Adding Semantics to Business Intelligence

Denilson Sell[1,2], Liliana Cabral[2], Enrico Motta[2,] John Domingue[2] and Roberto Pacheco[1,3]
[1]*Stela Group, Universidade Federal de Santa Catarina, Brazil*
[2]*Knowledge Media Institute, The Open University, Milton Keynes, UK*
[3] *INE, Universidade Federal de Santa Catarina, Brazil*
{denilson, pacheco}@stela.ufsc.br, {l.s.cabral, e.motta, j.b.domingue}@open.ac.uk

Abstract

Despite the importance of analytical tools to organisations, they still lack the inference power needed to solve the requests of decision makers in a flexible way. Our approach aims at integrating business semantics into analytical tools by providing semantic descriptions of exploratory functionalities and available services. We propose an architecture for business intelligence, which uses semantic web technology based on IRS-III. In addition, we present OntoDSS, a prototype tool based on this architecture that illustrates some of the functionalities that may be provided to decision makers within an application scenario.

1 Introduction

Fierce competition in the digital economy and increasing volume of available data are forcing organisations to find efficient ways of obtaining 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 the right information quickly and enable them to make well-informed decisions.

According to IDC, the market for Business Intelligence software is already worth more than $7 billion worldwide and is expected to double by 2006. Forrester Research states that more than 44% of companies will adopt BI software within 2005 (IBM, 2004). By applying semantic web technology to analytical tools we expect to tackle some of the main issues:

- Lack of flexibility for extension of the exploratory capabilities.
- No support for definition of business rules in order to get pro-active information and advises in the decision making;
- Lack of support to present information according to the different kinds of users that can be found in an organization.

Most of those tools come with a limited set of exploratory functionalities such as *drill-down*, *drill-up*, *slice* and *dice,* which operate over data sources only at the structural level. Some of these tools allow users to extend those functionalities but in non-scalable ways such as through specific programming languages that may not be known by the developers of the organization.

We have developed an architecture in which conceptualizations for business analysis can be captured, represented and processed in order to offer the decision maker more tailored and flexible exploratory functionalities. Ontologies and Semantic Web Services based on IRS-III [3] are applied to support the semantic extension of exploratory functionalities and to provide semantic filters and services when defining complex analysis. In addition, we present OntoDSS, a prototype tool based on this architecture that illustrates some of the functionalities that may be provided to decision makers within an application scenario.

The rest of the paper is organized as follows. In section 2, we present a brief introduction to business intelligence and analytical tools. Section 3 presents the semantic web technology applied. Section 4 presents an architecture for semantic based analytical tools. Section 5 presents a prototype tool implemented using this architecture. Section 5 describes the related work and a discussion about our approach. Finally, in section 6, we discuss the work done and present conclusions.

2 Business Intelligence and Analytical tools

Business Intelligence (BI) is defined as an integrated set of tools to support the transformation of data into information in order to support decision making. However, organizations are expecting a broader use of BI which comprehends also the ability to analyse information in the context of individual needs and the use of knowledge management technologies to speed up 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 and Semantic Web Services

The main objective of the Semantic Web is to enable the description of Web contents in such a way that it will be possible for programs to locate and reason over Web resources. Ontologies are one of the main artifacts used to leverage the current Web to the Semantic Web.

Another key technology used in our application is Semantic Web Services (SWS). They enable the semantic interoperability of distributed services on top of data (XML) and protocol (SOAP) standards. The semantic description of Web Service functionalities facilitates activities such as automatic discovery and composition of Web Services.

In our approach we use IRS-III [3], a framework and implemented infrastructure for Semantic Web Services compliant with WSMO [9]. The main components of WSMO are Ontologies, Goals, Web Services, and Mediators. Ontologies provide the basic glue for semantic interoperability and are used by the three other components. Goals represent the types of objectives that users would like to achieve via Web services. A Web service may be selected by a discovery process and then executed when a goal is required. Web services descriptions describe the functional behavior of an actual Web service. Mediators provide the means to link two components together, defining mappings between them.

