A Semantic Web based Architecture for Analytical Tools

Denilson Sell[1,2], Liliana Cabral[2], Enrico Motta[2], John Domingue[2],
Farshad Hakimpour[2] and Roberto Pacheco[1]
[1]*Stela Group and INE, Universidade Federal de Santa Catarina, Brazil*
[2]*Knowledge Media Institute, The Open University, Milton Keynes, UK*
{denilson, pacheco}@stela.ufsc.br, {l.s.cabral, e.motta, j.b.domingue, f.hakimpour}@open.ac.uk

Abstract

Despite the importance of analytical tools to organizations, there are challenges that should be tackled in order to leverage the impact of those tools in the decision making process. These challenges include difficulties to extend those tools according to the business requirements and lack of flexibility to customize information presentation according to users' profile. We argue that these issues are due to the lack of integration of business' semantics into the foundations of analytical tools. In this paper, we describe how Semantic Web technologies and business semantics were applied on the conception of an architecture for analytical tools. Our ultimate goals are to contribute to the resolution of the above issues and support decision makers from the investigation of their business to the implementation of actions according to insights obtained in their investigations. Such decisions and actions are captured to form a knowledge repository for supporting future decisions.

1 Introduction

Fierce competition in the digital economy and increasing volume of available data are forcing organizations to find efficient ways to gain valuable information and knowledge to improve the efficiency of their business processes. Business Intelligence (BI) solutions offer the means to transform data to information and derive knowledge through analytical tools in order to support decision making. Analytical tools should support decision makers to find information quickly and enable them to make well-informed decisions. However, the following critical issues should be tackled in order to make analytical tools more effective in the decision making process:

a) lack of flexibility for extension of the exploratory capabilities of analytical tools;
b) no support for definition of business logic in order to get pro-active information and advises in the decision making;
c) lack of support to present information according to the different kinds of users that can be found in an organization.

The first challenge is related to the way that analytical tools are conceived. Most of those tools come with a limited set of exploratory functionalities and do not provide scalable ways for extension of these functionalities. Current trends on application integration include XML, Web Services and now Semantic Web Services (SWS) as instruments to integrate data, applications and business processes. We argue that these technologies should be applied for the integration of existent code to analytical tools in order to extend the possibilities of analyses and support decision makers to convert insights in actions in a seamless process.

The second challenge is related to the existent gap between analytical tools and the business semantics. In order to improve the support for the decision making process, one should understand how conceptualizations that comprise business logic can be captured, represented, and processed by analytical tools in order to offer more tailored exploratory functionalities than just the generic OLAP functionalities. The developments in on-demand content and business logic availability through technologies such as ontologies and SWS offer the potential to allow organizations to create intelligence driven information chains enabling prompt information access for well-informed decisions. Equally important is the adaptation and assimilation of such technologies to enable the presentation of information according to users' profile, the subject related to the third challenge.

In order to solve the issues described above, we subscribe to semantic technologies to define an integrated architecture for analytical tools. The architecture is supported by business semantics that, in turn, are applied to contextualize the organizations' resources (i.e. logic, data sources and services). The architecture comprises a set of modules to support automated recommendation of analysis, inferences, relations and SWS according to the context of an analysis. SWS and logic reasoning are applied to support flexible extension of exploratory functionalities and powerful analyses. Functionalities for SWS composition and orchestration are integrated to the analytical environment in order to support the definition of complex analyses or to support actions over the business motivated by insights obtained through the exploratory facilities. Information about these analyses and actions made by decision makers are captured to form an important repository of tacit knowledge that can support future decisions. We present how the potentialities of our architecture can be used to leverage analytical tools through a prototype.

The rest of the paper is organized as follows. In section 2, we present a brief introduction to Business Intelligence. In Section 3, we introduce the Semantic Web technologies applied in our architecture to tackle the issues described above. Section 4 shows the architecture modules. Section 5 presents an analytical tool implemented on top of our architecture. Section 6 describes the related work and a discussion. Finally, in section 7, we present the conclusion and future work.

