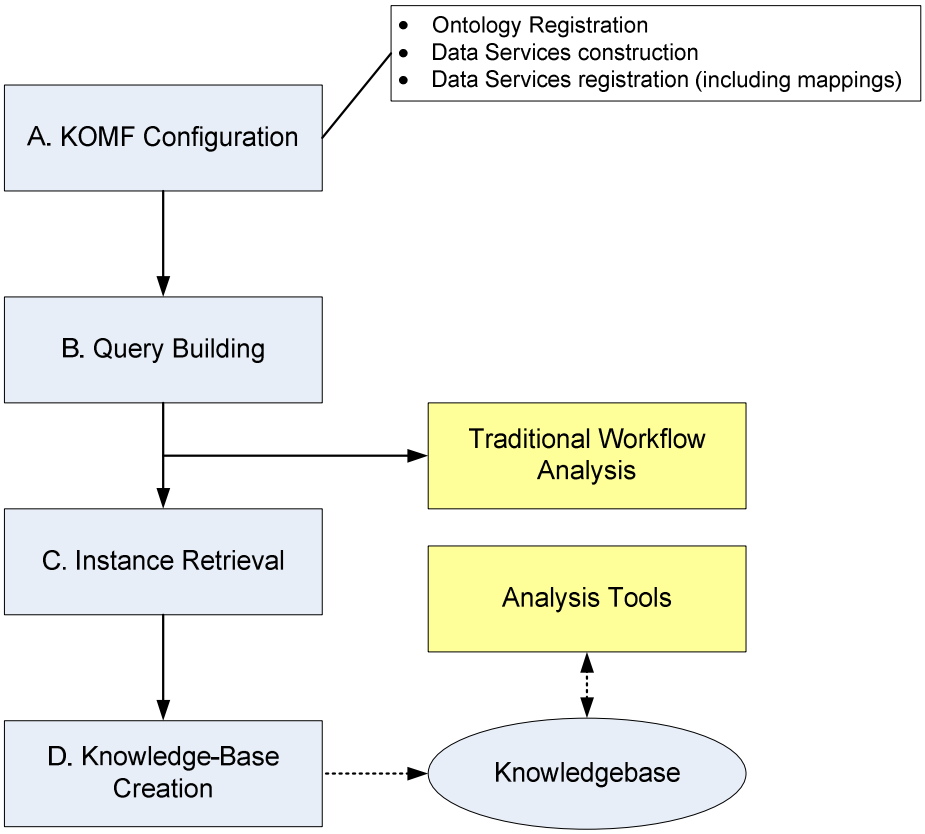

Fig. 1. KASBi, tool information flow.

3 KASBi

As described in the previous section, users can use KOMF to query heterogeneous data sources, and use this information to perform domain specific analysis. However, KOMF has limited reasoning capabilities. For this reason, the proposed tool introduces DBOWL as a persistent reasoner to perform more complex analysis.

Thus, the designed tool establishes a set of operations to be performed when a knowledgebase is to be constructed from diverse data sources (Figure 1). The tool follows four steps (see more details in Figure 2):

A. KOMF Configuration (A in Figure 3). This task aims to produce the necessary elements to integrate information from heterogeneous data sources. It involves firstly registering the domain ontology to represent the domain. The next step is to create the necessary data services, register them in the system and then set up the relationships between each data service schema and our domain ontology. After this configuration, users can send

queries in terms of the domain ontology that will be solved using the registered data services. This part requires a lot of work that remains mainly in the data service development and mapping definition (when using an existing ontology).

B. Query Building (B in Figure 3). As we aim to produce a knowledgebase centered on a specific need, it is necessary to design a query (or a set of queries) to retrieve all the information that will be later analyzed.

C. Instance retrieval (C in Figure 3). The designed query is executed using KOMF, obtaining a set of instances as RDF documents.

D. Knowledgebase Creation (D in Figure 3). The domain ontology and the retrieved RDF documents (for which the user requires a more sophisticated analysis) are used to generate the query-based knowledgebase using DBOWL.