In our approach, we use ontologies to create the necessary knowledge models for defining exploratory functionalities in analytical tools, making them driven by the business semantics. More specifically, we use OCML [4] for creating the business intelligence (BI) domain model, the service models and the application domain models. We are then able to define semantic functionalities including filters, relation navigation and semantic web services. By using the IRS-III framework we are able to connect these models in order to describe and invoke semantic web services.

4 An Architecture for Analytical Tools

In this section, we present the architecture we have developed for building semantic-based analytical tools. The architecture is composed of a set of loosely-coupled modules that are illustrated in Figure 1. We first describe the type of ontologies used in the architecture and then how the functional modules process these definitions.

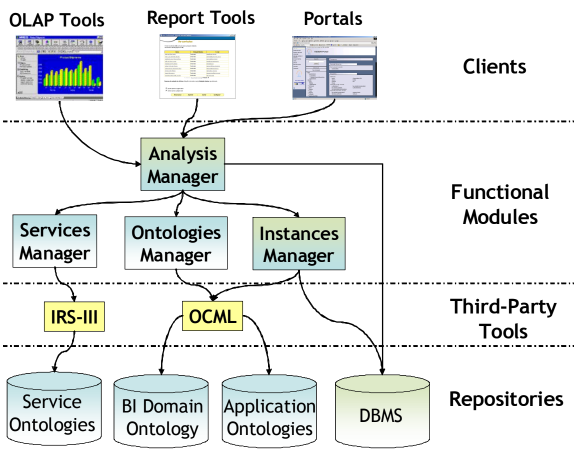

Figure 1. Illustration of the OntoDSS modules.

4.1 Architecture Ontologies

The BI domain ontology models business intelligence as described in section 2. This representation takes advantage of the structure of data sources in terms of dimensions and facts in the DW.

An analysis formatted by a user is defined in this ontology as shown in listing 1. This definition includes dimensions, measures, filters, privileges and parameters. Parameters are used to bind values in filter definition. Like dimensions and measures, parameters are represented by domain concepts defined in the Domain Ontology. Thus, one can find an analysis according to the relations (slots) in the analysis definition.

Listing 1. Partial OCML definition of the classes used in the Analysis definition.

```
(def-class Analysis () (
 (has_description :type string)
 (has_measures   :type Analysis_Measure)
 (has_dimension :type Analysis_Dimension)
 (has_filter    :type Analysis_Filter)
 (has_parameter :type Analysis_Parameter)
 (has_creator   :type User)
 (has_allowed_user   :type User)
 (has_allowed_role   :type Role)))

(def-class Analysis_Filter () (
 (has_attribute :type DB_Attribute)
 (has_operator :type DB_Operation)
 (has_value :type Class)
 (has_conector :type DB_Conector)))

(def-class Analysis_Parameter () (
 (has_filter :type Analysis_Filter)
 (is_single_value :type boolean)))
```

Some supporting classes in the BI ontology are used to associate attributes in dimensions and facts with concepts in the Application domain ontology as shown in listing 2.

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)))
```

Additional definitions are used to extend query results and support their presentation according to the user 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)))
```

Application domain ontologies model specific business concepts that can be bound to the BI ontology as values. Likewise, these ontologies provide the business terminology that will support the description of the SWS in the Services Ontology.

From the user point of view these concepts relate to the analysis they are applying in a specific scenario. For example, an application domain ontology for the R&D domain includes concepts such as university, researcher, and so on (see listing 4).

Listing 4. The research-group concept.

```
(def-class research_group ()
((has_student :type CV)
 (has_researcher :type CV)
 (has_project :type project)
 (has_institution :type institution)
 (has_leader :type CV)
 (has_year_formation :type string)
 (has_name_group :type string)
 (has_id_group :type string)))
```

Application-specific relations and rules can be defined for supporting filter definitions and extension of query results. Listing 5 defines for example the rival-institution relation, which states that universities are rivals if they are located in the same city.