2 Business Intelligence

Business Intelligence (BI) is defined as an integrated set of tools to support the transformation of data to information to support decision making. However, organizations are running to a broader view of BI which comprehends also the ability to analyze information in the context of individual needs and to the use of knowledge management technologies to speed the process of making knowledgeable decisions.

A typical BI architecture contains a *Data Warehouse* (*DW*), an *Extraction, Transformation and Loading* tool (*ETL*) and an *analytical tool*. DW is an integrated repository of data consolidated from different data sources through *ETL* tools. Usually, the approach used for data modeling in DW is the *star schema* [1], which defines that descriptions of the business (e.g. product description) are stored in *dimensions*, while the measures (e.g. amount of items sold) are kept in *fact* tables. DW supplies the data that is presented to the user through analytical tools. Different kinds of analytical tools such as the *On-Line Analytical Processing* (*OLAP*) tools are used to provide the means for users to define their *analyses* (i.e. reports or cubes) and explore the results through analytical functionalities. Typical analytical functionalities includes [1]: *slice* (i.e. reduces the dimensionality of a cube); *dice* (i.e. selects a set of data); *drill-up* (i.e. aggregates data along a hierarchy in a dimension); *drill-down* (i.e. increase the detail by descending along aggregation hierarchies); and *drill-across* (i.e. moves from a cube to another one).

3 Semantic Web Technologies

The main objective of the Semantic Web is to enable the description of Web resources in such way that it will be possible for machines to locate content and to reason over Web resources [2]. Ontologies are one of the main artifacts used to leverage the current Web to the Semantic Web. Ontologies are conceptualizations about a particular domain. Knowledge is represented in ontologies by the use of classes, properties and relations. Relations are used to describe conceptual links between classes and properties. Additional forms of relations include *transitive*, *symmetric*, *equivalence* and *difference*. The knowledge represented through ontologies can be explored by inference engines to generate further knowledge. We are currently applying OCML [4] to support the knowledge representation and inferences in our architecture to provide powerful exploratory functionalities in analytical tools.

Another key component to enable the interoperability on the Web is the Web Service technology. The main objectives of this technology are to allow the code reusing and to enable the integration of heterogeneous applications [2]. The application of ontologies for the description of Semantic Web Services (SWS) and correlated infrastructures facilitate activities including automatic discovery and composition of Web Services [2] [3]. In our approach, we subscribe to IRS-III SWS framework [3] in order to support the recommendation of SWS in the decision making time in our architecture. IRS-III enables the transformation of programming code into a Web service and the invocation of Web services via goals (i.e. abstract representations of tasks), supporting capability-driven service invocation in our architecture.

4 An Architecture for Analytical Tools

In this section, we present how we organize the modules of our architecture. The architecture comprehends a set of loosely-coupled modules that are illustrated in the Fig. 1. Due to the lack of space, we are going to emphasize the role of the ontologies, more specifically of the BI Ontology.

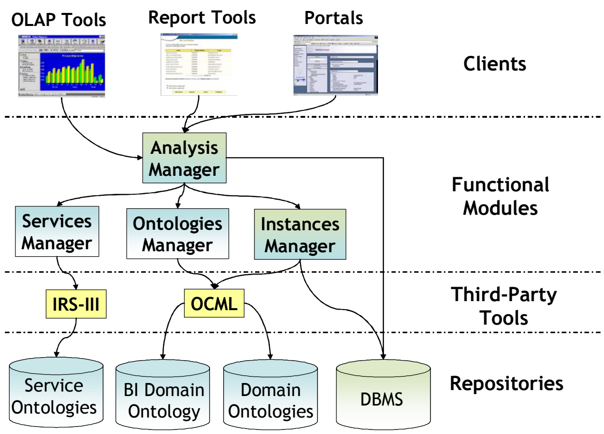

Figure 1. Illustration of the architecture modules.