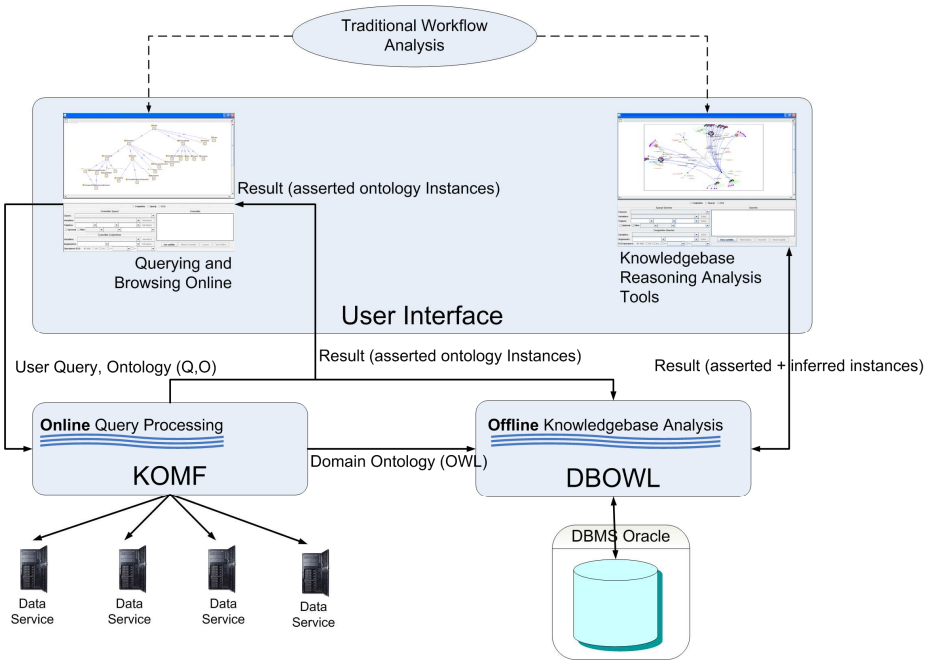

Fig. 2. KASBi implementation structure. The tool uses KOMF to retrieve information as ontology instances. When a user retrieves information that needs further analysis the tool allows him/her to create a persistent knowledgebase. This knowledgebase could be used to perform more deep and complex analysis over a specific set of information.

The proposed steps have been built using the works described in the previous section, KOMF and DBOWL (Figure 3). Thus, the user queries are sent to KOMF (see [9] for more details about the data service creation and mapping description in KOMF) to retrieve the required instances (those necessary for more sophisticated analysis) that will be stored in DBOWL (D in Figure 3). Then, analysis tools can take advantage of the reasoning capabilities of DBOWL. Both user interfaces can publish

their programming interface so they can be used in traditional life science workflows as another data source or data transformation tool.

The tool provides a user interface for visualizing the registered ontology and creating the user query (see Figure 3). This interface allows users to select concepts of the ontology to build the queries easily. Thus, this interface uses a heuristic to propose the user the links between predicates through the variables. For example if the domain ontology has the concepts *Polypeptide* and *Organism* (linked through the object property *bioSource*):

- When the user clicks on the Polypeptide concept, the tool proposes to introduce the predicate *Polypeptide (X)*;
- When the user clicks on the Organism concept, the tool proposes to introduce the predicate *Organism (Y)*;
- Finally, if the user clicks on the property bioSource, the tool proposes the predicate *bioSource (X, Y)*.

Once the knowledgebase has been created the users can use it to perform different analysis by means of analysis tools. For example, a visualization tool can be used to analyze the structure of the knowledge stored. This visualization tool can be configured to use different icons for different instance types, so end users can better understand the resulting graph.

Furthermore, new tools can be developed or existing tools can be adapted to analyze specific issues based on the expertise of domain experts.

The advantage of using DBOWL is that these tools (I in Figure 3) can take advantage of a persistent storage (II in Figure 3) and reasoning to infer new knowledge (III in Figure 3). Thus, results (IV in Figure 3) can contain asserted instances plus those obtained through reasoning.

