3

Adding Semantics to Business Intelligence: Towards a Smarter Generation of Analytical Tools

Denilson Sell[1,2,3], Dhiogo Cardoso da Silva[2], Fernando Benedet Ghisi[1,2], Márcio Napoli[1,2] and José Leomar Todesco[1,2]
[1]*Instituto Stela,*
[2]*UFSC – Universidade Federal de Santa Catarina,*
[3]*UDESC – Universidade do Estado de Santa Catarina,*
Brazil

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.

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, no support to analyze and interpret data 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. Our approach applies ontologies on the description of business rules, information sources and business concepts in order to support semantic-analytical functionalities that extend traditional OLAP operations. Such approach enables developers to customize BI solutions according to organizations' specific analytical requirements and allows developers to align BI solutions to the latest business analytic requirements. In addition, this approach made it possible to offer novel approaches to guide decision makers on the analysis of their business, including recommendation according to users' profile, a question answering approach to access business data and automatic generation of text summaries based on OLAP cubes.

The improvements on knowledge engineering and related technologies offer new approaches to tackle traditional issues in the context of information management. In this chapter, 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 a new generation of analytical tools that may drive decision makers from the investigation of their business to the implementation of actions according to insights obtained in their investigations.

The Semantic Business Intelligence (SBI) architecture presented in this chapter incorporates many features that distinguish it from the existing information management solutions and research. Our work aims at enabling the integration of business semantics, heterogeneous data sources, and knowledge engineering tools in order to support a smarter decision making.

In the first section, we present how we design our architecture and present each of its modules. 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 services according to the context of an analysis. Semantic web services and logic reasoning are applied to support flexible extension of exploratory functionalities and powerful analyses. Information about these analyses and actions made by decision makers are captured to form an important repository of explicit knowledge that can support future decisions.

We present how the potentialities of our architecture were used to leverage analytical tools in different scenarios. On top of our architecture, we developed different strategies in order to provide an intelligent behavior in the analytical environment.

One of the applications described in this chapter shows how we are applying natural language to support decision making and information retrieval. The need to obtain and use knowledge to support the decision making motivates the convergence of the new generations of Business Intelligence (BI) solutions with the Knowledge Engineering tools. Despite application of semantic technologies and methods of knowledge representation, BI research still lacks the use of natural language to conduct analysis. The metaphor of information searching conjectured on the Semantic Web is becoming a trend in the area of BI. Thus, we describe how our architecture made it possible the gathering of strategic information from corporate data sources driven by means of the semantic interpretation of natural language questions. This approach brings to the BI area of the discipline of Question Answering (QA) and the Semantic Web formalisms through an interdisciplinary approach. Some resources of knowledge representation, such as ontology, inference rules, idiomatic patterns and heuristics aid the architecture's functional modules with the interpretation of question and the return of the OLAP cube.

An analytical interface was constructed to allow the entry of questions in the Portuguese language, the interaction with the decision maker to resolve ambiguities and the visualizing hypercubes. As well as the way millions of users search for information on the Web, this research provides an innovative method to aid in the decision making process.

2. The semantic business intelligence architecture

The SBI architecture comprehends a set of loosely-coupled modules that are illustrated in the Fig. 1. The SBI ontologies comprise business semantics and describe the relationship among such semantics, BI terminology, operational semantics, and data sources.

SBI ontologies are used by the QueryManager to parse analytical tools and data requests, and to execute such requests on heterogeneous data sources, enabling the combination of unstructured and structured data on the very same analysis. The OntologyManager is the module that provides access to SBI ontologies. Such module relies on a Reasoner to support on-the-fly and batch based inferences over business semantics. These inferences make semantic driven slice and drill based on business rules possible.

In the following sections, we describe in details the SBI ontologies and its modules.

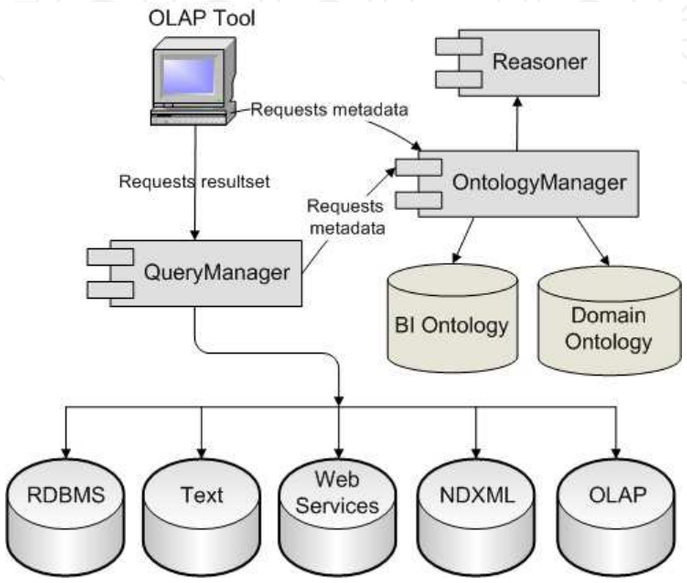


Fig. 1. Illustration of SBI main components

2.1 The SBI ontologies

In our approach, we use ontologies to capture business semantics and to define the necessary knowledge models for generating flexible and exploratory functionalities in analytical tools. In current version, the SBI ontologies have a version modeled in RDF (Lassila & Swick, 2004) and OWL (McGuinnes & Harmelen, 2004). In the following sections, we introduce the domain and the BI ontologies.

2.1.1 The domain ontology

The domain ontology supplies the business terminology used to enable data sources annotation. Therefore, users will be able to explore information repositories by using business concepts instead of technical descriptions. Also, the relations, rules, and logical expressions described in the domain ontology will support semantic drill and slice, query definition, and extraction of further details from data presented by analytical tools.

All required inferences to extend the analytical functionalities are supported by business rules and expressions represented in the domain ontology. Domain specific relations and rules can be defined to slice and drill OLAP cubes. The Listing 1 below defines the *Alumni* rule that infers institutions in which students have completed their degree. A new relation called *isAlumni* between *Person* class and *Institution* class is inferred by this business rule.

$$\text{Person}(?p) \wedge \text{Degree}(?d) \wedge \text{Institution}(?i) \wedge \text{hasDegree}(?p, ?d) \wedge \text{hasInstituion}(?d, ?i) \wedge \text{isCompleted}(?d, \text{'yes'}) \rightarrow \text{isAlumni}(?p, ?i)$$

Listing 1. The Alumni rule: former student of an institution

The notation above states that a person (?p) is a former student of an institution (?i) when she has completed her degree (?d). Business rules, such as alumni depicted in listing 1 are represented in the domain ontology using SWRL (Horrocks et al., 2004).

The mapping of domain concepts to the BI ontology is described in the next sections.

2.1.2 The BI ontology

The BI ontology models the concepts used to describe how the data is organized in 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 query results; b) support the presentation of query results using business terminology; c) provide an abstraction of data sources to guide the interaction of decision maker on the exploration of organizations' information sources. The main concepts related to business intelligence are modeled in the BI ontology. Fig. 2 shows its main constructors.

As depicted in Figure 2, the BI ontology maps OLAP concepts used by analytical tools to guide decision makers on analysis definitions and to provide semantic drill and slice operations. The information source concepts are used to represent data source structures and to map those structures to domain concepts represented in the domain ontology. The Table 1 presents more details about BI ontology concepts.