4.1 Domain Ontology

This ontology provides the formally specified terminology of the business domain that is being supported by the architecture. For example, a Domain Ontology for the R&D domain comprehends concepts such as university, researcher, and so on. This ontology provides the business terminology that will support the description of the SWS in the Services Ontology and to contextualize the data sources in the BI Ontology. Therefore, users will be able to browse company resources (i.e. logic, data and services) using business concepts instead of technical descriptions. Also, the relations, rules and logical expressions related to the business concepts will support query definition, query rewrite and extraction of further details over the results of queries. The integration of domain concepts into the remaining ontologies is described in the next sections.

4.2 Business Intelligence Ontology

The BI Ontology models the concepts used to describe how the data is organized in the data sources and to map such data to the concepts described in the Domain Ontology. These definitions are used to: a) support inferences using the Domain Ontology to extend the results of queries; b) support the presentation of query results according to users' profile; and c) guide the update of the instances in the Domain Ontology. Our approach to support query rewrite is similar to the one presented in [6], but we apply more extensively the business semantics described in the Domain Ontology to support the rewrite of queries' conditions and to combine OLAP features in this process. The concepts used to describe the analysis formatted by users are also defined in this ontology. These semantics support the automatic recommendation of analysis according to the context of users' explorations in order to guide the decision making, feature inexistent in current analytical tools.

Some systems are applying ontologies to describe the organization of data sources in order to support data integration and queries [5] [6]. However, since in these systems the data is not replicated as instances to form a knowledge base, inference requires highly customized engines. On the other hand, by replicating all the data as instances into the ontology, one can easily apply generic inference engines to explore the semantic representation. However, it is complex and expensive to maintain such replication. Typical OLAP queries are not trivially reproduced in query languages for ontologies. Such queries can comprehend several restrictions and grouping operators.

Therefore, in our approach, we process queries over the data sources, but using the Domain Ontology to support the rewrite of conditions in order to broad the results of a query and to support inferences over the results of the queries. We verified that most of the data that could be useful to support inferences and query rewrite are stored in the dimensions of a DW. Dimensions are typically small tables that are not object of constant updating [1]. Thus, we generate instances in the Domain Ontology using the data stored in the dimensions. These instances support users to filter or expand the results of queries relying on synonyms, hyponyms and other relations specified in the Domain Ontology. The rewrite process is guided by the users of analytical tools, who choose which inference will be made to extend the results of their queries. For instance, the user could be interested in finding products that are related to the same category of one particular product listed in the result of an analysis. In this example, it is assumed that the categorization is not described in the DW but expressed as a transitive relation in the Domain Ontology. The query rewrite is made according to the following steps:

1. By using a transitive relation described in the Domain Ontology, the inference engine finds the superclass of that product and which products subsumes the superclass found;
2. The primary keys of the products inferred are included in the condition of the original *select* command as an *in* clause;
3. Other restrictions related to the dimension product are removed from the query;
4. The new query is submitted to the database.

Basically, the query rewrite works as described for any kind of inference. The main difference between our approach and the one described in [6] is that we do not rewrite queries only by using hyponyms and synonyms, but using other kinds of relations and logic described in the Domain Ontology in order to support a user-guided query rewrite process. Transitive relations defined between business concepts are used to support exploratory operations such as *drill-down* and *drill-up* over the results of an analysis. Also, symmetric relations can support and extend operations such as *drill-across* over the data sources, without necessarily relying on the relationships between dimensions and fact tables. One can define a relation such as "product is supplied by company" and use this relation to analyze the prices offered by different suppliers, or by using the same relation, analyze the volume of sales by supplier. More sophisticated relations based on logical expressions can support even more powerful inferences over data sources. Such expressions can support questions such as "what was the income of my *competitors* in the last quarter?", being *competitor* a relation specified as "*companies* that sell the same *kind of product* in the same *region*". This relation is expressed in OCML as described in the Listing 1.