3.1 Case Studies

We show in this section some use cases that use a knowledgebase with useful information for systems biology researchers built taking advantage of the tool described. We have registered the domain ontology in the system to be able to extract instances (Figure 1). This ontology provides a set of concepts that are necessary to represent the information that we aim to extract from distributed databases. The ontology used in this example is BioPax Level 3 (http://www.biopax.org/), which covers metabolic pathways, molecular interactions, signaling pathways (including molecular states and generics), gene regulation and genetic interactions. Figure 4 shows the entities part of this ontology.

The following sub-sections describe some motivating examples using these ontologies. These examples are focused on showing how a reasoner can use inference to discover new knowledge and inconsistencies.

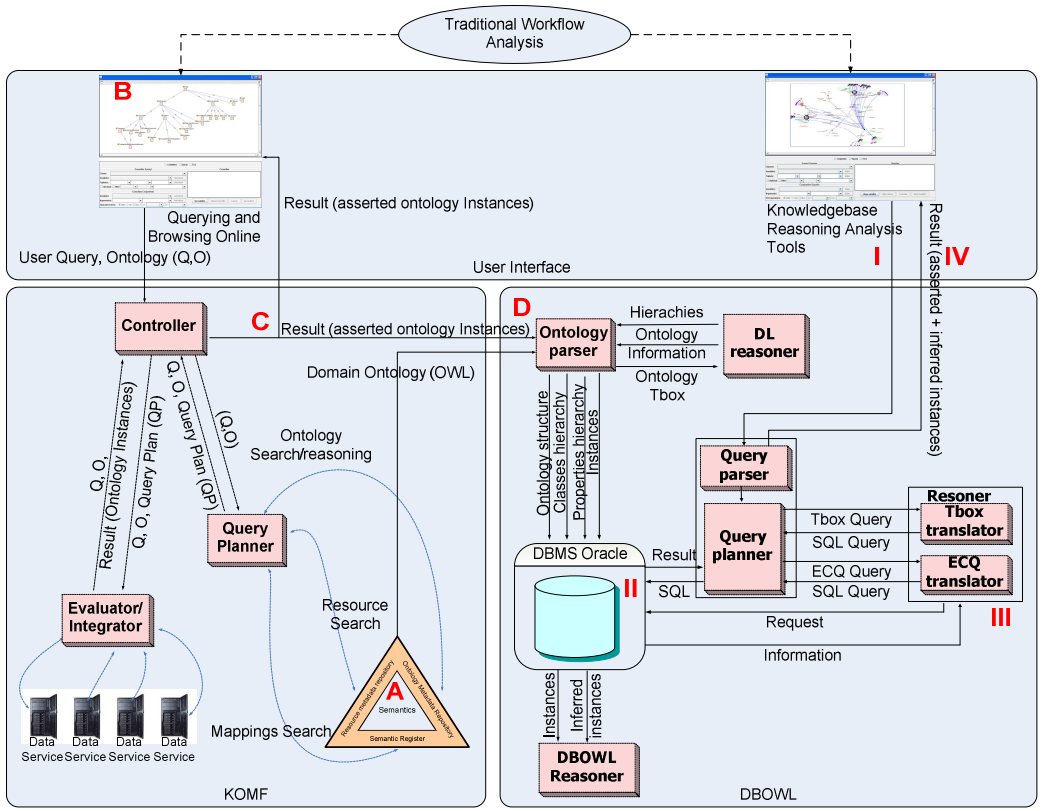

Fig. 3. KASBi implementation details. The internal elements of KOMF allow users to perform online queries, while DBOWL provides a persistent reasoner to perform more complex analysis over specific sets of information. In this way, results from the KOMF Controller can be used to create the knowledgebase using the DBOWL Ontology Parser (it requires an ontology and a set of instances of this ontology).