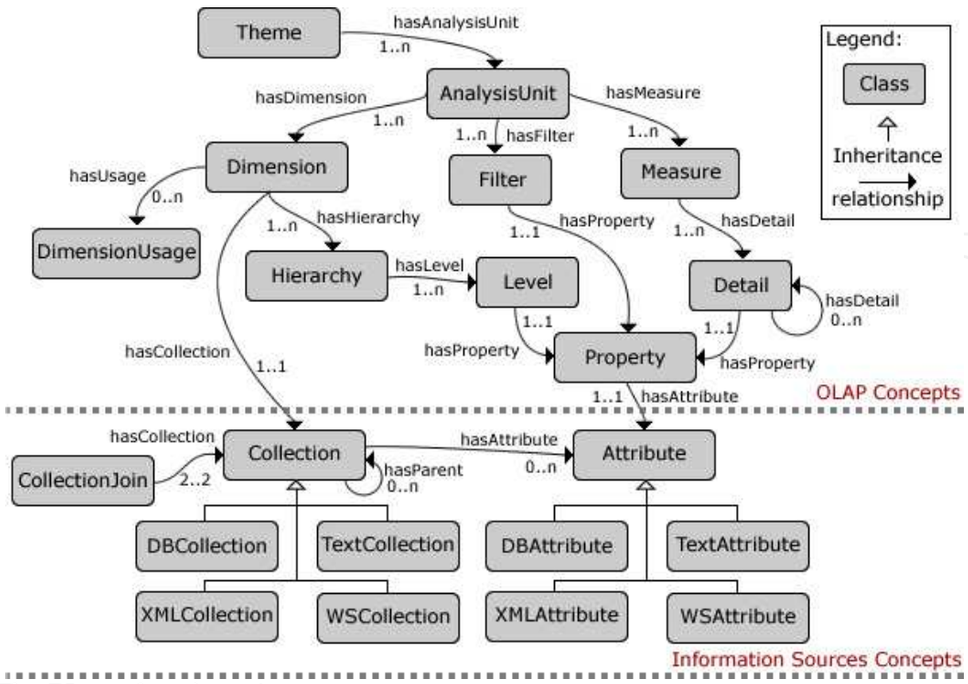


Fig. 2. The BI ontology's main constructors

Concept	Description
Theme	A Theme represents documents, fact and dimension tables associated with a business process (e.g. *R&D*)
AnalysisUnit	It defines fact tables and document collections related to a specific subject of a theme. In the R&D theme, for instance, one can find AnalysisUnits such as *School dropout.* An AnalysisUnit may have several dimensions and measures
Measure	This concept is used to represent quantitative values, aggregations or summarizations related to AnalysisUnit content (e.g. *Number of students*)
Filter	Filters are dimension attributes that could be applied to slice and dice data related to an AnalysisUnit (e.g. Students age, gender and so on)
Dimension	It describes analysis units dimensions. Dimensions can have many hierarchies and properties (e.g. *State*)
Hierarchy	This concept describes dimension hierarchies. Each hierarchy is composed of one or more levels (e.g. *City*, *State*, *and Country*).
Level	It represents a hierarchy level that is used on drill-up and drill-down operations.
Detail	It describes how an analysis unit can be detailed or presented in its atomic level. (e.g. *name*, *e-mail*).
Property	Property identifies the terminology used to present an information unit. It also maps instances of the attribute concept to instances of detail, filter, level, and measure concepts.
DimensionUsage	It describes how data collections are linked to analysis units.
Collection	This concept represents a data collection or a data provider and describes how these data sources relate to the concepts represented in the domain ontology.
CollectionJoin	It describes how a collection can be joined to another collection. It also identifies which properties and operations are used to join two collections.
Attribute	It corresponds to items contained in collections such as table fields, XML elements, entities extracted from documents, or spreadsheet columns. The Attribute concept also associates these elements with slots of concepts represented in the domain ontology.

Table 1. Description of BI ontology concepts

2.2 SBI Functional modules

SBI ontologies are used by functional modules to support analytical tools on the localization and exploration of data sources.

The QueryManager supports analytical tools by providing a transparent access to heterogeneous data sources and data providers based on a XML-based protocol. It enables the combination of query results from structured or non-structured data sources, independently of their localization. The QueryManager hides data sources complexities

from analytical tools. Requests of analytical tools are translated by QueryManager in queries that are processed on corresponding data sources. Analysis definitions (i.e. analysis units, filters, measures and so on) from analytical tools are received and translated in an XML message. The OntologyManager retrieves the information required from the BI ontology.

The OntologyManager component is responsible for manipulating the BI ontology and retrieving the necessary information to support the formulation of data requests. It retrieves details about data collections such as table names, their field definitions and their relationships from concepts defined in the domain ontology.

QueryManager performs intersections or unions between heterogeneous repositories by using common attributes of each result set returned by drivers. For instance, an inverted index structure is used just to find the papers ids produced by students, and these ids are used to join with other information about such students stored in a data mart. In this example, one can summarize the number of students by department that has written papers that mention the term "semantic web."

In our approach, for each type of data repository or data provider, we create a different driver to handle specific issues of that data source. For instance, in the textual driver, a set of algorithms were used for data indexing and retrieval (Beppler et al., 2005), while the RDBMS driver uses JDBC driver to access relational data bases.

2.2.1 Reasoning

Since SBI ontologies are described in formal language that enables the explicitation of business rules and the definition of axioms for specifying relationships between concepts, Semantic slice and drill are made possible by the architecture ontologies in which relations and rules are applied to filter or expand the results of queries relying on synonyms, hyponyms, and other relations specified in the SBI ontologies.

In order to infer new knowledge or, more precisely, to provide new ways to decision makers explore their data, we have integrated OntologyManager to Jena (McBride, 2002) and Pellet (Sirin et al., 2007).

In our approach, all results of inferences are stored in a data mart, more specifically in a star schema called *Triple Model*. The Triple Model is used as an extension of a dimensional model and its tables can be connected to the remaining tables in data marts. In this strategy, semantic inferences occur in batch as the traditional ETL processing. So, besides the strategic information available in the dimensional model, the OLAP tools can also access the inferred conclusions from business rules processed over the data stored in data marts.

The triple model is composed of an associative table, called *Triple Fact*, and a dimension, called *Inferred Predicate*. The triple fact table is responsible for storing inferred information that describes different relationships between two dimensions. The relationships inferred from business rules are stored in the inferred predicate dimension. In the triple model the reference of each dimension represents a subject or an object similarly to RDF statements.

Figure 3 illustrates the Triple Model integrated to a star schema. In this example, the DI_PERSON dimension is a subject and the DI_INSTITUTION dimension is an object of a statement stored in the triple fact table. Such relation has been inferred from business rules such as *works in*, *graduated in* and *alumnus of* described in the domain ontology.

