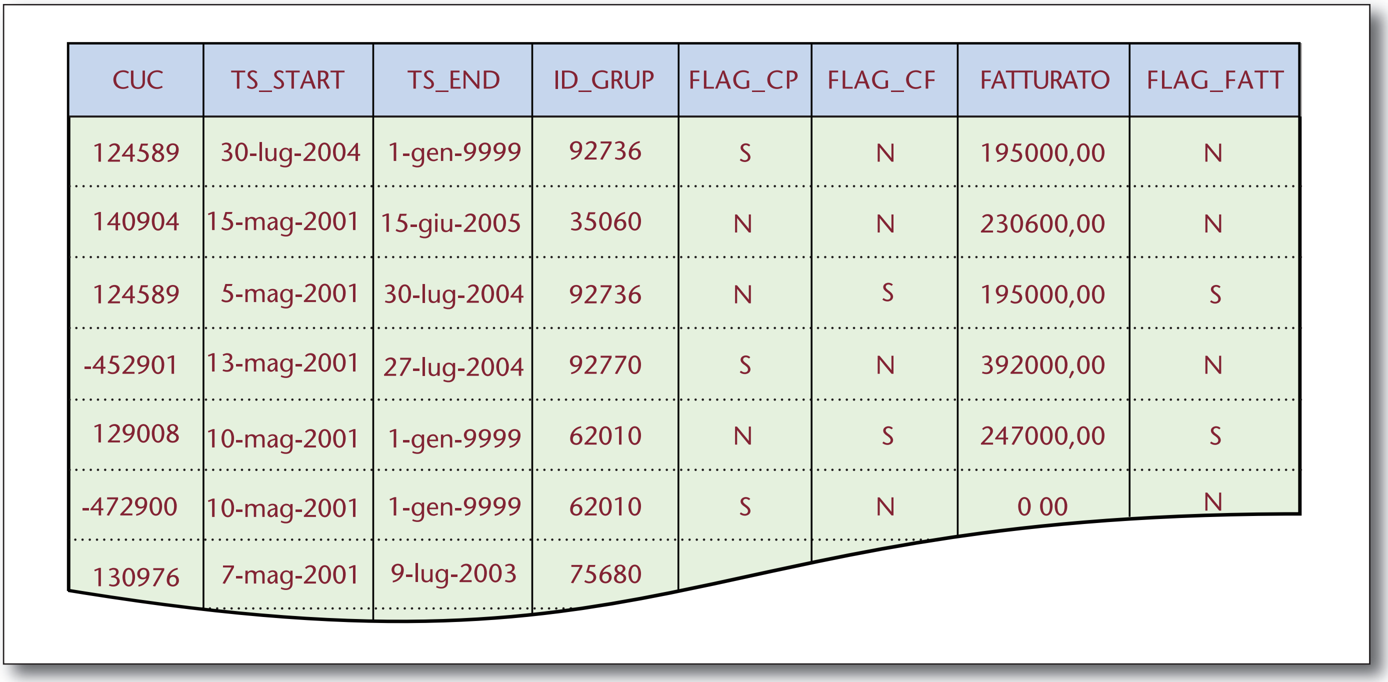
Managing Data Through the Lens of an Ontology

Maurizio Lenzerini

■ *Ontology-based data management aims at managing data through the lens of an ontology, that is, a conceptual representation of the domain of interest in the underlying information system. This new paradigm provides several interesting features, many of which have already been proved effective in managing complex information systems. This article introduces the notion of ontology-based data management, illustrating the main ideas underlying the paradigm, and pointing out the importance of knowledge representation and automated reasoning for addressing the technical challenges it introduces.*

While the amount of data stored in current information systems continues to grow, and the processes making use of such data become more and more complex, extracting knowledge and obtaining insights from these data, as well as governing both data and the associated processes, are still challenging tasks. The problem is complicated by the proliferation of data sources and services both within a single organization and in cooperating environments. Moreover, if we add to the picture the (inevitable) need for dealing with big data, and consider in particular the two *v*'s of volume and velocity, we can easily understand why effectively accessing, integrating, and managing data in complex organizations is still one of the main issues faced by the information technology (IT) industry today. Indeed, it is not surprising that data scientists spend a comparatively large amount of time in the data preparation phase of a project, compared with the data mining and knowledge discovery phase. Whether you call it data wrangling, data munging, or



CUC	TS_START	TS_END	ID_GRUP	FLAG_CP	FLAG_CF	FATTURATO	FLAG_FATT
124589	30-lug-2004	1-gen-9999	92736	S	N	195000,00	N
140904	15-mag-2001	15-giu-2005	35060	N	N	230600,00	N
124589	5-mag-2001	30-lug-2004	92736	N	S	195000,00	S
-452901	13-mag-2001	27-lug-2004	92770	S	N	392000,00	N
129008	10-mag-2001	1-gen-9999	62010	N	S	247000,00	S
-472900	10-mag-2001	1-gen-9999	62010	S	N	0 00	N
130976	7-mag-2001	9-lug-2003	75680				

Figure 1. Fragment of the Cust_table Table.

data integration, it is estimated that 50 to 80 percent of a data scientist's time is spent on collecting and organizing data for analysis.[1] If we consider that in any complex organization, data governance is also essential for tasks other than data analytics, we can conclude that the challenge of identifying, gathering, retaining, and providing access to all relevant data for the business at an acceptable cost is huge (Bernstein and Haas 2008).

The aforementioned considerations are valid even for very simple information systems, as the following example scenario illustrates. Figure 1 shows a portion of Cust table, a relational table contained in a real information system. The table maintains information about the customers of an organization, where each row stores data about a single customer. The first column contains the customer code, with the proviso that if the code is positive, then the record refers to an ordinary customer, and if it is negative, to a special customer. If the code is nonnumeric, then the customer type is unknown. Columns 2 and 3 specify the time interval of validity for the record. ID_GROUP indicates the group the customer belongs to (if the value of FLAG_CP is "S," then the customer is the leader of the group; if FLAG_CF is "S," then the customer is the controller of the group). FATTURATO is the annual turnover (but the value is valid only if FLAG_FATT is "S"). Obviously, each notion mentioned previously (like "special," "ordinary," "group," "leader," etc.) has a specific meaning in the organization, and understanding such meaning is crucial if one wants to correctly access or manage the data in the table and extract useful information out of it. Similar rules hold for the other 47 columns that, for lack of space, are not shown in the figure.