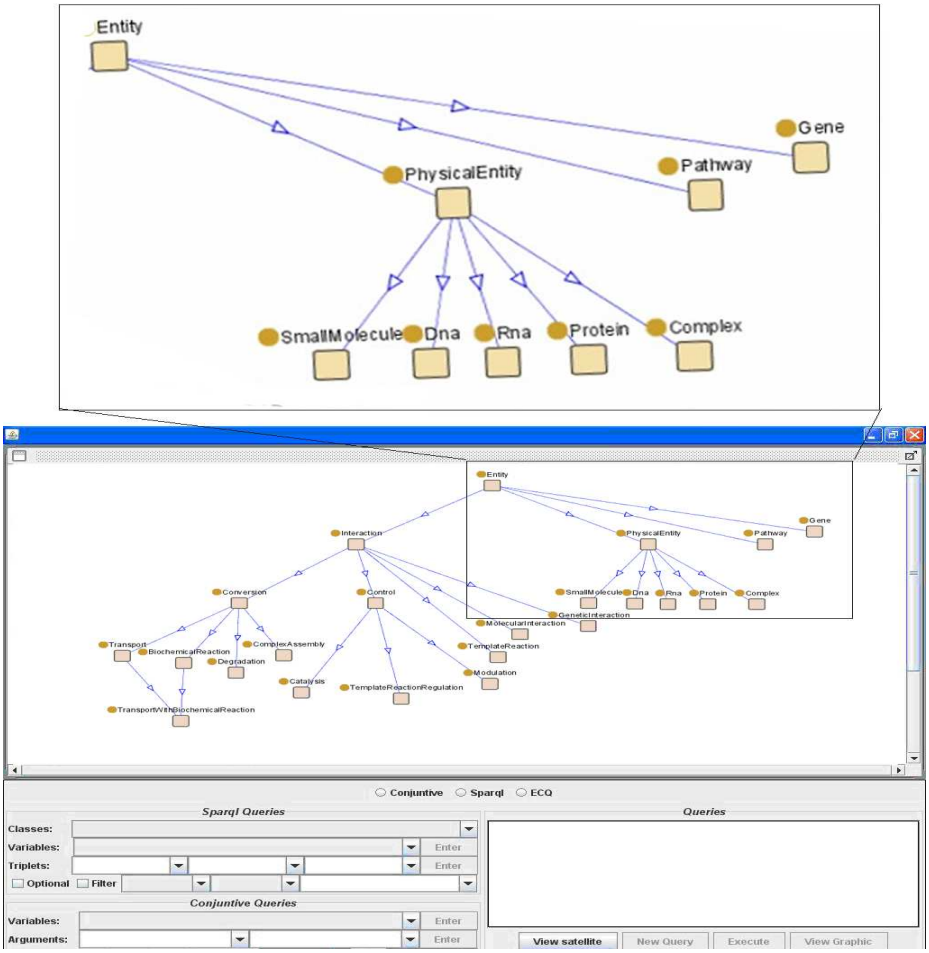

Fig. 4. Query Interface. This part of the tool enables building user queries easily.

3.1.1 Protein Bindings

In order to extract information related with the protein binding we have registered the databases BIND[1] and PDBBind[2]. Those databases have been mapped to the concept *Protein* and the object property *bindsTo*. Thus, elements of the data structure of these databases are mapped to the ontology concept participating in this property (Figure 5).

[1] http://www.bind.ca/
[2] http:// www.pdbbind.org/

Once the system has been configured, it is ready to receive queries. Thus, the user will design a query that aims to extract all the interesting information for a specific protein:

- `Ans(P) :- Protein(P1), name(P1,`**`name`**`), Protein(P), bindsTo(P1,P);`

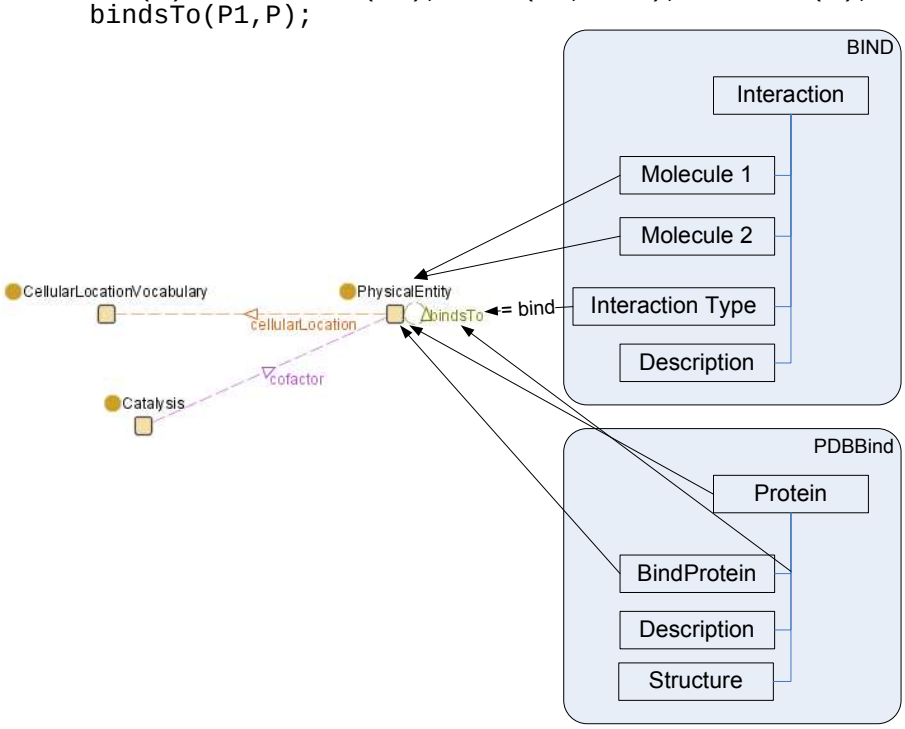

Fig. 5. Mappings between BioPax and databases with information about protein binding.

This query requires a protein name to retrieve the information about this protein. Thus, we have selected the proteins of interest for a specific database to extract all the useful information. The execution of this query with a specific protein name will return a set of RDF instances. These queries are planned by the mediator, by generating a decision tree with the set of sub-queries that must be executed in each data source. The resolution of queries will return a set of instances that will be used to create the knowledgebase.

Once the knowledgebase has been created, the analysis tools can take advantage of the reasoner. As the property "bindsTo" is a symmetric property, if we obtain that the protein P1 is bound to the protein P2, the reasoner can infer that P2 is also bound to protein P1 (even if this information is not explicitly stored in the database). For example, if we have that "GroES" binds to "GroEL", then the reasoner can infer that also "GroEL" binds to "GroES". For example, a user can generate a cluster of the Yeast PABP networks [17] searching for the binding proteins of each protein that he/she wants to include in the analysis (see the representation of the network in Figure 6).

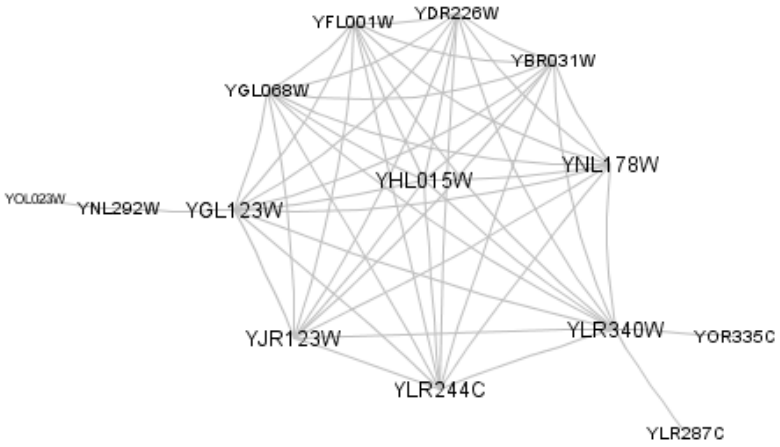

Fig. 6. Visualization of a binding network.

3.1.2 Organism identification

The definition of pathways is closely related to the organism in which this pathway is expressed. For this reason we have registered in KOMF the following data sources: KEGG[3] and BioCyc[4] (see representation of mappings in Figure 7).