Listing 1. The company-competitors relation.

```
(def-relation company-competitors (?c1 ?c2)
:sufficient (and (region ?r)
                 (product ?p)
                 (is_located_region ?c1 ?r)
                 (sells_product ?c1 ?p)
                 (is_located_region ?c2 ?r)
                 (sells_product ?c2 ?p)
                 (not (= ?c1 ?c2))))
```

The process above and the update of the instances in the Domain Ontology are guided by a set of classes that maps database and domain concepts. *Dimension* and *Fact* classes are described as children of the *DB_Collection* class. Mappings between the attributes in data sources and concepts in the Domain Ontology are made by the *DB_Attribute* class. The descriptions of these classes are presented in the Listing 2. The *has_attribute_name* property in the *DB_Attribute* class is used to identify a field of one particular table, while the *has_concept* and *has_slot* identify concepts in the Domain Ontology. The *has_pk_attribute* in the *DB_Collection* class is used to return the values used to rewrite the condition of queries.

Listing 2. Partial definition of classes used to map data sources and domain concepts.

```
(def-class DB_Collection (DB_Element)
((has_pk_attribute :type DB_Attribute)
 (has_attribute :type DB_Attribute)
 (has_collection_name :type string)))

(def-class DB_Attribute (Element)
((has_concept :type Class)
 (has_slot :type Slot)
 (is_summarized_by :type Summarization)
 (is_additive :type boolean)
 (has_presentation_format :type string)
 (has_attribute_name :type string)))
```

The BI ontology also comprehends concepts used to support the presentation of the data sources structure and of query results according to users' profile. The *label-role* relation described in the Listing 3 is used to infer the description for a *DB_Attribute* from a *Label* class according to the user role.

Listing 3. OCML definition of label-role relation.

```
(def-relation label-role (?class ?role ?label)
 :sufficient
  (and (has_attribute_name ?att ?class)
       (has_label_db_element ?db ?att)
       (has_label_user_role ?db ?role)
       (has_label_description ?db ?label)))
```

Analyses defined by the user in the analytical tools are represented as instances of the *Analysis* class. The definition of an analysis comprises dimensions, measures, filters, privileges and parameters. Parameters are used to bind values in filter definitions. Like dimensions and measures, parameters are represented by domain concepts defined in the Domain Ontology. Thus, one can find analysis according to the synonyms, hyponyms and other relations of the concepts used in the analysis definition.

Listing 4. Partial OCML definition of the classes used in the Analysis definitions.

```
(def-class Analysis () (
 (has_anl_description :type string)
 (has_anl_content   :type Analysis_Content)
 (has_anl_dimension :type Analysis_Dimension)
 (has_anl_filter    :type Analysis_Filter)
 (has_anl_parameter :type Analysis_Parameter)
 (has_anl_creator   :type User)
 (has_anl_allowed_user   :type User)
 (has_anl_allowed_role   :type Role)))

(def-class Analysis_Filter () (
 (has_filter_attribute :type DB_Attribute)
 (has_filter_operator :type DB_Operation)
 (has_filter_value :type Class)
 (has_filter_conector :type DB_Conector)))

(def-class Analysis_Parameter () (
 (has_parameter_filter :type Analysis_Filter)
 (is_parameter_single_value :type boolean)))
```

4.3 Instance Manager

The Instance Manager supports the replication of data from the dimensions into the Domain Ontology in a batch process. The Instance Manager is guided by the mappings described in the BI Ontology and by information about updating periodicity also described in the BI Ontology. The module is implemented as a collection of Java classes that connect to the desired data source through JDBC drivers and automatically extract the data needed and generate the instances (currently in OCML) in the ontology server.

4.4 Ontology Manager

The Ontology Manager is the module that provides a set of methods necessary to query and make inferences over the ontology repositories. This module was conceived to hide the complexities of the ontology query languages and ontology repositories from developers. It enables a loose integration among analytical tools, query engine and ontology repositories resulting in more flexibility for changes in the underlying repositories and query engine.

