

Relations entre bases de données et ontologies dans le cadre du web des données

Olivier Curé

▶ **To cite this version:**

Olivier Curé. Relations entre bases de données et ontologies dans le cadre du web des données. Base de données [cs.DB]. Université Paris-Est, 2010. tel-00843284

HAL Id: tel-00843284
https://theses.hal.science/tel-00843284

Submitted on 11 Jul 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Mémoire d'
Habilitation à Diriger des Recherches

présenté à l'
Université de Paris-Est

Spécialité
Informatique

On the Relationships Between Databases and Ontologies in the Context of the Web of Data

par
Olivier Curé

présenté le 11 octobre 2010 devant le jury composé de:

MM	Bernd AMANN	*Rapporteur*
	Jorge CARDOSO	*Rapporteur*
	Fabien GANDON	*Examinateur*
	Jean-Paul GIROUD	*Invité*
	Stefan JABLONSKI	*Examinateur*
	Robert JEANSOULIN	*Rapporteur*
	Odile PAPINI	*Examinateur*

To Philippe Curé

We achieve more than we know.
We know more than we understand.
We understand more than we can explain.
Claude Bernard quoted by Daniel Crevier
"The Tumultuous History of the Search for Artificial Intelligence," 1993.

Acknowledgments

I would like to thank

my parents
my wife and children
the members of the jury of this habilitation
Jean-Paul Giroud for twelve years of a rich collaboration
Robert Jeansoulin for opening doors to interesting projects
Jean Fruitet, former colleague at the Université Marne la Vallée
all colleagues at the Université Paris-Est Marne-la-Vallée
all research project partners

Contents

1 Introduction **8**
- 1.1 Introduction . . . 8
- 1.2 Motivating example . . . 8
- 1.3 Organization of this document . . . 11

2 The DBOM framework **13**
- 2.1 Introduction . . . 13
- 2.2 Mapping framework and the impedance mismatch issue . . . 15
  - 2.2.1 Syntax . . . 15
  - 2.2.2 Semantics . . . 18
- 2.3 Knowledge base maintenance through a trigger based strategy . . . 20
- 2.4 Dealing with uncertainties with preferences . . . 20
  - 2.4.1 R-preferences . . . 21
  - 2.4.2 A-preferences . . . 22
  - 2.4.3 Full-preferences . . . 22
- 2.5 Reasoning in DBOM . . . 23
  - 2.5.1 Inference-based instantiation of knowledge bases . . . 24
  - 2.5.2 Incremental generation from previous mapping assertion . . . 25
- 2.6 Presentation of papers . . . 30
- 2.7 Conclusion and perspectives . . . 31

3 Ontology mediation **33**
- 3.1 A Formal Concept Analysis approach to merge ontologies . . . 34
  - 3.1.1 Source TBoxes . . . 35
  - 3.1.2 Source ABoxes . . . 36
  - 3.1.3 Generation of the Galois connection lattice . . . 37
  - 3.1.4 Dealing with emerging concepts . . . 38
  - 3.1.5 Dealing with uncertainty . . . 43
- 3.2 Semantic integration in the DaltOn framework . . . 46
  - 3.2.1 DaltOn architecture . . . 46
  - 3.2.2 Architecture of the SeI component . . . 47
  - 3.2.3 Schema to ontology mapping . . . 50
  - 3.2.4 Methodology and heuristics . . . 52
- 3.3 Presentation of papers . . . 54
  - 3.3.1 FCA papers . . . 54
  - 3.3.2 DaltOn papers . . . 55
- 3.4 Conclusion and perspectives . . . 56

4 Ontology-based data quality enhancement of relational databases 57
4.1 Data quality enhancement using ontologies and inductive reasoning 58
4.2 Condition Inclusion dependencies (CINDs) 64
4.3 Conditional dependencies in the context of ontologies 68
4.3.1 Conditional queries as SPARQL queries 69
4.3.2 Conditional queries in an OBDA context 70
4.4 Presentation of papers 70
4.5 Conclusion and perspectives 72

5 Other works involving ontologies 73
5.1 Modeling an application ontology of underwater archaeological surveys of amphorae 73
5.1.1 Application ontology 74
5.1.2 Mapping and extension to CIDOC CRM 76
5.1.3 Reasoning with the application ontology 79
5.1.4 Future works and conclusion 79
5.2 Ontologies to design a Domain Specific Language 80
5.2.1 The Ocelet DSL 81
5.2.2 Ontology-based Ocelet modeling architecture 82
5.2.3 Conclusion and future works 84
5.3 Presentation of papers 85
5.4 Conclusion and perspectives 86

6 Curriculum Vitae 87
6.1 Education 87
6.2 Positions 87
6.3 Languages 88
6.4 Research activities 88
6.4.1 Integration and exchange of data 88
6.4.2 Ontology mediation 88
6.4.3 Knowledge representation 89
6.4.4 RDF triple storage 89
6.5 Teaching activities and student supervision 89
6.5.1 Supervision of MsC Theses 90
6.5.2 Ph.D. supervision 91
6.5.3 Ph.D. committee 91
6.6 Scientific collaborations 92
6.6.1 National collaborations 92
6.6.2 International collaborations 92
6.7 Administrative tasks 92
6.8 Software Developments, Publications and Communications . . . 93
6.8.1 Software Developments 93
6.8.2 International Journals 93
6.8.3 Book Chapters 94
6.8.4 Publications in Conferences (with review) 94
6.8.5 Dissertation 96
6.8.6 Deliverables for EU project 96
6.8.7 Others 97

7 Conclusion 98

Bibliography **100**

Chapter 1

Introduction

1.1 Introduction

This dissertation presents my interest in developing methods and algorithms necessary for realizing advanced applications for the Semantic Web [BLHL01]. This extension of the current Web aims to allow the integration and sharing of data across organizations and applications. A direct consequence of the success of this approach would enable to consider the Web as a global database containing the data stored on all connected machines. This aspect is well translated on the W3C Semantic Web Activity web site[1] which states that the Semantic Web is a Web of Data.

Thereby, this Web of Data will permit to submit structured queries on all accessible connected datasets and to retrieve relevant results coming from diverse and heterogeneous sources. A main issue related to this heterogeneity aspect concerns the notion of semantics. In the context of the Semantic Web, this is generally tackled with ontologies and associated mediation operations.

My research is anchored into these topics and this dissertation aims at presenting some of my investigations, results as well as to describe some of the applications I have designed and implemented.

1.2 Motivating example

The different contributions presented in this document have been largely motivated by the implementation of a medical informatics Web application [Cur02]. This application, named XIMSA (eXtended Interactive Multimedia System for Auto-medication), is the result of more than ten years of collaboration with the CNRS clinical pharmacology department at the CHU Cochin (Paris, France) and its former director, Pr. Jean-Paul Giroud (M.D., Ph.D, D.Sc, WHO expert). In order to fully motivate the next three chapters of this dissertation, we first present the main features proposed in this self-medication application and provide an overview of its architecture.

XIMSA aims at providing several services to the general public for an accompanied and responsible self-medication. The underlying goal is to improve

[1] http://www.w3.org/2001/sw/

the general public's comprehension of mild clinical signs and their relations to drug products. Figure 1.1 presents an overview of XIMSA's architecture where three main modules are presented: SEHR management, Diagnosis and Drug.

The `Drug` module proposes a single service, namely `search`, and as such is the simplest module of the application. The `search` service provides all the information concerning a peculiar drug product and directly uses the `medical` database without requiring any form of inferences.

The `Diagnosis` module chains three services. Among them, the `anamnesis` service invites an end-user (also referred as a patient) to select a symptom and then initiates interactions in order to characterize the gravity of this symptom. If the described symptom lies in the domain of self-medication, the `Drug provider` service searches for efficient drug products satisfying constraints associated to the information contained in this patient's Simplified Electronic Health Record (SEHR) document (i.e. mainly known diseases, allergies and current treatments). In case such drugs exist, the `Drug description` service organizes their presentation given an order based on a efficiency/tolerance ratio.

A SEHR is a document that stores information concerning a patient. These information concern personal information (e.g. gender, date of birth, etc.), clinical antecedents, allergies and history of drug treatments. This component enables us to personalize the questions asked by the anamnesis service, as well as improve the accuracy of drug propositions (we do not use the term 'prescription' to contrast with the task of a healthcare professional) and provide some personalized health related alerts, e.g. end of treatment with a given product (more details on this component can be found on [Cur04] and [Cur05a]). The `SEHR management` module proposes a maintenance service for each of these SEHR aspects. These services basically enable to perform CRUD (Create, Retrieve, Update and Delete) operations on the SEHR information.

Obviously, these last two modules contain services making intense use of the different available databases, i.e. medical (containing information about drug products and symptoms) and SEHR. But their functionalities go beyond simply retrieving or updating information from a set of databases. In fact, they require some forms of reasoning which are performed using a knowledge base together with the database instances. These inferences are provided by a Description Logic (DL) [BCM$^+$03] reasoner as well as some specific graph navigation operations. They are used by both the back and front office of the application.

Concerning the front office, inferences are used by the `diagnostic` module to personalize the questions asked to the patient within the `anamnesis` service and to propose a set of adapted and efficient drug products (`Drug provider` service). The `SEHR management` module also needs inferences to detect contra-indications at the drug and disease levels when a new drug is introduced in a patient's record. For the back office, reasoning services are exploited to ensure the data quality of underlying databases (Chapter 4 is dedicated to the presentation of our solutions)

The information and knowledge representation issues play an important role in the architecture of XIMSA. A top priority in this representation concern is to provide a declarative approach in contrast to a procedural one. This is motivated by the genericity, re-usability and interoperability of a declarative approach. Genericity concerns the ability to apply the application architecture to other domains, e.g. geography, biology. Re-usability implies that the

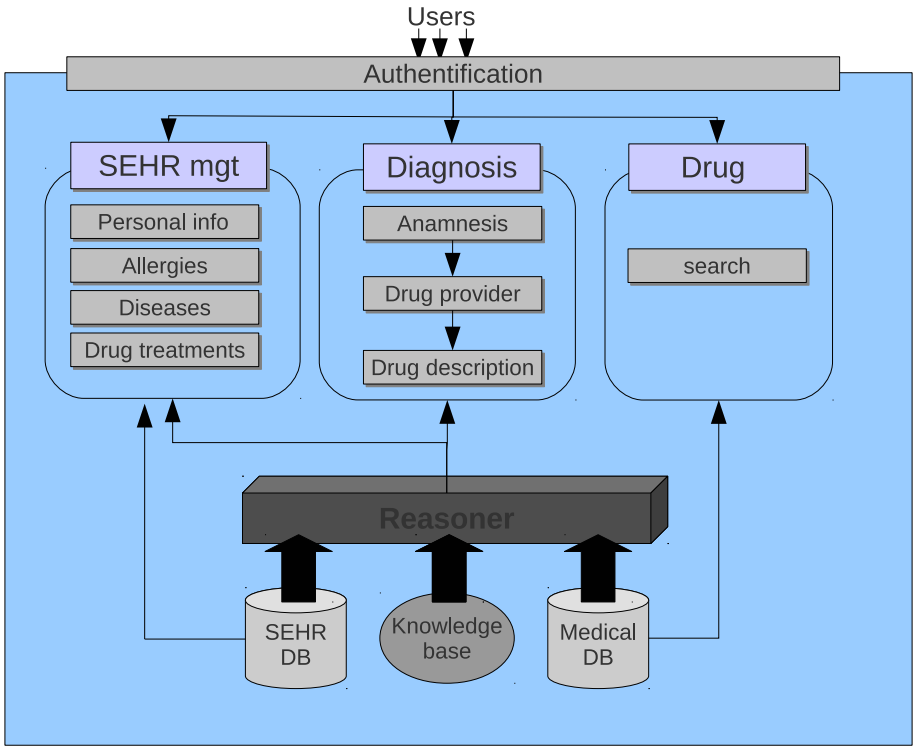


Figure 1.1: Architecture of the XIMSA application

knowledge bases defined for the self-medication application can be exploited in other applications. Finally, the interoperability aspect points to the availability of associated tools to manage and operate on the knowledge (e.g. editors, reasoners, query languages, APIs, etc.). The expressive power of the language used to describe the knowledge is also important and is directly related to the computational complexity of the different reasoning tasks.

