Business Intelligence Applications and the Web:

Models, Systems, and Technologies

Marta E. Zorrilla
University of Cantabria, Spain

Jose-Norberto Mazón
University of Alicante, Spain

Óscar Ferrández
University of Alicante, Spain & University of Utah, USA

Irene Garrigós
University of Alicante, Spain

Florian Daniel
University of Trento, Italy

Juan Trujillo
University of Alicante, Spain

BUSINESS SCIENCE Reference

Senior Editorial Director: Kristin Klinger
Director of Book Publications: Julia Mosemann
Editorial Director: Lindsay Johnston
Acquisitions Editor: Erika Carter
Development Editor: Myla Harty
Production Editor: Sean Woznicki
Typesetters: Chris Shearer, Milan Vracarich, Jr.
Print Coordinator: Jamie Snavely
Cover Design: Nick Newcomer

Published in the United States of America by
Business Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue
Hershey PA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: http://www.igi-global.com

Library of Congress Cataloging-in-Publication Data

Business intelligence applications and the web: models, systems and technologies / Marta E. Zorrilla ... [et al.] editors.
p. cm.
Includes bibliographical references and index.
ISBN 978-1-61350-038-5 (hardcover) -- ISBN 978-1-61350-039-2 (ebook) -- ISBN 978-1-61350-040-8 (print & perpetual access) 1. Business intelligence. 2. World Wide Web. I. Zorrilla, Marta E., 1971-
HD38.7.B8716 2011
658.4’7202854678--dc23

2011021833

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

Chapter 14
Semantic Web Technologies for Business Intelligence

Rafael Berlanga
Universitat Jaume I, Spain

Oscar Romero
Universitat Politècnica de Catalunya, Spain

Alkis Simitsis
Hewlett-Packard Co, USA

Victoria Nebot
Universitat Jaume I, Spain

Torben Bach Pedersen
Aalborg University, Denmark

Alberto Abelló
Universitat Politècnica de Catalunya, Spain

María José Aramburu
Universitat Jaume I, Spain

ABSTRACT

This chapter describes the convergence of two of the most influential technologies in the last decade, namely business intelligence (BI) and the Semantic Web (SW). Business intelligence is used by almost any enterprise to derive important business-critical knowledge from both internal and (increasingly) external data. When using external data, most often found on the Web, the most important issue is knowing the precise semantics of the data. Without this, the results cannot be trusted. Here, Semantic Web technologies come to the rescue, as they allow semantics ranging from very simple to very complex to be specified for any web-available resource. SW technologies do not only support capturing the "passive" semantics, but also support active inference and reasoning on the data. The chapter first presents a motivating running example, followed by an introduction to the relevant SW foundation concepts. The chapter then goes on to survey the use of SW technologies for data integration, including semantic

DOI: 10.4018/978-1-61350-038-5.ch014

data annotation and semantics-aware extract, transform, and load processes (ETL). Next, the chapter describes the relationship of multidimensional (MD) models and SW technologies, including the relationship between MD models and SW formalisms, and the use of advanced SW reasoning functionality on MD models. Finally, the chapter describes in detail a number of directions for future research, including SW support for intelligent BI querying, using SW technologies for providing context to data warehouses, and scalability issues. The overall conclusion is that SW technologies are very relevant for the future of BI, but that several new developments are needed to reach the full potential.

INTRODUCTION

The semantic web (SW) has been conceived as a means to build semantic spaces over web-published contents so that web information can be effectively retrieved and processed by both humans and machines in a great variety of tasks. The definition of these semantic spaces can have many different facets: to provide common terminology (e.g., thesauri), to semantically link published information (e.g., linked data) and to provide further knowledge to allow reasoning (e.g,. logical axioms). The SW is still an open research area although many interesting outcomes have been attained during the last years. Thus, we will use the term "SW technologies" rather than Semantic Web in order to refer to these results, since they can be applied to numerous tasks not necessarily associated to the web.

Despite the successful results of SW area, they have been timidly used in the data warehouse community. Multidimensional models (MD) and online analytical processing technologies (OLAP) have been successfully applied within the database community for analysis purposes, but always under a well-controlled and structured scenario. However, the eruption of XML and other richer semi-structured formats like RDF has shifted the attention of the data warehouse community to a much more heterogeneous and open scenario than that of traditional BI applications. Currently no one questions the need of adding all this external information to the traditional corporate analysis processes. On the other hand, there is a strong agreement in the community about bringing more semantics to the analytical processes. As data warehousing mainly involves the integration of disparate information sources, semantic issues are highly required for effectively discovering and merging data. These semantic issues are similar to those faced in the SW.

This chapter is aimed at giving a new perspective to the BI and the web, which is the main topic of the book. SW technologies have been recently applied to some BI tasks such as extract, transform, and load processes (ETL), MD design and validation, and so on. However, they are usually limited to traditional BI scenarios. In this chapter we also describe the SW technologies that can be useful in highly heterogeneous and open scenarios, and what are their strong and weak points.

As far as we know, this is the first review of the combination of SW and BI technologies. Given that there is a great interest within the BI area about analyzing web-published data, SW technologies seem to be a promising way to approach the involved semantic integration issues as well as new operational capabilities such as automatic classification and deductive reasoning over (integrated) data.

The chapter is organized as follows. First, we present a motivating scenario for combining BI and SW, including a running example. Second, the chapter introduces the relevant foundations of SW technologies, including the resource description format (RDF) and the ontology web language (OWL), standard reasoning services, and technologies for storing and querying semantic

annotations. Third, the chapter describes how to use SW technologies for data integration, including how to perform semantic data annotation and the operation of semantics-aware ETL processes. Fourth, the chapter goes into the heartland of BI, namely multidimensional data models and their relation to SW technologies, including how SW formalisms can be used to capture MD models, and how reasoning services can be applied to perform advanced reasoning about the models and their properties. Fifth, the chapter describes directions for future research in the area, including intelligent querying, contextualizing of data warehouses, and scalability issues. The chapter is rounded off with an overall conclusion.

MOTIVATING SCENARIO AND RUNNING EXAMPLE

BI technology is aimed at gathering, transforming and summarizing available data from existing sources to generate analytical information suitable for decision making tasks. A typical BI scenario can be roughly structured into three layers:

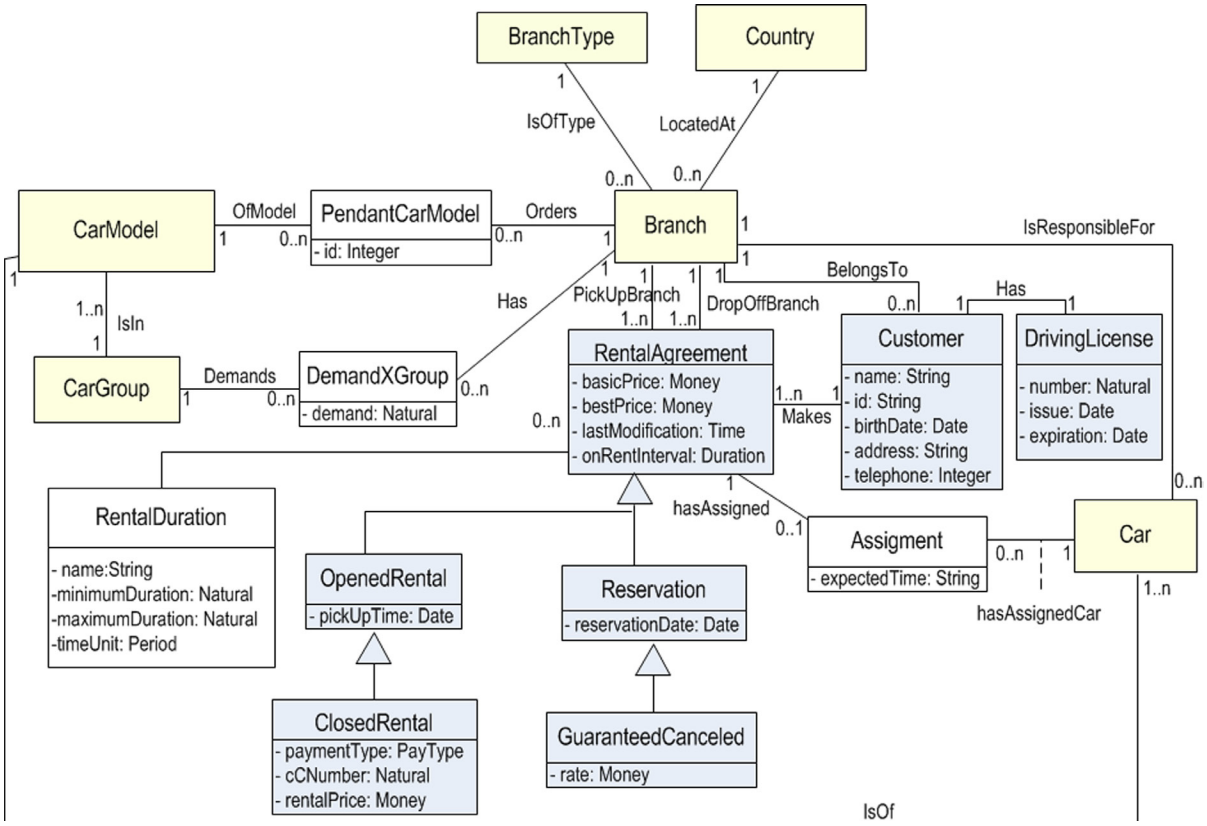
- the data sources layer, which regards all the potential data of any nature (e.g., relational, object-oriented, semi-structured, and textual) that can help to fulfill the analysis goals,
- the integration layer, which is in charge of normalizing and cleansing the data gathered from the sources, as well as of storing it in an appropriate format for the subsequent analysis, and
- the analysis layer, which contains a series of tools for generating the information from the normalized data so that it will be presented to analysts.

The most successful approach to BI has been the combination of data warehousing (DW) and online analytical processing technologies (OLAP). These approaches propose for the integration layer a special model, called multidimensional model, where factual data gathered from the data sources layer must be expressed in terms of numerical measures and categorical dimensions. The semantics of this model consists of representing any interesting observation of the domain (measures) at its context (dimensions). The typical processes in charge of translating data from the data sources layer to the integration layer are called ETL processes (extract, load, and transform). In this chapter we will focus on this kind of architecture, although most topics treated in it can be also applied to other integration architectures such as service-oriented ones.

For illustrating the concepts introduced in this chapter, we will use the BI scenario of a consortium of EU car rental companies. This consortium is interested in generating and sharing strategic information about the sector so that they can improve their respective market strategies. In this scenario, each partner is fully autonomous in the design of their IT infrastructures, while at the same time requiring minimal overhead when sharing data and documentation of interest. As a result, data sources are prone to diverge in almost all the aspects: syntactic, linguistic and semantic ones. Notice that in this scenario, the main issue is about the integration of data formats, schemas, vocabularies, and so on. However, there are also many aspects in common for all the partners, mainly the domain (rent-a-car) and the business strategic goals they are interested in. These agreements are gathered in a conceptual model, which is the base of the intended data warehouse (see Figure 1). This conceptual model captures the main elements of the scenario required by analysts, and it will guide the data warehouse design and implementation process.