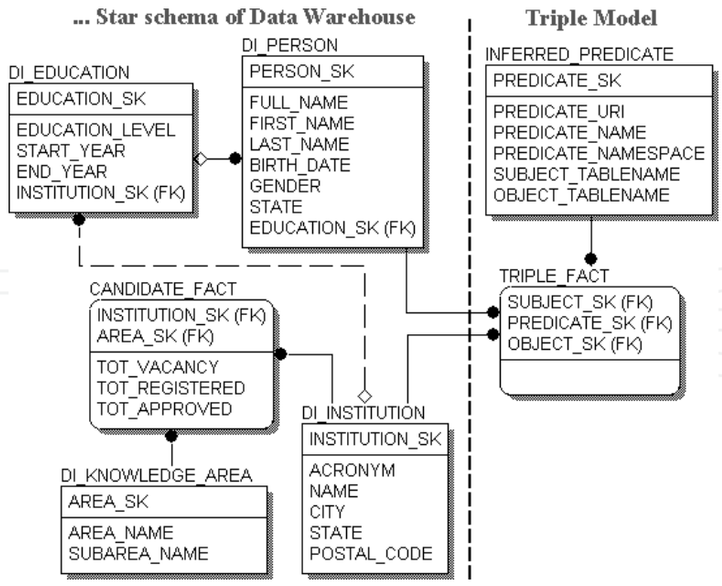


Fig. 3. The triple model and its integration with star schema

Our approach enables the creation of multiples associations between dimensions based on business rules. For instance, at any time, a knowledge engineer can add new business rules in the domain ontology to define new kinds of relationships between people and institutions.

The reasoning process is performed by a Reasoner such as Pellet (Sirin et al., 2007) integrated to the OntologyManager, as follows:

1. The OntologyManager imports the domain ontology model and its business rules. In this step, in the R&D scenario described before all classes, properties, axioms and business rules are brought to the OntologyManager work area.
2. The OntologyManager reads all BI ontology instances used to map the domain ontology to data sources. In this step, each concept defined in the R&D domain ontology will be associated to the Collection class' instances. In addition, all domain properties will be associated to instances of Attribute and CollectionJoin classes.
3. The OntologyManager retrieves the necessary data to create the domain ontology instances. The query is guided by the mapping between business concepts and data sources defined in the BI ontology.
4. New instances of domain ontology are created by OntologyManager based on the information retrieved in the last step.
5. Once created the instances, the reasoner is invoked to perform inferences based on business rules defined in the domain ontology. In this step, the *isAlumni* rule shown in the Listing 1 is applied.
6. At last, the OntologyManager stores inferred concepts from domain ontology into triple model. The new inferred concepts are saved into the inferred predicate dimension and the new relations between two dimensions are stored in the *Triple Fact*.

The inferred information stored in the Triple Model may be accessed by analytical tools to offer decision makers the possibility to slice, dice and drill over data sources by applying business rules defined in the domain ontology.

3. SBI and its question answering approach to support decision making

It is true that the simplicity of current Web search continuously contributes to the growth of its popularity. The ease of use in these search interfaces allows that by informing a few words in free text one can find almost any type of content so fast and ubiquitous. For its intuitive and natural way of providing access to information for people of virtually all ages, the same metaphor of Web search should be considered for the next generation of BI solutions. Such trend for the future of BI takes into account its proximity to resources and services of the Web, both in the use of heterogeneous sources as in the way of finding information (Bohringer et al., 2010; Howson, 2007).

The combination between the new research on BI and aspirations of the Semantic Web is focus of study that can be addressed further. To meet different stakeholders, it is necessary to the analytical tools rely on strategies for the representation of business knowledge and mechanisms the make it possible the use of that knowledge in the exploration of data sources. Just as the Semantic Web provides agile ways and navigation interfaces based on high semantic expressiveness to locate relevant content on the Internet, BI architectures must also make use of semantic to support the analytical processing. However, BI solutions lack the use of effective methods of exploration of content such as those already used by the billions of current Web users, yet without losing the potential conjectured by the Semantic Web (SmallTree, 2006).

Analytical tools usually require long and expensive training sessions due to: a) the potential number of users and time needed to train those users; b) the complexity of the tools; and c) the skills of each user. To reduce these costs, the use of natural language is considered one of the most appropriate and feasible strategy (Conlon et al., 2004). Therefore, the ability to express information needs through natural language should be introduced in the new BI architectures and is the goal of this work.

In this research, analysis submitted to the organization' data sources, instead of being guided through the conventional OLAP operations, are carried out through the semantic interpretation of a question expressed in natural language. That is, we provide access to OLAP cubes through questions stated by users in their language, in which the concepts and terminology of the business are expressed descriptively and independent of specific formalisms.

Unlike the strategies of searches driven by keywords, we apply an approach based on knowledge engineering methods and Question Answering (QA) techniques (Katz et al., 2001; Kauffman & Bernstein, 2007; Lopez et al., 2007). This approach is characterized by adopting more significant use of natural language or questions for returning a data cube that may have the information required by the decision maker.

3.1 Question answering support

The processing of natural language questions is performed by the modules of the architecture in three main steps: 1) a step associated with the construction and maintenance of the model and knowledge base, which is essential for the subsequent steps, 2) a second step related to the interpretation of the question and its formalization in a structure that represents its meaning, 3) a third and final step responsible for returning an OLAP cube.

The first step occurs prior to the decision-making process and should be performed regularly according to the evolution of the domain ontology and to the growth and changes in data sources of the organization. It aims at preparing the ontology and knowledge base used both in the process of analysis and interpretation of the question formulated by decision makers. In the second step, the question reported in natural language is analyzed and processed by a set of methods and technologies that rely on the semantic context defined by the organization's domain ontology. Here, some QA tasks are applied to the interpretation of questions aimed at translating the natural language in a formal language. This formal representation of the resulting question has the definition of quantitative measures, descriptive groupings and filters for the execution of OLAP operations (e.g. drill-down, roll-up, slice, dice, etc.). Once the query is built and formalized, in the last step the information request is performed on the data sources in order to answer the question.

The following figure shows the arrangement of the complementary elements of the SBI architecture as follows: processes and techniques - representing the tasks, techniques, procedures and processes performed by functional modules of the architecture, inputs and outputs - input data and results of the processes and techniques; functional modules - architecture subsystems or components developed by third parties that have some peculiarities in the roles they play and, repositories and data sources - include repositories of ontologies, models and knowledge base, configuration items and also the data sources.