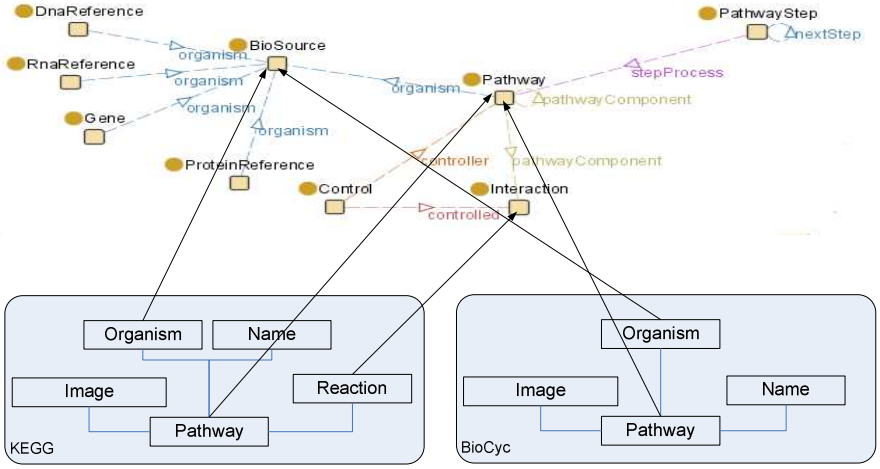

Fig. 7. Mappings between pathway data sources (KEGG and BioCyc) and BioPax.

[3] http://www.genome.jp/kegg/
[4] http://biocyc.org/

Thus, taking the BioPax ontology, we can design a query for extracting information about pathways in different organisms using the following query:

- `Ans(P) :- Pathway(P), BioSource(O), name(O,`**`name`**`), organism(P,O);`

This query requires the name of the organism to retrieve all the pathways described for its internal processes. The resulting set of RDF documents (an example is shown in Figure 8) for the execution using different organisms will require the use of a reasoner to detect possible duplications (caused by using different databases). Thus, we can use the fact that "organism" is a functional property. If the pathway P is described for the organism O1 and also for the organism O2, the reasoner can infer that O1 and O2 are the same organism (maybe even the same organism described using synonymous names).

For example, if the pathway "ENSEMBL ENSGALP00000028424 MGMT Methylated DNA protein cysteine methyltransferase EC 2 1 1 63" has as organisms "Gallus gallus" and "G. gallus domesticus" in two different databases, then the reasoner can infer that both organisms are the same (they are synonyms of the same organism).

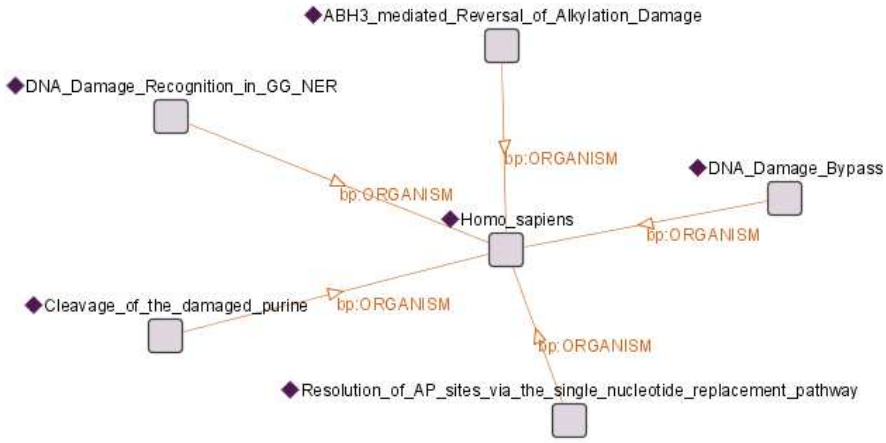

Fig. 8. Visualization of set of pathways for an organism (Homo Sapiens).