Before introducing the technical aspects of the SW technology, we discuss the role of knowledge-based representations in this scenario. In fact, for integration issues we need to define the agreed representations of the components involved in the scenario, namely:

Figure 1. A conceptual model for the proposed BI scenario



- Domain Ontology (DO), which describes the elements that characterize the business topics and subjects. In the example: cars, companies, locations, etc. Large taxonomies of products, taxonomies and generic business rules fall into this category.
- Technical Ontologies (TO), which describe the elements that characterize the information technology objects, mainly the schemas of the three BI layers. Logic representations of conceptual schemas fall into this category.
- Business Ontologies (BO), which describe the elements that characterize the business models, such as the semantics of the measures (Diamantini & Potena, 2008), their relation to strategic goals, etc. Also ontologies derived from the eXtensible Business Reporting Language (XBRL) fall within this category.

Following our running example, in Figure 1 yellow-colored classes correspond to knowledge pertaining to the DO elements, whereas blue-colored classes belong to TO elements, which are associated to the intended data warehouse. The business ontology will describe both the consortium's business activities and the BI measures used in their strategic goals (e.g., Return on Investment, Fraudulent Ratio, and so on). The other classes of the conceptual model can be represented either as properties between ontology classes (e.g. PendantCarModel) or ontology classes (e.g. RentalDuration).

SEMANTIC WEB FOUNDATIONS

This section covers the foundations of the Semantic Web, introducing the main concepts through the running example. First, we give a brief introduc-

tion of the semantic web technology origins and motivations. Second, we introduce the two main representation formalisms that are being used nowadays for semantic annotations, and their inference capabilities and limitations. Finally, we introduce the most widely used language for querying Semantic Web data, SPARQL, where we explain the structure, give examples, and discuss BI-related extensions. We then introduce the various approaches to building specialized RDF data stores, called triple-stores, including schema design and implementation choices.

Historical Background

Semantic Web technology is aimed at providing the necessary representation languages and tools to express semantic-based metadata. Prior to SW, there were several efforts to provide metadata formats to the web contents, resulting in well-known metadata formats such as Dublin-Core, whose main purpose was to improve information discovery and retrieval. However, these formats were shown very limited mainly due to their very poor expressivity and little web-awareness. As a result, the W3C proposed new representation formats, all relying on XML (Bray et al. eds., 2000), to overcome the limitations of existing metadata formats. The main idea behind these formats is that any concept or instance used for describing a web object must be referred through a unique resource identifier (URI). Thus, the most basic way to describe an object consists of creating a link to the URI that represents the intended semantics. With the resource description framework (RDF) (Keyne et al. eds., 2004), we can create more complex metadata elements allowing the representation of relationships between descriptors (e.g. triples). Additionally, the RDFS (Brikley & Guha eds., 2004) extension allows users to define a conformant schema for RDF descriptions. It is worth mentioning that the semantics of RDFS are quite similar to frame-based and object-oriented formalisms. More expressive semantic descriptions have been also proposed by adopting logic-based frameworks: DAML+OIL (Horrocks et al. eds., 2001) and the ontology web language (OWL) (Dean et al., 2004). Contrary to RDFS, all these languages rely on description logics, which are tractable subsets of the first order logic (FOL). In this context, metadata is governed by logic axioms over both classes and instances (assertions). Like in RDFS, logic axioms in these formats must be defined over web-based references (i.e. URIs).

In the data warehouse field, the definition and use of metadata also have strong requirements (Chaudri & Dayal, 1997). Indeed, the traditional division of DW metadata into three categories (i.e. administrative, business, and operational) resembles the division we have proposed in the previous section. In the same way that web developers required more powerful mechanisms to express and manage web metadata, DW developers are requiring more powerful tools to face the increasingly heterogeneous, dynamic and open BI domains.

SW Formats: RDF(S) and OWL

In RDF there are three kinds of elements (Keyne et al. eds., 2004): resources, literals, and properties. Resources are web objects (entities) that are indentified through a URI, literals are atomic values such as strings, dates, numbers, etc., and properties are binary relationships between resources and literals. Properties are also identified through URIs. The basic building block of RDF is the triple: a binary relationship between two resources or between a resource and a literal. The resulting metadata can be seen as a graph where nodes are resources and literals, and edges are properties connecting them. RDFS extends RDF by allowing triples to be defined over classes and properties. In this way, we can describe the schema that rules our metadata within the same description framework.

The ontology web language (OWL) mainly differs from RDFS in the underlying semantic

formalism, which is founded in description logics (DL) (Baader et al., 2003). Indeed, OWL languages provide RDF/XML serializations of different DL languages.

DL basic elements are concepts and individuals, being concepts the intentional representation of *individuals* sets. Concepts can be defined in terms of other concepts by using a series of constructors, which can be either set-oriented, like concept union (⊔), concept intersection (⊓), and concept complement (¬), or relation-oriented, like the existential ($\exists R.C$) and universal ($\forall R.C$) restrictions. Relations can be also defined in terms of other simpler relations by using role constructors, like the inverse of a role (R^{-}). Different DL families can be defined depending on the constructors provided by their languages. The basic family is ALC, which contains the previous operators except the inverse of roles.

Ontologies in DL consist of two parts: the terminological box (TBox), which contains a set of axioms describing concepts, and the assertional box (ABox), which contains concept and role assertions involving individuals (i.e. the data). In our running example, the following axiom of the TBox describes the concept RentalAgreement:

```
RentalAgreement≡
CommercialTransaction ⊓
∃assignment.Car ⊓
   1hasPrice.(1hasCurrency.Currency
   ⊓ 1amount.Float)
```

The qualifier "1" of the properties hasPrice, hasCurrency and amount indicates that they are functional properties for this concept definition, that is, they are to-one relationships.

The following assertions describe an instance for the previous definition:

```
{  RentalAgreement:Contract112,
assignment(Contract112, MMT34),
price(Contract112,_p001),
hasCurrency(_p001,Euros), hasAmount(_
p001,"100"^^"float")  }
```

Notice that this axiom does not necessarily follow the model given in Figure 1, as ontologies are intended to describe semantics not database schemas. Indeed, unlike RDF schemas, not all the axioms of the ontology are designed to describe the "structure" of classes, properties, and instances. Some axioms can express the application business logic. For example, the following axiom may be used to identify taxed transactions according to their currency:

```
∃hasPrice.(∃hasCurrency.(Currency  ⊓
∃usedIn⁻.(¬EuroZone)) ⊑
TaxedTransaction
```

In this way, we can encode implicit knowledge in our descriptions, resulting in much more concise metadata. For example, by just asserting that an instance *a* is of type "∃hasPrice.(∃hasCurrency. Dollars)", we can infer that "*a* is an amount of money expressed in the currency used in USA, and therefore it is associated to a taxed transaction". In the following section, we will introduce the inference tasks that can be performed by means of *reasoning* mechanisms.

DL axioms can be seen as a kind of logic rule of the form "body → head", because in DL the expression $C \sqcap D \sqsubseteq A$ is equivalent to the FOL expression $\forall x\, C(x) \wedge D(x) \rightarrow A(x)$. However, in DL we cannot express rules of the form: $\forall x.\, \forall y\, C(x) \wedge R(x, y) \rightarrow P(x, y)$, which can be useful to describe BI concepts as well as transformation rules necessary for integration tasks. For example, consider the following rule for the running example:

$\forall x.\forall y\ RentalAgreement(x) \wedge hasCustomer(x, y) \wedge NonUECustomer(y) \rightarrow TaxedTransaction(x, y)$

Several extensions of OWL have been proposed to support rules, mainly the semantic web rule language (SWRL) (Horrocks et al., 2004) and the recent integration between DL and rules (Motik & Rosati, 2010). However, these languages restrict the rule syntax in order to ensure they are safe: a safe rule has all individual variables bound to individuals named explicitly in the ontology. As a consequence, data variables are not allowed, and therefore they are not suitable for data-oriented transformations as those required in BI ETL flows.

Currently, there are several software platforms that give support to the development of ontologies by providing tools for edition, debugging, and querying. Among them, we emphasize Protégé and NeOn toolkits. Both platforms are based on java plug-ins, which can be easily added and removed according to the user requirements. These plug-ins allow a great variety of task associated to ontology development: editors, keyword searchers, module builders, ontology matching tools, reasoners, and so on.

Standard Reasoning Services

Logic-based systems can provide users with inference capabilities to manage the implicit knowledge derivable from the ontology axioms. Any inference can be expressed as $O \models \alpha$, where α is any axiom expressed with the same language as the ontology O. The typical inferences DL reasoning services usually provide are the following ones:

- Subsumption and equivalence inferences: $O \models C \sqsubseteq D$ and $O \models C \equiv D$
- Concept unsatisfiability: $O \models C \sqsubseteq \bot$
- Instance classification: $O \models C(a)$

A relevant task derived from the previous inference problems is that of query answering. Basically, it consists of retrieving all the subsumed/subsuming concepts along with the individuals of a given DL concept description.

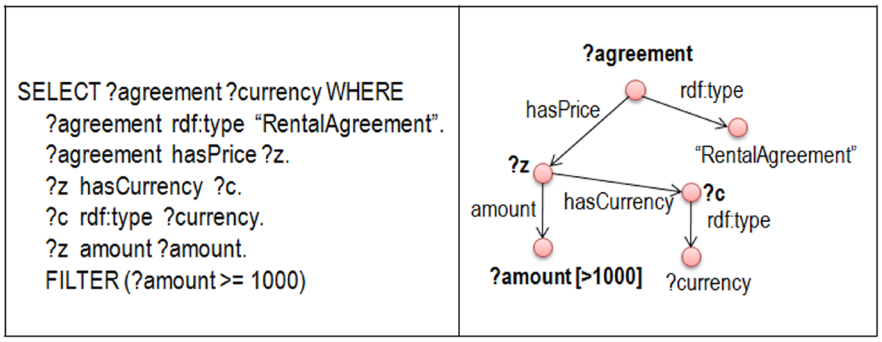
Reasoners

The main approaches to carry out DL inferences have mainly relied on the tableaux-based algorithms (Baader, 2009). A tableaux-algorithm is intended to incrementally build an ontology model (i.e. a finite interpretation of the ontology) by applying a series of transformation rules which express the semantics of each DL constructor in terms of models. Models in decidable DLs are always tree-shaped, where the branches represent relations between concepts (roles) and nodes represents interpretations of the involved concepts. Most of the popular reasoners for OWL-DL, like Pellet (Clark & Parsia, 2010), Racer (Haarslev & Möller, 2001) and FaCT++ (Horrocks, 1998), are implementations of these algorithms.

Computational Complexity Issues

Unfortunatelly, most of the significant OWL 1 languages proposed by the W3C (namely, OWL-DL and OWL-Lite) are actually coNP-hard in data complexity, i.e., when complexity is measured with respect to the size of the data layer only, which is indeed the dominant parameter in this context. This means that, in practice, computations over large amounts of data are prohibitively costly. A way to reduce the complexity of DLs is to impose restrictions on the ontology language, so as to guarantee that reasoning remains computationally tractable w.r.t. the TBox size. Possible restrictions that guarantee polynomial reasoning have been studied and proposed in the context of description logics, such as Horn SHIQ (Hustadt et al., 2005), EL++ (Baader et al., 2005) y DLP (Grosof et al., 2003). Among these fragments, we find a family of DLs, called DL-Lite (Artale, 2009), which is specifically tailored to capture basic ontology and conceptual data modeling

Figure 2. SPARQL query example and its graph pattern representation



languages, while keeping low complexity w.r.t. the size of the data. These logics allow for answering complex queries (namely, conjunctive queries, i.e., SQL select-project-join queries, and unions of conjunctive queries) in LogSpace with respect to data complexity. More importantly, after a pre-processing phase which is independent of the data, they allow for delegating query processing to the relational DBMS managing the data layer.