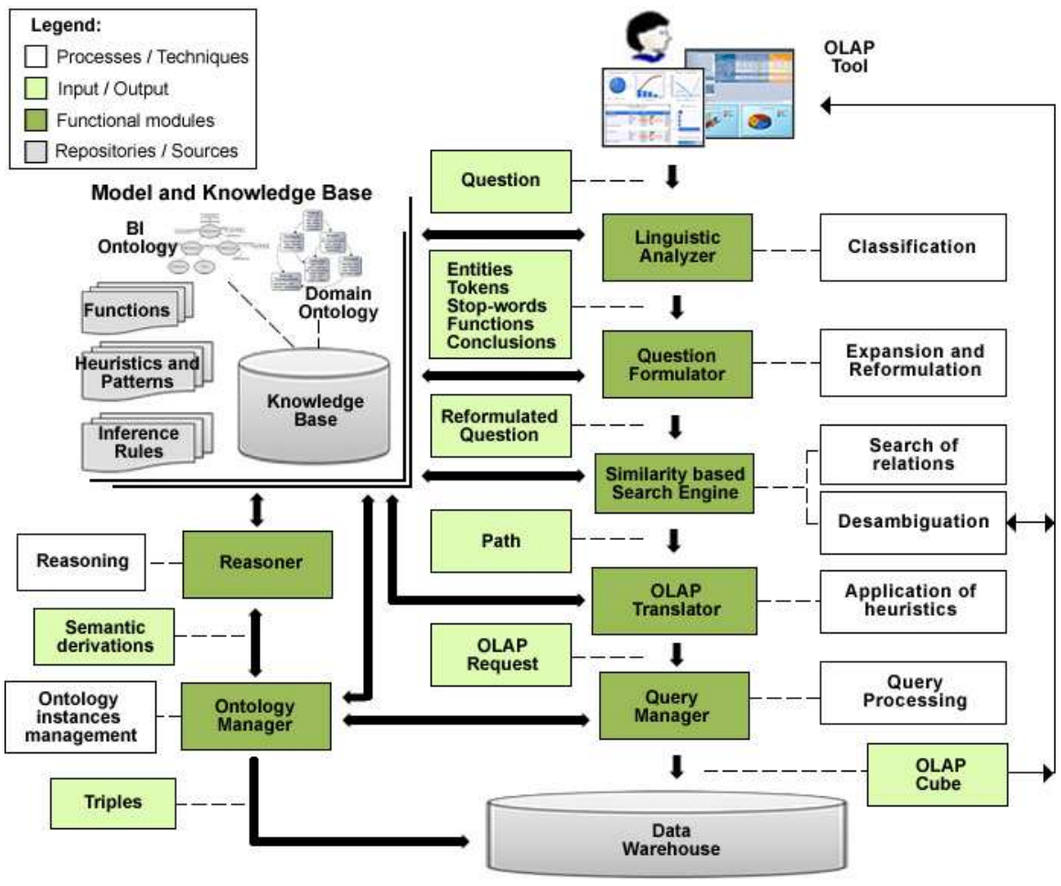


Fig. 4. Schematic view of QA modules

In a nutshell, the question after being informed by the decision maker in the OLAP module is assessed by the Linguistic Analyzer. This module classifies all the elements of the question as specific concepts, such as textual entities, stop-words, functions and other specific classifications reported below. From the classification of the tokens of the question, the module applies query expansion and reformulation techniques to detail and expand the question proposed originally. Based on the reformulated question, the module Similarity Search Engine tries to find relationships between concepts and entities in the question by searching the domain ontology. Once discovered the relationships between concepts in domain ontology, the OLAP Translator converts the question in a formal request, including the definitions of filters, projections and quantitative measures to be considered in the query. Then, the Query Manager performs the request on the DW to return the OLAP cube with the strategic information to decision makers. The semantic inferences performed in the process can be combined with information from the DW. The Knowledge Base serves as a central element in which the functional modules rely to complete each task. The following section describes the main SBI modules.

3.1.1 Knowledge base

The Knowledge Base comprises knowledge resources required in all the steps from the interpretation of decision makers' questions to the queries execution on data sources, as follows:

a. Knowledge Base and Inference Rules - repository formed by the instances of the Domain Ontology, the organization's business rules and instances of the BI Ontology.
b. Functions – consist of functions that must be associated with some specifics words used by the decision maker and with some domain ontology concept. A function assigns one output (e.g. dates, numbers, etc.) to each specific input text. For instance, the words *Today, Tomorrow, Yesterday* could produce a date as output in order to create an instance of the ontology class called *Time*.
c. Heuristics and Linguistic patterns – it represents regular expressions, lexical and syntax patterns, heuristics designed by a specialist according to the language and the distance between words. These resources must be stored on the Knowledge Base in order to support OLAP Translator identification of measures, projections and filters in the question introduced by the decision maker.

3.1.2 Linguistic analyzer

In the first step, the Linguistic Analyzer proceeds with a lexical, syntactic and semantic analysis of the question. The Linguistic Analyzer executes a set of tasks to analyze all textual elements contained in the question in order to classify them and get information needed to interpret the question formulated by the decision maker. As Question Answering systems, this work applies classification tasks and also introduces particular kind of classes to interpret the question and to support OLAP queries on Data Marts. These classes are grouped in six categories as shown in the Table 2.

Linguistic Analyzer executes some techniques of natural language processing to classify question terms, namely, POS-Tagging, Lemmatization, Stemming, Named-Entity Recognition, Co-reference and search into dictionaries.

Classification or feature	Description
Stop-word	Words with a high frequency in texts and usually with no relevant content in traditional QA works. In this research, the stop-words are particularly useful in the identification of OLAP constructors, such as measures, projections and filters.
Position or Order	It identifies the position or order of each token in the question. This information is used to recognize the linguistic patterns and heuristics, in which are stored on the Knowledge Base, according to distances between classes of words.
Function	When a term is associated with a function defined in the Model and Knowledge Base.
Conclusion of inference rule	When the term is contained in a conclusion of inference rules. (e.g. the concept "Alumni" presented in the listing 1)
Entity or Domain concept	It represents a domain ontology concept, such as, a class, a relationship, a property or an instance of a class.
Unknown token	Tokens not classified by the Linguistic Analyzer.

Table 2. Classifications and features of textual terms

During the classification task, some textual elements can generate ambiguities. That is, Linguistic Analyzer can identify two or more classifications for the same term. Likewise, a term classified as a domain concept may represent a class, a property of class, a relationship or an instance of class as well. These ambiguities are not resolved by Linguistic Analyzer and they are processed and eliminated by the module called Similarity Search Engine afterward.

After of the classification tasks, the question can be reformulated and expanded by the Query Formulator module presented below.

The Linguistic Analyzer performs a process with emphasis on each specific term of the question without focusing on the semantic relations among words. The Similarity Search Engine, by verifying the relationship between words and contextual information, can reduce or even eliminate the ambiguities. Therefore, the disambiguation is delayed and performed only once through the Similarity Search Engine. In practice, the Linguistic Analyzer identifies an ambiguity in the question when:

1. The textual entity is an instance of two or more classes of the domain ontology.
2. The textual entity is a class and is similar to two or more classes of the ontology. This case appears when two or more classes have the same name or synonyms in common and are mentioned in the question.
3. The textual entity is a relationship or property and belongs to two or more classes involved in the question. This case is commonly found as the concepts can share the same properties or have equivalent relationships in a given context.
4. The textual entity has similarity between classes, instances, properties or relationships of the domain ontology. This occurs when the term has the same textual description of a class, also a property or a class instance.

Once executed the process of linguistic analysis and obtained ratings for all the terms, the question can be reformulated or expanded through the Question Formulation module described in the following section.

3.1.3 Question formulation

Once the lexical-syntactic and semantic classifications are obtained from the elements of the text, the question is reformulated. This process aims to enrich and expand the original question in order to generate all the information necessary to create the OLAP request later. The reformulation process is also a characteristic of Question Answering systems. It comes to finding important facts related to the domain that have been omitted or reported differently by the user and that should be incorporated to complete and formalize the question. This work uses two types of reformulation that can be applied successively. The first strategy is based on the class hierarchy and synonymy relations and the other applies a rule-based inference approach.

The synonyms as well as superclasses and subclasses of the entities found in the question are necessary in the question reformulation because there are different ways to express the same request through natural language. Thus, the terminology that was reported in a given question can be exchanged for another that is more adherent or closer to the domain context.