Those who have experience with complex databases, or databases that are part of large information systems, will not be surprised to see such complexity in a single data structure. Now, think of a database with many tables of this kind, and try to imagine a poor client accessing such tables for data analysis. The problem is even more severe if one considers that information systems in the real world use different (often many) heterogeneous data sources, both internal and external to the organization. While many are the issues raised by this problem, I would like to go into more detail on some of them.

Accessing and Querying Data

As observed by De Giacomo et al. (2018), although the initial design of a collection of data sources might be adequate, corrective maintenance actions tend to reshape these sources into a form that diverges from the original structure. Also, they are often subject to changes so as to adapt to specific, application-dependent needs. Analogously, applications are frequently modified to accommodate new requirements, and guaranteeing their seamless usage within the organization is costly. The result is that the data stored in different sources and the processes operating over them tend to be redundant, mutually inconsistent, and obscure for large classes of users. So, query formulation often requires interacting with IT experts who know where the data are and what they mean in the various contexts, and can therefore translate the information need expressed by the user into appropriate queries. It is not uncommon to see organizations where this process requires domain experts to send a request to the data management staff and wait for several days, or even weeks, before they receive a (possibly inappropriate) query in response. In summary, it is often exceedingly difficult for end users to single out exactly the data that are relevant for them, even though they are perfectly able to describe their requirement in terms of business concepts.

Data Quality

It is often claimed that data quality is one of the most important factors in delivering high-value information services (Fan and Geerts 2012). However, the aforementioned scenario poses several obstacles to the modest goal of checking data quality, let alone achieving a good level of quality in information delivery. How can we possibly specify data quality requirements, if we do not have a clear understanding of the semantics that the data should bring? The problem is sharpened by the need for connecting to external data, originating, for example, from business partners, suppliers, clients, or even public sources. Again, judging the quality of external data, and deciding whether to reconcile possible inconsistencies or simply to add such data as different views, cannot be done without a deep understanding of the meaning of such data.

Open Data

Note that understanding and documenting the semantics of data is also crucial for opening data to external organizations. The demand for greater openness is irresistible nowadays. In many aspects of our society, there is growing awareness of and consensus on the need for data-driven approaches that are resilient, transparent, and fully accountable. But to achieve a data-driven society, it is necessary that the data needed for public goods be readily available (Wessels et al. 2017). Thus, it is not surprising that in recent years both public and private organizations have been faced with the issue of publishing open data, in particular with the goal of providing data consumers with suitable information to capture the semantics of the data they publish. But, again, associating a reasonably well-structured description of open data sets is very difficult if we do not have effective tools for documenting the meaning and the usage of the data sources from which such data have been extracted.

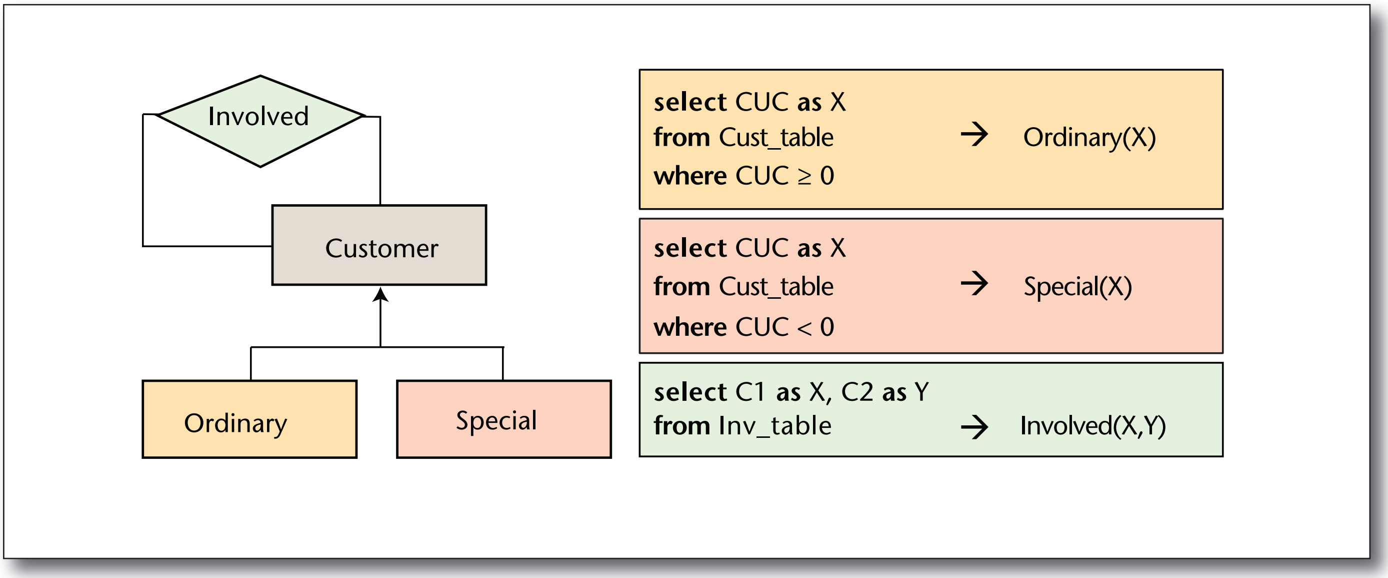
Process and Service Specification

Information systems are crucial artifacts for running organizations; and designing, documenting, managing, and executing processes is an important aspect of information systems. However, specifying what a process or service does, or which characteristics it is supposed to have, cannot be done correctly and comprehensively without a clear specification of which data the process will access and how it will possibly modify or update such data. The difficulties of doing that in a satisfactory way come from various factors, including the lack of modeling languages and tools for describing process and data holistically. However, the problems related to the semantics of data that we discussed previously undoubtedly make the task even harder (Berardi et al. 2003; Bagheri Hariri et al. 2013).

The Notion of an Ontology-Based Data Management System

All the previous observations show that a unified access to data, a comprehensive methodology for data preparation, and an effective governance of data-oriented processes and services are extremely difficult goals to achieve in modern information systems (Bernstein and Haas 2008). We argue that the ontology-based data management (OBDM[2]) paradigm (Lenzerini 2011) is a promising direction for addressing these challenges. The key idea of OBDM is to apply suitable techniques from the area of knowledge representation and reasoning in artificial intelligence for a new way to achieve data governance and integration, based on the principle of managing heterogeneous data through the lens of an ontology. Indeed, OBDM resorts to a three-level architecture comprising the ontology, the data sources, and the mapping between the two. First, the data layer is constituted by the existing data sources that are relevant for the organization; second, the ontology is a declarative and explicit representation of the domain of interest for the organization, specified by means of a formal and high-level description of both its static and its dynamic aspects; and third, the mapping is a set of declarative assertions specifying how the available sources in the data layer and the computational resources used in the organization relate to the ontology.

OBDM can thus be seen as a sophisticated form of information integration (Lenzerini 2002; Calvanese and De Giacomo 2005; Doan, Halevy, and Ives 2012), where the usual global schema is replaced by the conceptual model of the application domain, formulated as an ontology. With this approach, the integrated view that the system provides to information consumers is not merely a data structure accommodating the various data at the sources, but a semantically rich description of the relevant concepts in the domain of interest, as well as of the relationships between such concepts. The distinguishing feature of the whole approach is that users of the system are freed from the details of how to use the data sources, as they will express their needs (for example, a query) in the terms of the concepts, the relations, and the processes described in the domain model. The system



```
select CUC as X
from Cust_table          →   Ordinary(X)
where CUC ≥ 0

select CUC as X
from Cust_table          →   Special(X)
where CUC < 0

select C1 as X, C2 as Y
from Inv_table           →   Involved(X,Y)
```

Figure 2. Example of OBDM Specification: Ontology and Mapping.

will reason about the ontology and the mappings, and reformulate the needs in terms of appropriate calls to services provided for accessing the data sources. For the services expressed over the ontology to be translated into correct and efficient computations over the data sources, techniques typical of knowledge representation and automated reasoning are crucial. Note, however, that OBDM introduces new challenges to these areas. Indeed, while knowledge representation techniques are often confined to scenarios where the complexity resides in the rules governing the application, in OBDM one faces the problem of a huge amount of data in the data layer, which poses completely new requirements for the reasoning tasks that the system should be able to carry out. For example, the notion of data complexity, by which one measures the computational complexity on the basis of the size of the data layer only, is of paramount importance in OBDM.

From a more formal perspective, an OBDM specification $\mathcal{J}$ is defined as a triple $\langle \mathcal{O}\mathcal{S}\mathcal{M} \rangle$, where $\mathcal{O}$ is an ontology, $\mathcal{S}$ is a relational schema, called *source schema*, and $\mathcal{M}$ is a mapping from $\mathcal{S}$ to $\mathcal{O}$. In particular, $\mathcal{O}$ represents intensional knowledge about the domain, expressed in some logical language,[3] and $\mathcal{M}$ is a set of mapping assertions, again expressed in a logical language, each one relating a query over the source schema to a query over the ontology.

An OBDM system is a pair $(\mathcal{J}, \mathcal{D})$, where $\mathcal{J}$ is an OBDM specification and $\mathcal{D}$ is a database for the source schema $\mathcal{S}$, called *source database*, for $\mathcal{J}$. The semantics of $(\mathcal{J}, \mathcal{D})$ are given in terms of the logical interpretations that are models of $\mathcal{O}$, that is, that satisfy all axioms of $\mathcal{O}$ and all assertions in $\mathcal{M}$ with respect to $\mathcal{D}$. The notion of mapping satisfaction depends on the semantic interpretation adopted for mapping assertions. Commonly, such assertions are assumed to be sound, which intuitively means that the patterns specified over the sources imply a set of facts at the ontology level; in other words, data at the sources give rise to instance assertions in the ontology. Because of the logical nature of the domain description represented by the ontology, and the kind of mapping assertions considered, $(\mathcal{J}, \mathcal{D})$ is characterized by a set of models, denoted with $Mod_D(\mathcal{J})$.

We end this section by illustrating a simple example of an OBDM specification, referring in particular to the application scenario mentioned earlier. We will use the example in the next sections.

Source schema $\mathcal{S}$: We assume that, beside the table Cust_table illustrated in figure 1, we have another relational table available, called Inv_table, that stores pairs ⟨C1,C2⟩ such that the customer with code C1 involved customer with code C2 in a joint project. So, $\mathcal{S}$ is constitued by the relational schema {Cust_table, Inv_table}.

Ontology $\mathcal{O}$: A fragment of the domain ontology expressed in graphical form is shown in figure 2. The ontology sanctions that there are exactly two types of customers, namely ordinary and special, so that every customer is of one of these types. Also, the ontology defines Involved as a relationship between customers.

Mapping $\mathcal{M}$: The mapping, also shown in figure 2, asserts that the Cust_table table is mapped to the concepts Ordinary and Special, depending on the value of the field CUC, while data in the Inv_table are mapped to the relation Involved.

We obtain an OBDM system by pairing the previous specification with a specific $\mathcal{S}$-database, that is, a database coherent with the schema $\mathcal{S}$ that assigns an extension (set of tuples) to the tables Cust_table and Inv_table.

Query Answering

In OBDM systems, the main service of interest is query answering, that is, computing the answers to user queries, which are queries posed over the ontology. This process consists of returning what are known as *certain answers*, that is, the tuples that satisfy the user query in all the models in Mod D ($\mathcal{J}$). Notice the difference with query answering in traditional databases. While a database can be seen as a single model of a logical theory (see, for example, Reiter 1984), query answering in OBDM faces the problem of considering various models of the whole system, and is therefore a form of reasoning under incomplete information. It follows that query evaluation in OBDM is much more challenging than classical query evaluation over a database instance, and this complexity explains why automated deduction techniques are very relevant in this context.

To better illustrate the point, we reconsider the example of the previous section, and we assume that in a specific $\mathcal{S}$-database, -452901 and 124589 are two values appearing the CUC field of Cust_table, and ⟨124589,CCAAA⟩, ⟨CCAAA,-452901⟩ are two tuples appearing in the Inv_table. Note that, by the mapping assertions, the two tuples satisfy the predicate Involved in the ontology. Now, consider a query to check whether there exists an ordinary customer who involved a special customer in a project, expressed in logic as

$\exists X \exists Y$ Ordinary(X), Involved(X, Y), Special(Y)

If we evaluate the query simply by searching for the corresponding pattern in the data, we come up with the answer "false," because we cannot find any pair of elements to bind to the variables X, Y in such a way that the pattern specified by the query is satisfied in the data. However, if we consider the knowledge expressed by the ontology, then we know that, in every model of the ontology, the customer with code CCAAA is either ordinary or special. For the models where CCAAA is ordinary, the binding $X \rightarrow$ CCAAA, $Y \rightarrow$ –452901 makes the query true, whereas for the models where CCAAA is special, it is the binding $X \rightarrow$ 124589, $Y \rightarrow$ CCAAA that makes the query true. It follows that the certain answer to the query is "true."

What the previous example shows is that query answering in OBDM may require reasoning by cases on data (in the example, on the status of the customer CCAAA) and that this reasoning is necessitated, in particular, by the presence of certain representation patterns in the ontology (in the example, the pattern is "every customer is either special or ordinary"). It is not difficult to see that this need for reasoning by cases implies high computational complexity in the size of the data, and, unfortunately, the high cost does not seem to show up only in artificially constructed worst cases (see, for example, Schaerf 1993). The conclusion is that OBDM is yet another scenario where the trade-off between the expressive power of the modeling language and the complexity of reasoning is extremely relevant (Levesque and Brachman 1985).

Indeed, from the computational perspective, query answering depends on (1) the language used for the ontology, (2) the language used to specify the queries in the mapping, and (3) the language used for user queries. As for the first aspect, many years of research on description logics (Baader et al. 2003) has led to specific proposals of ontology languages suitable for OBDM. I want to briefly present one of the most successful, that is, the one based on a family of DLs, called *DL-Lite*[4], first introduced in Calvanese et al. (2004; 2005), which has also given rise to the OWL 2 QL profile[5] of the web ontology language OWL, standardized by the W3C. More specifically, I refer to *DL-Lite$_A$*, which is able to capture essentially all features of entity-relationship diagrams and UML class diagrams.[6]

As usual in DLs, DL-Lite$_A$ allows for representing the domain of interest in terms of concepts, denoting sets of objects and roles (or, relations), and denoting binary relations between objects. In DL-Lite$_A$, a concept is either an atomic concept C (that is, a unary predicate) or the projection $\exists R$ or $\exists R^-$ of a role R on its first or second component, respectively. A role can be either an atomic role R or an inverse role R^-, allowing for a complete symmetry between the two directions. DL-Lite$_A$ also includes value attributes relating objects in classes to domain values (such as strings or integers). The ontology is modeled by means of axioms that can express inclusion and disjointness between concepts or roles and the (global) functionality of roles (with some restrictions on the interaction between functionality and role inclusions to ensure tractability). In table 1, we illustrate the conceptual modeling constructs captured by DL-Lite$_A$ assertions and provide also their meaning expressed in first-order (FO) logic, where all variables are implicitly universally quantified. Type 1 corresponds to ISA/disjointness on concepts, type 2 to domain/range specification for a role, type 3 to mandatory participation in a role, type 4 to ISA/disjointness on roles, and type 5 to functionality assertion on a role. The DLs of the DL-Lite family, including DL-Lite$_A$, combined with specific languages for mapping the specification previously mentioned, have been designed so as to enjoy the first-order rewritability (FO-rewritability) property: given a UCQ q and an OBDM specification $\mathcal{J} = \langle \mathcal{OSM} \rangle$, it is possible to compile q, $\mathcal{O}$, and $\mathcal{M}$ into a new FO query q' formulated over $\mathcal{S}$. Such query q' has the property that, when evaluated over a database D for $\mathcal{S}$, it returns exactly the certain answers for q over the OBDM system $\langle \mathcal{J}, D \rangle$, for every data source D. Each such q' is called an (FO-)perfect rewriting of q with regard to $\mathcal{J}$. Most of the proposed techniques (Calvanese et al. 2007; Pérez-Urbina, Horrocks, and Motik 2009; Chortaras, Trivela, and Stamou 2011) to

Type	DL Syntax	FOL Semantics
1	$C_1 \sqsubseteq [\neg]C_2$	$\forall x.C_1(x) \rightarrow [\neg]C_2(x)$
2	$\exists R^{[-]} \sqsubseteq C$	$R^{[-]}(x, y) \rightarrow C(x)$
3	$C \sqsubseteq \exists R^{[-]}$	$C(x) \rightarrow \exists y.R^{[-]}(x, y)$
4	$R_1^{[-]} \sqsubseteq [\neg]R_2^{[-]}$	$R_1^{[-]}(x, y) \rightarrow [\neg]R_2^{[-]}(x, y)$
5	(funct $R^{[-]}$)	$R^{[-]}(x, y) \wedge R^{[-]}(x, z) \rightarrow y = z$

Table 1. DL-Lite$_A$ Assertions.

Symbols in square brackets may or may not be present, and $R^-(x, y)$ stands for $R(y, x)$.

achieve FO-rewritability start from a CQ or a UCQ (that is, a set of CQs) and end up producing a UCQ that is an expansion of the initial query. These techniques are based on variants of clausal resolution (Leitsch 1997): every rewriting step essentially corresponds to the application of clausal resolution between a CQ among the ones already generated and a concept or role inclusion axiom of the ontology. The rewriting process terminates when a fix-point is reached, that is, when no new CQ can be generated.

The results published by Calvanese et al. (2007) and Poggi et al. (2008) show that, following the technique illustrated earlier, conjunctive query answering is indeed first-order rewritable in DL-Lite, implying that answering (unions of) conjunctive queries can be reduced to query evaluation over a relational database, for which we can rely on standard relational DBMSs. This property also implies that CQ answering is in AC^0 (a subclass of LOGSPACE) in data complexity. Indeed, this implication is an immediate consequence of the fact that the complexity of the aforementioned phase of query rewriting is independent of the data source and that the final rewritten query is an SQL expression. An important question is whether we can further extend the ontology specification language of OBDM without losing the nice computational property of the query rewriting phase. Calvanese et al. (2013) show that adding any of the main concept constructors considered in description logics and missing in DL-Lite$_A$ (for example, negation, disjunction, qualified existential restriction, range restriction) causes a jump of the data complexity of conjunctive query answering in OBDM, which goes beyond the class AC^0. This issue has been further investigated by Artale et al. (2009). As for the query language, we note that going beyond unions of CQs is problematic from the point of view of tractability, or even decidability. For instance, adding negation to CQs causes query answering to become undecidable (Gutiérrez-Basulto et al. 2015).

This basic technique, introduced by Calvanese et al. (2007), has been the subject of many investigations in the last decade, with the goal of improving its performance (Pérez-Urbina, Horrocks, and Motik 2009; Chortaras, Trivela, and Stamou 2011; Kontchakov et al. 2011; Di Pinto et al. 2013; Gottlob et al. 2014a) and extending its applicability (Lenzerini, Lepore, and Poggi 2016). More generally, the issue of designing automated reasoning algorithms for query answering in OBDM has been addressed by many scientific works and projects. New ideas of how to answer queries for different ontology languages have been proposed (see, for example, Rosati and Almatelli 2010; Chortaras, Trivela, and Stamou 2011; Gottlob et al. 2014b; Lutz and Sabellek 2017) and various extensions to the basic ontology languages have been explored, such as extensions based on Datalog (see Calì et al. 2010) or on existential rules (see Gottlob, Manna, and Pieris 2015; Grau et al. 2013; König et al. 2015).

Finally, there has been interesting and promising work on extending query rewriting to more expressive, not necessarily first-order rewritable, ontology languages (Pérez-Urbina, Horrocks, and Motik 2009; Chortaras, Trivela, and Stamou 2011; Eiter et al. 2012; Calì, Gottlob, and Lukasiewicz 2012; Kaminski, Nenov, and Grau 2016; Bienvenu et al. 2014).

Other Services

While computing certain answers of queries under the classical semantics has been the main subject of the research investigation on OBDM, there are several other services that an OBDM system should provide. A brief overview of two services, and an exploration of one issue, follows.

Data Quality Assessment

Besides ontology-mediated querying and other data management tasks, recent works argue that OBDM is a promising tool for assessing the quality of data, especially in the presence of multiple, independent data sources (Console and Lenzerini 2014; Catarci et al. 2017). Some of the reasons are as follows: (1) basing data quality assessments on a formal conceptualization of the domain of interest allows us to easily blur out all the meaningless details of the single data source and focus on real data quality issues; (2) different data sources can be analyzed using the same yardstick, that is, the ontology, and hence accessed and compared in terms of their quality; and (3) the use of conceptualizations shared among the different assets of an organization allows for data quality assessments that are easy to present and use in many different contexts.

Quality assessment is carried out through different dimensions, such as consistency, accuracy, completeness, and confidentiality. We briefly discuss consistency, which is the quality dimension dealing with the coherence of data. Counterexamples to consistency show that data suffers from integrity problems, thus providing crucial information about the assets owning such data. In the literature, it is often advocated that consistency be assessed by checking whether data follow specific rules for integrity. However, in traditional approaches such rules are either implicit or specified depending on the single data source under analysis. On the contrary, OBDM promotes a new method, where the rules to be checked are derived directly from the ontology and where they have also been validated by the process of building the conceptual model of the domain. In addition, instead of implementing laborious quality-checking tasks for the various sources, the inference capabilities inherent in ODBM systems provide automated techniques for accessing consistency, singling out the various inconsistencies present in the data, even ranking them according to various predetermined criteria. For example, in the application scenario discussed in the introduction, we are not forced to implement a specific rule for checking whether a customer exists that is classified by the data sources as both an ordinary and a special customer. Indeed, we can rely on the automatic verification of the rule by means of the OBDM system as part of the consistency check of the whole OBDM system. We point out that the extensive research carried out in the last years has produced optimized algorithms for consistency checking, which scale nicely when applied to big data sources. Similar considerations hold for other data quality dimensions.

Inconsistency Tolerance

What are we supposed to do once we have found possible consistency problems in the data sources? It is commonly accepted that inconsistency causes severe problems in logic-based knowledge representation systems. Because an inconsistent logical theory has no classical model, it logically implies every formula, and therefore query answering over an inconsistent knowledge base becomes meaningless under classical logic semantics. Unfortunately, in real-world OBDM systems, inconsistencies are likely to occur between the domain knowledge represented by the ontology and that represented by the data at the sources, because data sources are generally maintained by single applications and so are kept coherent neither with other data sources nor with the axioms of the underlying ontology. Many research papers in the last years deal with this problem (Lembo et al. 2010; Rosati 2011; Lembo et al. 2011). In many of these approaches, the fundamental tool for obtaining consistent information from an inconsistent OBDM system is the notion of repair (Arenas, Bertossi, and Chomicki 1999). A repair of a data set contradicting a set of axioms is a database obtained by applying a minimal set of changes that restore consistency. There are several interpretations of the notion of minimality, and different interpretations give rise to different inconsistency-tolerant semantics. Under most interpretations of minimality, there are many possible repairs for the system, and the approach sanctions that what is consistently true is simply what is true in all possible repairs. Thus, inconsistency-tolerant query answering amounts to computing the tuples that are answers to the query in all possible repairs. Interesting papers investigating these notions in the context of OBDM have been written by Lembo et al. (2015) and Bienvenu, Bourgaux, and Goasdoué (2016).

Open Data Publishing

Current practices for publishing open data focus essentially on providing extensional information (often in very simple forms, such as CSV files), and they carry out the task of documenting data mostly by using metadata expressed in natural languages, or in terms of record structures. As a consequence, the semantics of data sets are not formally expressed in a machine-readable form. As we said before, OBDM opens up the possibility of a new way of publishing data, with the idea of annotating data items with the ontology elements that describe them in terms of the concepts in the domain of the organization. When an OBDM specification is available in an organization, an obvious way to proceed to open data publication is as follows: (1) express the data set to be published in terms of a SPARQL query over the ontology, (2) compute the certain answers to the query, and (3) publish the result of the certain answer computation, using the query expression and the ontology as a basis for annotating the data set with suitable metadata expressing its semantics. Using this method, the ontology is the heart of the task: it is used for expressing the content of the data set to be published (in terms of a query), and it is used, together with the query, for annotating the published data. First results on using OBDM for open data were reported in the paper by Cima (2017).

Conclusions

The OBDM paradigm is relatively new, but it is attracting a strong interest from several communities. Specific tools have been designed and delivered for query answering in OBDM[7] (Calvanese et al. 2011; 2017), and several projects have been carried out with the goal of adopting this paradigm in real-world applications (see, for example, Kharlamov et al. 2015; Antonioli et al. 2013; Daraio et al. 2016). From a research perspective, many groups worldwide have been working on research problems related to OBDM, producing an amazing number of scientific results.[8] Interesting open problems remain, and it is reasonable to foresee that new results will contribute to building novel tools or improving the current ones.

Interestingly, OBMD has helped to renew the interaction between the areas of data management and artificial intelligence. While in the last years such interaction was confined to methods and techniques for data mining and knowledge discovery, OBDM is pushing the community of knowledge representation and reasoning towards research topics that are closed to big data and data science. I think that this represents a great opportunity for our community, especially in the light of the importance that the notion of data-driven society is gaining.

Acknowledgements

I would like to warmly thank Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Antonella Poggi, and Riccardo Rosati, my invaluable collaborators on the topics discussed in this article in the last decade.

Notes

1. The 2017 Data Scientist Report, CrowdFlower.

2. The acronym is similar to *OBDA*, which stands for *ontology-based data access*. We use *OBDM* because we consider data access to be just one aspect, although important, of the more general notion of data management.

3. We consider languages that are fragments of OWL 2 (www.w3.org/TR/owl2-syntax), the ontology web language originated from description logics (Baader et al. 2007).

4. Not to be confused with the DLs studied by Artale et al. (2009), which form the DL-Lite$_{bool}$ family.

5. www.w3.org/TR/owl2-profiles.

6. Except for completeness of hierarchies, which is instead present in the ontology of the example.

7. See www.stardog.com.

8. See the series of description logics workshops at dl.kr.org/workshops.

References

Antonioli, N.; Castanò, F.; Civili, C.; Coletta, S.; Grossi, S.; Lembo, D.; Lenzerini, M.; Poggi, A.; Savo, D. F.; and Virardi, E. 2013. Ontology-Based Data Access: The Experience at the Italian Department of Treasury. In *Advanced Information Systems Engineering — 25th International Conference,* CAiSE 2013 Proceedings, 9–16. Lecture Notes in Computer Science 7908. Berlin: Springer

Arenas, M.; Bertossi, L. E.; and Chomicki, J. 1999. Consistent Query Answers in Inconsistent Databases. In *Proceedings of the 18th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, 68–79. New York: Association for Computing Machinery. doi.org/10.1145/303976.303983

Artale, A.; Calvanese, D.; Kontchakov, R.; and Zakharyaschev, M. 2009. The DL-Lite family and Relations. *Journal of Artificial Intelligence Research* 36: 1–69.

Baader, F.; Calvanese, D.; McGuinness, D. L.; Nardi, D.; and Patel-Schneider, P. F., eds. 2003. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge, UK: Cambridge University Press.

Baader, F.; Calvanese, D.; McGuinness, D.; Nardi, D.; and Patel-Schneider, P. F., eds. 2007. *The Description Logic Handbook: Theory, Implementations and Applications*, 2nd ed. Cambridge, UK: Cambridge University Press. doi.org/10.1017/CBO9780511711787

Bagheri Hariri, B.; Calvanese, D.; De Giacomo, G.; Deutsch, A.; and Montali, M. 2013. Verification of Relational Data-Centric Dynamic Systems with External Services. In *Proceedings of the 32nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, 163–174. New York: Association for Computing Machinery. doi.org/10.1145/2463664.246522

Berardi, D.; Calvanese, D.; De Giacomo, G.; Lenzerini, M.; and Mecella, M. 2003. A Foundational Vision of E-Services. In *Web Services, E-Business, and the Semantic Web: Second International Workshop, Revised Selected Papers*, 28–40. Lecture Notes in Computer Science 3095. Berlin: Springer.

Bernstein, P. A., and Haas, L. 2008. Information Integration in the Enterprise. *Communications of the ACM* 51(9): 72–79. doi.org/10.1145/1378727.1378745

Bienvenu, M.; Bourgaux, C.; and Goasdoué, F. 2016. Query-Driven Repairing of Inconsistent DL-Lite Knowledge Bases. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, 957–964. Palo Alto: AAAI Press.

Bienvenu, M.; ten Cate, B.; Lutz, C.; and Wolter, F. 2014. Ontology-Based Data Access: A Study Through Disjunctive Datalog, CSP, and MMSNP. *ACM Transactions on Database Systems* 39(4): 213–224. doi.org/10.1145/2661643

Calì, A.; Gottlob, G.; and Lukasiewicz, T. 2012. A General Datalog-Based Framework for Tractable Query Answering Over Ontologies. *Journal of Web Semantics* 14: 57–83. doi.org/10.1016/j.websem.2012.03.001

Calì, A.; Gottlob, G.; Lukasiewicz, T.; Marnette, B.; and Pieris, A. 2010. Datalog+/-: A Family of Logical Knowledge Representation and Query Languages for New Applications. In *Proceedings of the 25th Annual IEEE Symposium on Logic in Computer Science*, 228–242. Los Alamitos, CA: IEEE Computer Society. doi.org/10.1109/LICS.2010.27

Calvanese, D.; Cogrel, B.; Komla-Ebri, S.; Kontchakov, R.; Lanti, D.; Rezk, M.; Rodriguez-Muro, M.; and Xiao, G. 2017. Ontop: Answering SPARQL Queries over Relational Databases. *Semantic Web* 8(3): 471–487. doi.org/10.3233/SW-160217

Calvanese, D., and De Giacomo, G. 2005. Data integration: A Logic-Based Perspective. *AI Magazine* 26(1): 59–70. doi.org/10.1609/aimag.v26i1.1799

Calvanese, D.; De Giacomo, G.; Lembo, D.; Lenzerini, M.; Poggi, A.; Rodriguez-Muro, M.; Rosati, R.; Ruzzi, M.; and Savo, D. F. 2011. The Mastro System for Ontology-Based Data Access. *Semantic Web* 2(1): 43–53.

Calvanese, D.; De Giacomo, G.; Lembo, D.; Lenzerini, M.; and Rosati, R. 2005. DL-Lite: Tractable Description Logics for Ontologies. In *Proceedings, The 20th National Conference on Artificial Intelligence and the 17th Innovative Applications of Artificial Intelligence Conference*, 602–607. Menlo Park, CA: AAAI Press.

Calvanese, D.; De Giacomo, G.; Lembo, D.; Lenzerini, M.; and Rosati, R. 2007. Tractable Reasoning and Efficient Query Answering in Description Logics: The DL-Lite family. *Journal of Automated Reasoning* 39(3): 385–429. doi.org/10.1007/s10817-007-9078-x

Calvanese, D.; De Giacomo, G.; Lembo, D.; Lenzerini, M.; and Rosati, R. 2013. Data Complexity of Query Answering in Description Logics. *Artificial Intelligence* 195: 335–360. doi.org/10.1016/j.artint.2012.10.003

Calvanese, D.; De Giacomo, G.; Lenzerini, M.; Rosati, R.; and Vetere, G. 2004. DL-Lite: Practical Reasoning for Rich DLs. In *Proceedings of the 2004 international workshop on*

Description Logics. CEUR Workshop Proceedings 104. Aachen, Germany: RWTH Aachen University.

Catarci, T.; Scannapieco, M.; Console, M.; and Demetrescu, C. 2017. My (Fair) Big Data. In *2017 IEEE International Conference on Big Data*, 2974–2979. Piscataway, NJ: Institute for Electrical and Electronics Engineers.

Chortaras, A.; Trivela, D.; and Stamou, G. B. 2011. Optimized Query Rewriting for OWL 2 QL. In *Automated Deduction — CADE-23 — 23rd International Conference on Automated Deduction*, 192–206. Lecture Notes in Computer Science 6803. Berlin: Springer.

Cima, G. 2017. Preliminary Results on Ontology-Based Open Data Publishing. Paper presented at the 30th International Workshop on Description Logics. Montpellier, France, July 18-21.

Console, M., and Lenzerini, M. 2014. Data Quality in Ontology-Based Data Access: The Case of Consistency. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, 1020–1026. Palo Alto, CA: AAAI Press.

Daraio, C.; Lenzerini, M.; Leporelli, C.; Moed, H. F.; Naggar, P.; Bonaccorsi, A.; and Bartolucci, A. 2016. Data Integration for Research and Innovation Policy: An Ontology-Based Data Management Approach. *Scientometrics* 106(2): 857–871. doi.org/10.1007/s11192-015-1814-0

De Giacomo, G.; Lembo, D.; Lenzerini, M.; Poggi, A.; and Rosati, R. 2018. Using Ontologies for Semantic Data Integration. In *A Comprehensive Guide Through the Italian Database Research Over the Last 25 Years*, 187–202. Studies in Big Data 31. Berlin: Springer.

Di Pinto, F.; Lembo, D.; Lenzerini, M.; Mancini, R.; Poggi, A.; Rosati, R.; Ruzzi, M.; and Savo, D. F. 2013. Optimizing Query Rewriting in Ontology-Based Data Access. In *Joint 2013 EDBT/ICDT Conferences, EDBT '13 Proceedings*, 561–572. New York: Association for Computing Machinery. doi.org/10.1145/2452376.2452441

Doan, A.; Halevy, A. Y.; and Ives, Z. G. 2012. *Principles of Data Integration*. San Francisco: Morgan Kaufmann.

Eiter, T.; Ortiz, M.; Simkus, M.; Tran, T.-K.; and Xiao, G. 2012. Query Rewriting for Horn-SHIQ Plus Rules. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, 726–733. Palo Alto, CA: AAAI Press.

Fan, W., and Geerts, F. 2012. *Foundations of Data Quality Management*. Synthesis Lectures on Data Management. San Rafael, CA: Morgan & Claypool.

Gottlob, G.; Kikot, S.; Kontchakov, R.; Podolskii, V. V.; Schwentick, T.; and Zakharyaschev, M. 2014a. The Price of Query Rewriting in Ontology-Based Data Access. *Artificial Intelligence* 213: 42–59. doi.org/10.1016/j.artint.2014.04.004

Gottlob, G.; Manna, M.; and Pieris, A. 2015. Polynomial Rewritings for Linear Existential Rules. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, 2992–2998. Palo Alto, CA: AAAI Press.

Grau, B. C.; Horrocks, I.; Krötzsch, M.; Kupke, C.; Magka, D.; Motik, B.; and Wang, Z. 2013. Acyclicity Notions for Existential Rules and Their Application to Query Answering in Ontologies. *Journal of Artificial Intelligence Intelligence Research* 47: 741–808.

Gutiérrez-Basulto, V.; Ibáñez-García, Y. A.; Kontchakov, R.; and Kostylev, E. V. 2015. Queries with Negation and Inequalities over Lightweight Ontologies. *Journal of Web Semantics* 35(4): 184–202. doi.org/10.1016/j.websem.2015.06.002

Kaminski, M.; Nenov, Y.; and Grau, B. C. 2016. Datalog Rewritability of Disjunctive Datalog Programs and Non-Horn Ontologies. *Artificial Intelligence* 236: 90–118. doi.org/10.1016/j.artint.2016.03.006

Kharlamov, E.; Hovland, D.; Jiménez-Ruiz, E.; Lanti, D.; Lie, H.; Pinkel, C.; Rezk, M.; Skjæveland, M. G.; Thorstensen, E.; Xiao, G.; Zheleznyakov, D.; and Horrocks, I. 2015. Ontology Based Access to Exploration Data at Statoil. In *The Semantic Web - ISWC 2015*, 93–112. Lecture Notes in Computer Science 9367. doi.org/10.1007/978-3-319-25010-6_6

König, M.; Leclère, M.; Mugnier, M.; and Thomazo, M. 2015. Sound, Complete and Minimal UCQ-Rewriting for Existential Rules. *Semantic Web* 6(5): 451–475. doi.org/10.3233/SW-140153

Kontchakov, R.; Lutz, C.; Toman, D.; Wolter, F.; and Zakharyaschev, M. 2011. The Combined Approach to Ontology-Based Data Access. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence* , 2656–2661. Palo Alto, CA: AAAI Press.

Leitsch, A. 1997. *The Resolution Calculus*. Berlin: Springer.

Lembo, D.; Lenzerini, M.; Rosati, R.; Ruzzi, M.; and Savo, D. F. 2010. Inconsistency-Tolerant Semantics for Description Logics. In *Web Reasoning and Rule Systems — Fourth International Conference*, 103–117. Lecture Notes in Computer Science 6333. Berlin: Springer. doi.org/10.1007/978-3-642-15918-3_9

Lembo, D.; Lenzerini, M.; Rosati, R.; Ruzzi, M.; and Savo, D. F. 2011. Query Rewriting for Inconsistent DL-Lite Ontologies. In *Web Reasoning and Rule Systems - 5th International Conference*, 155-169. Lecture Notes in Computer Science Volume 6902. Berlin: Springer.

Lembo, D.; Lenzerini, M.; Rosati, R.; Ruzzi, M.; and Savo, D. F. 2015. Inconsistency-Tolerant Query Answering in Ontology-Based Data Access. *Journal of Web Semantics* 33: 3–29. doi.org/10.1016/j.websem.2015.04.002

Lenzerini, M. 2002. Data Integration: A Theoretical Perspective. In *Proceedings of the 21st ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems,* 233–246. New York: Association for Computing Machinery. doi.org/10.1145/543613.543644

Lenzerini, M. 2011. Ontology-Based Data Management. In *Proceedings of the 20th ACM Conference on Information and Knowledge Management*, 5–6. New York: Association for Computing Machinery.

Lenzerini, M.; Lepore, L.; and Poggi, A. 2016. Answering Metaqueries over HI (OWL 2 QL) Ontologies. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, 1174–1180. Palo Alto, CA: AAAI Press.

Levesque, H. J., and Brachman, R. J. 1985. A Fundamental Tradeoff in Knowledge Representation and Reasoning. In *Readings in Knowledge Representation*, edited by R. Brachman and H. Levesque, 41–70. Los Altos, CA: Morgan Kaufmann.

Lutz, C., and Sabellek, L. 2017. Ontology-Mediated Querying with the Description Logic EL: Trichotomy and Linear Datalog Rewritability. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 1181–1187. Marina del Rey, CA: IJCAI, Inc. doi.org/10.24963/ijcai.2017/164

Pérez-Urbina, H.; Horrocks, I.; and Motik, B. 2009. Efficient Query Answering for OWL 2. In *The Semantic Web — ISWC 2009*, 489–504. Lecture Notes in Computer Science 5823. Berlin: Springer.

Poggi, A.; Lembo, D.; Calvanese, D.; De Giacomo, G.; Lenzerini, M.; and Rosati, R. 2008. Linking Data to Ontologies.

Journal on Data Semantics X, 133–173. Lecture Notes in Computer Science 4900. Berlin: Springer. doi.org/10.1007/978-3-540-77688-8_5

Reiter, R. 1984. Towards a Logical Reconstruction of Relational Database Theory. In *On Conceptual Modeling: Perspectives from Artificial Intelligence, Databases, and Programming Languages*, edited by M. Brodie, J. Mylopoulos, and J. Schmidt, 191–238. Berlin: Springer. doi.org/10.1007/978-1-4612-5196-5_8

Rosati, R. 2011. On the Complexity of Dealing with Inconsistency in Description Logic Ontologies. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, 1057–1062. Palo Alto, CA: AAAI Press.

Rosati, R., and Almatelli, A. 2010. Improving Query Answering over DL-Lite Ontologies. In *Principles of Knowledge Representation and Reasoning: Proceedings of the 12th International Conference*, 290–300. Palo Alto, CA: AAAI Press.

Schaerf, A. 1993. On the Complexity of the Instance Checking Problem in Concept Languages with Existential Quantification. *Journal of Intelligent Information Systems* 2: 265–278. doi.org/10.1007/BF00962071

Wessels, B.; Finn, R. L.; Sveinsdottir, T.; and Wadhwa, K. 2017. *Open Data and the Knowledge Society.* Amsterdam, The Netherlands: Amsterdam University Press. doi.org/10.5117/9789462980181

Maurizio Lenzerini, AAAI and ACM Fellow, is a professor of computer science and engineering at the University of Rome, La Sapienza. His research interests include knowledge representation and reasoning, database theory, ontology languages, and reasoning about ontologies. His current projects focus on ontology-based data management, whose long-term goal is to exploit knowledge representation and automated reasoning techniques for addressing data access and integration issues in big data scenarios.