4.5 Service Ontology

The Services Ontology models the concepts used to describe SWS. Currently, this ontology corresponds to the set of ontologies used in the IRS-III framework. The Domain Ontology supplies the business concepts used to describe the inputs and outputs of the SWS and of their non-functional properties (see [3] for details about IRS-III definitions). We have also extended IRS-III ontologies in order to support composition of SWS in our tool. Details about such extensions are given in [8]. SWS descriptions support the discovery and composition of Web services according to an analysis being made by the user of the architecture. Finally, the semantics that describes the compositions of services made by users are also stored in this ontology. Such semantics are used to support the orchestration of compositions and to re-use previous approaches taken by decision makers to analyze the business and to take actions to solve issues identified in the analysis. The integration of composition of SWS into analytical tools support definitions of powerful and flexible analyses in which SWS implemented inside or outside the organization can be used to bring additional data, transform data into information or perform any other kind of transaction motivated by an insight obtained in the analytical tool.

4.6 Service Manager

The goal of the Service Manager is to enable the re-use of existent code to support the improvement of analytical tools' functionalities. It supports automatic recommendation of SWS according to the match of the concepts involved in an analysis and the semantic descriptions of Web Services. In addition, the Service Manager enables the composition and orchestration of SWS. A composition could be fed by the data being analyzed for a user or could provide the input parameters to another analysis. This module works as an interface to a SWS infrastructure. It corresponds to a Java API that is integrated to the Service Ontology in order to support the discovery, composition, invocation and monitoring of SWS. Right now, the API is integrated to IRS-III through the IRS-III Java API [3] and through our Java API for orchestration [8].

4.7 Analysis Manager

The Analysis Manager is the module that provides access to all the components in our architecture. It intermediates the access to the Ontology Manager, to the Service Manager and to the Instance Manager. Also, it provides a set of functionalities to support OLAP over data sources and implements the query rewrite process described previously.

5 OntoDSS - An Analytical Tool to Support Decision Making

In this section, we present OntoDSS, a prototype developed in Java designed to validate our architecture and illustrate the benefits of our approach to the decision making process. OntoDSS is divided in two main modules, namely: a) *Analysis Definition*, which supports users defining an analysis; and b) *Exploration Recommender*, which supports users on the composition of services in explorations more sophisticated and which recommends relations, inferences and services that could provide different perspectives in users' analyses. OntoDSS functionalities will be presented in the context of Scienti Network (SN) [9], an international knowledge network that comprehends data about the CVs of researchers. We describe how our approach support the needs of decision makers over a DW built from a sample of CVs extracted from SN. The domain concepts and corresponding mappings to the DW were defined in OCML. The instances in the Domain Ontology were generated by the Instance Manager.

5.1 Analysis Definition

The Analysis Definition module supports users on the definition of content and format of their analysis. Users browse the definition of their data sources using the BI Ontology in order to select the data items that they want to include in their analysis. The vocabulary used to present such definitions to the user is inferred from the relations between the data sources and users profiles in the BI ontology in order to facilitate the comprehension of the meaning of the information. For instance, a manager of a university will visualize the classification of persons in the DW as *researchers* and *students*. A company manager will visualize the same classification as *experts* and *trainees*.

5.2 Exploration Recommender

This module offers alternatives for exploration of the result of an analysis to a user. Besides the traditional OLAP functionalities, it provides exploratory functionalities relying on the semantics defined in the ontology repositories. Such functionalities are presented according to the context of the analysis. The Fig. 2 illustrates the results of an analysis defined in OntoDSS comprehending the number of researchers and students per university. OntoDSS automatically identifies which concepts were used in the analysis definition and use these concepts to recommend explorations to the user. Fig. 2 shows a list of options related to the *university* concept according to the user' selection. Several operations are supported by our architecture, as follows.