Recently, OWL 2 (W3C, 2009) introduces three profiles aimed to perform specific reasoning tasks (e.g. classification, query/answering, etc.) with tractable computational cost, namely:

- OWL2-EL profile, which is intended to produce complete inferences in polynomial time for large terminological requiring a limited expressivity resources (e.g. life sciences ontologies),
- OWL2-QL profile, which is intended to efficiently perform queries over large data bases with a expressivity equivalent to DL-Lite$_R$, and
- OWL2-RL, which is intended to make inferences over RDF (graph) data through rule-based query languages. More details about these profiles can be found at the official W3C site (Calvanese et al., 2008).

Currently, these profiles are supported by several reasoners: Mastro (Savo et al., 2010) is fully conformant to OWL2-QL, CEL (Baader et al., 2006) is a reasoner for the OWL2-EL profile, and OWLRL (Herman, 2008) implements a reasoner for the OWL2-RL profile.

Apart from these efforts, other recent approaches aim to implement efficient reasoners for the complete DL expressivity as well as new reasoning problems. For example, (Motik et al, 2010) propose an extension of the tableaux-algorithms, called hypertableux-algorithms, which allows reasoning with rules. The HermiT reasoner (Hermit, 2010) implements this approach.

Storing and Querying Semantic Annotations

An RDF store, also called triplestore, is a database management system in charge of storing huge amounts of RDF triples as well as of providing a query language for sub-graph retrieval. As RDF data is basically graph data expressed with triples of the form "subject-predicate-object" (SPO), the query language consists of sub-graph patterns also expressed as a set of triples containing variables over any of the triple arguments. Additionally, pattern variables can be restricted by filtering expressions. SPARQL (Prud'hommeaux & Seaborne, 2008) is the query language proposed by the W3C for RDF stores. Basically, a SPARQL statement is a SELECT-FROM-WHERE expression (see Figure 2).

In a SPARQL query, variables start with "?", the WHERE clause contains triple patterns and filtering expressions over variables which are separated by ".", and the SELECT clause contains the variables that will be used in the result set. Result sets are just tuples with all the bindings of the variables that satisfy the WHERE clause. Although not yet part of the W3C Recommendation, several extensions have been proposed for SPARQL, e.g., by the ARQ query engine, that make it more applicable for BI-like queries, including ORDERBY, DISTINCT and GROUPBY clauses to manage resulting tuples. While SPARQL is the most prominent, other families of query languages capable of querying semantic web data exist, including XML query languages such as XQuery (Boag et al., 2007), topic map query languages, other RDF query languages such as RQL (Karvounarakis et al., 2002), and OWL query languages. A detailed survey of these is found in (Bailey et al., 2009), which also notes that a drawback of SPARQL is the weak support for schema or ontology information. From our BI perspective, this is a problem, since BI data, including MD models, require very strong schema/ontology support.

RDF stores have the main drawback of lacking a conformant schema that facilitates its physical database design and query optimization techniques. The simplest way to implement a RDF store is to create a single relational table with three columns (S-P-O) and perform queries through self-join operations. This approach is clearly inefficient for both large triple stores and large graph queries. Several optimization approaches have been proposed in the literature, which have derived to different RDF store systems. A common optimization that still maintains a generic database schema is to put long text values such as URIs and string literals into separate tables. This is done, e.g., by the well-known triplestore 3store (Harris et al., 2003). A further optimization consists of grouping triples by predicate and then creating a table for each group. Jena (Wilkinson et al., 2003) and Sesame (Broekstra et al., 2003) use this strategy in their implementations. More sophisticated ways of clustering triples have been proposed in order to facilitate the generation of materialized joint views, like in the Oracle-RDF and C-Store (Abadi et al., 2007) implementations. The 3XL triple store (Liu et al. 2011) builds a specialized and optimized object-relational schema based on a supplied OWL-Lite ontology for the data in order to provide intelligent data partitioning, a strategy that proves very successful. Finally, the RDF-3X (Neumann & Weikum, 2008) and YARS2 (Harth et al., 2007) systems propose indexing the different SPO combinations under a B+-Tree, so that triple patterns can be solved through range queries.

The main issues addressed in query processing are the design of proper indexes for graph-pattern queries and gathering the most appropriate statistical information for join-order optimization. For the former, materialized join indexes, B+-trees and hash indexes have been proposed. For the latter, frequency statistics must go beyond S-P-O individual histograms and must take into account element co-occurrences for different graph shapes (Neumann & Weikum, 2010).

SW TECHNOLOGY FOR INTEGRATION ISSUES IN DATA WAREHOUSING

Current BI solutions face two main challenges. The first one is related to how to represent domain and business semantics, that is, how to model domain and business concepts and logic, so that decision makers can explore information repositories without using technical descriptions. The second challenge is concerned with the integration of heterogeneous information sources. The widespread adoption of new technologies in the Web, such as XML and other richer semi-structured formats like RDF, has widened the traditional BI scenario to a much more heterogeneous and open one. Dealing with the problem of accessing

structured and non-structured data in an integrated and transparent way is still a challenge. *Semantic annotation* has been proposed to overcome these issues by providing a semantic abstraction to support both homogeneous access to disparate data sources and resource discovery. In this context, semantic-aware ETL processes are those that take into account these semantic annotations to improve the integration processes required in BI solutions. This section reviews these topics.

Semantic Annotation

An annotation is usually conceived as a comment attached to a section of a document, or more generally, an object. Annotations may be provided in different forms and formats, ranging from links to the information resources to embed in the annotated object and also as unstructured text or with formal structures. In general terms, semantic annotation is conceived as the process of discovering and assigning to the entities in the text links to their semantic descriptions, which are usually defined in a knowledge base (Kiryakov et al., 2004; Reeve & Han, 2005). In principle, semantic annotation is applicable to any kind of text – web pages, text documents, text fields in databases, and so on. It can be seen as a metadata generation/acquisition process so that data can be leveraged into a more expressive semantic level. We put special emphasis in the language used to describe the schema of the annotation since the more formal the semantics of this language the more machine-processable are the annotations.

In the BI scenario where applications are domain-dependent (e.g. retail sales, R&D), semantic annotation should be focused on assigning both domain and business concepts from specific known resources (e.g. thesauri, ontologies) to the data structures of the information system. It can be seen as a mapping of the sources to a homogeneous conceptual space where we capture the meaning of the integrated elements. This process will be usually performed in a semi-automatic way.

State-of-the-art research in the BI area proposes the use of semantic annotation by means of ontologies as a semantic middleware for integrating data from heterogeneous information systems. In (Spahn et al., 2008) a layered architecture is proposed where each data source schema is independently mapped to a technical ontology (TO) and the various TOs are connected to a business ontology (BO) to relate technical concepts to business-level concepts. At the application layer, the user can specify queries based on graph representation of the BO, which contains business-relevant vocabulary that is familiar to business users and therefore intuitive and easy to understand. In (Sell et al., 2008) they differentiate between the domain ontology, which provides the terminology of the business domain, and the BI ontology, which models the concepts used to describe how the data is organized in data sources (i.e. OLAP concepts) and to map such data to the concepts described in the domain ontology. In (Simitsis et al., 2010) ontology-based semantic annotation of data stores is used for deriving the conceptual design of ETL processes through reasoning. In a pre-processing phase the elements of each individual data store are analyzed and mapped to concepts using a domain specific thesaurus using both string and structure-based schema matching techniques in a semi-automatic way. Then, an ontology is constructed with the previous concepts and is used to annotate the data stores using OWL-DL constructs. These annotations embed both domain and technical concepts. Figure 3 shows an excerpt of the semantic annotations over one data source and the data warehouse.

In a similar way, a semi-automatic method exploiting ontology-based semantic annotation for the design of multidimensional data warehouses is presented in (Romero & Abelló, 2010b). The method assumes several heterogeneous data sources are represented by an ontology and identifies potential facts, dimensions and measures based on functional dependencies. In (Romero et al., 2009), the inference capabilities of the domain

Figure 3. Double underlined concepts refer to the technical ontology (i.e. application specific concepts) whereas single underlined ones refer to the domain ontology (i.e. BI domain specific concepts)

Source_Contract ≡ RentalAgreement ⊓ ∀hasPrice.(∀hasCurrency.Dollar) ⊓
∀DropOffBranch.{BranchType1,BracnhType2}
DW_Contract ≡ RentalAgreement ⊓ ∀hasDuration.LessThan20days ⊓ ∀hasPrice.(∀hasCurrency.Euros) ⊓
∀DropOffBranch.{BranchType2}

ontology encoded in DL-Lite are exploited to derive a multidimensional model that fully takes into account the semantics of the domain.

In recent years a lot of research and development has been carried out in the area of automatic information extraction (IE) from Web pages, text resources, semi-structured data such as HTML tables or Wikipedia *infoboxes*. The main goal of these approaches is to provide a comprehensive knowledge base of facts about named entities, their semantic classes and their mutual relations. Most pattern-based approaches follow the basic method outlined in (Brin, 98). In (Kiryakov et al., 2003; Maedche et al., 2003) extraction rules arise from an initial set of tagged entities. Other relevant approaches include (Cimiano et al., 2004) which present a tool for automatic pattern-based annotation based on the available knowledge on the Web, Text2Onto (Cimiano & Völker, 2005) a tool for ontology learning with improved statistical assessment of fact candidates and patterns and Omnivore (Cafarella et al., 2009) which aim to extract arbitrary relations from natural language texts. Moreover, most research along these lines has considered Wikipedia as key asset for the extraction of knowledge. Examples of such efforts include (Atserias et al., 2008) which provide semantic annotations for the English Wikipedia, DBpedia (Auer et al., 2008) which harvests RDF subject-predicate-object triples from Wikipedia and similar sources, Kylin/KOG (Weld et al., 2008; Wu & Weld, 2007; Wu & Weld, 2008) an ambitious work whose goal is to extract arbitrary relations from natural language texts and Wikipedia *infobox* templates and YAGO (Suchanek et al., 2007; Suchanek et al., 2008) which integrates relational knowledge from Wikipedia with the WordNet taxonomy. The prevalent methods under these IE tools are a combination of rule-based pattern matching, natural language processing, and statistical machine learning.

Although one of the goals of the previous approaches is to leverage data with semantics (similar to the goal of semantic annotation) they conceive the harvesting of knowledge in a broad, universal way. They are domain-independent in the sense that they try to capture in an automatic way as many entities as possible and link them to knowledge resources mainly to advance the functionality of search engines to a more expressive semantic level. These differences w.r.t. the specific and domain-oriented nature of BI scenarios along with the lack of standards and integration with formal knowledge hinders its usage for semantic annotation in the aforementioned BI scenarios. However, some approaches have tackled this issue by customizing and adapting existing IE tools so that they can be effectively applied to a specific BI scenario. In (Saggion et al., 2007; Declerk, 2008) the use of ontology-based IE in the context of BI is proposed for the internationalization domain. This approach requires the construction of a domain ontology by knowledge engineers and domain experts that gathers all the required domain concepts and relations. Furthermore, IE tools like NER have been customized to target the specific domain entities and map them to the ontology concepts.