3.1.3 Annotation errors

The retrieval of information about different entities like Protein and Complex could be done using a generic query such as:

- `Ans(P) :- PhysicalEntity(P), name(P,`**`name`**`);`

The results of this query will include RDF documents with any kind of physical entities, and depending on the database used they can be instances of OWL classes such as Protein, Complex, DNA, RNA or Small Molecule.

However, once the knowledgebase is created the set of instances may contain errors. The use of the reasoner will solve this problem. If the physical entity P is an

instance of Protein and Complex classes, the reasoner can infer that the knowledgebase has inconsistencies (as far as Protein and Complex are defined as disjoint classes in this ontology).

For example the protein complex "Cytochrome b6f Complex" can be annotated in a database as a Protein and as Complex in a different database. Thus, this inconsistency will be detected by the reasoner, and the application using this information can act to solve this inconsistency.

4 Discussion

In this paper we have presented a novel system that combines the use of a mediation system (KOMF) with the reasoning capabilities of a persistent reasoner (DBOWL) to provide a way of finding new knowledge. The main drawback of this proposal is the configuration of KOMF that requires the development (or search) of a domain ontology, the implementation of data services for accessing the required data sources and the definition of mappings between the domain ontology and each data service schema.

However, the system enables a lot of opportunities to take advantage of the integrated information by means of a user interface for testing different queries. Then, the reasoner allows users to make some results available to search new knowledge or perform analysis tasks. As the reasoner is implemented over a relational database the reasoning part has a low computational cost and scales as much as a typical database. We have tested the scalability of the systems and it provides fast results (less than 3 seconds) for users querying knowledgebases with more than 5000 instances.

The system described shows a way to create knowledgebases from user queries. Then we have described some simple examples over the BioPax Level 3 ontology to motivate the type of reasoning that can be done.

However, our approach can be useful for real Systems Biology applications, especially for those aiming to provide end-user interfaces with extended capabilities. Thus, as stated in [18] the new technologies such as Ajax, SOA and Semantic Web along with enhanced functionality will make applications more interesting to researchers.

5 Conclusions

The life science domain has to face a new era in which the integration of information is an important issue due to the fast development of high throughput techniques, which are producing large amounts of data. Besides, traditional approaches must be improved to take into account the special characteristics of this domain.

This paper presents a tool that has two main pillars: an ontology-based mediator (KOMF) and a persistent reasoner (DBOWL). Thus, the use of KOMF enables the retrieval of information useful for end users, while DBOWL is used to create knowledgebases based on the user query (for making persistent the information that the user wants to analyze in more sophisticated ways).

This approach opens new capabilities for analyzing the information and for taking advantage of the knowledge represented by means of domain ontologies. Thus, a reasoner can be used to discover new knowledge and even inconsistencies between different databases.

As part of this tool we have described a user interface for creating the user queries, and visualizing the resulting information. Besides, some use cases are shown to demonstrate the need for a reasoner to find implicit knowledge and inconsistencies.

The proposed system will be available (http://khaos.uma.es/KASBi) as a demo version with a predefined ontology and a set of data services mapped to it (BIND, PDBBind, KEGG and BioCYC).

In the future the systems will also be distributed as an installable and configurable version, including the most recent improvements in each of its components.

Acknowledgments. Supported by the ICARO Project Grant, TIN2005-09098-C05-01 (Spanish Ministry of Education and Science), and Applied Systems Biology Project, P07-TIC-02978 (Innovation, Science and Enterprise Ministry of the regional government of the Junta de Andalucía).

References

[1] Tore Risch and Vanja Josifovski. Distributed data integration by object-oriented mediator servers. *Concurrency and Comoutation: Practice and Experience*, 14:1-21, 2001.

[2] Anthony Tomasic, Rémy Amouroux, Philippe Bonnet, Olga Kapitskaia, Hubert Naacke, and Louiqa Raschid. The distributed information search component (disco) and the world wide web. In *Proceeding of the 1997ACM SIGMOD International Conference on Management of Data*, pages 546-548, 1997.