The reformulation by inference rules is applied when the query terms are classified by Linguistic Analyzer as being *conclusions of inference rules*. At this stage the facts contained in the conclusions or consequents of the rules are used to reformulate the question. That is, the triple of concepts () that is in the consequent of a rule is used to replace the term classified as conclusion of the inference rule by Linguistic Analyzer.

From the classification and reformulation tasks made by linguistic analysis and query reformulation respectively, the next step is finding which path or set of relationships between concepts that best fit the question. This goal is accomplished by the module Similarity Search Engine that is described in the following section.

3.1.4 Similarity search engine

Based on the question reformulated in the previous step, the Similarity Search Engine performs a search on the model of the domain ontology to discover which path is closer to the context of the question. The textual elements used in the question are confronted with the concepts represented in the domain ontology by Similarity Search Engine in order to check the best path (or the only set of relationships between concepts used in the question) that can resolve the question. Therefore, the concepts of the domain ontology, along with their synonyms and hierarchies are used to evaluate possible alternatives to extract the information required by the decision maker.

The Similarity Search Engine analyses the sequence of concepts or classes (vertices) mentioned in the question and their relationships (edges) in the domain ontology. The Similarity Search Engine chooses the best way to solve the question considering all the terminology given by the decision maker. In this research, as applied by Lopez et al. (2007), the best path is characterized as the one that presents the greatest amount of relevant concepts and relationships according to the terms informed in the question.

The Similarity Search Engine may find more than one possible path to tackle the question. Thus, in addition to previously scenarios of ambiguities solved by Linguistic analysis, this module is responsible for resolving ambiguities among candidate paths. Therefore, two types of disambiguation are likely to be made by the Search Engine Similarity: disambiguation of concepts (i.e. considering classes and properties ambiguous) and disambiguation of paths (i.e. considering relationships ambiguous).

Both disambiguation processes may require decision maker intervention to complete. According to the user's choice in this process, the question is refined iteratively until there is no doubt the meaning of the elements mentioned in the question and about which is the best path to solve the information request.

The Similarity Search Engine, as its name indicates, performs searches on the domain ontology, relying on synonyms, class hierarchy, and other types of relationships defined in the ontology. Searches are supported by a textual index in order to speed up the path retrieval. After discovering the best path as well as the concepts related to the terms mentioned in the information request, the final query can be built and processed on the data sources. The work of translating the best path in a request to explore the Data Warehouse is performed by the OLAP translator described in the next section.

3.1.5 OLAP translator

The OLAP Translator, based on the path chosen in the last step, identifies the measures, groupings, filters and connections between concepts that may be applied in order to extract the data needed to solve the information request. a translation of the best path found by the search engines to an OLAP request, which will be performed later by DW Query Manager module.

Measures represent numerical measurements (sum, average, minimum, maximum, etc.) on a particular domain concept. Measures classified by the OLAP Translator are associated with facts and attributes of dimensions defined in the dimensional model. The concepts translated as groupings identify the descriptive information used to group or categorize the measures in the queries. These concepts should generally be associated with attributes of the DW dimensions and are the classes' properties.

When the concepts are translated as filters, the values related to these concepts are used as selection criteria in the OLAP request. Usually, the element of the question is translated as a filter when referring to a class instance, or when a value of a property identifier (name, initials, date, etc.) is given.

In addition to the measures, grouping and filters, the OLAP translator should also define how should be represented the relationships of these items. Each relationship between the classes denotes joins or connections that must be used between objects of the data sources. All relationships between concepts, including those resulting from the inference rules in the process of reformulation, must be presented on the result produced by the Similarity Search Engine.

In order to make the translation of the path in a query, OLAP translator uses a set of patterns and heuristics based on distance or position between the concepts of the question and stop-words. Although the words classified as stop-words are ignored by most IR

systems, they are essential at this stage of translation. Generally, research on QA applies stop-words to classify the type of question and also to identify the syntactic pattern to answer it correctly. In this work, the stop-words also help the discovery of the elements of OLAP query, such as measures, grouping, filters and joins.

All syntactic patterns and heuristics along with a list of stop-words used by OLAP translator must be configured according to the language in the knowledge base. This setting allows regular expressions and criteria based on the position or distance between tokens and stop-words to be used by the OLAP Translator. Thus, there is greater flexibility in the recognition of elements of the query according to the idiomatic patterns or writing mode of decision makers.

To perform the translation and to set standards in the Knowledge Base, this paper adopts three types of stop-words arranged as shown in Table 3.

Stop-word type	Query element	Description
Quantification	Measure	Stop-words that deal with numeric values summarization or calculations and data quantification. Expressions such as *how many*, *how much*, and its variants, such as *which and total* are considered.
Projection	Grouping	Stop-words used to categorize or group content typically descriptive, without the need to quantify them, such as the dimension attributes. Examples are the stop-words *by*, *for*, *as*, *grouped by*, *second*, etc., on questions like "how many students by age and by city study in the South?".
Selection	Filter	Stop-words used to filter a data set – relational operators such as *above*, *equal to*, *greater than*, *less than*, etc. - and even logical (logical operators), such as *AND* and *OR*. Example of question with relational stop-words: "How many specialists *over* 40 years and *below* 50 years have published articles in 2010?". Example of question with logical stop-words: "How many teachers *and* students study Human Physiology *or* Occupational Health?".

Table 3. Types of stop-words for OLAP translation

3.1.6 Applying SBI on science and technology management

This section shows an application of the architecture for the Science & Technology management, including the evaluation and analysis of intellectual productions and academic and professional activities of researchers, teachers and students of Federal University of Santa Catarina (UFSC). The sample data used as data source comes from the Lattes Institutional Platform[1] (LIP), from UFSC. Thus, the main concepts and terminology - such as person, degree, educational institution, academic and professional activity, production, among others - are modeled in the domain ontology.

[1] Lattes Institutional Platform - http://lattes.ufsc.br

From the domain ontology, a structure similar to a textual index was created so that the best path can be located by Similarity Search Engine. As in IR models, this index forms a matrix with the set of terms and their synonyms extracted from each path of the domain ontology, as shown below in Figure 5.

Figure 5 illustrates how some of the paths (concepts and relationships of domain ontology) are organized into an IR boolean model. This type of model is adopted in the construction of the architecture to demonstrate how the paths can be discovered in practice. However, the architecture does not limit the adoption of other forms and structures for organizing and finding paths. Thus, other mechanisms and methods can be used to support the construction of the Knowledge Base module, in order to fulfill the goal of the Similarity Search Engine module.

Due to the presence of synonyms and class hierarchies in the matrix of paths, the process of reformulation based on class hierarchy and synonyms is treated at once by Similarity Search Engine. Thus, the Similarity Search Engine developed also acts as a reformulator, except for cases of reformulation by inference rules.

Based on the matrix created, the question, after passing through the stages of linguistic analysis and reformulation, is used as input vector by Similarity Search Engine to perform a search. Here, all stop-words are ignored and only they are exploited later by OLAP Translator.