Listing 5. The rival institution relation.

```
(def-relation rival-institution (?i1 ?i2)
  :sufficient (and
              (city ?c)
              (has_address ?i1 ?a1)
              (has_city ?a1 ?c)
              (has_address ?i2 ?a2)
              (has_city ?a2 ?c)
                (not (= ?i1 ?i2))))
```

Service Ontologies contain Semantic Web Service definitions. Currently, these correspond to instances of goal, web service and mediator models used in the IRS-III framework. An Application Domain Ontology supplies the business concepts used to describe goals and web service capabilities (e.g. inputs, outputs, preconditions etc).

A SWS can be used to create built-in query definitions as well as composed services. Listing 6 shows de description of a SWS in IRS-III in terms of goal, web service and mediator. This service finds all researchers belonging to a research group. Researchers are identified by their CV description.

Listing 6. Partial definition of a SWS in IRS-III.

```
(def-class list-researchers-goal(goal)?goal
 ((has-input-role:value has-research-group)
  (has-input-soap-binding:value
         (has-research-group "sexpr"))
  (has-output-role :value has-list-cv)
  (has-output-soap-binding:value
         (has-list-cv "sexpr"))
  (has-research-group :type research_group)
  (has-list-cv :type cv)))

(def-class list-researchers-web-service
         (web-service) ?web-service
 ((has-capability:value
         list-res-web-service-capability)
  (has-interface:value
         list-res-web-service-interface)
  (has-non-functional-properties:value
     list-res-web-service-non-func-props)))

(def-class list-researcher-mediator
       (wg-mediator)?mediator
 ((has-source-component:value
         list-researchers-goal)
  (has-target-component:value
         list-researcher-web-service)))
```

4.2 Functional Modules

The Instance Manager supports the replication of data from the dimensions into an Application Domain Ontology. 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.

The Ontologies 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.

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. SWS can be a powerful mechanism for analytical tools because it allows access to external and distributed services or data sources that can be integrated to an analysis.

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 An application and prototype of an Analytical Tool

In this section, we present OntoDSS, a prototype tool based on our architecture in order to illustrate some of the functionalities provided by a semantic web based analytical tool.

OntoDSS functionalities will be presented in the context of a simple scenario. The scenario describes how our approach could support the needs of decision makers with different perspectives over a data mart built from a sample of CVs extracted from the Scienti Network (SN). SN is an international knowledge network in Science & Technology that comprehends data about the intellectual production and activities of students and researchers in Latin America [8].

Users decide about the content and format of their analysis. They browse the definition of their data sources in order to select the data items that they want to include in their analysis. Parameters are used to declare binding variables used in the filter expressions. For instance, an analysis of the volume of publications could be restricted to a specific period of time. Such period of time can be defined as a parameter, since it should vary constantly according to users' interests, and will be asked to the user every time that this particular analysis is executed. Parameters are also used by the tool to automatically suggest relevant analysis according to context of use of the tool.

Besides the traditional OLAP functionalities, OntoDSS provides a set of exploratory functionalities relying on the semantic descriptions defined in our architecture repositories.

Fig. 2 illustrates the results of an analysis defined by the user in OntoDSS. Such analysis includes universities and correspondent number of students and researchers. Users can right-click over any of the columns in the analysis to see what explorations are

recommended by the tool. OntoDSS automatically identifies which concepts were used in the analysis definition and use these concepts to recommend explorations to the user.
In the service option, OntoDSS lists all the SWS that have at least one input compatible, in this example, to the *university* concept. This match is supported also by the subsumption of the concepts defined in the Application Domain Ontology. The user can select one or more universities to be given as input to the SWSs recommended by the user. The result of the selected SWS is presented to the user and can be used as a parameter to another analysis. Analyses that have parameters compatible with the output of the SWS selected are also recommended to the user.