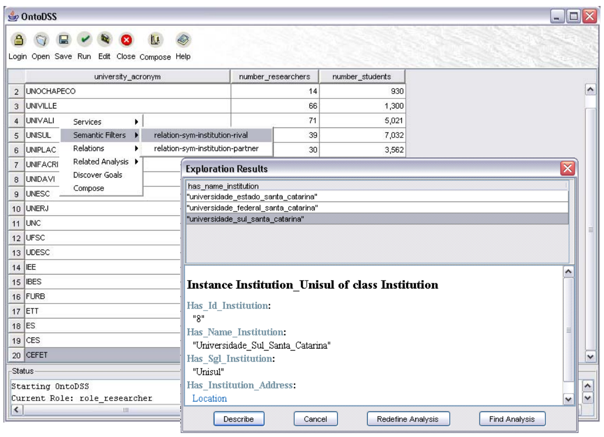

Figure 2. Illustration of recommendations according to the context of an analysis.

5.2.1 Services. In this option, all the SWS retrieved by the Services Manager are listed. The retrieved SWS have at least one input compatible, in this example, to the *university* concept. These services can then perform any kind of transaction such as access external or internal data sources to retrieve more information about the university given as input. The result of SWS is presented to the user and can be used as parameter to other analyses, which, in turn, are recommended through the match of the output of the SWS with the parameters of other analysis.

5.2.2 Semantic Filters. In this option the user will find which inferences can be used to make further explorations over data included in the report. Such inferences are represented as rules or logic expressions in the Domain Ontology and are evaluated by the inference engine. As illustrated in the Fig. 2, the first inference presented in the list was *competitors*, a relation described as "*universities* that offer *related courses* in the same *region*". The list of universities that holds this expression is presented in another dialog. The *Rewrite Analysis* button fires the reformulation of the original query in order to present a comparison between the universities retrieved in the inference. This functionality enables users to explore the business semantics to formulate more complex queries not supported by current OLAP functionalities.

5.2.3 Relations. In this option are listed all the relations that the university concept has. Such relations can guide the user in a tour through all the conceptual definitions provided in the Domain Ontology and may help users to analyze their business from all possible perspectives. Users can navigate through all the relations defined in the Domain Ontology not only to explore the fact tables, but also to explore the instances defined in this ontology. This functionality supports explorations more flexible than the current drill-down, drill-up and drill-across offered by OLAP tools, because it does not depend on relations and hierarchies defined in the database. One can easily define at any time a relation in the Domain Ontology and use this relation to support such explorations.

5.2.4 Related Analyses. In this option users can get a list of all the analyses that have at least one parameter that matches, in this example, the term university. This feature enables users to continue the exploration by opening analyses previously defined that are related to the current explorations. The recommendation of analysis is supported also by taxonomies and synonyms of the concepts used as parameter of other analysis.

5.2.5 Discover Goal. Users can find Goals and Web Services that could be useful to support further exploration of their analysis or to perform any other desired action using as input the data retrieved in the analysis. Users can define a search criterion by using SWS properties and logical operators. The Service Manager translates the arguments according to the syntax accepted by the SWS infrastructure being used.

5.2.6 Composer. This functionality was designed to support the definition of a complex analysis that could comprehend more than one service to process all the information and transactions required. The composer supports a user-guided composition approach, by recommending Web services and analyses according to the composition context (see [8] for a detailed description of our composition tool). The Fig. 3 depicts the composition tool and some of its functionalities. The tool guides users in a step-by-step composition process in which Goals, mediators and control flow operators are added to the composition. This interactive process is supported by the tool, which in each step recommends Goals by matching the inputs and outputs of the Goals that were previously selected considering also the subsumption of the input and output types. Once a composite service has been defined, the composition tool instantiates the workflow using our orchestration engine. The results of the composition can be used as parameters to another analysis, used to support the rewrite of the original analysis or simply be presented to the user. The Fig. 3 illustrates a composition made to handle a complex analysis. The first service checks which indicators are bellow the goals defined by the government regulations for a university. A condition is defined to guide the flow according to the indicators retrieved by the first service in order to find out more information about the indicators. The compositions defined can then be stored for posterior utilization thus forming a rich knowledge repository to support further decisions.