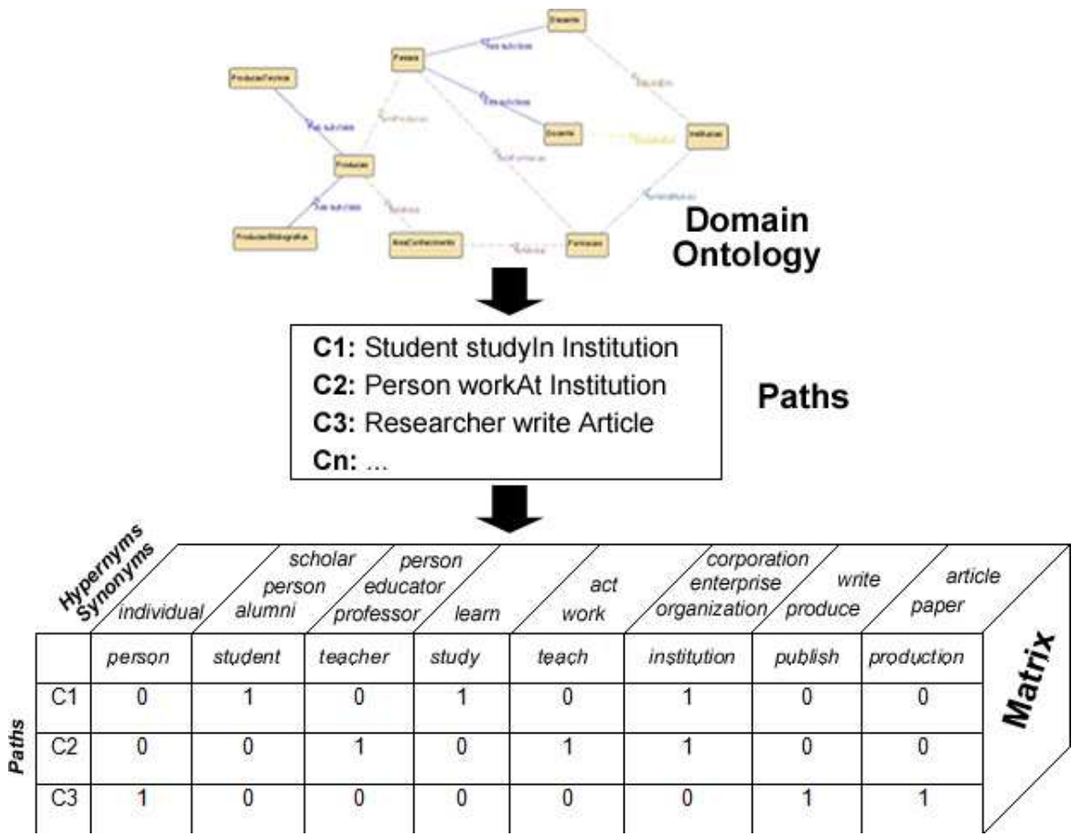


Fig. 5. Illustration of the matrix of paths obtained from the domain ontology

Based on the domain context and in the language used, Knowledge Base should also contain the knowledge of how to identify the elements of OLAP queries, such as measures, groupings, filters and joins. This knowledge is formed by the lexical-syntactical patterns, the relative positions of the query terms and by heuristics based on the types of stop-words previously described in Table 3. The representations of these patterns and heuristics are formalized in the Knowledge Base by means of regular expressions that match the positions of recognized entities of the question and the types of stop-words. The patterns and heuristics used by the OLAP Translator module are detailed in Table 4 according to each element of the associated query (measure, grouping, filter or join).

Nº	Query element	Description of the pattern or heuristic
1	Measure	All terms classified as classes and class properties positioned immediately to the right of stop-words of quantification, which are followed or not by the tokens AND or OR, are classified as measures.
2	Grouping	All terms classified as classes and class properties located to the right of stop-words of projection, which are followed or not by the tokens AND or OR, are classified as groupings.
3	Grouping	The direct classes of terms classified as instances of classes are always used as groupings.
4	Filter	All terms classified as instances of classes or identified as class property values are used as filters. Note: If the term is not to the right of a stop-word of selection (see classification in the Table 4.), the criterion of equality (=) is used to filter the content; otherwise, the stop-word of selection will be considered. The tokens AND and OR present between the values of properties or instances of classes are used as logical operators of the filter criteria.
5	Join or relationship	All relationships between the concepts of domain ontology are used as joins or relationships in OLAP request. In own BI Ontology mapping, these relationships must be matched with joints or relationship between the tables and dimensions in the query.

Table 4. Patterns and heuristics used to formulate OLAP requests

To clarify how heuristics and patterns are applied, consider the following question: "*How many students*[1]*, by gender*[2] *and academic training*[2]*, study*[5] *at CSE*[3,4] *or CFH*[3,4]*?*". The terms superscript numbers refer to the number of the respective pattern or heuristics in Table 3. In this example, the term "students" refers to the class "Student" and is classified as a measure because it has proximity to the right of the stop-word *How many* (by Rule 1). The attribute set in the BI Ontology as a standard measure of the class "*Student*" should be used to quantify the information in the query. Yet, the two terms "*gender*" and "*academic training*", although with the same numbering (number 2), the first represents a property of the class "*Person*" and the second represents directly the class "*Degree*" on the domain ontology.

In this case, the first (the term "*gender*") is used directly as a grouping, the second (the term "*education*"), the default attribute of the class "*Education*" is defined as a group that should

be applied. The terms "CSE" and "CFH", instances of the class "*Instituition*" in this example, are used as filters (by rule 4) and also presented in the return of the query (by rule 3). Again, the attributes of the class "*Instituition*" as defined in the BI Ontology should be used as a filter and aggregate by the OLAP translator. As the terms "CSE" and "CFH" are not involved with the stop-words selection, the equality criterion (=) is used in the filter of data comparison in the query. The logical OR is used to build the filter criteria, because the token "or" is informed in the answer between the terms in question. Finally, by applying the pattern number 5 in Table 4, the word "*study*", which symbolizes a relationship in the domain ontology, is used as a join for the OLAP request. Once this relationship is given in the path returned by the Similarity Search Engines, the OLAP Translator recognizes that the term "*study*" relate the concepts "*Person*" and "*Instituition*" in this example.