Most of the classical IE annotation tools require a complete syntactic analysis, and in some

cases, a semantic analysis too, which is usually an expensive operation not affordable for even medium-sized document collections. Methods based on manual pattern definition do not suffer from these issues, but require human effort and intervention for updating and customizing patterns to each application scenario. Finally, machine learning methods usually rely on training corpora, which is not always available. As an alternative to the previous approaches in (Danger & Berlanga, 2009) a tool for the extraction of complex instances from free text on the Web is presented. The approach is based on non-monotonic processing and uses a logic-based reference ontology, entity recognizers and disambiguators, in order to adequately create and combine instances and their relations. The complementary work in (Nebot & Berlanga, 2009) enables the customized use of available knowledge resources such as thesauri or ontologies to assist in the annotation process. The method allows the user to select and build tailored and logics-enabled ontologies from large knowledge repositories. Later, the extracted ontology can be used as an alternative to the use of training corpora in machine learning methods (Danger & Berlanga, 2009).

New research possibilities may arise if we consider the semantic descriptions delivered by the previous tools as a new type of data sources susceptible of being analyzed by BI applications. Along this line of research we can find a few approaches aimed at analyzing semantic annotations encoded in logic languages such as RDF(S) and OWL. (Nebot et al, 2009) propose a multidimensional framework for analyzing semantic annotations from a logical viewpoint using ontologies. In this approach semantic annotations are based on application and domain ontologies. The user can build a multidimensional integrated ontology (MIO) containing the required analysis measures and dimensions taken from the available ontologies. This approach is similar to the previous semantic BI approaches in the sense that it uses ontologies as an integrating tool among different sources. In such scenario where multiple ontologies co-exist together, ontology alignment and merging strategies play a key role.

Semantic-Aware ETL Processes

During the initial steps of an ETL project, the main goal is to construct a conceptual ETL design that identifies the useful to the project data sources and describes the corresponding data transformations needed to map these sources to the target data warehouse concepts. For achieving that, it is imperative to identify and understand the semantics of both the data sources and the target data stores. Several approaches have already been proposed for using Semantic Web technology to the design and construction of the ETL part. Naturally, most of them deal with the conceptual part of the ETL design, since the Semantic Web paradigm seems as a promising means to overcome the lack of handy ways for capturing the semantics of an ETL process.

The prevailing –so far– idea in using Semantic Web technology for ETL suggests using a global ontology for mapping all the involved data stores to it. This idea resembles the "local-as-view" paradigm (Lenzerini, 2002), where the application ontology, constructed as a conceptual model of the domain, corresponds to the global schema and the semantic descriptions of the data stores, in terms of classes and properties defined in the ontology, correspond to the views describing the local schemata. However, the use of an OWL ontology, instead of a global schema provides a formal model on which automated reasoning mechanisms may be applied. Furthermore, in ETL it is not sufficient to consider the integration problem as a query rewriting problem, since the transformations taking place in a real-case ETL scenarios usually include operations, such as the application of functions, that cannot be captured by a query rewriting process (see Skoutas & Simitsis, 2007).

(Niemi et al., 2007) discusses methods for OLAP cube construction using Semantic Web technology. They use a generic OLAP ontology as an upper ontology for all OLAP cubes. This ontology defines only general OLAP concepts and it is independent of the application area. Per application need, they consider domain-specific ontologies (e.g., CarModel, Branch, Country) based on the upper one. Such ontologies are defined based on common concepts that have global definitions shared among all domain-specific ontologies. The OLAP cubes are described based on the domain-specific ontologies and the data sources are defined using the global concepts. However, since in practice it is possible that some data sources are not defined using the domain-specific ontology and they are described in another way –for example, using the upper ontology– we may need to define ontology mapping transformations describing how the source data should be converted to conform to the global domain ontology. In order to integrate data from different sources, the authors consider an RDF data format and an RDF query language. The approach proposed by (Niemi et al., 2007) is as follows. They suggest starting with mapping the sources to an OWL/RDF ontology. Then, the user needs to design the structure of the OLAP cube. Next, the OLAP cube (presented using the XML serialisation of RDF) is constructed based on RDF queries issued on the data sources. At the instance level, the combined result of such queries represents an instance of the OLAP cube. As a constraint, the method requires that there is a common ontology base for the data sources and that each data source can be described in sufficient level of detail for enabling a mapping to RDF format.

An extension to this work discusses in more detail the method for automating the construction of OLAP schemas (Niinimäki & Niemi, 2009). Again, the source and target schemas are considered as known. The mapping among the source data and the OLAP schema is done by converting the data in RDF using ontology maps. Then, the relevant source data are extracted using RDF queries generated using the ontology describing the OLAP schema. At the end, the extracted data are stored in a database and analyzed using typical OLAP techniques. Both works aim at an end-to-end design approach, but they have two main limitations. First, they both require prior knowledge of the source and target schemas and second, they consider simple data transformations.

Another research approach to ETL design using Semantic Web technology elaborates more on the complexity of the data transformations required for integrating source data from heterogeneous sources into a data warehouse (Skoutas & Simitsis, 2006; Skoutas & Simitsis, 2007). This research work deals with one of the major challenges of the ETL design: the structural and semantic heterogeneity. For example, two sources *S1* and *S2* may contain the same kind of information under two different schemata: S1.Rents(rentID, cartype, carplate, carmileage, customerID, …) and S2.Rents(rentID, carID, customerID, …); or they may use different representation formats, like: carmileage in kilometers (km) in *S1* and in miles (mi) in *S2*. The core idea of this work is the use of ontologies to formally and explicitly specify the semantics of the data source and the data warehouse schemas and thus, to automate in a large extent the ETL generation. This work also assumes that the source and target schemas are previously known.

In more detail, the first step of this approach is to construct an ontology to model the domain of discourse as described by the data store schemas and the application specifications. For that and in order to deal with different naming schemes, first, an application vocabulary is constructed. The vocabulary involves information like terms denoting the primitive concepts of the domain of discourse (e.g., car, branch), the features that characterize such concepts (e.g., carID, carmileage), the different representation formats that may be used for a feature (e.g., for carmileage {km, mi}), the allowed values that an enumerated feature may

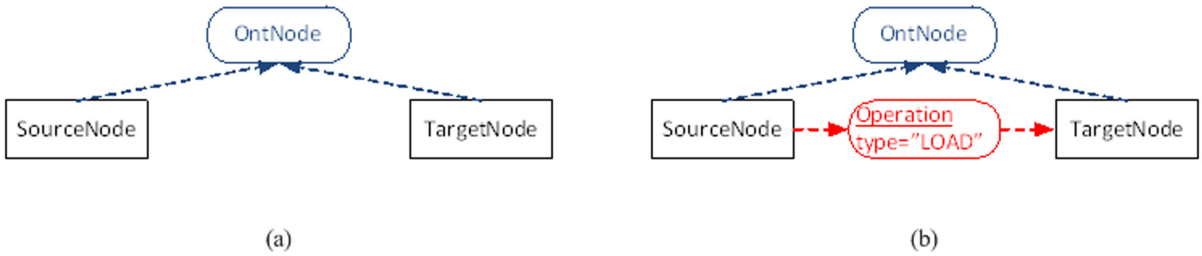
take (e.g., for branch location {Athens, Barcelona, New York}), and functions associating features to concepts, representation format to features, values to representation formats or features, and so on. Then, the data sources and the data warehouse are annotated w.r.t. the application vocabulary. Each data store contains a set of relations that comprise one or more attributes. The process of annotating a data store refers to providing two types of information: establishing the appropriate mappings between the data store relations and attributes and the concepts and features of the vocabulary; and describing each relation in terms of the cardinality, representation format and (range of) values of its associated features. Based on these, this works elaborates on how the application ontology is generated.

The second step of the process is the generation of conceptual ETL design. This step involves an automatic means for deriving the mappings from the source attributes to the attributes belonging to the data warehouse, along with the appropriate ETL transformations. First, the method determines from which sources –i.e., from which attributes/ relations of these sources– information needs to be extracted in order to populate each attribute/ relation in the data warehouse. Next, the method determines the transformations required to integrate data from the source to the target relations using the specified mappings and the relative position of the source and target classes in the class hierarchy. This is realized in three phases. In the first, transformations like project, concatenate, and join are identified. In the second, other transformations like select, convert (functions), not null, aggregate, and so on are also discovered. Finally, the design is complemented with transformations like add attribute, union, distribute, detect duplicates, and so on. Although more complex transformations (e.g., pivot and slowly changing dimensions) are not captured by this process, the most frequently used ones can be automatically discovered. For example, assuming that location information was stored in the two sources *S1* and *S2* under different formats: S1.Branch(…, location,…) and S2.Branch(…, street, number, …) and that the target data store has a different schema T(…, city, street,…), then a convert operation, namely concatenation c($attr_1$_val, …, $attr_2$_val, property_val), will be identified as: c(street, number, street).

Several extensions to the above two research works have been proposed. (Skoutas et al., 2009) proposes using a graph-based representation as a conceptual model for the source and target data stores and based on that, the ETL transformations can be identified by means of graph transformations. In other words, this work describes how the operations comprising the ETL process can be derived through graph transformation rules, the choice and applicability of which are determined by the semantics of the data with respect to an attached domain ontology. Thus, starting from an initial graph comprising the source and target data stores subgraphs, the ontology subgraph, and the semantic annotations, this works shows how to produce a final graph that contains the ETL process subgraph. In this context, the specification of the ETL process can be seen as a set of paths directed from source data store nodes towards target data store nodes. The nodes along these paths denote ETL operations and the edges connecting the nodes indicate the data flow. Figure 4 presents a simple rule for inserting a LOAD operation in the presence of a direct relationship, where the source and target elements correspond to the same concept; in Figure 4(a) a match is found and a rule is triggered causing the insertion of a LOAD operation Figure 4(b). Several transformations can be discovered by this process as Load, Filter, Convert, Extract, Split, Construct, and Merge.

The techniques discussed so far consider as given the source and target schemas and search for the mapping of one to the other. A more recent approach starts with the business requirements and the source data stores, and works toward the identification of both the target schema and the transformations needed for the realization of the

Figure 4. Rule for inserting LOAD operation (Skoutas et al., 2009)



ETL process (Romero et al., 2010a). The method starts with a single business requirement, and after its validation, and possibly completion, it relates it to the source data stores in order to annotate a generic ontology. Then, the concepts are characterized either as factual or dimensional and correctness w.r.t. multidimensional design principles is checked. In the meantime, the annotations are used for extracting schema modification operators; e.g., selection, projection, union, set operations, aggregation, functions. At a next step, additional transformations can be added either based on functional requirements of the data –e.g., "make sure that each customer is considered once"– or based on standard business and design needs; e.g., "replace the production keys with surrogate keys" or "take care of the slowly changing dimensions". As a final step, all the results produced for each single business requirement are consolidated in order to produce the final multidimensional design and ETL process.

The abovementioned efforts work toward the facilitation and automation of ETL design, and thus, aim at making the life of the designer easier. Another crucial matter is to allow business people to comprehend and evaluate the design outcome. Several design notations, like UML or BPEL, require some knowledge and experience for their understanding. Providing a textual description of ETL designs is the most natural way for their representation. The ontology-driven techniques for conceptual ETL design can be exploited for producing such textual descriptions (Simitsis et al., 2008; Simitsis et al., 2010). In particular, after the generation of the ETL design, the constructed ontology can be parsed and a template based method can be used for representing information about the data stores –using the data store annotations– and about the generated ETL process. Then, several reports can be customized using a template language for showing a list of annotations, ETL transformations, ETL statistics, and so on.