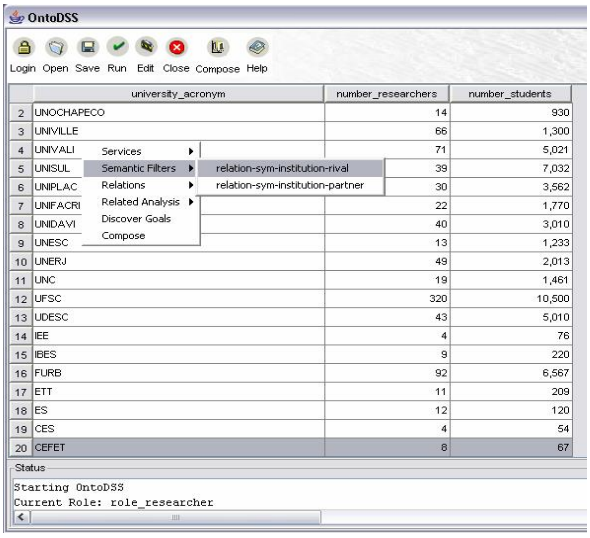

Figure 2 - Illustration of OntoDSS functionalities.

Users can also use semantic filters to make further explorations over one or more universities included in the report. As illustrated in the Fig. 2, the first filter presented in the list was rival. The list of universities that holds this expression is presented in another dialog. The user can then look the details of each instance retrieved and ask to compare such universities clicking on the Rewrite Analysis button, which reformulates the original query in order to present just the universities retrieved in the inference.

The relation option allow users to navigate the Application Domain Ontology. They can analyse, for instance, the researchers and students that have a link to an university.

Another option to the user is to get a list of all the analyses that have at least one parameter that matches, in this example, the term university. This feature enables the user to continue the exploration by opening analysis previously defined. Therefore, the decision maker can have different insights over the object being analyzed by just navigating from one analysis to another. The recommendation of analysis is made by matching terms used in the parameters of other analysis.

6 Discussion and Conclusions

We argue that business semantics should be applied as the backbone for contextualization and integration of data and services in organizations, and present a solution in the context of 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. Also, 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 Application Ontology using the data stored in the dimensions. We are replicating the data of dimensions in the BI Ontology and processing queries over the fact tables directly over the relational database.

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 Application Ontology to support the rewrite of query conditions and to combine OLAP features in this process. The rewrite process is guided by the users of analytical tools, who choose which semantic filter will be applied to extend the results of their queries. For instance, the user could be interested in findind 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 this categorization is not described in the DW but expressed as a relation in the Application Ontology.

Relations also support exploratory operations such as *drill-down* and *drill-up* over the results of an analysis 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 analyse the prices offered by different suppliers, or the volume of sales by supplier.

Semantic Web Services are applied to support flexible extension of exploratory functionalities. The semantic description of services, analysis and data are used on the recommendation of actions to the users.

We provide an integrated architecture for analytical tools that is adaptable to the needs of organizations and individual users. The Business Intelligence ontology is used to support the exploration of data sources as structured in DW and the personalization of the presentation according to users' profile.

As illustrated in our prototype, users receive suggestions of relevant filters, relations and services that can present different perspectives of an analysis. The recommendation is made automatically according to business concepts being explored in an analysis.

One of the advantages of our approach is that modification of the tool behavior no longer depends on many hours of hard coding, and maintenance of the system can be done from remote place.

Acknowledgement

This work is supported by CNPq, Brazil and by DIP (Data, Information and Process Integration with Semantic Web Services) project (EU FP6 - 507483).

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. CEUR Workshop Proceedings, vol. 122

[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] R. Pacheco, "Rede Scienti". VI Congreso Regional de Información in Ciencias de la Salud. Puebla, Mexico. 2003.

[9] WSMO Working Group (2004). D2v1.0. Web Service Modeling Ontology (WSMO). WSMO Working Draft. http://www.wsmo.org/2004/d2/v1.0/

.