[3] H. Garcia-Molina, Y. Papakonstantinou, D. Quass, A. Rajaraman, Y. Sagiv, J. Ullman, V. Vassalos, and J. Widom. The tsimmis aproach to mediation: Data models and languages. *Journal of Intelligent Information Systems*, 8(2):117-132, 1997.

[4] L. Haas, D. Kossmann, E. Wimmers, and J. Yang. An optimizer for heterogeneous systems with nonstandard data and search capabilities. *Data Engineering Bulletin*, 19:37-44, 1996.

[5] Tomasz Ksiezyk, Gale Martin, and Qing Jia. Infosleuth: Agent-based system for data integration and analysis. In *Proceedings of the 25th International Computer Software and Applicartions Conference on Invigorating Software Development*, page 474, 2001.

[6] Domenico Beneventano, Sonia Bergamaschi, Silviana Castano, Alberto Corni, R. Guidetti, G. Malvezzi, Michele Melchiori, and Maurizio Vincini. Information integration: The momis project demonstration. In *Proceedings of the 26th Intrnational Conference on Very Large Data Bases*, pages 611-614, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[7] Christof BornhÄovd ad Alejandro P. Buchmann. A prototype for metadata-based integration of internet sources. In *Proceedings of the 11th International*

Conference on Advanced Information Systems Engineering, volume 1626 of *LNCS*, 1999.

[8] Eduardo Mena, Vipul Kashyap, Amit P. Sheth, and Arantza Illarramendi. OBSERVER: An approach for query processing in global information systems based on interoperation across pre-existing ontologies. In *Conference on Cooperative Information Systems*, pages 14-25, 1996.

[9] Othmane Chniber; Amine Kerzazi; Ismael Navas-Delgado y José F.Aldana-Montes. KOMF: the Khaos ontology-based mediation framework. NETTAB2008: Bioinformatics Methods for Biomedical Complex System Applications. Varenna, Italia. 2008.

[10] Ismael Navas-Delgado, Raúl Montañez, Almudena Pino-Ángeles, Aurelio A. Moya-García, José Luis Urdiales, Francisca Sánchez-Jiménez, José F. Aldana-Montes. AMMO-Prot: ASP Model Finder. BMC Bioinformatics. Biomed Central Ltd. 2008. ISSN: 1471-2105 (JCR Impact factor: 3.617).

[11]. Haarslev, V., M¨oller, R. RACER System Description. Proceedings of International Joint Conference on Automated Reasoning, IJCAR'2001, Springer-Verlag, 2001.

[12] Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur and Yarden Katz. Pellet: A practical OWL-DL reasoner, Journal of Web Semantics, 5(2), 2007.

[13] Maria del Mar Roldán-García, Jose F. Aldana-Montes. DBOWL: Towards a Scalable and Persistent OWL reasoner. The Third International Conference on Internet and Web Applications and Services. ICIW 2008. June 8-13, 2008. Athens, Greece.

[14] A. Freier, R. Hofestädt, M. Lange, U. Scholz and A. Stephanik. BioDataServer: A SQL–based service for the online integration of life science data. In Silico Biology, 2002. Online Journal: http://www.bioinfo.de/isb/2002/02/0005/.

[15] Maria del Mar Roldán-García, Joaquin J. Molina-Castro, Jose F. Aldana-Montes. ECQ: A Simple Query Language for the Semantic Web. 7th International Workshop on Web Semantics, WebS '08. DEXA 2008. September, 1-5, 2008. Turin, Italy.

[16] Protegé. http://protege.stanford.edu

[17] Nazila Salamat-Miller, Jianwen Fang, Christopher W. Seidel, Aaron M. Smalter, Yassen Assenov, Mario Albrecht and C. Russell Middaugh. A Network-based Analysis of Polyanion-binding Proteins Utilizing Yeast Protein Arrays. Molecular & Cellular Proteomics 5:2263-2278, 2006.

[18] Dong-Yup Lee , Rajib Saha , Faraaz Noor Khan Yusufi , Wonjun Park , and Iftekhar A. Karimi. Web-based applications for building, managing and analysing kinetic models of biological systems. Briefings in Bioinformatics Advance Access published on September 19, 2008.