Practically, this declarative approach and the need to perform sound and complete inferences implies to consider a logic-based solution to represent information and knowledge in a Web environment. We have selected the technologies proposed by the W3C's Semantic Web activity, namely RDF(S) and OWL for ontology technologies and SPARQL as a query language, to design the architecture of XIMSA [Cur03]. The ontology component can be considered the corner stone of a Semantic Web application architecture. It consists in a formal description of the concepts and their relationships of a given domain, e.g. medicine. When described in an expressive logical formalism, an ontology [Gru93] enables many different forms of inferences to be performed over its knowledge. We have adopted this approach because of its adequacy to our representation needs, the quality of W3C's recommendations and the availability of robust, scalable and open-source tools, e.g. ontology editors, reasoners, APIs and storage solutions.

A main advantage in designing such an architecture in the medical domain is the availability of terminologies, thesaurus and other forms of ontologies.

Some of them are the result of years of development in large and recognized institutes, e.g. World Healthcare Organizations (WHO), National Institute of Health (NIH) or National Health Service (NHS). Hence, a main task for this self-medication application consists in integrating some of these ontologies, to adapt them to the specific context of our application domain and use them to design inference-based functionalities for both the application's back and front office.

1.3 Organization of this document

The next three chapters of this dissertation present my research activities on some relationships between ontologies and databases. They all have a common structure consisting of an introduction with motivations, presentation of my contributions and obtained results, presentation of the papers published on this research aspect and finally a conclusion with directions for future works. These three chapters are organized as follows:

Chapter 2 presents the DBOM system (DataBase Ontology Mapping) which supports the generation of RDFS/OWL compliant knowledge bases from relational databases. This tool proposes several features which enable to:

- synchronize a knowledge base with a set of databases at the data level,
- manage the uncertainty emerging from schema mappings when several databases are used to construct a knowledge base,
- support several reasoning services that enable to instantiate efficiently knowledge bases and to generate mapping assertions using previously defined mappings and databases at both the data and metadata levels.

DBOM has been used extensively for the development of XIMSA to generate expressive ontologies from terminologies stored in the databases, e.g. on pharmaceutical molecules, therapeutic classes and diseases.

Chapter 3 deals with two ontology mediation solutions. The first one focuses on merging ontologies using methods borrowed from Formal Concept Analysis (FCA). The second one supports data integration in the context of a process modeling project (named DaltOn) and has been designed conjointly with the Database and Information System laboratory of the University of Bayreuth, Germany. In XIMSA, several ontologies related to the pharmaceutical domains needed to be aligned and merged. These tasks were performed using the FCA merging approach.

Chapter 4 presents several works conducted on the data quality domain. It aims at detecting inconsistent data and missing values in relational databases using associated ontologies. A first contribution consists in an induction-based approach to enrich and refine ontologies from databases. These ontologies are later used to ensure detection of inconsistencies in databases and to support data cleansing solutions. A second contribution proposes methods to discover conditional inclusion dependencies in the context of relational database management systems (RDBMS). Together with existing methods for discovering conditional functional dependencies, it is possible to design innovative tools for improving the data quality of relational databases. Finally, these methods have been adapted to the Web of Data by representing dependencies in SPARQL and in

an epistemic query language (namely SparSQL) adapted to the Ontology-Based Data Access (OBDA) approach. The data quality of the information stored in XIMSA is paramount to the application since incorrect or incomplete data can put in danger the life of patients. Most of the data quality solutions presented in Chapter 4 are processed regularly on our databases.

Finally, Chapter 5 summarizes different research activities involving ontologies but which are not related to our self-medication application. It mainly concerns research conducted within collaborative projects on the European FP6 VENUS project and ANR STAMP. My participation on the first project is twofold: it concerns the definition of an archaeological application ontology via a mapping to the CIDOC CRM standard and the proposition of reasoning techniques, based on the use of OWL 2 fragments, to detect inconsistencies of the knowledge base. My contribution to the latter project consists in proposing an ontology-based solution to design and check the consistency of a Domain Specific Language for the domain of dynamic landscape modeling.

The last chapters contain my curriculum vitae which presents my work as a researcher and teacher at the University of Paris Est Marne la Vallée. It proposes a complete list of my publications and involvement in administrative and research tasks. Follows a conclusion and future works still in direction of enhancing Web of Data applications.

Chapter 2

The DBOM framework

2.1 Introduction

The role of ontologies is foundational in providing semantics to vocabularies used to access and exchange data within the Semantic Web. The creation and maintenance of large scale, concrete and useful knowledge bases is a major challenge which will condition the success of the Web of Data [BGG+03]. Since designing ontologies from scratch is considered a complex, error-prone and time-consuming task, any available structured background information can be helpful sources.

Relational databases (in the chapter, whenever we refer to databases, we mean relational ones) are obvious candidates for such background information as they are currently considered the most popular model to store data. Moreover, they possess some interesting properties which encourage the design of ontologies and the instantiation of knowledge bases. One of these properties emerges from the structural similarity between Description Logics (henceforth DL) [BCM+03], currently the most popular model for knowledge bases of the Semantic Web, and relational databases. In fact, they both propose two forms of components: (i) intensional, i.e. schema for databases and domain terminology (called the TBox) for DL, and (ii) extensional, i.e. database instance and assertions that make use of the DL terminology (called the ABox). On the semantics point of view, the authors of [MHS07] emphasize similarities between databases and DLs, as they are both interpreted according to standard first-order logic semantics. Hence DL knowledge bases can be considered as expressive but decidable database schema languages. In contrast, database schemata often do not provide explicit semantics for their data.

In this Chapter, we are interested in generating both the TBox and ABox components of a DL knowledge base from existing databases. To this end we will use: (i) the intensional part of several databases to design a practical ontology, via the declaration of a schema mapping, i.e. a set of mapping assertions, (ii) the extensional part of databases and mapping assertions to instantiate a knowledge base.

We adopted a materialization approach of the ABox, i.e. tuples remain stored in the target structure, as opposed to a virtual approach where the tuples remain in sources and are accessed directly when queries are performed

on the target structure. This is mainly motivated by the fact that some of the databases we are using are not always accessible at run time. In fact, we are periodically supplied by external medical information providers with some database snapshots. Additionally, in order to benefit from DLs expressiveness and associated schema reasoning procedures, it is necessary to enrich the target ontology with DL axioms.

We consider that this enrichment step is at most a semi-automatic process since it usually requires interactions with end-users. For instance, to obtain an expressive ontology, e.g. in OWL2, it is necessary to provide a definition to a concept right after its creation via its mapping assertion processing. That is, consider the creation of mapping assertion defining a `Drug` concept and its processing within the DBOM system. The `Drug` concept is now part of the ontology and one can add restrictions to it, e.g. a drug is a chemical thing that treats a symptom and contains chemical molecules, etc.

We consider that the goal of this approach is to create valuable knowledge bases which are being used in data-centric applications requiring inferences, e.g. [CS06a] and [JVR+09]. Hence, our solution may not be adapted to the definition of upper level ontologies, i.e. describing very general concepts valid across several domains and providing an extension point that propagates coherence constraints and interoperability to its extension, e.g. DOLCE [1] and Suggested Upper Merged Ontology (SUMO) [2].

The DBOM (DataBase Ontology Mapping) system uses this approach and enables users to create, instantiate and maintain expressive and richly axiomatized OWL knowledge bases from relational databases. This system provides an incremental solution to map several database sources to a single ontology and to instantiate it by executing a set of SQL queries. A first important issue in this setting consists in considering the impedance mismatch problem. This problem generally refers to the disparity existing between set-oriented relational database access and iterative one-record-at-a-time host language access. In the context of our application, this is due to the fact that databases store data while knowledge bases represent objects. We handle this issue by providing a mapping language that transforms data retrieved from tuples of the database to objects of the knowledge base (Section 2.2).

The rest of this chapter presents the following contributions:

- a trigger-based strategy to synchronize database sources and a knowledge base at the data level (Section 2.3).
- several preference-based solutions to handle uncertainties that can arise when processing a set of mapping assertions (Section 2.4).
- several inference-based functionalities that enable DBOM to (i) efficiently instantiate a knowledge base and (ii) to automatically generate schema mapping assertions based on previously defined mappings using database constants (Section 2.5).

[1] http://www.loa-cnr.it/DOLCE.html
[2] http://www.ontologyportal.org/

2.2 Mapping framework and the impedance mismatch issue

We argue that DBOM is a hybrid approach between Data Exchange (DE) [Kol05] and Data Integration (DI) [Len02] because it possesses some of the key properties of both these systems. This is clearly demonstrated with the following comparison:

- as in both DE and DI, the source schema is given and the schema mapping is a set of formulas constructed by a human expert. In Section 2.5, we extend this approach by proposing a semi-automatic solution to generate a fragment of these formulas.
- as in DI, the target schema, i.e. the intensional knowledge of a DL, is constructed from the processing of the source schema given a schema mapping.
- as in DE, the target instances are materialized, while they are usually virtualized in the case of DI.

A first important issue considered in the DBOM setting is impedance mismatch which is tackled in the next section through the presentation of a mapping language. Our mapping specification enables us to characterize some values as identifiers which are generally mapped to primary keys of the databases. Moreover, the system supports the representation of compound identifiers, i.e. primary keys composed of several attributes.

2.2.1 Syntax

The mapping approach we have adopted in our system is composed of a set of relational databases, an ontology and a GAV (Global-As-View) schema mapping language with sound sources. This approach requires that the target schema is expressed in terms of queries over the sources [Len02]. Intuitively, an end-user maps a DL concept, respectively a DL role, to a query and in a second step, she maps all distinguished variables of the query to data type properties which have the DL concept as domain, respectively concepts of the ontology, which have previously been defined in the ontology. We now formalize this with a definition of the mapping system.

Definition 1 *: The DBOM system is defined as the following triple:* $\mathcal{DS} = \langle \mathcal{S}, \mathcal{K}, \mathcal{M} \rangle$*, where:*

- $\mathcal{S}$ *is a set of sources* $\{S_1, .., S_n\}$ *corresponding to relational databases that we assume locally satisfy their set of integrity constraints.*
- $\mathcal{K}$ *is the target and corresponds to DL knowledge base. That is, it contains a (possibly empty) TBox (an ontology) as well as a (possibly empty) ABox (a set of assertions over the terms of the ontology).*
- $\mathcal{M}$ *is a set of mapping assertions between* $\mathcal{S}$ *and* $\mathcal{K}$*. This mapping is represented as a set of logical assertions in which views, i.e. queries, expressed over elements of* $\mathcal{S}$ *are put in correspondence to an element of the TBox of* $\mathcal{K}$*.*

Being a hybrid approach between DE and DI, DBOM allows for several interesting knowledge base design approaches:

- creation of a knowledge base from scratch, i.e. $\mathcal{K}$ is empty (i.e. both the TBox and ABox are empty), at the start of the design process. In general, end-users begin with the definition of some primitives concepts and data type properties then they can map novel concepts and roles to elements of the sources, incrementally instantiate the knowledge base and provide axioms to the terminology, e.g. restrictions to the concepts, domain and range to roles.

- start from a non empty TBox and enrich it with new concepts and roles, generate concept/role assertions from tuples stored in the relational database sources via the execution of mapping assertions.

We can now present the form of mapping assertions that have been adopted in our system. The schema mapping language adopted in DBOM corresponds to, respectively in DE and DI terminology, a source to target tuple-generating dependencies (s-t tgd) or GAV in which the right-hand side of the implication consists of a single atomic formula.