All the previous approaches are conceived for dealing with relational data sources. Recently, (Nebot & Berlanga, 2010) have presented a method to generate fact tables directly from semantic data expressed in RDF(S) and OWL. The proposed method starts with the target MD, which must be expressed in terms of concepts and properties of the source ontologies, and then it performs a series of transformations that guarantee that the generated factual data conforms to both the MD and the source semantics. This method could be incorporated to existing semantic-aware ETL processes in order to integrate SW annotations of unstructured and semi-structured data into data warehouses.

MULTIDIMENSIONAL MODELS AND THE SW

This section presents a brief introduction to the multidimensional model, and we highlight the strong dependence of multidimensional modeling of data warehouses on data found within the organizations. However, nowadays it is compulsory to consider external data to produce the

data warehouse multidimensional schema. At this point, SW technologies arise as a valuable asset to help in the integration process of external data to complement the organizations own data. This section is divided in 3 subsections:

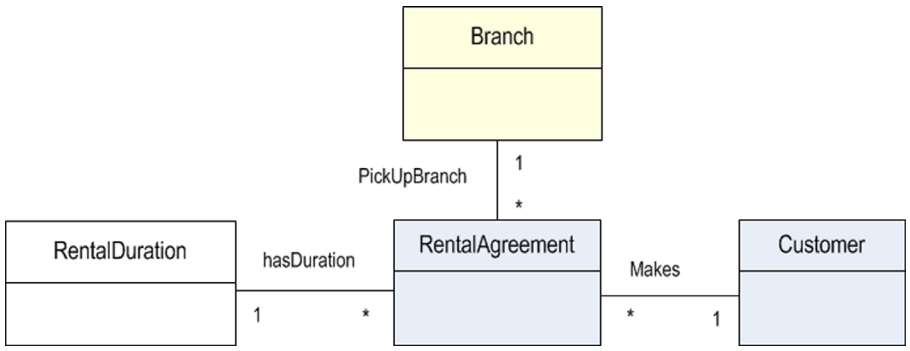
- In the "Relationships between MD models and SW formalisms" subsection we present the current straightforward solutions relating multidimensional modeling and the SW technologies.
- The "Advanced reasoning services for MD models" subsection discusses which advanced features from the SW become essential when modeling the data warehouse. We highlight two main reasoning tasks tightly related to data warehousing: reasoning on data aggregation and transitive functional dependencies.
- Finally, we wrap up the discussion in the "Advanced Reasoning and MD Modeling" subsection by presenting how these advanced features have been exploited by current modeling methods.

Multidimensional modeling is a well-known paradigm in the area of data warehouses and databases in general. It was firstly introduced by Ralph Kimball at the logical level (Kimball, 1996) and later by Matteo Golfarelli at the conceptual level (Golfarelli et al., 1998). Since then, many approaches have introduced, or improved, multidimensional models either at the logical or the conceptual level (Pedersen et al., 1999; Vassiliadis, 1998; Abelló et al., 2006). Multidimensionality is based on the dichotomy Fact-Dimension. This paradigm aims at analyzing the instances of a kind of fact (or subject of analysis), from different points of view (i.e., the analysis dimensions). For example, we may want to analyze "rental agreements" (our kind of fact) depending on the customer profile, the duration of the agreement, the branch where it was picked up, the branch where it was dropped off, and so on. Factual instances can be placed in a multidimensional space (whose axis are the analysis dimensions), known as data cube. Thus, each factual instance is identified by a point on each analysis dimension.

Several measures (i.e., variables or metrics) use to be available for each fact instance. For example, we may be interested in analyzing the basic price (with no benefit considered) or the best price (the lowest price offered yet providing benefit) of the rental agreements. Furthermore, the multidimensional model also provides foundations to study / analyze the available measures at various aggregation levels hierarchically structured in the dimensions. For example, we would like to study the basic price we may offer regarding the kind of customer (instead of a specific customer), country (instead of the city from where the deal was done) and kind of branch (instead of the specific branch where the car was picked up). Indeed, aggregation is one of the main characteristics of the multidimensional model, setting foundations for the well-known roll-up and drill-down operators.

Usually, the multidimensional analysis of data has been restricted to the well structured information sources within the company. Nevertheless, (Inmon et al., 2008) outlines the opportunity and importance of using unstructured and semi-structured data (either textual or not) in the decision making process. These data could still come from the sources in the company, but also from the Web. It is clear the benefit of enriching our multidimensional model with information coming from the Web, since it can provide new points of view, new aggregation levels, or even new measures to analyze. Anyway, either coming from inside or outside the company, data must be annotated somehow in order to be used for decision making (it is the addition of data and metadata that generates information). Among the first attempts to include web data in analysis schemas were (Jensen et al., 2001) and (Vrdoljak et al., 2003). These works generate multidimensional schemas from XML. However, this can be clearly improved by using current semantic web

Figure 5. UML multidimensional schema



formalisms. A review and deep discussion of data warehouse approaches for XML and Web data can be found in (Pérez-Martínez et al, 2008a).

Relationships Between MD Models and SW Formalisms

To the best of our knowledge, there is only one work, (Hacid et al., 1997), that shows the relationship of multidimensionality and the SW by proposing a model based on DL. This work emphasizes on multidimensional operations and defines a data cube as follows:

```
Cube  ≡ ∀hasCell.(∀hasDuration.
RentalDuration ⊓ ∀makes.Customer ⊓
∀pickupBranch.Branch)
```

The problem with such definition is that strict role-typing is assumed (meaning that the range as well as the domain of a role is a subset of a concept and it cannot freely relate any instance in the database), which is a rather common assumption in the conceptual modeling area, but not for DL ontologies. For example, the has role in Figure 1 has two different ranges DemandXGroup and DrivingLicense. Thus, it cannot be represented by assuming strict role-typing.

In the last years, some multidimensional models based on UML have been presented. For example, (Abelló et al., 2006) proposed a UML meta-model extension to capture and describe the multidimensional concepts in a UML class diagram (like that in Figure 5).

Relevantly, nowadays it is well known how to capture and represent UML and ER class diagrams by means of DL (Berardi et al., 2005). Indeed, most of the UML features are known to be captured by well-behaved DLs such as the DL-Lite family (Artale et al., 2009). This fact opens a new bridge between the semantic web and multidimensional formalisms, as the work presented in (Abelló et al. 2006) can be easily translated into DL. Thus, the multidimensional schema in Figure 5 would be translated into:

```
Cube ≡ RentalAgreement ⊓
∃hasDuration.RentalDuration ⊓
∃makes.Customer ⊓ ∃pickupBranch.
Branch
```

However, two issues still remain open: how to reason with aggregation relationships (which, in any case, can be represented by roles) and fact IDs (to be captured as keys).

Advanced Reasoning Services for MD Models

Logic frameworks are the foundations of the **SW** that provide standard reasoning services that, in turn, provide a theoretical and algorithmic basis

that can be used for the design and evolution of the data warehouse.

Reasoning Services to Support Data Aggregation

In the field of ontology population we can find some approaches dealing with aggregated ontology instances. For example, (Danger & Berlanga, 2008) presents a system to extract complex ontological instances and relations from unstructured environments such as the web. This work, based on the DL language SHOIQ(D), presents a set of operations for merging and aggregating instances. Thus, aggregation of instances (by means of paths between concepts) can be asserted on a reference ontology. However, this work is thought to capture aggregation relationships between instances when parsing unstructured text related to an ontology, but no further reasoning aggregation services can be exploited. Many inference problems can be reduced to satisfiability and containment but this is not the case of aggregation, a new kind of inference problem crucial in the data warehouse scenario. (Jarke et al, 2000) outlines four different open problems associated to aggregation:

1. To decide whether queries (or views) are satisfiable,
2. whether one is contained in another,
3. whether one is refined by another (i.e., one can be calculated from the other despite not being a subset), and
4. to answer a query.

(Baader et al., 2003) introduced aggregates over concrete domains. The resulting language is called $ALC(\Sigma)$, and extends the basic language ALC with concrete domains, functional roles (i.e., relations between instances and values from these domains) with predicates expressing value comparisons, and a limited set of aggregation functions, namely: sum, min, max and count. Aggregates are introduced through complex functional roles of the form $\Gamma(R \circ u)$, which relate each instance with the aggregate Γ over all the values reachable from R followed by the functional role u (note that $\circ$ doesn't stand for role composition, but for an ad-hoc constructor). For example, we can define the following complex functional role sum(monthPickUpTime ◦ basicPrice) to find out which was the income based on the rental agreements done per month, where the complex functional role sum(monthPickUpTime ◦ basicPrice) relates an individual to the sum over all values reachable from monthPickUpTime (extracted from the pickUpTime date from the OpenedRental class) followed by basicPrice (inherited from RentalAgreement). Thus, a new, complex functional role is built using the aggregation function sum, the role name PickUpTime, and the functional role basicPrice.

However, DLs formalisms present important limitations for representing complex measures and aggregations. (Baader et al., 2003) also demonstrate that handling aggregates in DLs usually leads to undecidability for problems (1) and (2), even for very simple aggregates such as sum and count. Moreover, decidable cases present a level of computational complexity too high for practical real-world applications and thus, there are no reasoners able to deal with the advanced features required by these new constructors.

Problem (3) is also known as query rewriting from materialized views (i.e., if a query can be refined from materialized views, given that answering queries from views is cheaper than answering them from the sources). For example, consider that the user asks for a query Q asking for the average best price offered by every rental agreement done, per year and kind of customer (ranged by age). Let us consider now two complementary materialized views V and V', which are defined as rental agreements in EU countries and rental agreements in non-EU countries, per day and customer, respectively. Then, Q can be answered by computing the average best price (grouping the instances by year and date of birth

and then, summing up the best prices of each group and dividing the result obtained by the number of deals on the group) from both views. In general, according to (Mannino, 2007), a view could be used to answer a certain aggregation query if:

1. the selection predicate of the query is subsumed by that of the view,
2. the aggregation level (i.e., the data granularity demanded) is coarser or equal than that in the query and
3. if the aggregation function used is the same (or compatible) with that of the view.

For example, the first condition states that a view storing data related to EU cannot be used to answer a query regarding the USA, as the query selection predicate is not contained into that of the view (thus, data needed is missing). The second condition states that data at month granularity level cannot be used to answer queries at day level. However, it can be used to answer queries at year level. Finally, a query using the average function to aggregate data can be answered from a view capturing both the sum and the number of instances involved, but it would not be possible if, for example, the number of instances is not captured in the view. In general, query rewriting is too complex to be exhaustively checked. Thus, DBMSs use heuristics to rewrite queries in terms of materialized views.

Recently, the problem of answering queries over ontologies (4) has also been addressed. Firstly, very expressive DLs (such as fragments of OWL 1 DL) have been considered and the data complexity (i.e., measured in the size of the intensional level) of the problem was characterized. It has been shown that for expressive ontology languages like SHIQ, answering unions of conjunctive queries (the typical scenario considered in this area) is coNP-complete in data complexity (Glimm et al., 2008) and (Ortiz et al., 2008). For this reason, most recent works have focused on less expressive languages providing a nice trade-off with data complexity on query answering. One relevant result is that of the DL-Lite family (Artale et al., 2009), which is known to be LogSpace in data complexity (i.e., it can be reduced to standard query evaluation over DBs) when considering union of conjunctive queries. In this scenario, (Calvanese et al., 2008) studied the complexity of extending query answering of union of conjunctive queries with aggregates. However, the authors shown that there are some assumptions related to aggregate queries that are difficult to overcome in formalisms such as DL. DL ontologies deal with incomplete information (i.e., the intensional level –i.e., IL- is a partial description of the domain of interest that the ontology completes by characterizing the space of all the possible, compatible intensional levels – i.e. IL'-). This fact is known as the open-world assumption in contrast to the closed-world assumption assumed for relational databases. For this reason, the authors argue that assuming certain answers (**i.e.,** asking for an aggregate query Q over each IL') may produce meaningless answers, as it may return a different answer for each IL' and eventually obtain an empty intersection. The authors explore the conditions to be fulfilled so aggregate queries make sense; i.e., that the answer exists and can be computed (Calvanese et al., 2008) and (Thorne et al., 2009). However, although, in these cases, the computational complexity is yet LogSpace regarding data complexity, the result happens to be too restrictive and not feasible for most real cases.