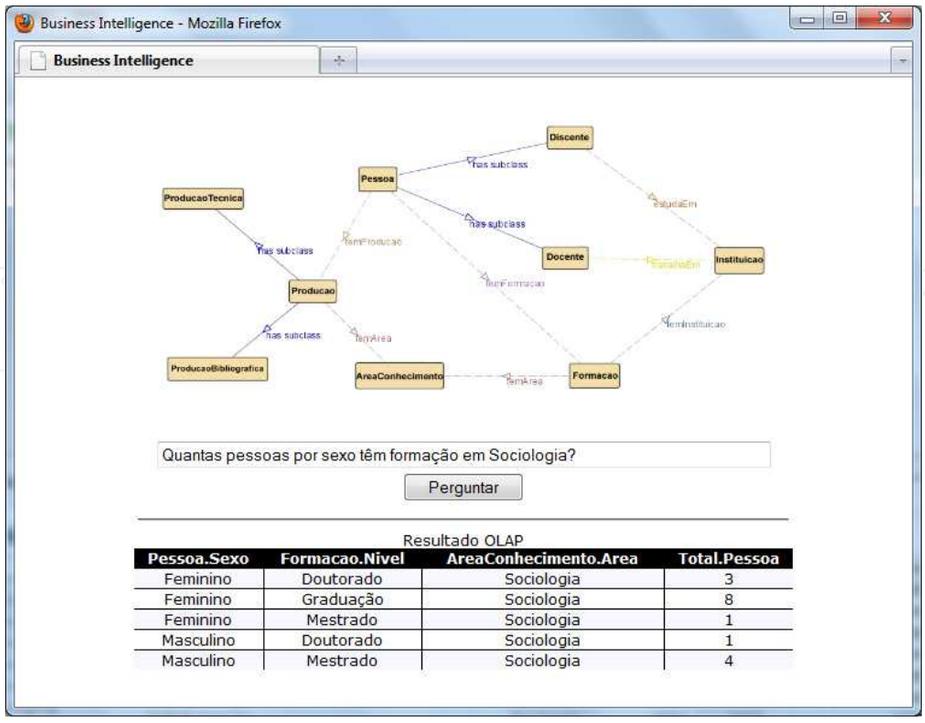
As the dictionary that helps the terms classification, the stop-words are arranged according to their type in the Knowledge Base. These stop-words are defined according to the previous classification given in Table 4. Then the set of stop-words used for the examples of questions in the S&T scenario is shown in Table 5.

<table>
<tr><th>Quantification</th><th>Projection</th><th colspan="4">Selection</th></tr>
<tr><td rowspan="6">how many;
how much;
amount of; which the amount of;
total of;
number of</td><td rowspan="6">by;
according to;
grouped by;
as;</td><td colspan="2">Relational Operator</td><td colspan="2">Logic Operator</td></tr>
<tr><td>Term</td><td>Operator</td><td>Term</td><td>Operator</td></tr>
<tr><td>Above, greater than</td><td>></td><td rowspan="2">and</td><td rowspan="2">AND</td></tr>
<tr><td>Below;
less than;</td><td><</td></tr>
<tr><td>From</td><td>≥</td><td rowspan="2">or</td><td rowspan="2">OR</td></tr>
<tr><td>Equal to</td><td>=</td></tr>
</table>

Table 5. List of stop-words used according to type

The stop-words applied as filters are related one by one with a specific operator. For example, the stop-word formed by the tokens "Above" is associated to the operator ">" the stop-word "From" and its variants are associated with operator "≥", and so on. Others stop-words could be added as required and organization´s standard writing. However, this work is limited to the set of stop-words shown in Table 5.

The following figure shows a prototype analytical tool to support users interaction on top of SBI QA components, where the domain ontology of S&T is illustrated along with the regions of the input query and display the results. This figure shows an example of a question in the context of S&T with its own answer. The steps to obtain the OLAP cube from this question are detailed below.



Pessoa.Sexo	Formacao.Nivel	AreaConhecimento.Area	Total.Pessoa
Feminino	Doutorado	Sociologia	3
Feminino	Graduação	Sociologia	8
Feminino	Mestrado	Sociologia	1
Masculino	Doutorado	Sociologia	1
Masculino	Mestrado	Sociologia	4

Fig. 6. Illustration of a prototype analytical interface

The question "How many people by gender have education in Sociology?" shown in Figure 6, is a simple example which presents no ambiguities, conclusions of inference rules and no functions. That is, only the class properties, classes and their relationships domain ontology of S&T are involved in the question.

Initially, this question, after being informed at the interface must go through the process of lexical analysis, syntax and semantics of Linguistic Analyzer. By consulting the dictionary and concepts of the S&T domain ontology, the Linguistic Analyzer determines the classification of each term in question, in this case: "*How many*" (stop-word of quantification); "*people*" (*Person* class); "*by*" (stop-word of projection); "*gender*" (class property of *Person*); "*have*" (*token* not recognized); "*education*" (*Education* class); "*in*" (*token* not recognized) and; "*Sociology*" (instance of *KnowledgeArea* class). The *tokens* not recognized (on this example, *have* and *in)* are not between the dictionary terms and classes of the ontology, so, do not have defined classification.

Before finding the best path based on the domain ontology, the Query Reformulation module in this example replace the instance "Sociology" by respectively class "*KnowledgeArea*". Since the question does not have classified terms as conclusions of rules there is not reformulation based on inference rules. However, for this example reformulation by using synonyms and class hierarchies are performed. This reformulation is performed by Similarity Search Engine that performs also the role of the Query Reformulation module.

The reformulated question that should be used as input to the search for Similarity Search Engine show only the terms: "*Person gender have education in KnowledgeArea*". Note that some

terms classified as stop-words were ignored in the input vector for the search. The terms *have* and *in*, even not being classified, are used in the search, and special characters are removed.

Only the paths that have the highest number of concepts identified from the input vector are returned by the Similarity Search Engines. So, the best path the S&T domain ontology in this example is (represented in N3): *(Person hasEducation Education) - (Education hasArea KnowledgeArea)*. Note that the other paths, such as those formed by only a single vertex (*Persona; Education or KnowledgeArea)* and those formed by the triple; (*Person hasEducation Education)* or (*Education hasArea KnowledgeArea)* should not return in the search. Thus, in this case a single path is obtained without the need for participation of the decision maker for the disambiguation of entities and paths. Otherwise, the alternatives found are presented to the user that should pick one of the options.

With the best path defined, the set of patterns and heuristics presented in Table 4 is applied by the OLAP translator. Thus, from the classified elements and the types of stop-words, the generated OLAP request has as a measure: the *Person* class, as grouping: the property *gender* and the classes *Education* and *KnowledgeArea*, and as a filter: the term *Sociology*, instance of the *KnowledgeArea* class. The relationships (*hasEducation* and *hasArea*) are also translated in the request as relationships (joins) that connect the concepts of the domain.

Thus, Query Manager works with the Ontology Manager module to assemble and execute the derived query with the dimensions or fact tables associated with the concepts identified in the last step (in this case, *Person*, *gender*, *hasEducation, Education, hasArea and KnowledgeArea*). The instances of BI Ontology that map these concepts to the DW structure are retrieved by Ontology Manager. After locating these instances, the Ontology Manager tells to the Query Manager the dimensions, attributes, fact tables and how they are interconnected to create the SQL query.

As seen, the properties of classes are most often associated with the dimension attributes in the BI Ontology. However, only one property (*gender* property from *Person* class) were reported and recognized in question. According to the translation from OLAP Translator, a default attribute must be set to measure, group or filter for the class. Thus, when a class has no property explicitly informed the default attribute is set to the BI Ontology to be used in the query.

Thus, considering the configuration of the BI Ontology and the question of this example, the dimension attribute *PERSON_SK* from *DI_PERSON* dimension is used as the standard measure and corresponding to the *Person* class. Since the class *Education*, which corresponds in BI Ontology to *DI_EDUCATION* dimension, has as attribute group *EDUCATION_LEVEL*. Finally, the *KnowledgeArea* class has as group and also filter *AREA_NAME* attribute from *DI_KNOWLEDGE_AREA* dimension.

To find out the joins that link the dimensions *DI_PERSON*, *DI_EDUCATION* and *DI_KNOWLEDGE_AREA* in the query, Query Manager uses the relationships *hasEducation* and *hasArea* obtained from the identified path. Also, Query Manager gets the information through the BI Ontology through Ontology Manager to identify the joins between the tables. This information configured in the BI Ontology indicates which attributes of the dimensions

is used to describe the join and the type of join (*inner join*, *left join*, etc.). Finally, the resulting SQL query is performed by Query Manager in order to answers the question.