Definition 2 *: The mapping assertions in $\mathcal{M}$ for $\mathcal{DS}$ take the following form:*

$$\forall \overline{x}(\phi(\overline{x}) \rightarrow \exists \overline{y}\psi(\overline{x}, \overline{y}))$$

where ϕ denotes a conjunctive query over source relations and ψ denotes either a DL Concept expression or a DL Role (in this case $\overline{y}$ is empty and the existential quantifier is not needed). We denote as ϕ the premise of a mapping rule and ψ its conclusion. The two forms of ψ, therefore named $\mathcal{DS}$-premises, correspond to:

- *Concept where tuples retrieved from the premise serve to create instances of this concept together with data type properties whose domain is this concept. If some of these data type properties are not mapped to distinguished variables of the query, we do not instantiate a property for processed individuals. This approach fits well with the Open World Assumption (OWA) semantics associated to such knowledge bases.*

- *Role where tuples retrieved from the premise serve to relate existing ABox instances via this object property.*

Finally, we present the attribute mappings that are associated to each form of $\mathcal{DS}$-premises:

- if the query attribute is a primary key in its source relation, then the end-user may select this value to be an object identifier in the knowledge base. This definition is supported by a special property defined in an associated mapping ontology, namely "dbom:id". This property creates a linked list of objects where each object has a "dbom:hasIdValue" property whose value is retrieved from the query. This approach enables us to relate source relations with compound primary keys to objects in the knowledge base.

- if the query attribute is not a primary key, the end-user maps it to a data type property.

In DL, roles correspond to binary predicates where a first component relates to the domain and a second to the range. The same approach applies to *Role* $\mathcal{DS}$-premises. As previously explained, our system supports compound keys and thus a non-empty set of distinguished variables may identify the domain and range. It is obvious in our system that the distinguished variables of an *Role* mapping assertion must correspond to attributes that have been previously, in a *Concept* $\mathcal{DS}$-premise mapping, mapped to the *dbom:id* property. The main idea of the system is to define the set of distinguished variables that correspond to the domain, respectively the range, and based on the values retrieved from the execution of this query, identify the knowledge base individuals and relate them via the object property. In Section 2.5, we highlight that this instantiation requires some forms of reasoning.

Example 2.2.1 *Let* $\mathcal{DS}^1 = (\mathcal{S}^1, \mathcal{K}^1, \mathcal{M}^1)$ *be a DBOM integration system where* $\mathcal{S}^1$ *consists of a single source with three relations (starting with lower case letters). The* `drug` *relation is of arity 3 and contains information about drugs with their codes, names and prices. The* `ephMRA` *relation, arity 2, contains codes and names of the Anatomical Classification, i.e. a standard that represents a subjective method of grouping certain pharmaceutical products, proposed by the European Pharmaceutical Market Research Association. Finally, relation* `ephDrug`*, of arity 2, proposes relationships between drug codes and codes of the Anatomical Classification.*

The ontology schema is made of the following members (properties start with the 'has' prefix and concept start with capital letters). The data type properties `hasDrugName`*,* `hasDrugPrice` *whose domains will be inferred to be instances of the* `Drug` *concept and ranges are respectively a drug name and a drug price. The* `hasEphName` *is a data type property with instances of the* `EphMRA` *concept as domain and name as range. Finally, we need an object property, namely* `hasEphMRA`*, to relate instances of the* `Drug` *concept to instances of the* `EphMRA` *concept. Note that this property can only be introduced after the processing of the assertions creating the* `Drug` *and* `EphMRA` *concepts. That is, in the following* $\mathcal{M}^1$ *mapping, (3) must be introduced after the processing of (1) and (2).*

The mapping $\mathcal{M}^1$ *is defined by:*

$$(1)\ \forall x, y, z\ drug(x, y, z) \rightarrow Drug(x, y, z)$$
$$(2)\ \forall x, y\ ephMRA(x, y) \rightarrow EphMRA(x, y)$$
$$(3)\ \forall x, y\ ephDrug(x, y) \rightarrow hasEphMRA(x, y)$$

This mapping is completed by the following:

- *in assertion (1) of* $\mathcal{M}^1$*, the x,y and z of* $Drug(x, y, z)$ *are related respectively to dbom:id, hasDrugName and hasDrugPrice.*
- *in assertion (2) of* $\mathcal{M}^1$*, the x and y of* $EphMRA(x, y)$ *are related respectively to dbom:id and hasEphName.*
- *finally, in assertion (3) of* $\mathcal{M}^1$*, the x and y of* $hasEphMRA(x, y)$ *correspond respectively to a Drug individual identified by x and a EphMRA individual identified by y.*

In the DBOM implementation, the mapping is expressed using SQL queries instead of conjunctive queries and an interactive GUI is proposed to define all elements of the mapping (Figure 2.3). In the previous example the left hand side of mapping assertion (1) would be written with the following SQL query:

```
SELECT code, name, price FROM drug;
```

This query would be mapped to the `Drug` concept and attributes of the select clause would be mapped to data type properties having the `Drug` concept as domain.

2.2.2 Semantics

First, it is important to note that a standard first-order semantics can interpret DL TBoxes and relational schemata because they both distinguish legal structures, i.e. those that satisfy all axioms, named models in DL and database instances in relational databases, from illegal ones, i.e. structures that violate some of them. Thus we can use a first-order semantics with the domain of interpretation being a fixed denumerable set of elements Δ.

In order to specify the semantics of $\mathcal{DS}$, we first have to consider a set of data at the sources, and we need to specify which are the data that satisfy the target schema with respect to such data at the sources. We call $\mathcal{C}$ a *source model* for $\mathcal{DS}$.

Starting from this specification of a *source model*, we can define the information content of the target $\mathcal{K}$. From now on, any interpretation over Δ of the symbols used in $\mathcal{K}$ is called a *target interpretation* for $\mathcal{DS}$.

Definition 3 *: Let $\mathcal{DS} = \langle \mathcal{S}, \mathcal{K}, \mathcal{M} \rangle$ be our data integration system, $\mathcal{C}$ be a source model for $\mathcal{DS}$, a target interpretation $\mathcal{A}$ for $\mathcal{DS}$ is a model for $\mathcal{DS}$ with respect to $\mathcal{C}$ if the following holds:*

1. *the ABox $\mathcal{A}$ is consistent with respect to the TBox of $\mathcal{K}$.*
2. *$\mathcal{A}$ satisfies the mapping $\mathcal{M}$ with respect to $\mathcal{C}$.*

The first condition in Definition 3 can be handled with the DL standard ABox consistency reasoning solution. For the kind of DL we are using in our implementation, namely $\mathcal{SHIF}$(D) and $\mathcal{SHOIN}$(D), corresponding respectively to OWL Lite and OWL DL [BvHH+04], this reasoning problem is decidable (resp. ExpTime-complete and NExpTime-complete). But DL reasoners generally implement heuristics which make standard reasoning services efficient on practical knowledge bases [BCM+03].

We also need to consider the constraints that need to be satisfied by $\mathcal{K}$. To do so we need to adopt a database-like constraint approach in the DL knowledge base. This approach is satisfied by the adoption of the Unique Name Assumption (UNA), i.e. requiring each object instance to be interpreted as a different individual.

The introduction of UNA is supported by the mapping ontology *dbom:id* property. This property is considered as an object property and needs to satisfy the following axioms:

$$(1)\ \forall x, y_1, y_2 (\neg dbom : id(x, y_1) \vee \neg dbom : id(x, y_2) \vee y_1 = y_2)$$
$$(2)\ \forall x, y_1, y_2 (\neg dbom : id(y_1, x) \vee \neg dbom : id(y_2, x) \vee y_1 = y_2)$$

These constraints state that the domain of an object property is identified by a single range (1) and that a range identifies a single domain (2). In the implementation of our system, the *dbom:id* property is defined in terms of cardinality constraints of OWL properties, that is owl:functionalProperty, i.e. equivalent to (1) and owl:inverseFunctionalProperty, i.e. equivalent to (2). Using this approach, we are still able to design ontologies in the decidable fragment of OWL, namely OWL DL or OWL2 DL.

In this Section, we consider that the mapped databases are not overlapping. This means that at most one mapping assertion is defined per ontology element (concept and roles), i.e. there is at most one mapping assertion per element on the right hand side. Given this constraint, the instantiation of mapped ontology elements from database instances is given by the tuples retrieved with the conjunctive queries (i.e. SQL queries in the DBOM implementation). In the next section, we relax this constraint and introduce preferences to prefer a mapping assertion over other ones for a given ontology element.

Example 2.2.2 *We consider the $\mathcal{DS}^1$ system from Example 2.2.1 and the $\mathcal{S}^1$ database instance of Figure 2.1*

Figure 2.1: Extract of database instance of $\mathcal{S}^1$

(a) Drug relation

code	name	price
33316809	Nodex	1.69

(b) EphMRA relation

code	name
R5D1	Plain antitussives
R5D2	Antitussives combinations

(c) EphDrug relation

drugCode	ephMRACode
33316809	R5D1

According to the semantics of $\mathcal{DS}^1$, the ABox displayed in Figure 2.2 is generated.

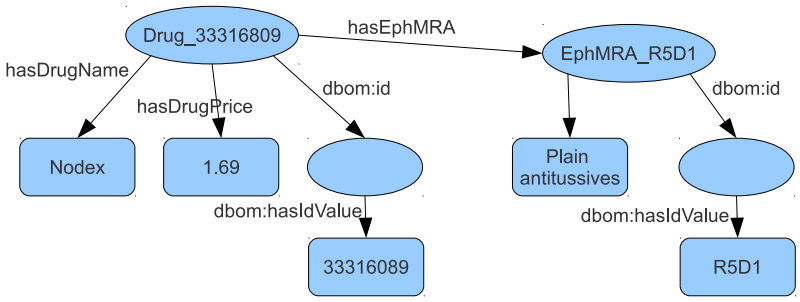


Figure 2.2: ABox graph generated for example 2.2.2

2.3 Knowledge base maintenance through a trigger based strategy

Since the ABox of the generated knowledge base is materialized, a solution is needed to synchronize objects of the ABox with tuples of the database. This solution is supported by a trigger based strategy. Intuitively, it aims at exploiting the dynamic aspect of the relational database sources together with schema mapping to maintain an up-to-date ABox. This approach implies that whenever a database tuple related to a DL entity (concept, role) is updated (via a SQL query), the ABox is synchronized accordingly to ensure that it remains in a consistent state with respect to its sources. The dynamic aspects considered are triggers and referential actions such as `ON UPDATE` and `ON DELETE`. Thus this solution only tackles modifications of instances in relational database tables, meaning that updates of schema through Data Definition Language (DDL) queries are not taken into account.

Generally ABox instance maintenance is usually performed via a complete processing of the mapping. But it is much more efficient to propose a differential approach where only the modified tuples of the sources imply a modification to the knowledge base.

In [CS05], the trigger strategy definition is expressed with the formalization of a translation approach which relates a relational model to DBOM's mapping model.

Based on this translation approach, a set of standard SQL triggers are automatically generated by the DBOM system. They are defined as `AFTER` triggers, because the system needs to have the data stored in the database before processing updates of the ABox, and as `ROW LEVEL` triggers, meaning that the trigger action is executed for every row that is inserted, deleted or updated by the query as opposed to statement level where the trigger is executed only once [EN06]. This `AFTER/ROW LEVEL` policy allows to consider, when not limited by recursive trigger characteristics, referential actions such as `ON DELETE` | `ON UPDATE` action.

The triggers are implemented by two generic Java methods (one for ontology concepts and another one for roles) defined in the DBOM framework and whose main purpose is to find candidate ontology concepts and roles to be updated. Once a set of update candidates are identified, the modifications can be performed at the knowledge base level. For this purpose, several heuristics have been defined to solve the problem of efficiently and accurately mapping an update at the database level to the knowledge base level. They are decomposed into rules that solve insert, delete and update trigger events. For each of these three categories, we distinguish between triggers that call the property method and those that call the concept method (more details are provided in [CS05]).

2.4 Dealing with uncertainties with preferences

As said earlier, DBOM aims to integrate several relational database sources on an OWL knowledge base. A situation frequently encountered in integration solutions is the execution of uncertain mappings. Such mappings are often generated from overlapping sources where data are possibly not up-to-date and unreliable. In such a situation, one needs to give a special attention to instan-

tiate the most reliable, up-to-date and consistent knowledge base. In order to guarantee the consistency of the processed target knowledge base, end-users can set preferences over the mapping assertions.

Preferences have some of their origins in decision theory where they support complex, multi factorial decision processes [Fis70]. Preferences have also motivated research in the field of databases starting with [LL87]. In [Kie02], the authors distinguish between quantitative and qualitative approaches to preferences. In the quantitative approach, a preference is associated with an atomic scoring function. This approach usually restricts the approach to total orderings of result tuples. The qualitative approach is more general than the quantitative as it proposes partial ordering of results.

We now motivate the reasons for a quantitative approach in the $\mathcal{DS}$ system:

- it responds effectively to our need to correct inconsistencies when a given individual, identified by a given key, may be generated from several mapping assertions.
- the task of setting preferences is only required for ontology elements defined by different source views. Thus the total order aspect of this quantitative approach does not make the preference setting task more restrictive.
- in practice, the task of setting preferences to views of a given ontology element may not be complex for a mapping designer. Users responsible for the design of mappings generally know the sources well and are able to tell which source is more reliable than others.

We have described several levels of preferences which we describe in the following sub-sections.

2.4.1 R-preferences

In [CJ07c], we presented a solution that enables us to define preferences over conjunctive queries of mapping assertions, named *R-preferences*. This approach considers that data retrieved from a source all have the same reliability value.

Definition 4 *: Let $\mathcal{DS}$ be a DBOM system as defined in Definition 1. We consider the situation where an element of $\mathcal{K}$ (i.e. Concept or Role) is defined with several mapping assertions, denoted ma_i over sources of $\mathcal{S}$. In order to avoid inconsistencies, the end-user defines a total order relation, denoted $>_R$, on the mapping assertions of this element. For two mapping assertions ma_1 and ma_2, if $ma_1 >_R ma_2$ then*

- *all objects retrieved from a mapping assertion with no counter parts (i.e. objects identified by the same identifying values) in other mapping assertions are created in the ABox.*
- *all objects that could be retrieved from both mapping assertions, are created exclusively from values of ma_1 since it is the preferred one.*

2.4.2 A-preferences

A main limitation with *R-preferences*, is that it is not possible to compose objects from different sources. That is all values of an object come from a single source. In order to propose a fine-grained composition of objects, we enriched our system with a preference setting over attribute of the views. They are denoted *A-preferences* and are set on the distinguished variables of mapping assertions. They enable to create fine-grained objects from source tuples. With this approach, one can express that for a given object, the system prefers the value of an attribute stored in one source to another attribute stored in another source. Thus the system is able to compose an object via retrieving preferred values from different sources.

Definition 5 *: Let $\mathcal{DS} = \langle \mathcal{S}, \mathcal{K}, \mathcal{M} \rangle$ be our data integration system with A-preferences over mapping assertions. We consider the situation where an element of $\mathcal{K}$ (i.e. Concept or Role) is defined with several mapping assertions, denoted ma_i over sources of $\mathcal{S}$. In order to avoid inconsistencies, the end-user defines a total order relation, denoted $>_A$, on the distinguished variables of mapping assertions of this element. For two mapping assertions ma_1 and ma_2 and respective attributes set (a_1, b_1, c_1) and (a_2, b_2, c_2) where c_i corresponds to an identifier (mapped to dbom:id), if $ma_1.a_1 >_A ma_2.a_2$ and $ma_2.b_2 >_A ma_1.b_1$ then*

- *all objects retrieved from a mapping assertion with no counter parts in other mapping assertions are created in the ABox.*
- *all objects that could be retrieved from both mapping assertions, are composed by relating a_1 from ma_1 and b_2 from ma_2 to respective data type properties.*

2.4.3 Full-preferences

In our extended mappings, both preferences can coexist within a mapping and we call *Full-preferences* the union of *R-preferences* and *A-preferences*. In order to ease the definition of mapping assertions, we have implemented a Graphical User Interface (GUI) that enables users to set the *R-preference* of a mapping assertions as the default value for all *A-preferences* of this same mapping assertion. The end-user can later overrule all the *A-preferences* she wants.

In the context of *Full-preferences*, the complexity of computing all solutions for a set S of sources and A of attributes is related to the number of possible permutations when the order of values matters and a value can be selected several times in a rearrangement. The number of such permutations is: $\| S \|^{\|A\|}$. Although still proposing an intractable solution, we reduce the complexity of the problem to $2^{\|S\|}$ in the worst case and propose heuristics which makes the issue of computing *Full-preferences* over practical integration cases relatively efficient. In [Cur08] we proposed an algorithm that searches for attribute permutations with respect to existing overlapping of the sources. In this algorithm, subsets are returned in descending order on their size and the empty set terminates the set of candidate solutions. A heuristic first returns the singleton sets and then it analyzes if *A-preferences* do not overrule *R-preferences* such that the empty set can be returned more efficiently, without considering all subsets. Otherwise

other heuristics return sets composed of sources with the highest *R-preferences* first. In order to improve our system's performance when instantiating the ABox, we use parameterized queries which are compiled once and then re-used with different parameter values without recompilation.

2.5 Reasoning in DBOM

In order to present the reasoning functionalities included in the DBOM system, it is first important to emphasize on the semantic enrichment of the ontology elements. This enrichment is facilitated by the implementation of DBOM as a Protégé plug-in. Hence, end-users can benefit from all the standards functionalities of this ontology editor, e.g. enjoying interactions with the built in OWLClasses, Properties and Instances tabs, interfacing with reasoners, benefiting from visualization tools.

In Figure 2.3, five distinct areas are highlighted in a DBOM window. The left hand side of this window is concerned with sources management and ontology representation while the right hand side deals with mappings.

The upper left zone (1) supports the addition and removal of relational database sources. Just below this area (2), database(s) and ontology schemata are presented. The upper section (3) supports the creation of new ontology members (e.g. concept, properties) and displays information about individuals. In the context of the creation of a concept (resp. role), end-users can specify a super class (resp. super property). Then it is possible to enter as many mapping assertions for this new ontology member (`Request` tabs of (4)). In the example of Figure 2.3, a `RespiratorySystemDrug` concept has been created and area (4) defines a mapping assertion (taking the form of an SQL query) defined on source 1.

Note that by default, the confidence value is set to 50% (corresponding to the default of R-preferences). Then each database attribute of the SELECT clause is mapped to an ontology property or the `dbom:id` identification system in are (5). Note that attributes `dbom:id` logically have no preference value. A default value of 50% is set to A-preferences for all attributes not mapped to `dbom:id`. End-users have the ability to refine the A-preference value of each mapped attribute and thus overrule the R-preference.

All ontology members created with a DBOM mapping are integrated in the standard Protégé tabs (OWLClasses or Property tabs). For instance, the `RespiratorySystemDrug` concept of Figure 2.3 is inserted in the tree structure displayed in the OWLClasses tab. Then it is possible to add DL axioms to this concept, e.g. `RespiratorySystemDrug` $\sqsubseteq$ Drug $\sqcap \exists$ treat.RespiratorySystem. Or for a property to specify its domain, range as well as characteristics as functional, inverse functional, etc.

At this stage of the presentation of DBOM, reasoning already plays an important role. Via interactions with standard DL reasoners (e.g. Pellet, Racer or HermiT), one can compute concept subsumption, define and check the consistency of the defined ontology. But we will present in the next section two other cases requiring inferences.

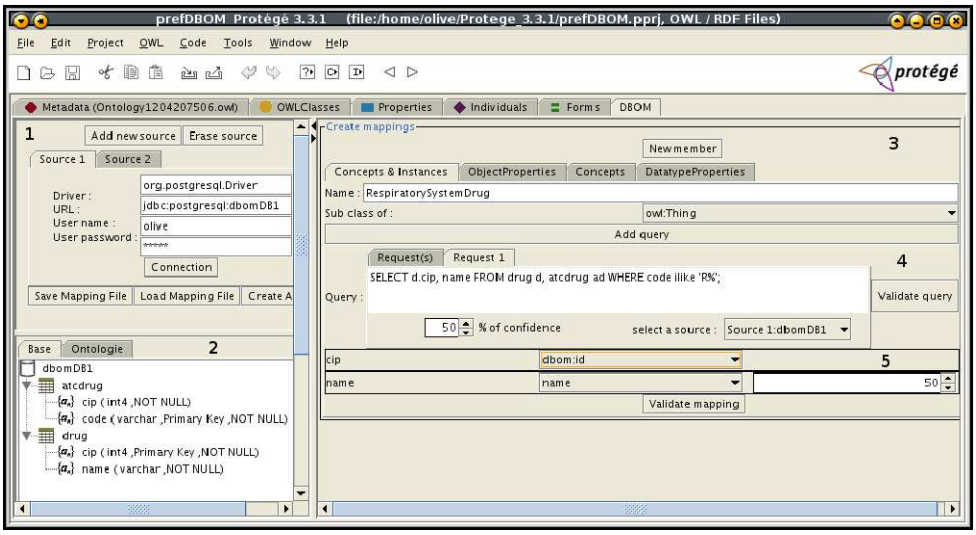


Figure 2.3: Screenshot of the DBOM Protégé plug-in

2.5.1 Inference-based instantiation of knowledge bases

In order to generate a consistent knowledge base from a set of mapping assertions and database sources, DBOM performs multiple inferences. They correspond to checking the type of concepts involved in property assertions with respect to the domain and range defined in the ontology. In the background of this approach, standard DL reasoning procedures are used, e.g. concept subsumption, instance checking (i.e. check whether an assertion is entailed by an ABox) and realization (given an individual and a set of concepts, provide the most specific concepts this individual is an instance of). In the following example, we present a concrete situation where inferences are needed.

Example 2.5.1 *Consider the following ontology extract:*

1. $\texttt{RespiratorySystemDrug} \sqsubseteq \texttt{Drug}$
2. $\texttt{NervousSystemDrug} \sqsubseteq \texttt{Drug}$
3. $\texttt{RespiratorySystemDrug} \sqsubseteq \neg\ \texttt{NervousSystemDrug}$
4. $\top \sqsubseteq \forall\ \texttt{hasContraIndication}.\texttt{Drug}$
5. $\top \sqsubseteq \forall\ \texttt{hasContraIndication}^{-1}.\texttt{Drug}$

Where line (1) defines a $\texttt{RespiratorySystemDrug}$ *concept being a subconcept of the* $\texttt{Drug}$ *concept (resp. on line (2) with* $\texttt{NervousSystemDrug}$*). Line (3) states that the* $\texttt{RespiratorySystemDrug}$ *and* $\texttt{NervousSystemDrug}$ *concepts are disjoint. Finally lines (4) and (5) define the domain and range of the* $\texttt{hasContraIndication}$ *object property.*

Given a set of mapping assertions, DBOM proceeds as follows: mappings concerned with concepts are treated first then follows mapping concerned with

object properties. This approach enables us to first generate individuals and then to related them with some roles. This means that in the case of Example 2.5.1, `RespiratorySystemDrug` and `NervousSystemDrug` instances are first created, designing a graph of nodes corresponding to individuals, concepts and literals. Finally, the mapping assertion concerning the `hasContraIndication` property states that the domain and range of this property are the `Drug` concept. But the current graph does not contain any explicit instances of this concept. Hence reasoning other the concept hierarchy of the domain and range concepts are needed to relate existing individuals of the graph, i.e. instances of the `RespiratorySystemDrug` and `NervousSystemDrug` concepts.

2.5.2 Incremental generation from previous mapping assertion

An important problem encountered in data integration systems corresponds to the generation of schema mapping. The problem of discovering such schema mappings is known to be a complex one which can be addressed with practical heuristics and approximation algorithms. We have proposed a method to derive new mappings from previously defined ones with constant values. In order to perform these derivations, we also exploit constraints and data associated to the sources. Additionally our solution generates labels for the elements of the mediated (target) schema. This approach is motivated by our experience in designing fined-grained ontologies (e.g. in medicine, biology, archeology, ecology) which mapping assertions usually involve constant values present in the database sources.

Mapping assertion generation Intuitively, our method takes a limited set of user generated mappings as an input and derives new mappings based on uncovered constants of database instances. The effectiveness of this mapping generation depends on the semantic information associated to the constant values. We consider that in databases containing classifications or terminologies (e.g. scientific domains), this semantic information is valuable and not exploited enough.

Our approach requires to distinguish between two kinds of mappings denoted M_d and M_u.

Definition 6 *: Let M_u denote the mappings that have been defined by **end-users** or created from a computer system. We denote by M_d the set of mappings **derived** by our solution. We state that $M = M_d \cup M_u$ and $M_d \cap M_u = \emptyset$.*

For a mapping M, we denote with *LHS(M)*, the left hand-side of a mapping M, i.e. the (SQL) query over the sources, and denote by *RHS(M)*, the right hand-side of M, i.e. the expression over elements of the ontology. In this work, we restrict *LHS(M)* to SPJ (Select Project Join) queries with equalities. For instance, we do not consider aggregation functions or GROUP BY in *LHS(M)*. Our system exploits meta data of the databases in order to derive new mappings. This information corresponds to the most commonly used constraints over the relational model: primary and foreign key constraints.

Another aspect that needs to be considered is equivalence of conjunctive queries which plays an important role in discovering constant values with high semantic information. In fact, two queries LHS(M_1) and LHS(M_2) are equivalent if their answer sets are identical for a given instance of the database.

Example 2.5.2 *Consider two equivalent queries expressed over the database schema of example 2.2.1:*

Q_1 : SELECT d.code, d.name, d.price FROM drug d, ephDrug ed, ephmra e WHERE ed.drugCode=d.code AND ed.ephmracode=e.code AND e.name LIKE 'Plain antitussives';

Q_2 : SELECT d.code, d.name, d.price FROM drug d, ephDrug ed WHERE ed.drugCode=d.code AND ed.ephmraCode LIKE 'R5D1';

Both SQL queries present a constant, i.e. 'Plain antitussives' in Q_1 and 'R5D1' in Q_2, which does not have the same 'direct' semantic information. But in the context of the query and database schema, we can associate an equivalent semantic information to 'R5D1' and 'Plain antitussives' since both columns are related by a functional dependency. Similar equivalence relationships can be discovered using foreign key constraints.

We now present a structure Ω which stores the information associated to the occurrence of constant values in the conjunctive query of a mapping.

Definition 7 *Ω corresponds to an ordered set of triples $\{\omega_1, .., \omega_n\}$ where ω_i is defined as a triple $\langle R_j, A_k, c\rangle$. In this triple, R_j corresponds to a relation of the source, A_k is an attribute of R_j with $1 \leqslant k \leqslant |R_j|$ (arity of the relation R_j), and c is the constant value associated to A_k in the conjunctive query and thus belongs to Δ.*

In our approach, we are interested in computing such structures for both M_u and M_d and thus denote them respectively with Ω_u and Ω_d. Once we have generated the Ω_u from a given *LHS(M_u)*, we start a second step aiming to derive new mappings. The method searches for new constant values for Ω_d which are different from the ones used in Ω_u. In order to find these values, we execute an aggregation query based on *LHS(M_u)*. This query returns the number of instances retrieved for a rewriting of *LHS(M_u)* for each constant value discovered and sorts the results in descending order on the number of instances of these groups. This ordering is justified by the assumption that concepts or roles with the most instance assertions should be the most pertinent in the context of creating an ontology. The aggregation for *LHS(M_u)* of Q_1 and Q_2 correspond respectively to the following SQL queries:

(1) SELECT name, count(*) FROM ephmra GROUP BY name HAVING name NOT LIKE 'Plain antitussives' ORDER BY count(*) DESC;

(2) SELECT ephmracode, count(*) FROM ephmra GROUP BY ephmracode HAVING ephmracode NOT LIKE 'R5D1' ORDER BY count(*) DESC;

Queries with $|\Omega_u| \geqslant 2$, i.e. the size of the set Ω_u, may retrieve very large sets of candidate values, i.e. based on the cartesian product of all constant values of Δ for attributes in Ω_u. In order to minimize the number of irrelevant mappings generated, our system interacts with the end-user to select relevant attributes. Basically, for a given mapping, it enables an end-user to select a subset of relations and attributes of Ω_u for which new constants should be searched.

The interaction with the end-user is handled by a GUI taking the form of a list of relation, $\Omega_u.R_j$, and attribute, $\Omega_u.A_k$ couples. Each entry of the list is associated to a check box component. This enables users to select/deselect attributes in an effective way. The GUI also interacts with the end-user's selection to display the number of mappings that will be automatically generated if this

selection is validated. Finally, when $|\Omega_u| = 0$, the system does not search for new mappings as the *LHS(M_u)* already retrieves all instances of a given relation R_i of S.

The results obtained from the execution of queries (1) and (2) on our database extract, respectively enable to generate the following *LHS(M_d)*:

(3) SELECT d.code, d.name, d.price FROM drug d, ephDrug ed, ephmra e WHERE ed.drugCode=d.code AND ed.ephmracode=e.code AND e.name LIKE 'Antitussives combinations';

(4) SELECT d.code, d.name, d.price FROM drug d, ephDrug ed WHERE ed.drugCode=d.code AND ed.ephmraCode LIKE 'R5D2';

Both queries are being generated using a rewriting of *LHS(M_u)* by substituting constant values of Ω_u with the new constant values of Ω_d. We now consider these M_d queries as candidate queries for ontology mappings.

Label generation In cases a M_u mapping introduces new ontology elements, e.g. 'PlainAntitussive' concept in M_1, then it is necessary to search for relevant ontology element labels for M_d. For instance what *RHS(M_d)* can we associate to the SQL queries (3) or (4)?

In order to perform this task, the system searches for relationships between the constants in *LHS(M_u)* and the ontology elements introduced in *RHS(M_u)*. Then it applies these relationships on the *LHS(M_d)* queries to discover ontology element labels to complete the *RHS(M_d)* queries.

We distinguish two main relationships between constants: (i) lexicographic equivalence, i.e. a constant and an ontology element correspond to the same string. We also consider some forms of simple lexicographic transformations and concatenation over Ω_u constant values. (ii) non lexicographic equivalence, i.e. a constant value from the SQL query does not correspond to any label of the ontology elements. In this situation, we are searching for other forms of equivalence, e.g. synonymy or hyponymy.

When an equivalence relationship is detected, then the symbol for the ontology element is obvious and can be computed from the constant value. In our running example, a lexicographic equivalence is detected in mapping M_1. Thus we are able to easily propose a label for the concept of the derived mappings associated to the LHS queries (3) and (4): *AntitussiveCombination*.

We adopt the same approach if multiple constants appear in M_u (i.e. $|\Omega_u| \geqslant 2$). That is if one of the constant values in Ω_u equals one of *LHS(M_u)*, then we mark its triple (R_k, A_n, c) and for all derived *LHS(M_d)*, we set the ontology element symbol with the constant value of the marked Ω_d.

When no lexicographic equivalence holds, it is necessary to discover a relationship between some or all of the constant values and the ontology element labels in the end-user mapping. Once these relations have been discovered, we can apply them to the corresponding constants of M_d and find labels for the associated ontology elements. In order to perform this discovery, we exploit information coming from some background knowledge, e.g. WordNet (an electronic lexical database). Using WordNet, we consider only nouns as label candidates and we use the Java WordNet Library (http://jwordnet.sourceforge.net) to recognize variants of the same noun obtained by applying the usual inflection rules. Our method considers WordNet as a graph where vertices correspond to nouns or synsets. The edges between these nodes are relations of the kind hyponyms, hypernyms, meronyms, etc.. Using this graph representation, it is possible to navigate through this background knowledge and find relations between nodes.

The labeling solution is decomposed into two algorithms: `getRelation` and `getLabel` which are detailed below.

In a nutshell, the `getRelation` algorithm accesses a list of nouns that match the labels of the newly introduced elements of *RHS(M_u)*. For instance 11 synsets are available for the noun 'Man'. Then the algorithm uses all the information available from Ω_u, i.e. relation, attribute names and constants, to find the most appropriate synset. The selection of the most appropriate one is performed using a score function. Once a synset has been selected, we need to search for the most appropriate relations, i.e. hypernym, hyponym, meronym,etc., available. Again, this is done using a function that scores for each relation the number of matches with the terms of Ω_u. Finally the most relevant relation is returned. Several heuristics are added to our `getRelation` algorithm. For instance, if no hypernyms, hyponyms or meronyms, etc. are found using WordNet, then we return a '`null`' relation which implies that constant values of Ω_u are proposed as ontology labels. Also, if the scores of all or some the hypernyms, hyponyms, meronyms, etc. relations are equal then we return the hyponym relation.

Once we have characterized the relation between the ontology symbol and the query of M_u, we use this relation with the constant values of M_d to find a set of symbol candidates for M_d. This task is performed by the `getLabel` algorithm for each end-user selected Ω_u element. The inputs of this algorithm are the Ω_d elements and the WordNet relation returned from `getRelation`. A first step is to retrieve a set of words corresponding to the constant value of the Ω_d element. Then the most adequate noun is selected and a set of synsets is retrieved from WordNet based on the selected noun and the previously discovered WordNet relation. Then synsets are rated in a way similar to the `getRelation` algorithm, i.e. using a score function that searches for matches between synset descriptions, labels and elements of Ω_d triples. Finally a set of synset labels is returned by the `getLabel` function. In cases where several constant values have been selected by the end-user, several execution, one for each selected Ω_d triple selected, is performed and the set of candidate labels then corresponds to the union of the returned labels.

Mapping selection and refinement Once a derived mapping M_d is completed, i.e. both the LHS and RHS queries have been generated, the next logical operation is to make it persistent in the integration system. Basically, this is performed by storing the derived mapping in a mapping repository and recording the new ontology elements in the TBox. But before performing these operations, we need to make sure that the derived mappings satisfy the end-user's intention. The consideration of this aspect prevents the system from generating large TBoxes where only a subset of the concepts and roles are relevant to the ontology designer. This is the case if one wants to design an ontology that only considers a restricted part of the domain of a database, e.g. design an over-the-counter drug ontology from a complete drug database. Hence it is necessary to consider only a relevant subset of the mappings that can be effectively derived. Basically, this is usually a consequence of an under restricting of M_u's SQL query. In order to correct this situation, the only reliable source of information is the end-user.

We propose a user-friendly GUI (Figure 2.4) for the acceptance of mappings which permits an effective scan and selection of candidate mappings as well as an easy solution to refine SQL queries associated to these mappings.

This GUI highlights:

1. a pattern of LHS SQL query of the mappings where all constant values are substituted by a symbol (Cx).

2. an evaluation of the number of mapping to process and number of emerging concepts and roles.

3. a simplified view of mappings via pairs consisting of a set of constant values and a set of ontology element labels.

4. a check box for each mapping view which allows to select a mapping.

5. text areas where the end-user can enrich the FROM and WHERE clauses of the aggregation SQL query responsible for finding constant values.

Based on this GUI, it is clear that the system proposes two different approaches to refine, adopt or reject a set of mappings. A first approach consists in selecting, via the check box, a set of satisfying mappings. This approach is useful when the number of proposed mappings is low, in practice we have studied that the upper bound is about thirty mappings. We have seen in the previous sections that our solution can easily derive hundreds or thousands mappings from certain classifications. In this situation, it is not manageable for the end-user to manually check such an amount of check boxes. Hence, we propose a text area enabling to restrict the set of discovered constant values. This is performed by introducing new tables in the FROM clause and conditions in the WHERE clause of the aggregation SQL query. The main drawback of this approach is to require that end-users are able to write SQL queries and know well the database schema.

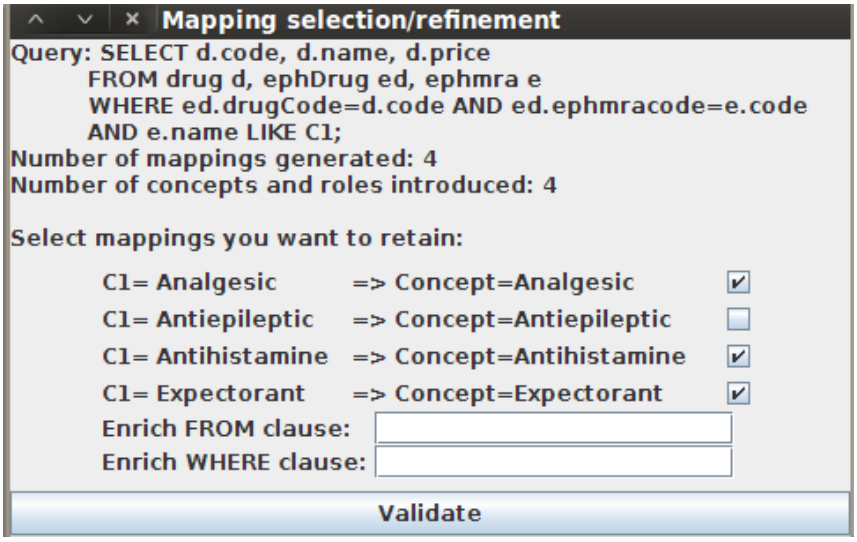


Figure 2.4: DBOM's Mapping selection and refinement GUI

2.6 Presentation of papers

[Cur05b] **Olivier Curé**. Semi-automatic data migration in a self-medication knowledge-based system.
In Wissensmanagement (LNCS Volume),pages 373–383, 2005.

This article presented at the Knowledge Management in Medicine session at WM 2005 motivates the idea of representing some of the information of our self-medical application, namely XIMSA, with ontologies. The paper details the architecture and functionalities provided in XIMSA (e.g. the Simplified Electronic Healthcare Record (SEHR) and the diagnosis module). Then the overall architecture of DBOM is discussed in the context of this medical application.

[CS05] **Olivier Curé** and Raphael Squelbut. A database trigger strategy to maintain knowledge bases developed via data migration.
In EPIA, pages 206–217, 2005.

This paper motivates the approach of using a trigger based strategy to synchronize data states between a set of relational databases and a knowledge base. A survey of the most influential system in the field of mapping databases and ontologies is provided and enables us to perform several comparisons with DBOM. Finally a formal definition of the trigger strategy is presented with details on applied heuristics and implementation information.

[CS06a] **Olivier Curé** and Raphael Squelbut. Data integration targeting a drug related knowledge base.
In EDBT Workshops, pages 411–422, 2006.

This article was presented at the Information In Healthcare (IIHA) workshop at the EDBT conference. It focuses on methods defined in XIMSA's Drug Consumption Checker Application (DCCA). This service, aimed at the general public, controls the adequacy of a drug (self) prescription. Thus the architecture of the DCCA service is supported by a drug ontology and a Simplified Electronic Health Record (SEHR). DBOM's syntax and semantics is introduced in order to design and maintain the different knowledge bases involved in this task.

[CS06b] **Olivier Curé** and Raphael Squelbut. Integrating data into an OWL knowledge base via the DBOM protégé plug-in.
In 9th International Protégé conference, 2006.

Protégé is an open-source ontology and knowledge base framework developed at the Stanford Center for Biomedical Informatics Research and is considered as the leading project in this field. The architecture of this editor encourages the implementation of plug-ins and a conference. This conference is held annually to bring together researchers developing or using Protégé methodologies and tools. In this context, we demonstrated our DBOM plug-in as well as presented details on its implementation and the way it interacts with the standard component of the editor (e.g. interactions with OWLClasses, Properties, etc.).

[CJ07c] **Olivier Curé** and Florent Jochaud. Preference-based integration of relational databases into a description logic.
In DEXA, pages 854–863, 2007.

This article motivates the need to deal with uncertainties emerging from the mapping assertions of DBOM when several database sources are used to instantiate DL concepts or properties. In this paper, only query level preferences (*R-preferences*) are introduced.

[CB08] **Olivier Curé** and Jean-David Bensaid. Integration of relational databases into OWL knowledge bases: demonstration of the DBOM system.
In ICDE Workshops, pages 230–233, 2008.

This demonstration paper was presented at Information Integration Methods, Architectures, and Systems (IIMAS) at the ICDE conference. It emphasizes all the functionalities presented in this chapter. That is the ability to map several database sources with *Full preferences*, reasoning in the context of instantiating a knowledge base, generation of triggers on the sources on the different preference forms available in DBOM and on the associated reasoning mechanisms.

[Cur08] **Olivier Curé**. Data integration for the semantic web with full preferences.
In ONISW, pages 61–68, 2008.

This paper extends the R-preference approach presented in [CJ07c] with *A-preferences* and presents the algorithms necessary to process *Full preferences*, the union of *R preferences* and *A preferences*. This paper also evaluates the efficiency of these algorithms and the satisfaction of end-users.

[Cur09c] **Olivier Curé**. Incremental generation of mappings in an ontology-based data access context.
In OTM Conferences (2), pages 1025–1032, 2009.

This paper presents the incremental generation of mappings approach. All steps of this generation are detailed and algorithms are proposed in the context of Ontology-Based Data Access (OBDA). OBDA provides access to a set of (relational) data sources through a mediating ontology, which acts as a semantic layer between the user and the data. Hence OBDA proposes a mapping solution that has some similarities with DBOM. Anyhow, a main difference between these two approaches is that DBOM, via its materialization of the target knowledge base can be considered a hybrid solution between data integration and exchange system while OBDA is a strict data integration system. Nevertheless, the incremental generation of mappings is easily translatable to the DBOM context.

2.7 Conclusion and perspectives

The DBOM system proved to be quite useful on many projects we have been working on. Obviously, it has been the case on the XIMSA application which integrates several ontologies from classifications contained in databases. But it has

also been the case on biology and ecology projects experimented with the DaltOn framework (Chapter 3), geographical information on ANR STAMP (Chapter 5) and several on-going projects developed in the Géomatique, Télédétection et Modélisation des Connaissances (GTMC - Geomatics, Remote sensing and Knowledge modeling) research team.

Future works on the DBOM system include the support for a dual materialization/virtualization of the ABox, introducing novel approaches to discover mapping assertions and proposing a data provenance tool associated to ontologies generated through DBOM mapping. We now detail each of these research perspectives.

The former perspective will enable a given end-user to choose between a virtualization, i.e. a real data integration approach where the data are kept in the database sources and retrieved only at query-runtime (freeing the system from any synchronization tasks) and the current materialization approach, i.e. the knowledge is stored locally on the system and a synchronization approach (e.g. Section 2.3) is required to get up-to-date information.

Concerning the second research perspective, we consider that the number of available ontologies, databases and mappings of all sorts can be analyzed and used in order to propose a set of candidate mappings. Currently, we do not believe that this process can be fully automatized but a semi-automatic approach (e.g. in the direction of the work presented in the 2.5 section) where the end-user selects and refines relevant mapping assertions is realistic and may prove to be quite useful in practical cases.

The provenance aspect is becoming a priority in many data and meta data management systems. We believe that with its preference-based mapping assertion solution and trigger-based synchronization approach, DBOM possesses valuable data and meta-data to provide an efficient and user-friendly data provenance solution. Hence future works will address this issue.

Chapter 3

Ontology mediation

We have already emphasized that the number of ontologies available on the Web is important and constantly growing. An important portion of available ontologies are the results of initiatives that propose storage, browsing and retrieval solutions for thousands of ontologies and billions of RDF triples, e.g. Tones[1], the Open ontology Repository (OOR)[2] and LinkedData[3]. But these vocabularies cannot reply to everyone's need and it is frequently the case that application designers generate their own ontologies. These ontologies may be created from scratch or by an alignment and extension from existing ontologies. A direct consequence of this situation is that many ontologies coexist in some specific domains, e.g. geographical information and medicine.

With so many ontologies being produced, it is unavoidable that some of their content overlap and possibly disagree. In many scenarios, it is required to semantically relate these ontologies. Thus ontology mediation [Ehr07] becomes a main concern. Ontology mediation enables one to share data between heterogeneous knowledge bases, and allows applications to reuse data from different knowledge bases. Ontology mediation takes two distinguished forms: (i) ontology mapping, where the correspondences between elements of two ontologies are stored separately from the ontologies. The correspondences are generally represented using axioms formulated in a peculiar mapping language. (ii) ontology merging, which consists in creating a new ontology from the union of source ontologies. The merged ontology is supposed to capture all the knowledge of the sources.

Ontology mediation is an active research field where many kinds of solutions have been proposed: schema-based, instance-based, machine learning-inspired, hybrid approaches; see [KS03], [ES07] for surveys on this domain. Thus the methods used in ontology mediation usually depend on the kind of information one can access about the local ontologies. For instance, the availability of instance datasets are highly desirable. But the efficiency of these methods also depends on the kind of source ontologies the system is dealing with. In terms of expressiveness, [McG03] presents an ontology spectrum characterizing different forms of ontology, i.e. ranging for the Entity Relationship model to expressive logical formalisms.

[1] http://owl.cs.manchester.ac.uk/repository/
[2] http://openontologyrepository.org
[3] http://linkeddata.org/home

In this chapter, we present two approaches related to ontology mediation where ontologies correspond to expressive Description Logics and can be serialized in OWL2. The organization is the following: in Section 3.1, a Formal Concept Analysis approach is proposed to merge such ontologies and in Section 3.2 we present an ontology mapping based component designed within a process modeling system that supports data transformation and transport from one process to another.

3.1 A Formal Concept Analysis approach to merge ontologies

In this section, we present a solution to the ontology merging problem which is based on the techniques of Formal Concept Analysis (FCA) [GW99]. As a machine learning technique, FCA is the process of abstracting conceptual descriptions from a set of objects described by attributes and hence enables us to reveal some associations between elements of the original structures. These associations can be considered as alignments between a set of ontologies and can thus be used to merge them. Intuitively, this means that we merge several ontologies in a context consisting of a set of objects (the extent), a set of attributes (the intent), one for each ontology, and a set of correspondences between objects and attributes.

Our solution extends existing FCA-based systems (e.g. [SM01]) for ontology merging in the following way:

1. we provide a method to create concepts not originally in the source ontologies,

2. we define emerging concepts in terms of concepts and roles of the source ontologies and

3. we handle the creation of merged ontologies based on the uncertainty underlying the extension and alignment of source concepts.

The step (1) is the classical approach named *ontology alignment* in FCA literature. The steps (2) and (3) are an extension of this alignment and exploit concept descriptions, DL reasoner functionalities and notions from possibility theory [DD04].

In order to understand the details of our approach, some notions need to be introduced. FCA is based on the notion of a *formal context*.

Definition 8 *A formal context is a triple $\mathcal{K} = (G, M, I)$, where G is a set of objects, M is a set of attributes and I is a binary relation between G and M, i.e. $I \subseteq G \times M$. For an object g and an attribute m, $(g, m) \in I$ is read as "object g has attribute m".*

Given a formal context, we can define the notion of formal concepts:

Definition 9 *For $A \subseteq G$, we define $A' = \{m \in M | \forall g \in A : (g, m) \in I\}$ and for $B \subseteq M$, we define $B' = \{g \in G | \forall m \in B : (g, m) \in I\}$. A formal concept of $\mathcal{K}$ is defined as a pair (A, B) with $A \subseteq G$, $B \subseteq M$, $A' = B$ and $B' = A$.*

The hierarchy of formal concepts is formalized by $(A_1, B_1) \leq (A_2, B_2) \iff A_1 \subseteq A_2$ and $B_2 \subseteq B_1$. The concept lattice of $\mathcal{K}$ is the set of all its formal concepts with the partial order $\leq$. This hierarchy of formal concepts obeys the mathematical axioms defining a lattice, and is called a concept lattice (or Galois lattice) since the relation between the sets of objects and attributes is a Galois connection.

We now introduce the notion of Galois connection [Bir73] which is related to the idea of order and plays an important role in lattice theory, universal algebras and recently in computer science. Let $(P, \preceq)$ and $(Q, \preceq)$ be two partially ordered sets (poset). A Galois connection between P and Q is a pair of mappings (Φ, Ψ) such that $\Phi : P \to Q$, $\Psi : Q \to P$ and:

- $x \preceq x'$ implies $\Phi(x) \succeq \Phi(x')$,
- $y \preceq y'$ implies $\Psi(y) \succeq \Psi(y')$,
- $x \preceq \Psi(\Phi(x))$ and $y \preceq \Psi(\Phi(y))$

for x,x' $\in$ P and y,y' $\in$ Q.

Several algorithms have been proposed to compute a concept lattice, some optimized ones are proposed in [Cho06]. Intuitively, such an algorithm starts with the complete lattice of the power set of all individuals (the extent), respectively for attributes (the intent) and retains only the nodes closed under the connection. That is beginning with a set of attributes, the algorithm determines the corresponding set of objects which itself provides an associated set of attributes. If this set is the initial one, then it is closed and preserved otherwise the node is removed from the lattice.

In this section, we present the process of merging two source ontologies. In fact, the method can be used for several ontologies as long as these ontologies share elements of their datasets. That is ABoxes of these ontologies contain assertions about the same objects.

3.1.1 Source TBoxes

Let us consider two geographical applications that manipulate space parcel data. Each application uses an independent ontology formalism to represent the concepts related to its data. Also the teams of experts that designed each ontology may not agree on the semantics of some concepts. Nevertheless, the two applications need to exchange information, and thus require that some correspondences are discovered between their DL concepts.

Example 3.1.1 *The following two ontology extracts, O_1 and O_2, are used all along this section. In order to ease the understanding and reading of our example, all concepts and roles are under scripted with the number of their respective ontology, i.e. '1' for O_1 and '2' for O_2.*

Terminological axioms of ontology O_1:

1. $CF_1 \equiv F_1 \sqcap \exists vegetation_1.C_1$
2. $BLF_1 \equiv F_1 \sqcap \exists vegetation_1.M_1$
3. $C_1 \sqcap M_1 \sqsubseteq \bot$

This extract of ontology O_1 defines two concepts, CF_1, standing for Coniferous Forest, and BLF_1, standing for Broad Leaved Forest, as well as the following concepts: F_1 (Forest), C_1 (Coniferophyta) and M_1 (Magnoliophyta). Line #1 states that the coniferous forest concept is defined as the intersection of the concept Forest of O_1 and the concept having at least one vegetation being a coniferophyta. Line #2 defines the concept of a broad leaved forest accordingly with magnoliophyta. Line #3 states that the concepts coniferophyta and magnoliophyta are disjoint.

Terminological axioms of ontology O_2:

4. $CF_2 \equiv F_2 \sqcap \forall vegetation_2.C_2 \sqcap \ \exists vegetation_2.C_2$
5. $BLF_2 \equiv F_2 \sqcap \forall vegetation_2.M_2 \sqcap \ \exists vegetation_2.M_2$
6. $MF_2 \equiv F_2 \sqcap \exists vegetation_2.C_2 \sqcap \ \exists vegetation_2.M_2$
7. $C_2 \sqcap M_2 \sqsubseteq \bot$

The study of O_2 emphasizes that designers do not entirely agree on the semantics of forest related concepts of O_1. On line #4, the concept of a coniferous forest is defined has being a forest composed of at least coniferophyta vegetation and exclusively of this kind of vegetation. Line #5 defines the concept of broad leaved forest accordingly with magnoliophyta. In order to represent other kinds of forests, the designers of O_2 define a mixed forest concept as the intersection of being a forest with at least one coniferophyta vegetation and at least one magnoliophyta vegetation. Finally Line #8 states that the concepts coniferophyta and magnoliophyta of O_2 are disjoint.

Merging the ontologies O_1 and O_2 with some other ontologies would require that the TBoxes for these new ontologies are available and are no more expressive than $\mathcal{ALC}$.

3.1.2 Source ABoxes

Given the kind of TBoxes presented in the previous section, e.g. $\mathcal{ALC}$, we consider DL knowledge bases with non-empty ABoxes. In a first step, we map the information of the two ABoxes on a common set of observed objects.

The information of these ABoxes can be stored in a structured or unstructured format. It is interesting to note that the activity of several research teams in the DL and Semantic Web community focuses on studying cooperations between the domains of databases and knowledge bases represented in a DL. For instance, the authors of [PLC+08] recently claimed that the ideal solution would be to have the individuals of the ABox stored in a relational database and represent the schema of this database in a DL TBox. Also tackling this same objective, the team supporting the Pellet reasoner, one of the most popular OWL reasoners, recently released *OWLgres* which is being defined by their creators as a 'scalable reasoner for OWL2' (the latest version of the OWL). A main objective of this tool is to provide a conjunctive query answering service using SPARQL and the performance properties of relational database management systems. Hence, using such an approach, the set of observed objects may be retrieved from existing relational database instances.

Table 3.1: Sample dataset for our ontology merging example

	CF_1	BLF_1	F_1	CF_2	BLF_2	MF_2	F_2
1	x		x	x			x
2	x		x	x			x
3	x		x			x	x
4		x	x		x		x
5		x	x		x		x
6		x	x			x	x

The mapping we propose between both ontologies can be represented by a *matrix*, either generated by a specific tool and/or by interactions with end-users. In order to map concepts of both ontologies via the selected set of observed objects, a reference reconciliation tool may be used [DHM05]. Using an approach that exploits a relational database as the data container for the ontology ABox enables to use existing FCA tools. This is the case of the ToscanaJ suite [4] which provides features for database connectivity.

We present a sample of this mapping in Table 3.1: the rows correspond to the objects of $\mathcal{K}$, i.e. common instances of the KB's ABox, and are identified by integer values from 1 to 6 in our example. In the context of geographical information, these values identify spatial parcels. The columns correspond to FCA attributes of $\mathcal{K}$, i.e. concept names of the two TBoxes. In the same table, we present, side by side, the formal concepts coming from our two ontologies, i.e. CF_1, BLF_1, F_1 from O_1, and CF_2, BLF_2, MF_2, F_2 from O_2. Thus this matrix characterizes the type of spatial parcels in terms of two different ontologies.

Merging more than two ontologies would require that the individuals of the ABox belong to the extension of the concepts of these ontologies. That is concepts from a third ontology can be added to the columns of Table 3.1 and objects of the ABox (rows of the table) are instances of these new concepts.

3.1.3 Generation of the Galois connection lattice

The matrix is built using the information stored in the TBox and ABox of both ontologies:

- first, for each row, mark the columns where a specific instance is observed, e.g. the object on line #1 is an instance of the CF_1 and CF_2 concepts. Thus ABox information is used in this step.
- then, complete the row with the transitive closure of the subsumption relation between ontology concepts, e.g.: line #1 must also be marked for DL concepts F_1 and F_2, as respective ontologies entail that: $CF_1 \sqsubseteq F_1$ and $CF_2 \sqsubseteq F_2$. Here, the concept hierarchy of TBoxes is exploited.

It is interesting to note that lines #3 and #6 emphasize different assumptions for their respective parcels. For instance, the parcel corresponding to line #3 has been defined as a coniferous forest using the classification of O_1 while, possibly due to a vegetation not limited to coniferophyta, it has been defined as a mixed

[4] http://toscanaj.sourceforge.net/

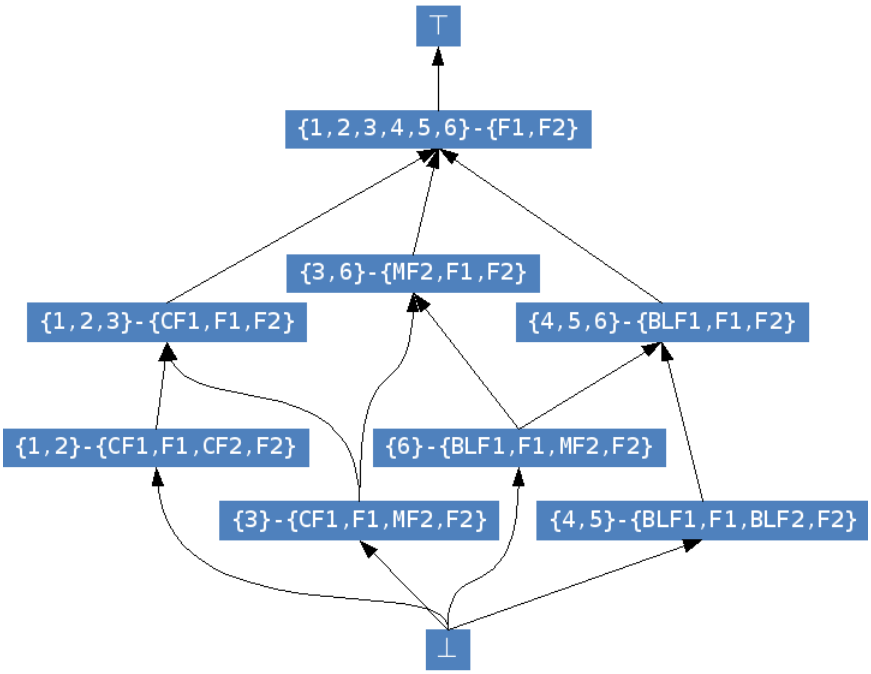


Figure 3.1: Galois connection lattice

forest using O_2. The same kind of approach applies to the parcel associated to line #6.

Using Table 3.1 with the Galois connection method [DP90], we obtain the lattice of Figure 3.1, where a node contains two sets: a set of objects (identified by the integer values of the first column of our matrix) from $\mathcal{K}$ (extension), and a set of DL concepts from the source ontologies (intension), identified by the concept labels of source ontologies.

3.1.4 Dealing with emerging concepts

In order to concentrate solely on the intensional aspect of the lattice, i.e. the TBox, we now remove the extensional part of each node of the lattice. Hence, the only set present in each node corresponds to concept names (Figure 3.2). Considering that the relationship holding between two nodes in this lattice corresponds to an inheritance property, it is possible to minimize each node's set by removing concept names that are present in an inherited node. The method we propose consists in deleting repeated occurrences of a given concept name along a path of the lattice and thus to obtain a minimal set of concept names for each node. Next, we define this notion of minimality:

Definition 10 *Given a node N in the Galois connection lattice and a set of concept symbols S contained in its intension fragment. We consider that S is minimal for N if and only if there is no S' for N such that $|S'| < |S|$, where $|S|$ denotes the size of S.*

Due to the lattice structure obtained by applying the Galois connection method, we can proceed by using a top-down navigation, i.e. starting from the top concept (Top), on the concepts of the merged ontology. Basically, this algorithm (named *optimizeLabel* and presented in Algorithm 1) proceeds as follows: for a given formal concept C of the lattice, it computes all its children c (line #1) and checks if a concept symbol used to characterize C is used in the concept name set for c (line #2). If this is the case, this symbol is removed from their set of c (line #3) otherwise the set of symbols of c remain unchanged. Finally, the method is applied recursively to each concept c until all concepts are processed (line #5).

Algorithm 1	optimizeLabel (Concept C)
1	FOR EACH child c of C DO
2	IF label(C) $\in$ label(c) THEN
3	remove label(C) from label(c)
4	END IF
5	optimizeLabel(c)
6	END DO

Processing this algorithm on our running example, yields Figure 3.2 where lattice nodes contain singleton sets, corresponding to concept names from some of the source ontologies or newly introduced symbols, e.g. α, which replace empty sets. Several kinds of nodes, in terms of the size of a name set, can be generated with this method. Basically, it is important to distinguish between the following three kinds of nodes:

1. a singleton: a name of a concept from some of the source ontologies, because it can be distinguished from any of its successors by this specific name, e.g. this is the case for the $\{CF_1\}$. lattice node.

2. an empty set, denoted by a variable (α), because it cannot be directly distinguished from any of its possible successors. We have 2 such nodes in Figure 3.2, namely α and β.

3. a set of several concept symbols, all belonging to source ontologies, because the mediation based on the given ABoxes, has not been able to split the concepts into several nodes. Indeed, it is as if the two names are glued together in a single concept name. In our running example, we have one such node with concept set $\{F_1, F_2\}$.