Transitive Functional Dependencies

It is well-known that patterns used to look for multidimensional concepts in multidimensional design are based on functional dependencies (Kimball et al., 1996; Romero et al., 2010b) because of two reasons. On the one hand, the multidimensional space is arranged by the analysis dimensions of a given fact. Each instance of data is identified (i.e., placed in the multidimensional space) by a point in each of its analysis dimensions (i.e., the

multidimensional space axis). Conceptually, it entails that the fact must be related to each analysis dimension by a to-one conceptual relationship. That is, a functional dependency. Furthermore, since two different instances of data cannot be placed in the same point of the multidimensional space, it is compulsory that a set of dimensions (known as the multidimensional base) functionally determine the fact. On the other hand, measures can be thought as class attributes (i.e., OWL properties with data types as range) and thus, they are functionally dependent on the fact. For this reason, computing functional dependencies is essential for data warehouse modelling.

Functional dependencies discovery has been tightly related to the databases field and it has been typically addressed either at the logical or physical level. Addressing this task at the logical level entails that results obtained are tied to the design decisions made when devising the system. Most approaches try to overcome the lack of semantics in a logical schema by addressing this task at the physical level (e.g., (Jensen et al., 2004) already addressed this approach for data warehousing), but these result in computationally expensive solutions that register drops in performance when a large number of attributes or instances are processed.

Given that DL ontologies provide a semantically rich formalism, (Romero et al., 2010b) discusses the benefits of computing the functional dependencies closure (i.e., by considering role chains, also known as role compositions) at the conceptual level, if the ontology provides multiplicities for the roles. Composition allows expressing joined relationships making the intermediate involved concepts implicit, but it was not supported by OWL 1 and it is not yet fully supported by current reasoners. For this reason, (Romero et al., 2010b) presented an ad hoc algorithm to compute transitive functional dependencies based, partially, on standard reasoning services. A refined approach to fully exploit DL reasoning services and compute functional dependencies is discussed in (Romero et al, 2009). This work shows that role composition can be simulated by means of a well-behaved DL such as DL-Lite$_A$ (Artale et al., 2009), by exploiting its query answering services for conjunctive queries. Similarly, (Danger & Berlanga, 2008) presents the adaptation of an algorithm to select functional dependencies. This approach focuses on building dimension hierarchies, which are shaped to maximize the information gain.

Recently, OWL 2 provides a construct to assert a property that is the composition of several properties. Such axioms are known as complex role inclusions in SROIQ (in which OWL 2 is based on). However, SROIQ defines regularity conditions for decidability (mainly, prevent cyclic definitions involving hierarchies with property chains).

A specific case of functional dependency is the key concept, which is relevant for modeling tasks to identify instances. Thus, it is worth remarking that some recent works have addressed the issue of asserting keys on ontologies: OWL 2 allows asserting them under certain safe rules (Motik et al., 2005). With the same spirit, (Calvanese et al., 2008) shows how path-based identification constraints can be considered in both, very expressible DLs such as ALCQIbreg and the DL-Lite family. In general, path-based constraints are a powerful class of identification constraints (which allow using roles, inverses, and paths) that capture sophisticated forms of identifications (although it does not suit to the traditional key definition). This kind of constrains though, happen to be problematic in the general case and the authors propose a restricted form of these constraints, called local, that does not increase the complexity of reasoning both in very expressive DL and in the tractable DL-Lite family. Local path-based constraints still remain interesting for most real scenarios.

Advanced Reasoning and MD Modeling

Nowadays we can find several approaches exploiting the advanced reasoning services presented in

previous section. We can classify its use in three areas: discovering dimension hierarchies from ontology fragments, fact extraction and integrating concepts from several ontologies to produce the multidimensional schema (i.e., carry out the fact extraction and dimension discovery from several sources).

Mainly, automatically discovering dimension hierarchies has attracted researchers' attention. Regardless of the data source formalisms, it can be thought as a way to summarize schemas (in the sense of allowing the user to grasp, at a glance, the information contained by the schema), as considered in (Yang et al., 2009). This approach summarizes relational databases by exploiting a metric distance over the schema to cluster the most relevant tables (i.e., facts) and present a summarization of the ontology topology around it (i.e., the analysis dimensions). In the data warehousing area, (Romero et al., 2010b) present a similar way of proceeding. There, distance metrics and multidimensional patterns are introduced to identify facts, measures and dimensional concepts (which, eventually, will form the analysis dimensions). (Danger & Berlanga, 2008) also present its own approach to identify dimension hierarchies based on functional dependencies. Both approaches foundations are based on discovering functional dependencies as discussed in previous section.

Note, however, that these two approaches assume a single domain ontology. However, in certain scenarios, like biomedicine, this is not a fair assumption, as several domain ontologies are available and should be considered. To overcome this limitation, (Nebot et al., 2009) introduces the Semantic Data Warehouse (SDW), which can contain semantic annotations defined in several large inter-linked ontologies. Thus, it presents an approach to tackle the semantic integration of ontologies. Specifically, the authors focus on only integrating the right amount of knowledge needed for specific multidimensional analysis (for example, data related to opened rental agreements, or data related to customers, etc.) and a method for designing, validating and building these schemas is detailed.

Oppositely, (Romero et al., 2010a) presents a different approach for the same scenario. There, the authors propose to extract a multidimensional schema from each domain ontology (for example, as suggested in (Romero et al., 2010b)) and later conciliate those results in a single, detailed multidimensional schema. Thus, the semantic integration of the resulting schemas is performed a posteriori.

FUTURE RESEARCH DIRECTIONS

In this section, we discuss future and emerging trends in the collaboration of both OLAP and SW areas. First, we discuss the use of Semantic Web techniques for OLAP query recommendation. Second, the use of Semantic Web data to provide context information for multidimensional data warehouses is discussed. Finally, issues related to the scalable and efficient storage of Semantic Web data are treated.

Query Recommendation

The use of semantic web technologies has increased the amount of steps that can be (semi-) automatically performed in the data warehousing system design process. This scenario opens a new and interesting researching topic: query recommendation. Query recommendation has gained relevance in the database community in the last years. In the data warehousing community, (Giacometti et al., 2009a; 2009b) introduced their approach for multidimensional query recommendation. These works take advantage of the OLAP tool query log to exploit the knowledge it contains and assess the user in his / her future queries. However, it remains open if the knowledge extracted from performing some tasks automatically (mainly, when involving the user requirements in the design phase) can help the user when querying the DW.

A promising approach would be considering SW technologies to help in this process.

Contextualization of Data Warehouses with SW Data

Next generation of BI systems require further research on data warehouse contextualization. This is an alternative way of doing global search on separate structured and unstructured data sources and integrating both types of information in order to semantically enrich the process of analyzing business data. The process of analyzing contextualized data was initially defined by (Priebe & Pernul, 2003) with the purpose of communicating the user context among different *portlets* representing different data sources and, therefore, to support such integration in a generic way. With this approach, the system can provide the user with the documents that are related to the information that is being currently displayed in an OLAP report. In order to solve the problem of the heterogeneity of both systems, they propose to use ontological concept mapping.

The formal definition of contextualized warehouse was later proposed in (Pérez-Martínez et al., 2008a). This work consists in the integration of a corporate warehouse of structured data with a warehouse of text-rich XML documents. With this purpose they define a new information retrieval model to select the context of analysis from the document warehouse and to associate to each fact of the analysis cube the set of documents that are more related (Pérez-Martínez et al., 2009). In a contextualized warehouse, the user specifies an analysis context by supplying a sequence of keywords (i.e., an IR condition like "financial crisis"). The analysis is performed on a new type of OLAP cube, called R-cube, which is materialized by retrieving the documents and facts related to the selected context. A new set of OLAP operators allows users to find out the relationship that can occur between the data in the cubes and the information in the documents.

Another contribution someway related to data warehouse contextualization has been recently made by (Castellanos et al., 2010). With the purpose of identifying external events that may affect the enterprise operations, in this work, novel techniques of information extraction and correlation measurement are applied to extract relevant information from two disparate sources of unstructured data, and determine which documents are correlated. They have also developed several functions for information extraction and analytics.

All these systems have in common that the integration of data needs to be made on the fly, in a dynamic and efficient way. This is the only way of ensuring that the answer can satisfy the requirements of each specific analysis operation involved in a decision making process. The application of the technology developed in the context of Semantic Web research to this problem is a work that remains to be done and that surely would bring further benefits. As an example, semantic annotation technologies can help to identifies generic and domain specific entities, relationships as well as semantic time expressions. This would allow us to improve the integration and joint analysis of structured and unstructured data coming from heterogeneous data sources.

SW Storage Issues

A RDF store somewhat resembles a data warehouse in the sense that they are intended to store huge amounts of read-only data under very simple schemas (just three columns). Here, a very interesting direction is the application of data warehousing techniques like bulk-loading in order to manage huge amounts of triples more efficiently. For example, the 3XL system (Liu et al., 2011) have shown that using bulk-loading techniques in combination with main memory storage and intelligent data partitioning can result in huge speedups for bulk operations such as loading and querying large amounts of triples.

Unlike traditional multidimensional data warehouses, data in triple stores is not subject-oriented, and there is not a predefined set of dimensions and measures to which all data must refer. Identifying which concepts and properties are of interest for representing dimensions and measures is indeed one of the key points for integrating OLAP techniques and SW data. For example, the approach recently presented in (Niinimaki, M. & Niemi, 2009), proposes to build DWs from RDF data through SPARQL queries that identifies the target dimensions and measures. Similarly, (Nebot & Berlanga, 2010) proposes to define multidimensional schemas from the concepts of the ontology to which the triple store refers, and then semi-automatically generate an OLAP cube according to that schema.

A further promising direction is the application of bitmap indexing techniques to semantic web data management, including efficient reasoning. Bitmap indices have traditionally been applied for the dimensional data found in data warehouses. Recent advances in compressed bitmap indices (Deliege et al., 2010) have shown that significant speedups can be achieved for both storage and query speed when performing complex operations, and it is believed that these advantages can be exploited for more efficient SW data management.

Finally, a huge challenge related to both scalability and semantics lies in the transition of business intelligence into so-called "cloud intelligence", where the full potential of cloud computing is realized (Pedersen, 2010). Challenges related to the SW includes using SW technologies to achieve location and device independence, providing intelligence as a service, scaling intelligence services to a global level through techniques such as map-reduce and beyond, and providing agility, the ability to assemble the necessary resources on demand, not only in terms of computing power, but also in terms of data sources.

CONCLUSION

Business intelligence requires the integration of massive data coming from disparate data sources. Traditionally, these data sources were limited to corporate transactional databases, relying on the relational data model mostly. As a consequence, BI research has been mainly focused on this data model so far. However, the increasing availability of valuable knowledge resources, public databases, and a great variety of information sources in the Web are requiring new BI models and techniques for dealing with structured and unstructured data at the same time.

The Semantic Web is clearly targeted to facilitate the integration of all these web resources by providing semantic annotations that follow some agreed ontologies. In this chapter, we have firstly reviewed the major efforts to bring semantics to both web resources and BI application data, which is a previous step to a true integration of both worlds. Then, we have presented the main approaches that have utilized SW technology to face classical issues of data warehouse design and implementation. These approaches have brought BI methods closer to SW data and vice versa. However, several issues must be addressed before achieving the actual integration of BI and SW, to mention a few: new user recommendation methods, data warehouse contextualization, massive SW data storage for analytical tasks, and performing BI in the cloud regarding fully distributed data and services. These issues and many others to come constitute an open intelligent information systems research area of great interest.

REFERENCES

W3C OWL Working Group (Eds.). (2009). *OWL 2 Web ontology language document overview*. Retrieved from http://www.w3.org/TR/ owl2-overview/

Abadi, D. J., Marcus, A., Madden, S., & Hollenbach, K. J. (2007). Scalable Semantic Web data management using vertical partitioning. In *Proceedings of VLDB Conference*, (pp. 411–422).

Abelló, A., Samos, J., & Saltor, F. (2006). YAM^2: A multidimensional conceptual model extending UML. *Information Systems*, *31*(6), 541–567.

ARQ. (n.d.). A SPARQL processor for Jena. Retrieved from http://jena.sourceforge.net/ ARQ/

Artale, A., Calvanese, D., Kontchakov, R., & Zakharyaschev, M. (2009). The DL-Lite family and relations. *Journal of Artificial Intelligence Research*, *36*, 1–69.

Atserias, J., Zaragoza, H., Ciaramita, M., & Attardi, G. (2008). Semantically annotated snapshot of the English Wikipedia. *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*.

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., & Ives, Z. (2008). Dbpedia: A nucleus for a Web of open data. *Proceedings of the Conference on The Semantic Web*, (pp. 722–735).

Baader, F., Brandt, S., & Lutz, C. (2005). Pushing the EL envelope. *In Proceedings of the 19th International Joint Conference on Artificial Intelligence* (IJCAI) (pp. 364-369), Edinburgh, Scotland.

Baader, F., Calvanese, C., McGuinness, D. L., Nardi, D., & Patel-Schneider, P. F. (Eds.). (2003). *The description logic handbook: Theory, implementation, and applications*. Cambridge University Press.

Baader, F., Lutz, C., & Suntisrivaraporn, B. (2006) CEL-A polynomial-time reasoner for life science ontologies. In *Proceedings of the Third International Joint Conference of Automated Reasoning, IJCAR 2006*, (pp. 287-291).

Baader, F., & Sattler, U. (2003). Description logics with aggregates and concrete domains. *Information Systems*, *28*(8), 979–1004.

Berardi, D., Calvanese, D., & De Giacomo, G. (2005). Reasoning on UML class diagrams. *Artificial Intelligence*, *168*(1-2), 70–118.

Boag, S., Chamberlin, D., Fernándex, M. F., Florescu, D., Robie, J., & Siméon, J. (2007). *XQuery 1.0: An XML query language*. Retrieved from http://www.w3.org/ TR/xquery/

Bray, T., Paoli, J., Sperberg-McQueen, C. M., & Maler, E. (Eds.). (2000). *Extensible markup language* (XML) 1.0 (2nd ed.). W3C Recommendation, 6 October 2000. Retrieved from http://www.w3.org/ TR/REC-xml/

Brickley, D., & Guha, R. V. (Eds.). (2004). *RDF vocabulary description language 1.0: RDF schema*. Retrieved from http://www.w3.org/ TR/ rdf-schema/

Brin, S. (1998). *Extracting patterns and relations from the World Wide Web*. International Workshop on The World Wide Web and Databases, (pp. 172–183).

Broekstra, J., Kampman, A., & van Harmelen, F. (2003). An architecture for storing and querying RDF data and schema information. In *Spinning the Semantic Web* (pp. 197–222). Sesame.

Cafarella, M. (2009). Extracting and querying a comprehensive Web database. In *Proceedings of the Conference on Innovative Data Systems Research (CIDR)*. Asilomar, CA. Retrieved from http://www.cs.washington.edu/ homes/mjc/papers/ cafarella-cidr09.pdf

Calvanese, D., Carroll, J., De Giacomo, G., Herman, I., Parsia, B., Patel-Schneider, P., & Ruttengerb, A. (2008). *OWL 2 Web ontology language: Profiles*. Retrieved from http://www.w3.org/TR/2008/WD-owl2- profiles-20081008/

Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., & Rosati, R. (2008). Path-based identification constraints in description logics. In *Proceedings of the 11th International Conference on Principles of Knowledge Representation and Reasoning* (KR) (pp. 231-241). Sydney, Australia.

Calvanese, D., Kharlamov, E., Nutt, W., & Thorne, C. (2008). *Proceedings of the 2nd International Workshop on Ontologies and Information Systems for the Semantic Web (ONISW)* (pp. 97-104). Napa Valley, California, USA.

Castellanos, M., Wang, S., Dayal, U., & Gupta, C. (2010) SIE-OBI: A streaming information extraction platform for operational business intelligence. In *Proceeding of SIGMOD Conference 2010*, (pp. 1105-1110).

Cimiano, P., Handschuh, S., & Staab, S. (2004). Towards the self-annotating Web. In *Proceedings of the 13th International Conference on World Wide Web* (pp. 462-471). New York, NY: ACM. doi:10.1145/988672.988735

Cimiano, P., & Völker, J. (2005). Text2Onto. In *Proceedings of the Conference on Natural Language Processing and Information Systems* (pp. 227-238). Retrieved from http://dx.doi.org/10.1007/ 11428817_21

Clark & Parsia. (2010). *Pellet: The OWL 2 reasoner*. Retrieved from http://clarkparsia.com/pellet/

Danger, R., & Berlanga, R. (2008). A Semantic Web approach for ontological instances analysis. *Software and Data Technologies. Communications in Computer and Information Science*, *22*, 269–282.

Danger, R., & Berlanga, R. (2009). Generating complex ontology instances from documents. *Journal of Algorithms*, *64*(1), 16–30.

Dean, M., Schreiber, G., Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., et al. Stein, L.A. (2004). *OWL Web ontology language reference*. W3C Recommendation 10 February 2004. Retrieved from http://www.w3.org/TR/owl-ref/

Declerck, T., Krieger, H., Saggion, H., & Spies, M. (2008). Ontology-driven human language technology for semantic-based business intelligence. In *Proceeding of the 2008 Conference on ECAI 2008: 18th European Conference on Artificial Intelligence* (pp. 841-842). IOS Press. Retrieved from http://portal.acm.org/ citation.cfm?id=1567281.1567497

Deliège, F., & Pedersen, T. B. (2010). Position list word aligned hybrid: Optimizing space and performance for compressed bitmaps. In *Proceedings of EDBT*, (pp. 228-239).

Dublin Core Metadata Initiative. (n.d.). *Website*. Retrieved from http://dublincore.org/

Giacometti, A., Marcel, P., & Negre, E. (2009a). Recommending multidimensional queries. In *Proceedings of the 11th International Conference on Data Warehousing and Knowledge Discovery* (DaWaK 2009), (pp. 453-466).

Giacometti, A., Marcel, P., Negre, E., & Soulet, A. (2009b). Query recommendations for OLAP discovery driven analysis. In *Proceedings of the ACM 12th International Workshop on Data Warehousing and OLAP* (DOLAP 2009), (pp. 81-88).

Glimm, B., Horrocks, I., Lutz, C., & Sattler, U. (2008). Conjunctive query answering for the description logic SHIQ. *Journal of Artificial Intelligence Research*, *31*, 151–198.

Golfarelli, M., Maio, D., & Rizzi, S. (1998). The dimensional fact model: A conceptual model for data warehouses. *International Journal of Cooperative Information Systems*, *7*(2-3), 215–247.

Grosof, B. N., Horrocks, I., Volz, R., & Decker, S. (2003). Description logic programs: Combining logic programs with description logic. *In Proceedings of the 12th International World Wide Web Conference* (WWW) (pp. 48-57). Budapest, Hungary.

Haarslev, V., & Möller, R. (2001). Description of the RACER system and its applications. In *Working Notes of the 2001 International Description Logics Workshop*. Retrieved from CEUR-WS.org

Hacid, M.-S., & Sattler, U. (1997). An object-centered multi-dimensional data model with hierarchically structured dimensions. In *Proceedings of the IEEE Knowledge and Data Engineering Exchange Workshop (KDEX)* (pp. 65-72).

Harris, S., & Gibbins, N. (2003). 3store: Efficient bulk RDF storage. In *Proceedings of PSSS*, 2003.

Harth, A., Umbrich, J., Hogan, A., & Decker, D. (2007). YARS2: A federated repository for querying graph structured data from the Web. In *Proc. of 6th International Semantic Web Conference/ 2nd Asian Semantic Web Conference*, (pp. 211-224).

Herman, I. (2008). *RDFS and OWL 2 RL generator service*. Retrieved from http://www.ivan-herman.net/Misc/2008/owlrl/

Hermit. (2010). *Hermit OWL reasoner*. Retrieved from http://hermit-reasoner.com/

Horrocks, I. (1998). Using an expressive description logic: Fact or fiction? *In Proceedings of 6th Conference on Principles of Knowledge Representation and Reasoning*, (pp. 636-649). Morgan Kaufmann. Retrieved from http://owl.man.ac.uk/factplusplus/

Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosof, B., & Dean, M. (2004). *SWRL: A Semantic Web rule language combining OWL and RuleML*. W3C Consortium, Member submission. Retrieved from http://www.w3.org/ Submission/ SWRL/

Horrocks, I., van Harmelen, F., & Patel-Schneider, P. (2001). *Reference description of the DAML+OIL ontology markup language*. Retrieved from http://www.daml.org/ 2000/12/reference.html

Hustadt, U., Motik, B., & Sattler, U. (2005). Data complexity of reasoning in very expressive description logics. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence* (IJCAI) (pp. 466-471), Edinburgh, Scotland.

Inmon, W. H., Strauss, D., & Neushloss, G. (2008). *DW 2.0: The architecture for the next generation of data warehousing*. Morgan Kauffman.

Jarke, M., Lenzerini, M., Vassiliou, Y., & Vassiliadis, P. (Eds.). (2000). *Fundamentals of data warehouses*. Springer.

Jensen, M. R., Holmgren, T., & Pedersen, T. B. (2004). Discovering multidimensional structure in relational data. In *Proceedings of the 6th International Conference on Data Warehousing and Knowledge Discovery (DaWaK)* (pp. 138-148).

Jensen, M. R., Møller, T. H., & Pedersen, T. B. (2001). Converting XML data to UML diagrams for conceptual data integration. In *Proceedings of the 1st International Workshop on Data Integration over the Web (DIWeb)* (pp. 17-31), Interlaken, Switzerland.

Karvounarakis, G., Alexaki, S., Christophides, V., Plexousakis, D., & Scholl, M. (2002). RQL: A declarative query language for RDF. In *Proceedings Eleventh International World Wide Web Conference*, (pp. 592-603).

Kimball, R. (1996). *The data warehouse toolkit: Practical techniques for building dimensional data warehouses*. John Wiley.

Kiryakov, A., Popov, B., Terziev, I., Manov, D., & Ognyanoff, D. (2004). Semantic annotation, indexing, and retrieval. *Web Semantic*, *2*(1), 49–79.

Klyne, G., Carroll, J., & McBride, B. (Eds.). (2004). *Resource description framework (RDF) concepts and abstract syntax*. W3C Recommendation 10 February 2004. Retrieved from http://www.w3.org/ TR/rdf-concepts/

Lenzerini, M. (2002). Data integration: A theoretical perspective. In *Proceedings of the Twenty-first ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems* (PODS), (pp. 233-246).

Liu, X., Thomsen, C., & Pedersen, T. B. (2010). *3XL: Supporting efficient operations on very large OWL lite triple-stores. Information Systems 36(4):765-781 (2011)*. Preprint available as DBTR no. 28. Retrieved from http://dbtr.cs.aau.dk

Maedche, A., Neumann, G., Staab, S., & Saarbruecken, G. (2003). *Bootstrapping an ontology-based information extraction system* (pp. 345–359). Intelligent Exploration of the Web.

Mannino, M. V. (2007). *Database design, application development, & administration*. McGraw Hill.

Motik, B., & Rosatti, R. (2010). Reconciling description logics and rules. *Journal of the ACM*, *57*(5), 1–63.

Motik, B., Sattler, U., & Studer, R. (2005). Query answering for OWL-DL with rules. *Journal of Web Semantics: Science*, *Services and Agents on the World Wide Web*, *3*(1), 41–60.

Nebot, V., & Berlanga, R. (2009). Efficient retrieval of ontology fragments using an interval labeling scheme. *Information Sciences*, *179*(24), 4151–4173.

Nebot, V., & Berlanga, R. (2010). Building data warehouses with semantic data. *In Proceedings of the 1st International Workshop on Business Intelligence and the Web (BEWEB)*. Lausanne, Switzerland.

Nebot, V., Berlanga, R., Pérez, J., Aramburu, M., & Pedersen, T. (2009). Multidimensional integrated ontologies: A framework for designing semantic data warehouses. *Journal of Data Semantics*. *Special Issue on Semantic Data Warehouses*, *13*, 1–36.

NeOn Project team. (2010). *Neon tool-kit*. Retrieved from http://neon-toolkit.org/wiki/Main_Page

Neumann, T., & Weikum, G. (2008). RDF-3X: A RISC-style engine for RDF. In *Proc. Of the VLDB Endowment, 1*(1), 647-659.

Neumann, T., & Weikum, G. (2010). The RDF-3X engine for scalable management of RDF data. *The VLDB Journal*, *19*(1), 91–113.

Niemi, T., Toivonen, S., Niinimäki, M., & Nummenmaa, J. (2007). Ontologies with Semantic Web/Grid in data integration for OLAP. *International Journal on Semantic Web and Information Systems*, *3*(4), 25–49.

Niinimäki, M., & Niemi, T. (2009). An ETL process for OLAP using RDF/OWL ontologies. *Journal of Data Semantics*. *Special Issue on Semantic Data Warehouses*, *13*, 97–119.

ORACLE-RDF. (n.d.). *Oracle technical network, semantic technologies center*. Retrieved from http://www.oracle.com/technology /tech/semantic_technologies /index.html

Ortiz, M., Calvanese, D., & Either, T. (2008). Data complexity of query answering in expressive description logics via tableaux. *Journal of Automated Reasoning*, *41*(1), 61–98.

Pedersen, T., & Jensen, C. (1999). *Multidimensional data modeling for complex data*. IEEE International Conference on Data Engineering (ICDE) (pp. 336-345).

Pedersen, T. B. (2010). Research challenges for cloud intelligence (invited talk). In *Proceedings of the Workshop on Business IntelligencE and the WEB (BEWEB),* 2010.

Pérez-Martínez, J. M., Berlanga, R., & Aramburu, M. J. (2009). A relevance model for a data warehouse contextualized with documents. *Information Processing & Management*, *45*(3), 356–367.

Pérez-Martínez, J. M., Berlanga, R., Aramburu, M. J., & Pedersen, T. B. (2008a). Contextualizing data warehouses with documents. *Decision Support Systems*, *45*(1), 77–94.

Pérez-Martínez, J. M., Berlanga, R., Aramburu, M. J., & Pedersen, T. B. (2008b). Integrating data warehouses with Web data: A survey. *IEEE Transactions on Knowledge and Data Engineering*, *20*(7), 940–955.

Priebe, T., & Pernul, G. (2003). Towards integrative enterprise knowledge portals. In *Proceedings of the Twelfth International Conference on Information and Knowledge Management* (*CIKM*) (pp. 216-223)

Protégé. (2010). *Protégé Project*. Stanford Center for Biomedical Informatics Research. Retrieved from http://protege.stanford.edu/

Prud'hommeaux, E., & Seaborne, A. (2008). SPARQL query language for RDF. Retrieved from http://www.w3.org/TR/rdf-sparql-query/

Reeve, L., & Han, H. (2005). Survey of semantic annotation platforms. In *Proceedings of the 2005 ACM Symposium on Applied Computing* (p. 1638).

Romero, O., & Abelló, A. (2007). Automating multidimensional design from ontologies. In *Proceedings of the ACM Tenth International Workshop on Data Warehousing and OLAP* (pp. 1-8). Lisbon, Portugal: ACM. doi:10.1145/1317331.1317333

Romero, O., & Abelló, A. (2010a). Automatic validation of requirements to support multidimensional design. *Data & Knowledge Engineering*, *69*(9), 917–942.

Romero, O., & Abelló, A. (2010b). A framework for multidimensional design of data warehouses from ontologies. *Data & Knowledge Engineering*, *69*(11), 1138–1157.

Romero, O., Calvanese, D., Abelló, A., & Rodríguez-Muro, M. (2009). Discovering functional dependencies for multidimensional design. In *Proceeding of the ACM Twelfth International Workshop on Data Warehousing and OLAP* (pp. 1-8). Hong Kong, China: ACM. doi:10.1145/1651291.1651293

Romero, O., Simitsis, A., & Abello, A. (2010). (to appear). GEM: Requirement-driven generation of ETL and multidimensional conceptual designs. *DaWaK*, *2011*.

Saggion, H., Funk, A., Maynard, D., & Bontcheva, K. (2007). Ontology-based information extraction for business intelligence. In *Proceedings of ISWC/ASWC* (pp. 843-856)

Savo, D. F., Lembo, D., Lenzerini, M., Poggi, A., Rodriguez-Muro, M., & Romagnoli, V. … Stella, G. (2010). Mastro at work: Experiences on ontology-based data access. In *Proc. of the 23rd International Workshop on Description Logics, CEUR Electronic Workshop* (pp. 20-31). Waterloo, Canada: CEUR. Retrieved from http://www.dis.uniroma1.it/quonto/

Sell, D., Silva, D. C. D., Beppler, F. D., Napoli, M., Ghisi, F. B., Pacheco, R. C. S., & Todesco, J. L. (2008). SBI: A semantic framework to support business intelligence. In *Proceedings of the First International Workshop on Ontology-Supported Business Intelligence* (pp. 1-11). Karlsruhe, Germany: ACM. doi:10.1145/1452567.1452578

Simitsis, A., Skoutas, D., & Castellanos, M. (2008). Natural language reporting for ETL processes. In *Proceedings of the 11th ACM International Workshop on Data Warehousing and OLAP* (DOLAP), (pp. 65-72).

Simitsis, A., Skoutas, D., & Castellanos, M. (2010). Representation of conceptual ETL designs in natural language using Semantic Web technology. *Data & Knowledge Engineering*, *69*(1), 96–115.

Skoutas, D., & Simitsis, A. (2006). Designing ETL processes using Semantic Web technologies. In *Proceedings of the 9th ACM International Workshop on Data Warehousing and OLAP* (DOLAP), (pp. 67-74).

Skoutas, D., & Simitsis, A. (2007). Ontology-based conceptual design of ETL processes for both structured and semi-structured data. *International Journal on Semantic Web and Information Systems*, *3*(4), 1–24.

Skoutas, D., Simitsis, A., & Sellis, T. (2009). Ontology-driven conceptual design of ETL processes using graph transformations. *Journal of Data Semantics*. *Special Issue on Semantic Data Warehouses*, *13*, 120–146.

Spahn, M., Kleb, J., Grimm, S., & Scheidl, S. (2008). Supporting business intelligence by providing ontology-based end-user information self-service. In *Proceedings of the first international workshop on Ontology-Supported Business Intelligence* (pp. 1-12). Karlsruhe, Germany: ACM. doi:10.1145/1452567.1452577

Suchanek, F. M., Kasneci, G., & Weikum, G. (2007). YAGO: A core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web* (pp. 697-706). Banff, Alberta, Canada: ACM. doi:10.1145/1242572.1242667

Suchanek, F. M., Kasneci, G., & Weikum, G. (2008). YAGO: A large ontology from Wikipedia and WordNet. *Web Semantics*, *6*(3), 203–217.

Thorne, C., & Calvanese, D. (2009). Controlled aggregate tree shaped questions over ontologies. In *Proceedings of 8th International Conference on Flexible Query Answering Systems (FQAS)* (pp.394-405).

Vassiliadis, P. (1998). Modeling multidimensional databases-Cubes and cube operations. In *Proceedings of the IEEE International Conference on Scientific and Statistical Database Management (SSDBM)* (pp. 53-62).

Vrdoljak, B., Banek, M., & Rizzi, S. (2003). Designing Web warehouses from XML schemas. In *Proceedings of the 5th International Conference on Data Warehousing and Knowledge Discovery* (DaWaK) (pp.89-98), Prague, Czech Republic.

Weld, D. S., Hoffmann, R., & Wu, F. (2008). Using Wikipedia to bootstrap open information extraction. *SIGMOD Record*, *37*(4), 62–68. doi:. doi:10.1145/1519103.1519113

Wilkinson, K., Sayers, C., Kuno, H. A., & Reynolds, D. (2003). Efficient RDF storage and retrieval in Jena2. In *Proceedings of SWDB*, (pp. 131–150).

Wu, F., & Weld, D. S. (2007). Autonomously semantifying Wikipedia. In *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management* (pp. 41-50). Lisbon, Portugal: ACM. doi:10.1145/1321440.1321449

Wu, F., & Weld, D. S. (2008). Automatically refining the Wikipedia infobox ontology. In *Proceeding of the 17th International Conference on World Wide Web* (pp. 635-644). Beijing, China: ACM. doi:10.1145/1367497.1367583

Yang, X., Procopiuc, C. M., & Srivastava, D. (2009). Summarizing relational databases. In *Proceedings of the 35th International Conference on Very Large Data Bases* (VLDB), (pp. 634-645). Lyon, France.

KEY TERMS AND DEFINITIONS

Business Intelligence: Business intelligence (BI) is a broad category of applications and technologies for gathering, integrating, analyzing, and providing access to data to help enterprise users make better business decisions. BI applications include the activities of decision support systems, query and reporting, online analytical processing (OLAP), statistical analysis, forecasting, and data mining.

Data Semantics: The semantics refers to the meaning of data. Such an abstraction can be formalized as a mapping between an object modeled, represented and stored in an information system and a set of agreed concepts (objects, relationships, behavior) representing a conceptualization of the real-world.

Semantic Web: The Semantic Web is an extension of the current Web that provides an easier way to find, share, reuse and combine information. It refers to the group of methods and technologies to allow machines to understand the meaning - or "semantics" - of information on the World Wide Web. The term was coined by the W3C director Tim Berners-Lee. He defines the Semantic Web as a web of data that can be processed directly and indirectly by machines. It is based on machine-readable information and builds on XML technology's capability to define customized tagging schemes and RDF's flexible approach to representing data.