6 Discussion and Related Work

Semantic Web technologies have been applied in different forms to tackle traditional issues related to information systems but not specifically in the context of analytical tools. Ontologies have been applied in systems like Observer [5] and Tambis [6] in order to support the translation of queries in sub-queries that are processed over distributed data sources. Ontologies are also applied in query rewrite using synonyms and hyponyms to extend the results of queries [7]. In our approach, queries are performed over data sources because complex OLAP queries are not supported by current ontology query languages. In addition, current strategies for ontology persistence are also far to offer the same performance provided by database vendors. We are replicating only the data of dimension tables in the Domain Ontology to support further inferences over the results of queries. The full replication of the DW in an ontology repository is unfeasible. Dimensions contain most of the data used in our inferences.

Current analytical tools do not offer scalable ways for the aggregation of new functionalities. Frequently, users have to develop the required extension from the scratch, even when there are code already implemented that could be used. We offer a scalable solution for this issue in our architecture through SWS and ontologies. Existent code found in the organization or on the Web can be semantically described and easily integrated to analytical tools through our architecture. Such semantic description supports the discovery, composition and invocation of services related to users' analysis. In addition, definitions of business concepts and logic are done directly in the Domain Ontology, so one can modify or include definitions and change the behavior of analytical tools at any time. Therefore, modifications no longer depend on hours of coding, and maintenance of the system can be done remotely

Our composition tool is a step towards an automatic composition of services. Such tool suggests goals according to a composition context and support definition of mediators and control operators in an interactive composition process. Other composition tools such as CAT [10] and Mindswap Composer [11] also support a semi-automatic composition guided by the user, but do not support neither mediators nor control components. The adoption of mediators gives more flexibility to users, since it is inevitable to select services defined and implemented by different parties while building a composition. Mediators enable also the integration of the composition tool to the analytical environment. Virtually, there are no references in the literature describing the integration of SWS composition tools to analytical tools. The result of a composition can be used as input for an analysis and the outputs of an analysis can be used as inputs for the composition. In addition to support the definition of complex explorations, such integration could support users to perform an action over their business according to an insight obtained in an analysis. For instance, a manager can define a composition using several SWS to find products whose sales have decreased, to check the price of the competitors, to verify the internal cost of the products and to check whether it is feasible to offer more competitive prices. Finally, users can use other SWS to change the price of the product according to the final decision. Compositions like this represent the approaches taken by decision makers to analyze and make actions over the business and can be used to support future decisions and actions. This interactive process enables the capture of tacit knowledge from decision makers forming a rich knowledge repository to the organizations.