4. Conclusion

The improvements on knowledge engineering and related technologies offer new approaches to tackle traditional issues in the context of BI and analytical processing. Just as the Semantic Web provides agile ways and navigation interfaces based on high semantic expressiveness to locate relevant content on the Internet, BI architectures should also make use of semantic to support the analytical processing. However, BI solutions lack the use of effective methods of exploration of content such as those already used by the billions of current Web users, yet without losing the potential conjectured by the Semantic Web.

SBI architecture was applied in several e-gov projects and is used, for instance, by three ministries in Brazil (education, environmental, and S&T) to publish data to the Brazilian society and to support internal decision making. The results of the application of our approach in such projects shown that an approach based on ontologies make it easier to handle business rules changes and to offer a more tailored vision over public data on several BI projects.

Our approach incorporates many features that distinguish it from the existing BI solutions and research. The present work aims at enabling the integration of business semantics, heterogeneous data sources, natural language and analytical tools in order to support a smarter decision making.

SBI proved, through case studies, to be a liable alternative for the construction of flexible BI solutions aligned to business logic and decision maker's needs. The following features were made possible by our approach:

- Information is presented to the users using their own vocabulary and in logical views that make it easier to locate information and understand their meaning;
- The definition of business concepts is used to present structured and non-structured data sources available in the organization or remotely;
- Structured and non-structured data can be combined in the same analysis;
- Knowledge and business rules definitions can be altered any time, providing more flexibility to align analytical tools to the latest business rules.
- New possibilities of slice and drill were made possible by combing business semantics and two different reasoning strategies.
- Strategic information is gathered from corporate data sources driven by means of the semantic interpretation of natural language questions.
- Information is summarized by means of textual summaries.

We described how SBI combines knowledge engineering and Question Answering techniques through an interdisciplinary approach. Ontologies, inference rules, idiomatic patterns and heuristics are applied by the architecture's modules on the interpretation of question expressed by decision makers and to produce OLAP cubes to provide all the data needed by SBI users.

Future work comprehends capture of further information about decision makers' interactions with the available functionalities of semantic analytical tools. We are

investigating how to extract rules from this information aiming to support automatic analysis of the business and recommendation of actions.

5. References

Beppler, F. D., Todesco, J. L., Gonçalves, A. L., Sell, D., Morales, A. B. T., & Pacheco, R. C. S. (2005). Uma Arquitetura para Recuperação de Informação Aplicada ao Processo de Cooperação Universidade-Empresa, *Proceedings of the KM BRASIL*, São Paulo, November 2005.

Böhringer, M., Gluchowski, P., Kurze, C., & Schieder, C. A. (2010). Business Intelligence Perspective on the Future Internet, *Proceedings of the Sixteenth Americas Conference on Information Systems*, Lima, Peru, August 2010.

Conlon, S.J., Conlon, J.R., & James, T.L. (2004). The economics of natural language interfaces: natural language processing technology as a scarce resource. *Decision Support Systems*, Vol. 38, No. 1, October 2004.

Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosof, B., & Dean., M. (21 May 2004). SWRL: A Semantic Web Rule Language Combining OWL and RuleML, In: *W3C Member Submission*, 22 May 2010, Available from: <http://www.w3.org/Submission/SWRL>.

Howson, C. (2007). *Successful Business Intelligence: Secrets to Making BI a Killer App*, McGraw-Hill, 978-0071498517, New York.

Katz, B., Lin, J., & Felshin, S. (2001). Gathering Knowledge for a Question Answering System from Heterogeneous Information Sources, *Proceedings of the ACL Workshop on Human Language Technology and Knowledge Management*, Toulouse, France, July 2001.

Kaufmann, E., & Bernstein, A. (2007). How Useful Are Natural Language Interfaces to the Semantic Web for Casual End-Users?, *Proceedings of the 6 th International Semantic Web Conference and 2nd Asia Semantic Web Conference*, Busan, Korea, November 2007.

Lassila, O, & Swick, R. R. (February 2004). RDF/XML Syntax Specification (Revised), In: *W3C Recommendation*, 22 May 2010, Available from: <http://www.w3.org/TR/REC-rdf-syntax/>.

Lopez, V., Uren, V., Motta, E., & Pasin, M. (2007). AquaLog: An ontology-driven question answering system for organizational semantic intranets. *Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 5, No. 2, (June 2007), pp. (72-105), 1570-8268.

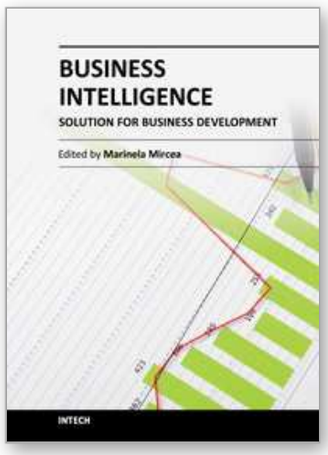
McBride, B. (2002). Jena: A Semantic Web Toolkit. *Internet Computing, IEEE*, Vol. 6, No. 6, (Nov/Dec, 2002), pp. (55-59), 1089-7801.

McGuinness, D. L., & Harmelen, F. v. (February 2004). OWL Web Ontology Language Overview, In: *W3C Recommendation*, 22 May 2010, Available from: <http://www.w3.org/TR/owl-features/>.

Sell, D., Silva, D. C., Beppler, F. D., Napoli, M., Ghisi, F. B., Pacheco, R. C. S., & Todesco, J. L. (2008). SBI: a semantic framework to support business intelligence, *Proceedings of the First International Workshop on Ontology-supported Business Intelligence*, 978-1-60558-219-1, Karlsruhe, Germany, October 2008.

Sirin, E., Parsia, B., Grau, B. C., Kalyanpur, A., & Katz, Y. (2007). Pellet: A Practical OWL-DL Reasoner. *Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 5, No. 2, (June 2007), pp. (51-53), 1570-8268.

Smalltree, H. (June 2006). Business intelligence search: Five myths, In: *SearchBusinessAnalytics.com*, 17 August 2010, Available from: <http://searchbusinessanalytics.techtarget.com/news/1507286/Business-intelligence-search-Five-myths >.



Business Intelligence - Solution for Business Development

Edited by Dr. Marinela Mircea

ISBN 978-953-51-0019-5
Hard cover, 108 pages
Publisher InTech
Published online 01, February, 2012
Published in print edition February, 2012

The work addresses to specialists in informatics, with preoccupations in development of Business Intelligence systems, and also to beneficiaries of such systems, constituting an important scientific contribution. Experts in the field contribute with new ideas and concepts regarding the development of Business Intelligence applications and their adoption in organizations. This book presents both an overview of Business Intelligence and an in-depth analysis of current applications and future directions for this technology. The book covers a large area, including methods, concepts, and case studies related to: constructing an enterprise business intelligence maturity model, developing an agile architecture framework that leverages the strengths of business intelligence, decision management and service orientation, adding semantics to Business Intelligence, towards business intelligence over unified structured and unstructured data using XML, density-based clustering and anomaly detection, data mining based on neural networks.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen