Updating data and knowledge bases

Towards introducing updates in Ontology-Based Data Management systems

Antonella Poggi

SAPIENZA
UNIVERSITÀ DI ROMA

23rd Italian Symposium on Advanced Database Systems (2015) – June 14th, 2015

Overview

1. Introduction
   - Motivation
   - Introduction to updates
2. Approaches to updates over KBs
3. Updating DL KBs
4. The view update problem
5. Conclusions

Ontology-Based Data Access

An Ontology-Based Data Access (OBDA) system is a three-layered information system that:

- consists of:
  - an ontology (or conceptual schema, or knowledge base intensional level), which is an abstract description of the domain of interest;
  - a set of data sources containing the actual data; and
  - a set of mappings, i.e., assertions expressing the relationships between the ontology and the actual data;
- provides access to actual data through query answering over the ontology.

$\leadsto$ can be seen as a data integration system, whose "global schema" is independent from the data sources and *talks* about objects of the domain, and about relationships among these objects, where the latter are described in terms of an expressive logic-based language

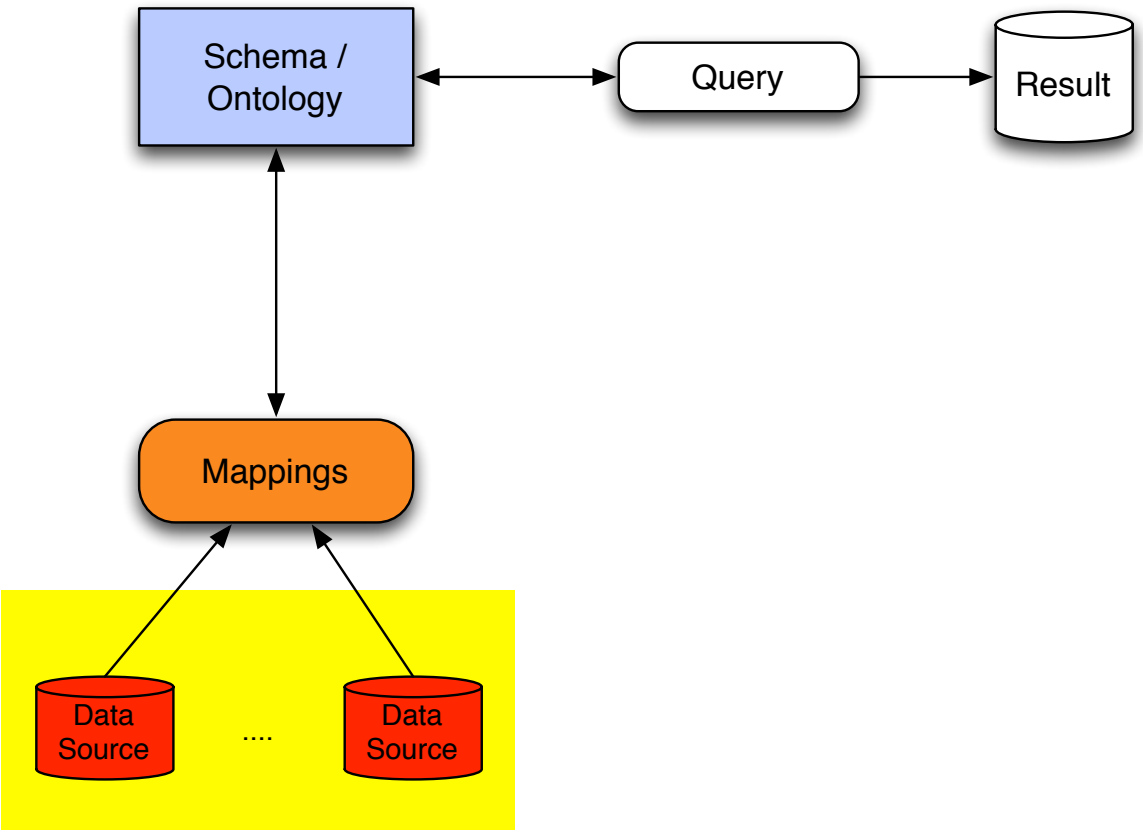
Semantics of OBDA systems

- In contrast to (complete) databases, an OBDA system specification represents a set of possible worlds
  - because of the presence of the ontology, which is a logical theory that is satisfied by several models
  - because of the presence of the mappings, for which there exist several satisfying models

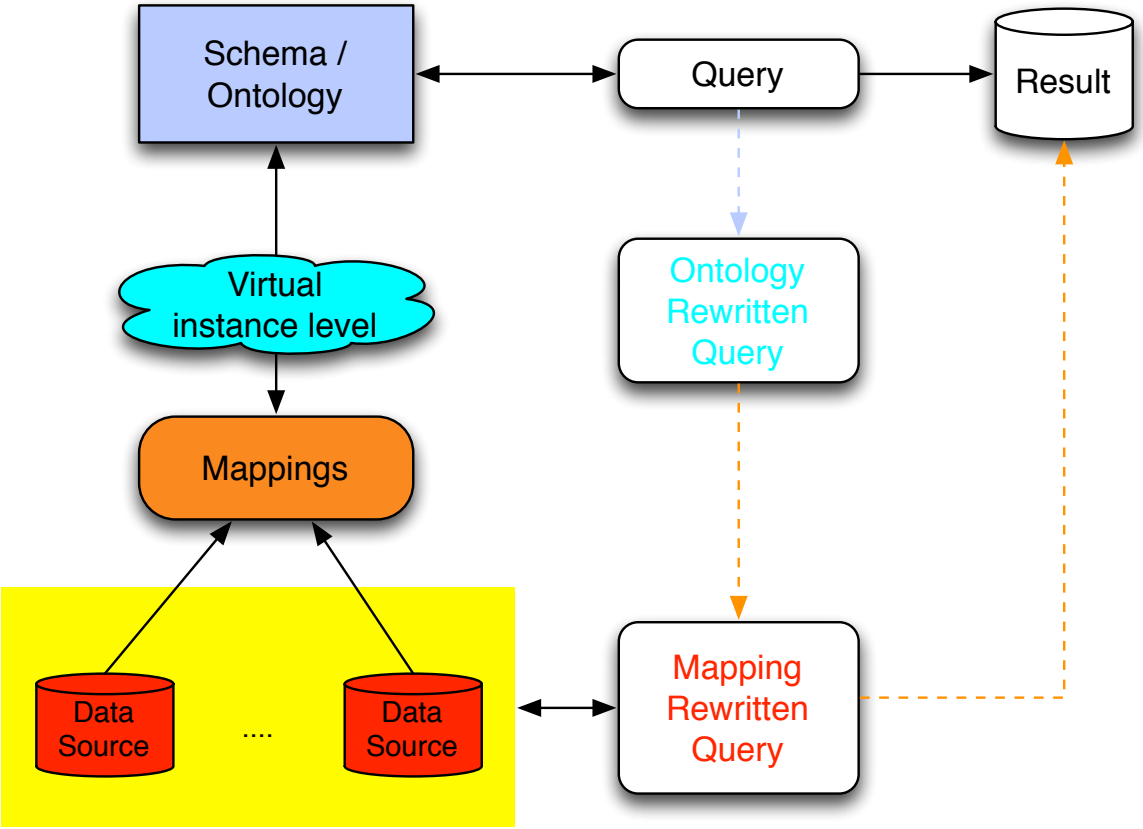
$\leadsto$ Query answering over OBDA systems amounts to logical inference over the theory consisting of the ontology, the mappings, and the data

$\leadsto$ equivalently: query answering amounts to query answering over incomplete information [21]

Query answering over OBDA systems



Query answering over OBDA systems



About research on OBDA query answering

- Research on OBDA has built upon seminal results on:
  - automated reasoning over knowledge bases
  - data integration, query answering using views, incomplete databases
- Research on OBDA has further produced:
  - design of (optimized) ontology query rewriting algorithms [5, 22, 24]
  - design of (optimized) mapping query rewriting algorithms [11]
  - implementation of OBDA systems [23, 4, 7]
  - studies on real-world experiences [1, 25, 2]

Towards Ontology-Based Data Management systems

- OBDA systems provide access to data
  $\leadsto$ what about switching to a "fully-fledged" information system built upon the OBDA three layers?
  $\leadsto$ this leads to Ontology-Based Data Management(OBDM) systems [18], offering services that go beyond query answering:
  - open data provision
  - data governance facilities, e.g., data quality check, data cleaning
  - update
  - ...

OBDM systems of interest (for this tutorial)

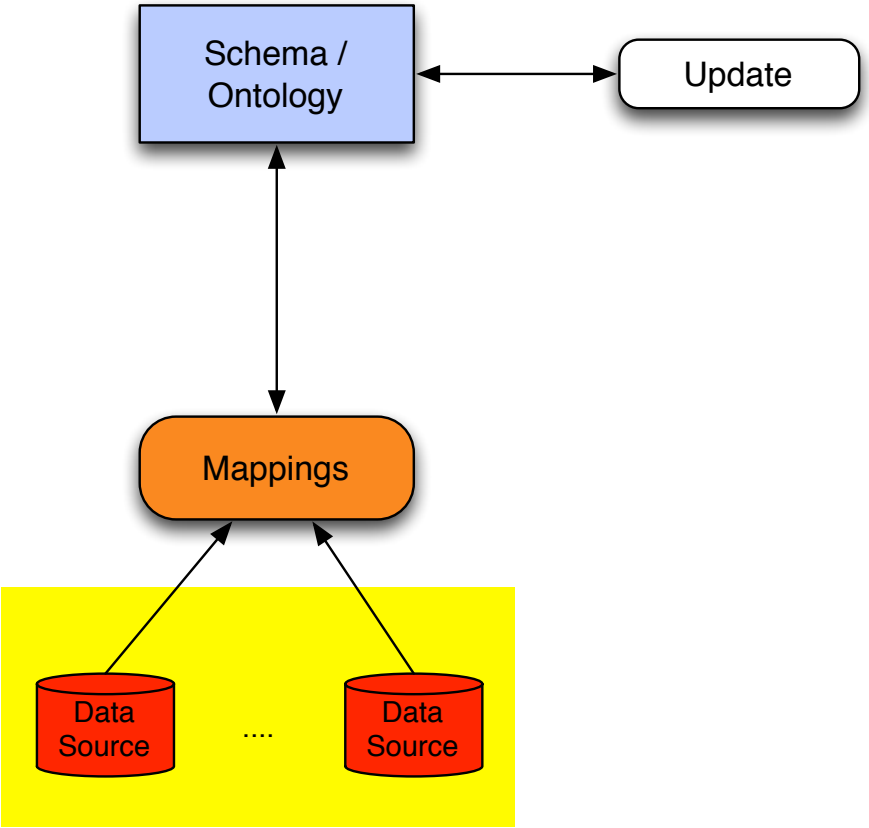
From now on, we will restrict to OBDM systems in which:

- the ontology is expressed as the intensional level of Description Logic (DL) KB
  - A DL is a fragment of first-order logic
  - A DL KB is a pair $\mathcal{O} = \langle \mathcal{T}, \mathcal{A} \rangle$, where $\mathcal{T}$, also called TBox, specifies the intensional level, and $\mathcal{A}$, also called the ABox, specifies the instance level
- the sources are relational
- the mappings follow the global-as-view approach, i.e., they have the form

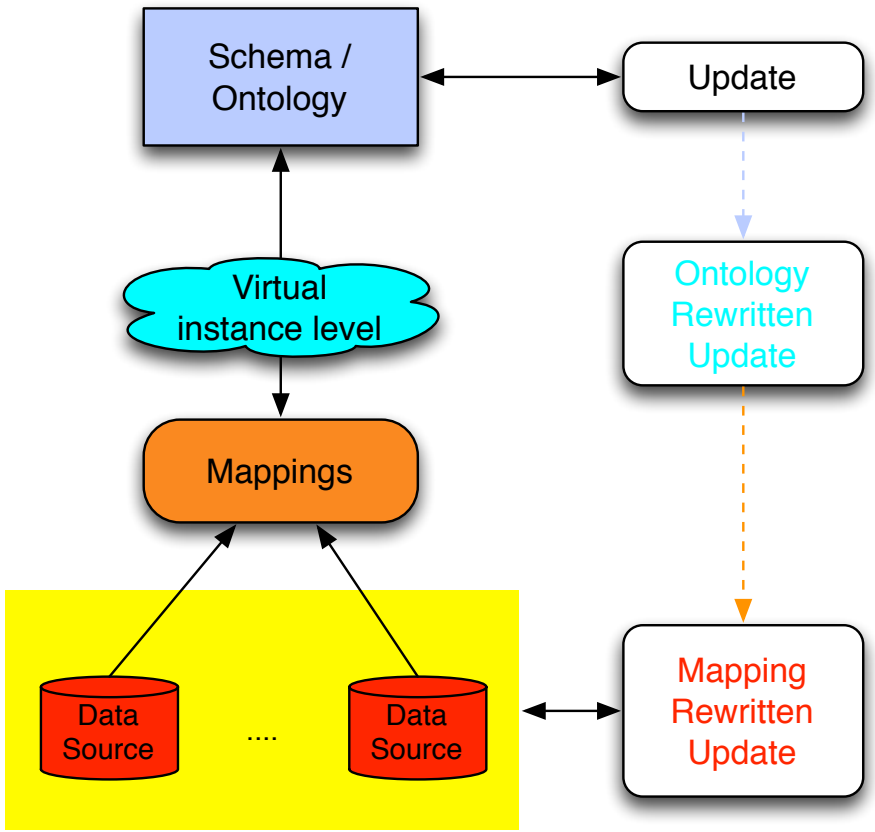
$$E(\vec{x}) \rightarrow V(\vec{x}, \vec{y}),$$

where E is an ontology element and V is a relational view over the sources, whose extension provides a subset of the instances of E

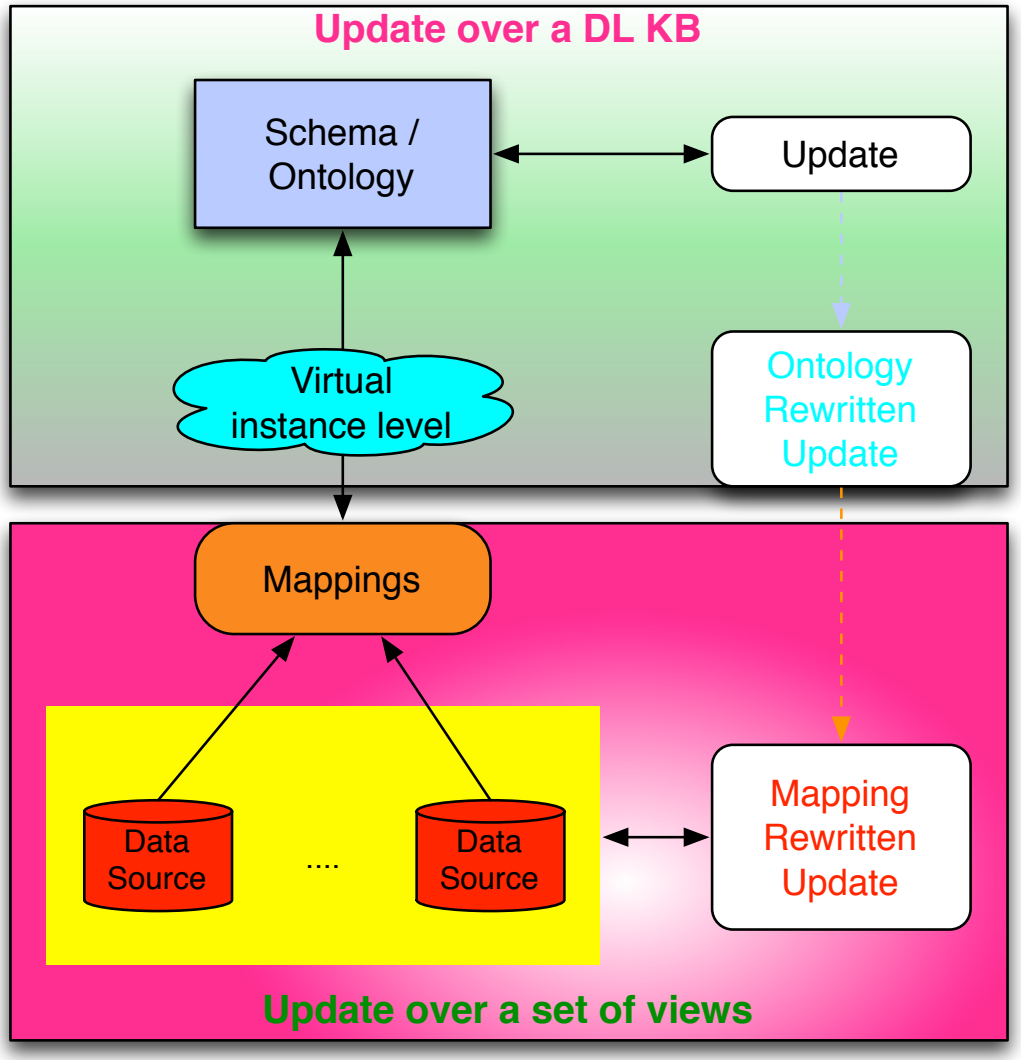
Updates over OBDM systems (first attempt)



Updates over OBDM systems (first attempt)



Updates over OBDM systems (first attempt)



Why is this hard?

1. In the presence of incomplete information, the update semantics is rather unclear [13, 26, 17, 12]
   $\leadsto$ different approaches to update a knowledge base are possible
2. Updating a DL knowledge base amounts to update a theory expressed in a specific language
   $\leadsto$ the result of the update might not even be expressible in the ontology language
3. Updating a set of views is an "old" problem, a.k.a. the view update problem [3, 8, 16] that presents several challenges
   $\leadsto$ propagating new information, expressed over the views, to the underlying actual data sources might not even be possible, e.g., without introducing any other "side-effect"

Overview

1. Introduction
   - Motivation
   - Introduction to updates
2. Approaches to updates over KBs
3. Updating DL KBs
4. The view update problem
5. Conclusions

Basics of updates (1/2)

- In general, an update may concern new knowledge that may or may not be consistent with respect to the original theory, in the sense it may happen that no model exists satisfying both the original theory and the update
- All approaches to updates (or, in general, to *evolution*) state that the original theory should change as little as possible if new information is incorporated
- Intuitively, one wants to keep as much knowledge as possible from a theory that has been updated
  $\leadsto$ if the new knowledge is inconsistent with respect to the theory, then one wants to keep as much knowledge as possible that does not contradict the new knowledge

Basics of updates (2/2)

Desiderata: compute a new consistent theory such that:

- it incorporates the new knowledge
- it *minimally differs* from the initial theory

Note that the semantics of updates, similarly to the semantics of consistent query answering, crucially depends on the notion of minimal distance between theories and/or set of models.

From now on in this tutorial...

In the following, we will assume that updates over OBDM consist of insertions of instance level assertions, e.g., new knowledge about an object being an instance of a concept of the ontology

- we will consider insertions to a DL KB
  - the case of deletion is analogous
- we will focus on both insertions and deletions of tuples over a set of views

Overview

1. Introduction
2. Approaches to updates over KBs
   - Classification criteria
   - Formula-based update operators
   - Model-based update operators
3. Updating DL KBs
4. The view update problem
5. Conclusions

Classification criteria (1/2)

Several approaches to knowledge base updates were proposed in the literature [12].
In the following, we will classify the main approaches to updates according to the following criteria:

- Formula-based vs. model-based: is the semantics of the update defined in terms of changes to the theory itself or in terms of changes to its models?

Classification criteria (2/2)

- Measure of closeness: which measure is used to determine the theory, consistent with the update, that is closest to the original theory?
  - cardinality-based measure: the distance between two sets is expressed by the number of items in which they differ
  - inclusion-based measure: a set S_1 is closer to a set S than a set S_2 if the elements in which S_1 and S differ is a proper subset of the elements on which S_2 and S differ, i.e. if

$$S_1 \Delta S \subset S_2 \Delta S$$

(where $A \Delta B$ is the symmetric set difference $(A \cup B) - (A \cap B)$)

Formula-based update operators

If a theory $\mathcal{T}$ is inconsistent with a new fact F, a straight solution to gain consistency is to repair the theory itself, by removing the minimal number of facts that contradict the new fact. This simple idea underlies the so-called Set-Of-Theories (SOT) approach [13]
Formally:

$$\mathcal{T} \circ_{SOT} F = \{\mathcal{T}_i \cup \{F\} | \mathcal{T}_i \text{is a maximal subset of} \mathcal{T} \text{consistent with } F\}$$

$\Rightarrow$ a formula is true iff it is true in all the theories that result from the update
Problem: one would like the update to return a unique resulting theory!
Two formula-based update operators have then been proposed, both providing a distinct solution to the problem of the possibly multiple theories resulting from the SOT approach: the SOT cross-product and the SOT WIDTIO update operators.

The SOT update operator - Example

Let $\mathcal{T} = \{a \wedge b \Rightarrow c, a, b\}$ and $F = \neg c$.
The maximal subsets of $\mathcal{T}$ that are consistent with F are

$$\{a \wedge b \Rightarrow c, a\}, \{a \wedge b \Rightarrow c, b\}, \{a, b\}$$

Hence:

$$\mathcal{T} \circ_{SOT} F = \{\{a \wedge b \Rightarrow c, a, \neg c\}, \{a \wedge b \Rightarrow c, b, \neg c\}, \{a, b, \neg c\}\}$$

The SOT cross-product update operator

Suppose that the SOT update returns a finite number of theories $\mathcal{T}_1, \ldots, \mathcal{T}_n$.
Then, the SOT cross-product update operator is defined as follows:

$$\mathcal{T} \circ_{SOTcross-prod} F = \{p_1 \vee \ldots \vee p_m \mid p_i \in \mathcal{T}_i, 1 \leq i \leq m\}$$

i.e., it returns the theory whose formulas are all disjunctions with one formula from each theory $\mathcal{T}_i$.
It can be shown that $\circ_{SOTcross-prod}$ fully captures the semantics of the SOT approach to updates, i.e., the set of models of the resulting theory is exactly the union of all models of the theories resulting from the SOT update, i.e.

$$\cup_{i=1,\ldots,n} Mod(\mathcal{T}_i) = Mod(\bigvee\{\mathcal{T}_i \mid i \in 1, \ldots, n\})$$

The WIDTIO approach

Of course, computing a unique theory as discussed above may not be feasible in practice, given that the number of formulas may grow exponentially in the number of theories resulting from the SOT update. The WIDTIO (When In Doubt Throw It Out) approach [14, 15] solves this problem by defining the following update operator $\circ_{SOTW}$:

$$\mathcal{T} \circ_{SOTW} F = \bigcap_{i=1,\ldots,n} \mathcal{T}_i$$

i.e., only the formulas that appear in all the repairs of the theory are retained

- much less expensive to compute
- big loss of knowledge in "bad" cases (all knowledge can be retracted in the worst-case!)

The SOT WIDTIO update operator - Example

Consider again $\mathcal{T} = \{a \wedge b \Rightarrow c, a, b\}$ and $F = \neg c$.
Then, by following the WIDTIO approach, we have:

$$\mathcal{T} \circ_{SOTW} F = \{a \wedge b \Rightarrow c, a, \neg c\} \cap \{a \wedge b \Rightarrow c, b, \neg c\} \cap \{a, b, \neg c\} = \{\neg c\}$$

Note, in particular, that $a \vee b$ is not implied by $\mathcal{T} \circ_{SOTW} F$.

Pros and cons of formula-based approaches

- Cons: in general, formula-based approaches may violate the Dalal's *Principle of irrelevance of syntax* [9], i.e., the same update may have different effects on equivalent but distinct theories
- Pros: formula-based approaches look easier and often more intuitive to implement
- Pros: formula-based approaches allow to assign priorities to facts, to be taken into account when updating the theory [13].

Model-based update operators

The semantics of model-based update operators is defined in terms of the models of the theory to be updated.
In particular, a model-based update aims at updating each model of theory *separately*.
The model-based update operator that was considered more in the literature was introduced by M. Winslett [26].

The Winslett's update operator

Define the distance between a model M and a fact F as follows:

$$\Delta_M^{min}(F) = min_{\subseteq}(\{M \Delta M' : M' \in Mod(F)\})$$

i.e., as the set of minimal symmetric differences between M and the models of F diverge.
Now, suppose to have a theory $\mathcal{T}$ and a new fact F. Winslett's update operator is defined as follows:

$$Mod(\mathcal{T} \circ_W F) = \bigcup_{M \in Mod(\mathcal{T})} \{M' \in Mod(F) : M \Delta M' \in \Delta_M^{min}(F)\}$$

i.e., the theory resulting from the Winslett's operator is such that it captures, for each model M of $\mathcal{T}$, the models of F that minimally differ from M, according to the inclusion-based measure of closeness among models.

The Winslett's update operator - Example

Consider $\mathcal{T} = \{a \equiv \neg b\}$ and $F = a$. Note that F is consistent with $\mathcal{T}$. On facts a, b, $\mathcal{T}$ has two models that are $M_1 = \{a\}$ and $M_2 = \{b\}$.

- Consider M_1: the model of F that minimally differs from M_1 is M_1 itself
- Consider M_2: the model of F that minimally differs from M_2 is $\{a, b\}$.

Therefore:

$$\mathcal{T} \circ_W F = a$$

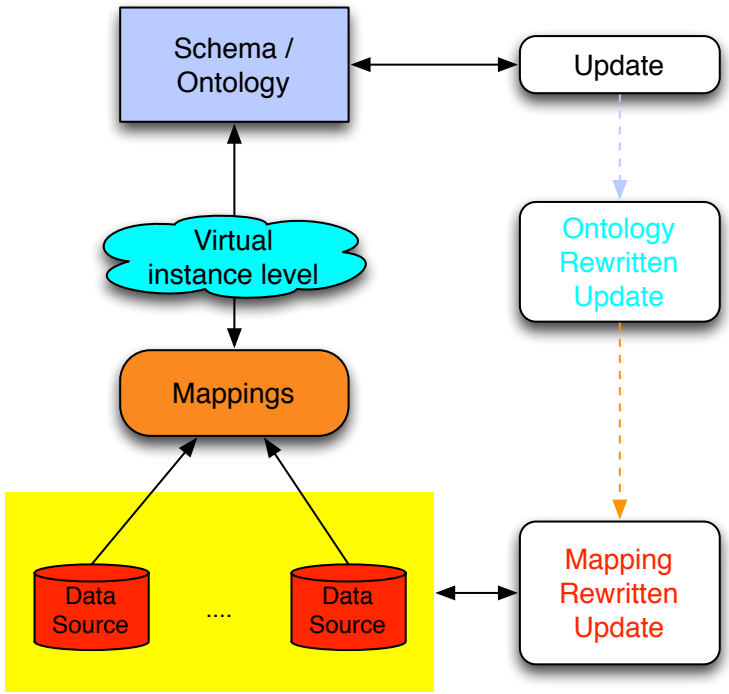
Overview

1. Introduction
2. Approaches to updates over KBs
3. Updating DL KBs
   - Problem definition
   - Applying model-based approaches to DL evolution
4. The view update problem
5. Conclusions

DL KBs instance level update problem

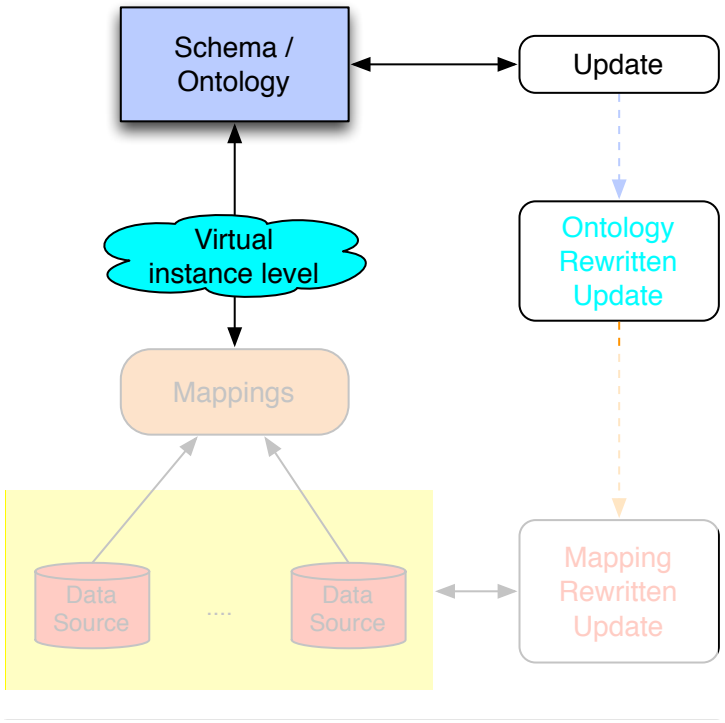
Given a DL KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, and a set $\mathcal{F}$ of instance assertions, compute the consistent KB $\mathcal{K}' = \langle \mathcal{T}, \mathcal{A}' \rangle$ that *is as close as possible* to $\mathcal{K}$ and is such that $\mathcal{K} \models \mathcal{F}$.

Recall our OBDM scenario



- Update: *insertions* of instance assertions
- Ontology rewritten update: set of update operations, i.e., *insertions* and *deletions* of instance assertions, over the virtual instance level

Recall our OBDM scenario



- Update: *insertions* of instance assertions
- Ontology rewritten update: set of update operations, i.e., *insertions* and *deletions* of instance assertions, over the virtual instance level

Applying model-based approaches to DL evolution

Defining the evolution operators in terms of sets of models, gives rise to the following evolution expressibility problem:

> *Given an ontology $\mathcal{O}$ expressed in a language $\mathcal{L}$, a fact F and an evolution operator $\circ$ such that $\mathcal{O} \circ F = \mathcal{M}$ where $\mathcal{M}$ is the set of models obtained through the update.*

$\rightsquigarrow$Expressibility problem: Is there an ontology $\mathcal{O}'$ expressed in $\mathcal{L}$ such that $Mod(\mathcal{O}') = \mathcal{M}$?

It has been shown that for very simple DL languages, the update is not expressible (see e.g., [20]).

Instance level updates over simple DL KBs

- Expressibility problem: model-based approaches to instance-level update have been proved to be not suitable: over simple DL KBs, updates are not expressible [10].
- Counterintuitive behaviour

Expressibility problem of Winslett update operator

Example

$\mathcal{T}: \quad \forall x \exists y HasHusband(x,y) \sqsubseteq Male(x), \quad \forall x Single(x) \sqsubseteq \neg \exists y HasHusband(x,y)$
$\mathcal{A}: \quad \{\exists y HasHusband(Mary,y)\},$

insert $F = \{Single(Mary)\}$

As the result of the update:

- $Single(Mary)$ must be logically implied
  $\leadsto \exists y HasHusband(Mary,y)$ must be removed from the ontology
- to remain as close as possible to the original KB (according to Winslett update semantics), we would like to express the fact that $Male$ contains at least an instance
  $\leadsto$ this cannot be expressed by an instance level assertion!

Counterintuitive behaviour of the Winslett's update operator

Example

$\mathcal{T}: \quad \forall x Male(x) \sqsubseteq Human(x), \quad \forall x Female(x) \sqsubseteq Human(x), \quad \forall x Male(x) \sqsubseteq \neg Female(x)$
$\quad \forall x \exists y Human(x) \sqsubseteq HasFather(x,y), \quad \forall x \exists y HasFather(y,x) \sqsubseteq Male(x)$
$\mathcal{A}: \quad \{Male(Mario), Human(Andrea)\},$

insert $F = \{Female(Andrea)\}$

There exists at least one model M of $\langle \mathcal{T}, \mathcal{A} \rangle$ in which $HasFather(Mario, Andrea)$ is true. Now, suppose to update such a model to make it satisfy F. We have to remove from M the fact $HasFather(Mario, Andrea)$.

A counterintuitive behaviour of the Winslett's update operator (cont'd)

Example (cont.)

Then, we obtain the following models of $\langle \mathcal{T}, F \rangle$ that are at minimum distance from M:

- M': obtained by removing the fact $Human(Mario)$
- the family of models M''_{c_i}: obtained by adding the facts $HasFather(Mario, c_i)$, for all constants c_i within the alphabet of constants

Hence, by stating $Female(Andrea)$ we loose certainty about the fact $Human(Mario)$!

"Formula-based" approach

- In order to overcome model-based approaches drawbacks, in [6, 19] the authors propose to adapt to DL KBs, the SOT approach
  $\leadsto$ an update can return multiple KBs which are at minimal distance from the original KB!
- Two different approaches have been proposed to solve the problem of multiple update results:
  - in [19], the authors propose to follow an adaptation of the WIDTIO approach
  - in [6], the authors propose to choose one KB nondeterministically, which they call bold semantics.

$\leadsto$ Challenge: compute the result without computing all ABoxes accomplishing the insertion minimally.

Overview

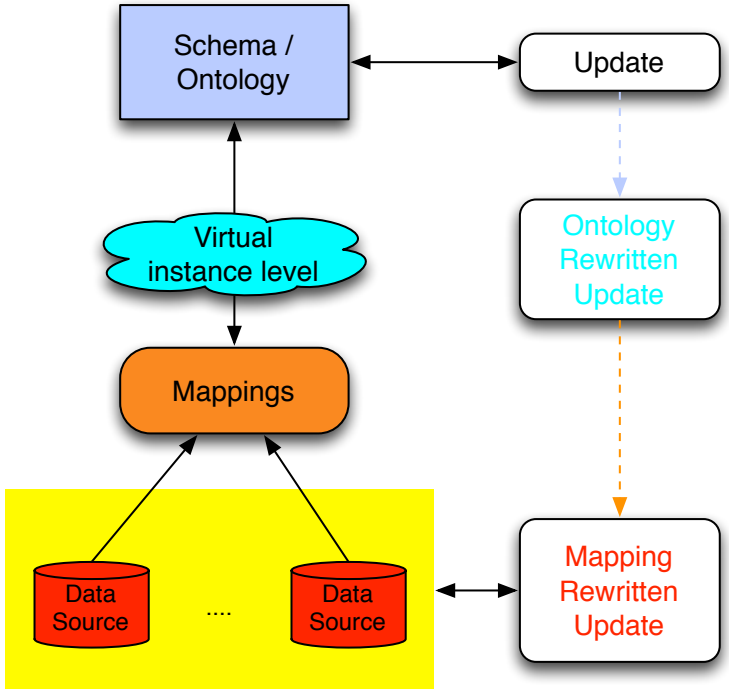
1. Introduction
2. Approaches to updates over KBs
3. Updating DL KBs
4. The view update problem
   - Problem definition
   - Requirements
5. Conclusions

The view update problem

Given a set of views $\mathcal{V}$ over a set of database relations $\mathcal{D}$, and an update operation U (e.g., a set of insertions) over the views, compute the update operation U' over the database relations such that given the set of relations $\mathcal{D}'$ resulting from U', the views $\mathcal{V}$ over $\mathcal{D}'$ reflect exactly the update U.

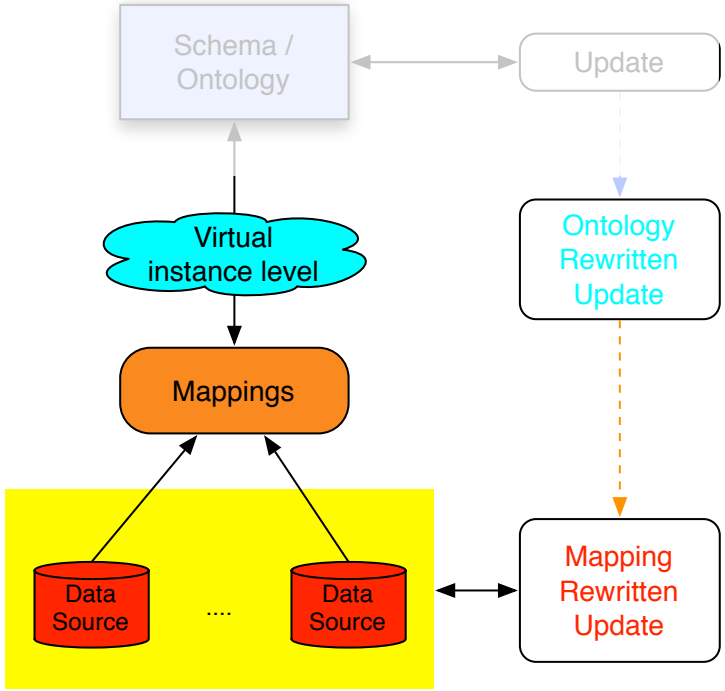
N.B. The seminal work of Bancilhon and Spyratos [3] lays the theoretical foundations of the view update problem. Refer to [16] for a recent study on the problem (from which we borrowed the following examples).

Recall our OBDM scenario



- Ontology rewritten update: *insertions* and *deletions* of instance assertions over the virtual instance level
- Mapping rewritten update: set of update operations, i.e., insertions

Recall our OBDM scenario



- Ontology rewritten update: *insertions* and *deletions* of instance assertions over the virtual instance level
- Mapping rewritten update: set of update operations, i.e. insertions and deletions of tuples, over the data sources

Consistency

V(x,y,z)=P(x,y),C(y,z)

ORG	UNI	CHAIR
Torlone	Roma 3	Atzeni

P

ORG	UNI
Mecella	Roma 1
Torlone	Roma 3

C

UNI	CHAIR
Roma 3	Atzeni

- Insert into **V** the green tuple
- Propagate the update over **P** and **C** by inserting the green tuples
  $\leadsto$ the red tuple is inserted as a side-effect

Consistency: the changes of the database relations reflect exactly the changes of the view relation.

Consistency

V(x,y,z)=P(x,y),C(y,z)

ORG	UNI	CHAIR
Torlone	Roma 3	Atzeni
Lembo	Roma 1	Lenzerini

P

ORG	UNI
Mecella	Roma 1
Torlone	Roma 3

C

UNI	CHAIR
Roma 3	Atzeni

- Insert into **V** the green tuple
- Propagate the update over **P** and **C** by inserting the green tuples
  $\leadsto$ the red tuple is inserted as a side-effect

Consistency: the changes of the database relations reflect exactly the changes of the view relation.

Consistency

V(x,z)=P(x,y),C(y,z)

ORG	UNI	CHAIR
Torlone	Roma 3	Atzeni
Lembo	Roma 1	Lenzerini
Mecella	Roma 1	Lenzerini

P

ORG	UNI
Mecella	Roma 1
Torlone	Roma 3
Lembo	Roma 1

C

UNI	CHAIR
Roma 3	Atzeni
Roma 1	Lenzerini

- Insert into **V** the green tuple
- Propagate the update over **P** and **C** by inserting the green tuples
  $\leadsto$ the red tuple is inserted as a side-effect

Consistency: the changes of the database relations reflect exactly the changes of the view relation.

Consistency

V(x,z)=P(x,y),C(y,z)

ORG	UNI	CHAIR
Torlone	Roma 3	Atzeni
Lembo	Roma 1	Lenzerini
Mecella	Roma 1	Lenzerini

P

ORG	UNI
Mecella	Roma 1
Torlone	Roma 3
Lembo	Roma 1

C

UNI	CHAIR
Roma 3	Atzeni
Roma 1	Lenzerini

- Insert into **V** the green tuple
- Propagate the update over **P** and **C** by inserting the green tuples
  $\leadsto$ the red tuple is inserted as a side-effect

Consistency: the changes of the database relations reflect exactly the changes of the view relation.

Univocality

V(x,z)=P(x,y),C(y,z)

ORG	CHAIR
Torlone	Atzeni

P

Org	UNI
Mecella	Roma 1
Torlone	Roma 3

C

UNI	CHAIR
Roma 3	Atzeni

- Insert into **V** the green tuple
- Many different way of propagating the update over **P** and **C** exist!

Univocality: there exists only one way, if any, of consistently propagating any given update .

Univocality

V(x,z)=P(x,y),C(y,z)

ORG	CHAIR
Torlone	Atzeni
Lembo	Lenzerini

P

ORG	UNI
Mecella	Roma 1
Torlone	Roma 3

C

UNI	CHAIR
Roma 3	Atzeni

- Insert into **V** the green tuple
- Many different way of propagating the update over **P** and **C** exist!

Univocality: there exists only one way, if any, of consistently propagating any given update .

Univocality

V(x,z)=P(x,y),C(y,z)

ORG	CHAIR
Torlone	Atzeni
Lembo	Lenzerini

P

ORG	UNI
Mecella	Roma 1
Torlone	Roma 3
Lembo	y

C

UNI	CHAIR
Roma 3	Atzeni
y	Lenzerini

- Insert into **V** the green tuple
- Many different way of propagating the update over **P** and **C** exist!

Univocality: there exists only one way, if any, of consistently propagating any given update .

Univocality

V(x,z)=P(x,y),C(y,z)

ORG	CHAIR
Torlone	Atzeni
Lembo	Lenzerini

P

ORG	UNI
Mecella	Roma 1
Torlone	Roma 3
Lembo	y

C

UNI	CHAIR
Roma 3	Atzeni
y	Lenzerini

- Insert into **V** the green tuple
- Many different way of propagating the update over **P** and **C** exist!

Univocality: there exists only one way, if any, of consistently propagating any given update .

Overview

1. Introduction
2. Approaches to updates over KBs
3. Updating DL KBs
4. The view update problem
5. Conclusions

OBDM: several open problems

- Update through OBDM systems builds upon many challenging problems, some of which still require investigation; for example:
  - investigate a model-based semantics for DL knowledge bases, that does not give rise to any counterintuitive behavior
  - investigate approaches to updates that possibly affect the intensional level of KB in order to accomplish an instance level update
  - ...
- the combination of the problem of updating a DL KB with the view update problem poses many new challenges...

References I

[1] A. Amoroso, G. Esposito, D. Lembo, P. Urbano, and R. Vertucci.
Ontology-based data integration with MASTRO-I for configuration and data management at SELEX Sistemi Integrati.
In *Proc. of the 16th Ital. Conf. on Database Systems (SEBD)*, pages 81–92, 2008.

[2] N. Antonioli, F. Castanò, S. Coletta, S. Grossi, D. Lembo, M. Lenzerini, A. Poggi, E. Virardi, and P. Castracane.
Ontology-based data management for the italian public debt.
In *Proc. of the Eighth International Conference, FOIS 2014*, pages 372–385, 2014.

[3] F. Bancilhon and N. Spyratos.
Update semantics of relational views.
ACM Trans. on Database Systems, 6(4):557–575, 1981.

References II

[4] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, A. Poggi, M. Rodriguez-Muro, R. Rosati, M. Ruzzi, and D. F. Savo.
The Mastro system for ontology-based data access.
Semantic Web J., 2(1):43–53, 2011.

[5] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati.
Tractable reasoning and efficient query answering in description logics: The *DL-Lite* family.
J. of Automated Reasoning, 39(3):385–429, 2007.

[6] D. Calvanese, E. Kharlamov, W. Nutt, and D. Zheleznyakov.
Evolution of *DL-Lite* knowledge bases.
In *Proc. of the 9th Int. Semantic Web Conf. (ISWC)*, volume 6496 of *Lecture Notes in Computer Science*, pages 112–128. Springer, 2010.

References III

[7] C. Civili, M. Console, G. De Giacomo, D. Lembo, M. Lenzerini, L. Lepore, R. Mancini, A. Poggi, R. Rosati, M. Ruzzi, V. Santarelli, and D. F. Savo.
MASTRO STUDIO: Managing ontology-based data access applications.
Proc. of the VLDB Endowment, 6:1314–1317, 2013.

[8] S. S. Cosmadakis and C. H. Papadimitriou.
Updates of relational views.
J. of the ACM, 31(4):742–760, 1984.

[9] M. Dalal.
Investigations into a theory of knowledge base revision.
In *Proc. of the 7th Nat. Conf. on Artificial Intelligence (AAAI)*, pages 475–479, 1988.

References IV

[10] G. De Giacomo, M. Lenzerini, A. Poggi, and R. Rosati.
On instance-level update and erasure in description logic ontologies.
J. of Logic and Computation, Special Issue on Ontology Dynamics, 19(5):745–770, 2009.

[11] F. Di Pinto, D. Lembo, M. Lenzerini, R. Mancini, A. Poggi, R. Rosati, M. Ruzzi, and D. F. Savo.
Optimizing query rewriting in ontology-based data access.
In *Proc. of the 16th Int. Conf. on Extending Database Technology (EDBT)*, pages 561–572, 2013.

[12] T. Eiter and G. Gottlob.
On the complexity of propositional knowledge base revision, updates and counterfactuals.
Artificial Intelligence, 57:227–270, 1992.

References V

[13] R. Fagin, J. D. Ullman, and M. Y. Vardi.
On the semantics of updates in databases.
In *Proc. of the 2nd ACM SIGACT SIGMOD Symp. on Principles of Database Systems (PODS)*, pages 352–365, 1983.

[14] M. Ginsberg.
Counterfactuals.
Artificial Intelligence, 30(1):35–79, 1986.

[15] M. L. Ginsberg and D. E. Smith.
Reasoning about action I: A possible worlds approach.
Technical Report KSL-86-65, Knowledge Systems, AI Laboratory, 1987.

[16] P. Guagliardo.
The logic of view updates.
PhD thesis, Free University of Bozen-Bolzano, 2014.

References VI

[17] H. Katsuno and A. Mendelzon.
On the difference between updating a knowledge base and revising it.
In *Proc. of the 2nd Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR)*, pages 387–394, 1991.

[18] M. Lenzerini.
Ontology-based data management.
In *Proc. of the 20th Int. Conf. on Information and Knowledge Management (CIKM)*, pages 5–6, 2011.

[19] M. Lenzerini and D. F. Savo.
On the evolution of the instance level of *DL-Lite* knowledge bases.
In *Proc. of the 24th Int. Workshop on Description Logic (DL)*, volume 745 of *CEUR Electronic Workshop Proceedings*, `http://ceur-ws.org/`, 2011.

References VII

[20] H. Liu, C. Lutz, M. Milicic, and F. Wolter.
Updating description logic ABoxes.
In *Proc. of the 10th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR)*, pages 46–56, 2006.

[21] R. Reiter.
On closed world data bases.
In H. Gallaire and J. Minker, editors, *Logic and Databases*, pages 119–140. Plenum Publ. Co., 1978.

[22] M. Rodriguez-Muro and D. Calvanese.
High performance query answering over *DL-Lite* ontologies.
In *Proc. of the 13th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR)*, pages 308–318, 2012.

References VIII

[23] M. Rodriguez-Muro, R. Kontchakov, and M. Zakharyaschev.
Ontology-based data access: Ontop of databases.
In *Proc. of the 12th Int. Semantic Web Conf. (ISWC)*, volume 8218 of *Lecture Notes in Computer Science*, pages 558–573. Springer, 2013.

[24] R. Rosati.
Prexto: Query rewriting under extensional constraints in *DL-Lite*.
In *Proc. of the 9th Extended Semantic Web Conf. (ESWC)*, volume 7295 of *Lecture Notes in Computer Science*, pages 360–374. Springer, 2012.

[25] D. F. Savo, D. Lembo, M. Lenzerini, A. Poggi, M. Rodríguez-Muro, V. Romagnoli, M. Ruzzi, and G. Stella.
MASTRO at work: Experiences on ontology-based data access.
In *Proc. of the 23rd Int. Workshop on Description Logic (DL)*, volume 573 of *CEUR Electronic Workshop Proceedings*, `http://ceur-ws.org/`, pages 20–31, 2010.

References IX

[26] M. Winslett.
A model-based approach to updating databases with incomplete information.
ACM Trans. on Database Systems, 13(2):167–196, 1988.

Grazie