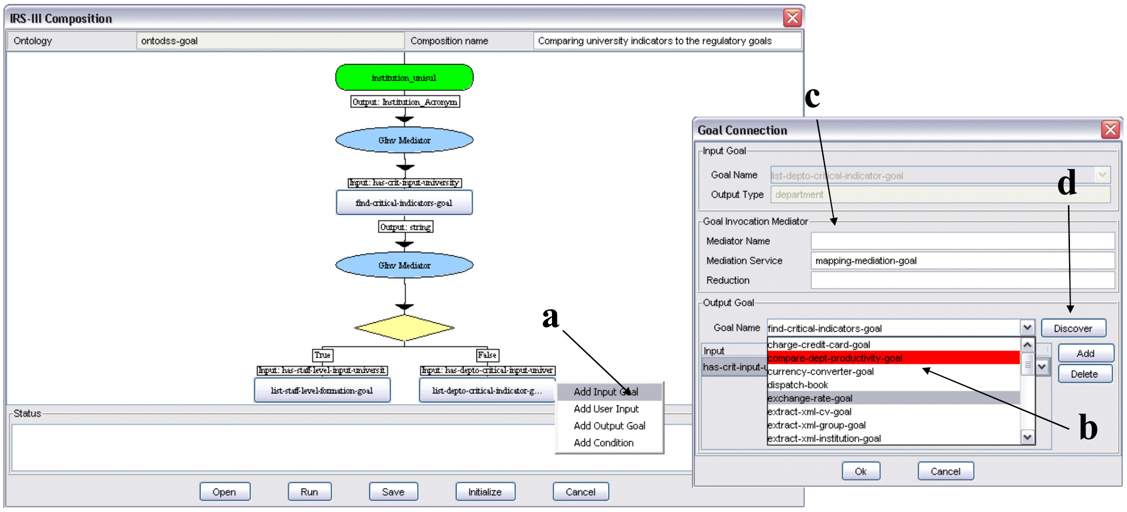

Figure 3. Illustration of a composition. Users interactively define a composition (a) receiving recommendations of Goals (b). Users can define mediators (c) or call the discovery functionality (d).

7 Conclusion and Future Work

Our architecture supports a seamless integration of business semantics, data and services into the decision making process. We described how the incorporation of inference power in the foundation of our architecture enabled the development of an analytical tool that can be easily adapted and extended according to the decision makers needs. Information is presented to users according to their own vocabulary, facilitating the comprehension of the meaning of the information. Powerful exploratory capabilities combining OLAP functionalities with ontological inferences are offered to the user. SWS are applied to provide flexible extensions of exploratory functionalities. The semantic descriptions of services, analysis and data support the automatic recommendations of relevant resources and guide users in an interactive decision making process. A composition tool is offered in order to support the definition of complex analyses or to support actions over the business. Such compositions form an important tacit knowledge repository that describes the approach taken by decision makers to analyze the business and take actions. This knowledge repository can support future decisions or provide the means for automation of actions through agents. Ongoing work includes the formal definition of the implemented semantic-OLAP functionalities. Future work comprehends capture of further information about decision makers' interactions with the available functionalities. We are investigating how to extract rules from this information and from the compositions made aiming to support automatic analysis of the business and recommendation of actions.

Acknowledgement

This work is supported by CNPq, Brazil and by the Data, Information and Process Integration project.

References

[1] R. Kimball. "The Data Warehouse Lifecycle Toolkit". John Wiley & Sons Inc., New York, 1998.
[2] H.P. Alesso and C.F.Smith. "Developing Semantic Web Services". A.K Peters, Massachusetts, 2004.
[3] J. Domingue et. al., "IRS-III: A Platform and Infrastructure for Creating WSMO-based Semantic Web Services", Proc. of the Workshop on WSMO Implementations, Frankfurt, Germany, September 2004.
[4] E. Motta, "Reusable Components for Knowledge Modelling". IOS Press, Amsterdam, The Netherlands, 1999.
[5] E. Mena et. al., "Observer: An Approach for Query Processing in Global Information Systems". Proc. Of Conference on Cooperative Systems, pp. 14-25, 1996.
[6] N.W.Paton et. al.,"Query Processing in the Tambis Bioinformatics Source Integration System". Statistical and Scientific Database Management, pp. 138-14, 1999.
[7] C.B. Necib and J. Freitag. "Ontology based Query Processing in Database Management Systems". Proc. of the International Conference on Ontologies, Databases and Applications of Semantics. Catania, Italy, November 2003.
[8] D. Sell et. al., "Interactive Composition of WSMO-based Semantic Web Services in IRS-III". Proc. of 1st AKT Workshop of Semantic Web Services. 2004.
[9] R. Pacheco, "Rede Scienti". VI Congreso Regional de Información in Ciencias de la Salud. Puebla, Mexico. 2003.
[10] J. Kim, M. Spraragen and Y. Gil, "An Intelligent Assistant for Interactive Workflow Composition", Proc. of the Int'l Conf. on Intelligent User Interfaces, Portugal, 2004.
[11] E.Sirin, B.Parsia and J.Hendler, "Filtering semantic Web services with interactive composition techniques". IEEE Intelligent Systems,Vol.19, Issue 4, 2004, pp. 42-49.