All singletons are maintained in the resulting merged ontology and we are now aiming to provide a concept description to the remaining concepts, case 2 and 3 of our node categorization. The first step toward our solution is to expand the concepts of the merged ontology according to their respective TBoxes. That is, we replace each occurrence of a name on the right hand-side of a definition by the concepts that it stands for. A prerequisite of this approach is that we are dealing with acyclic TBoxes. Thus this process stops and the resulting descriptions contain only primitive concepts on the right hand-side.

We first deal with the nodes which are formed of several concept symbols, denoted σ_i, e.g. the node labeled F_1, F_2 in Figure 3.2. Due to the fact that the algorithm adopted results from the generation of the Galois connection lattice, these nodes appear at the top of the lattice and do not have multiple

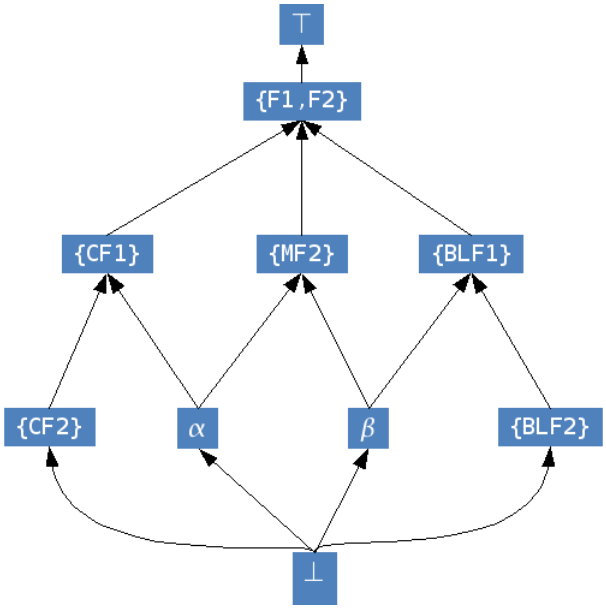


Figure 3.2: Galois connection lattice with "empty nodes"

inheritance to concepts that are not of this form. Thus we adopt a top-down approach from the top concept ($\top$) of our merged ontology. We consider that the concepts associated are equivalent, e.g. $F_1 \equiv F_2$, since they have exactly the same extension. We also propose a single concept symbol σ, e.g. F (Forest) for F_1, F_2, and associate information to this concept stating that this concept is equivalent to the original concepts for interoperability reasons, e.g. $F \approx F_1$ and $F \approx F_2$. Now all occurrences of the concept σ_i are translated into the concept symbol σ in the concept descriptions of the merged ontology.

We can now concentrate on the nodes with empty sets, e.g. α and β. According to the Galois based lattice creation, these nodes cannot be at the root of the lattice. This means that they inherit from some other concept(s). We use the description of these inherited concept(s) to provide a description. Using this method, the concepts α and β of Figure 3.2 have the following description:
$\alpha \equiv CF_1 \sqcap MF_2 \equiv F \sqcap \exists vegetation_1.C_1 \sqcap \exists vegetation_2.C_2 \sqcap \exists vegetation_2.M_2$
$\beta \equiv BLF_1 \sqcap MF_2 \equiv F \sqcap \exists vegetation_1.M_1 \sqcap \exists vegetation_2.C_2 \sqcap \exists vegetation_2.M_2$

All concepts from the merged ontology have been associated to a concept description, except of course the primitive concepts. Alignments between primitive concepts and roles of the source ontologies are able to refine the merged ontology. Later in this section, we will propose solutions to finding these alignments and dealing with their uncertainty, but we now present the impact of providing such correspondences between TBox elements.

Suppose that we are being provided the following alignments: $C_1 \equiv C_2$, $M_1 \equiv M_2$ and even $vegetation_1 \equiv vegetation_2$. So we can easily introduce some concept symbols to simplify the different equivalences:
(8) $C \equiv C_1 \equiv C_2$, $M \equiv M_1 \equiv M_2$ and $vegetation \equiv vegetation_1 \equiv vegetation_2$

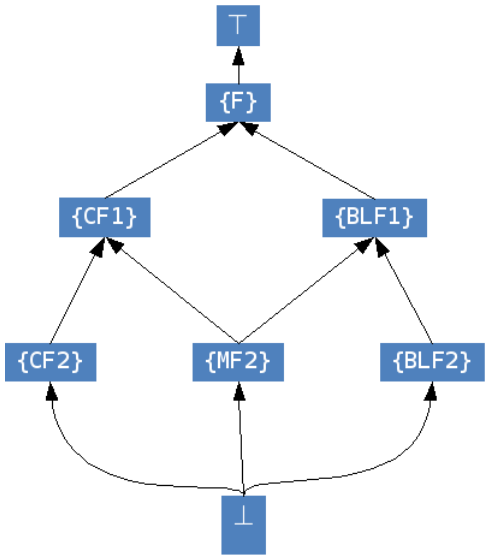


Figure 3.3: Lattice corresponding to merged ontology O_{m1}

We are then able to modify the descriptions of the merged ontology and we denote this TBox as O_{m1}:

9. $CF_1 \equiv F \sqcap \exists vegetation.C$
10. $BLF_1 \equiv F \sqcap \exists vegetation.M$
11. $CF_2 \equiv CF_1 \sqcap \forall vegetation.C \sqcap \ \exists vegetation.C$
12. $BLF_2 \equiv BLF_1 \sqcap \forall vegetation.M \sqcap \ \exists vegetation.M$
13. $MF_2 \equiv F \sqcap \exists vegetation.C \sqcap \ \exists vegetation.M$
14. $\alpha \equiv F \sqcap \exists vegetation.C \sqcap \ \exists vegetation.M$
15. $\beta \equiv F \sqcap \exists vegetation.C \sqcap \ \exists vegetation.M$
16. $C \sqcap M \sqsubseteq \bot$

We can notice that the descriptions for the concepts α, β and MF_2 are the same. Thus we can state that $MF_2 \equiv \alpha \equiv \beta$. Finding such equivalences, or subsumption relationships, is easily processed by a DL reasoner. This result is comforted by the fact that starting from the ontologies O_1 and O_2 and the alignments of (8), any DL reasoner is able to provide the ontology O_{m1}, assuming that we have the alignment $F \equiv F_1 \equiv F_2$ (which has been deduced from our Galois lattice). The lattice corresponding to this new ontology is depicted in Figure 3.3.

Of course, alignments different from (8) can be proposed between primitive concepts and roles of O_1 and O_2. For instance, if we consider the alignments in (17), then the optimized merged ontology again correspond to Figure 3.3.
(17) $M_2 \sqsubseteq M_1, C \equiv C_1 \equiv C_2$ and $vegetation \equiv vegetation_1 \equiv vegetation_2$

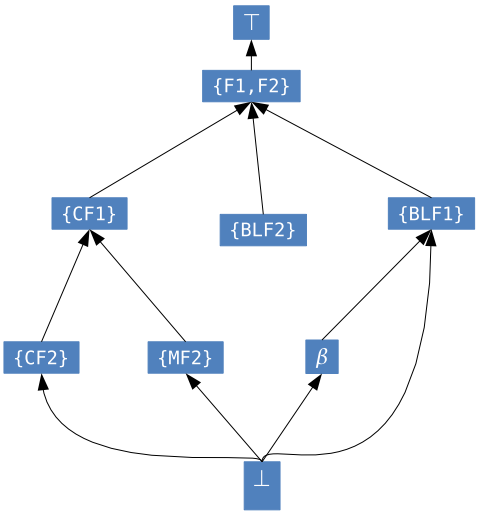


Figure 3.4: Lattice corresponding to merged ontology O_{m2}

Concentrating on the relationships between M_1 and M_2, alignments other than (8) and (17) can generate different merged ontologies. Let consider the alignments in (18) where the only difference with (8) and (17) is that now M_1 is a subconcept of M_2:
(18) $M_1 \sqsubseteq M_2, C \equiv C_1 \equiv C_2$ and $vegetation \equiv vegetation_1 \equiv vegetation_2$

Then looking at the descriptions of BLF_1 and BLF_2 (respectively (2) and (4)), we can no longer state that $BLF_1 \sqsubseteq BLF_2$. We consider that the alignments of (18) do not contradict our FCA-based method but instead refine the constructed lattice of Figure 3.2. In fact, this lattice is the result of applying a Galois connection based algorithm from a given dataset. This dataset can be considered to be a model of the merged ontology but it is only one of the possible models for this ontology. The statements in (18) say that instances of BLF_2 need not to be instances of BLF_1, a situation that was not present on our dataset (Table 3.1).

Moreover, the statements of (18) allow us to state that $MF_2 \sqsubseteq CF_1$ and $\alpha \equiv MF_2$ but prevent us from saying that $MF_2 \sqsubseteq BLF_1$. The lattice corresponding to this new merged ontology, which we denote O_{m2}, is presented in Figure 3.4.

In terms of the DL model-theoretic semantics [BCM+03], the $MF_2 \sqsubseteq \neg BLF_1$ axiom makes the ABox represented in Table 3.1 inconsistent with the merged ontology of Figure 3.4. Recall that an ABox $\mathcal{A}$ is consistent with respect to a TBox $\mathcal{T}$, if there is an interpretation that is a model of both $\mathcal{A}$ and $\mathcal{T}$. Intuitively, $MF_2 \sqsubseteq \neg BLF_1$ states that it is not possible to be an instance of both BLF_1 and MF_2 in a given model which is not case of object #6 in our dataset.

This raises the issue of the confidence one has on the existence of an object, of an alignment and to their relationship. For instance, we can have a greater confidence on the statements of (18) than on the existence of object #6. This would yield a merged ontology similar to the one presented in Figure 3.4 but without the β concept. We will provide details on the notion of confidence

values when introducing our solution to deal with uncertainties in Section 3.1.5.

In summary, we can generate different ontologies based on the fact that we are able to propose different alignments, to assign them confidence values and to assign confidence values to some objects of our sample dataset matrix. In order to provide alignments between source ontologies, we consider the following two approaches: these alignments originate from external ontologies or they are provided by the end-user.

Alignments originating from external knowledge

The alignments of primitive concepts and roles can be provided by an external knowledge source. This is in fact frequently the case when designing ontologies. Early in the design process, a background ontology, preferably recognized as a standard in the application domain, is identified and imported in the source ontologies. It is likely that the source ontologies we consider for fusion, import some common parts of a given background ontology, e.g. it can be the case in spatial information with the SWEET ontology. Then the alignment of some imported primitive concepts and roles is straightforward and less subject to some uncertainty since their interpretations are identical.

End-user defined alignments

In cases alignments cannot be provided by some background knowledge, end-users can define their own correspondences between concepts and roles. In such a situation, different end-users may provide differing alignments. Also, an end-user may not be totally confident on an alignment she is providing. This uncertainty aspect needs to be handled by the system in order to propose the most adequate merged ontology.

3.1.5 Dealing with uncertainty

In the previous section, we highlighted several situations characterized by some forms of uncertainty. In particular, we highlighted uncertainties at the 'object-level', that is we are not totally confident in the correctness of some of our dataset objects. We also emphasized on uncertainties at the 'alignment-level', that is one can be more or less confident on the correspondences set between concepts and roles of the source ontologies. In order to deal with these uncertainties, we use possibilistic logic to encode both object and alignment confidences within a DL knowledge base context.

Concerning the setting of confidences on objects of the source datasets, we do not believe that an automatic solution can produce reliable and relevant confidence values. Hence, it is necessary to integrate the end-user, generally a domain expert, in the process of setting these certainty levels. Two solutions can be envisioned: (i) ask the end-user to assign confidence values to all the tuples of the dataset, (ii) assume that the dataset is sound and ask the end-user to set certainty degrees only on the tuples that are causing inconsistencies.

Solution (i) cannot be realistically implemented since the dataset may be very large and the end-user may not have the time and knowledge to assign a confidence value to each tuple. In this perspective, solution (ii) is much more realistic and efficient since we are asking the end-user to study only a subset of

the dataset objects. This is based on the assumption that the data contained in practical databases is sound and that only a subset of it is erroneous. Hence, this approach requires that all objects are first set to a default value of 1, i.e. assuming soundness, recall that confidence are set in [0,1]. It also implies that the system provides a solution to check consistency of the knowledge base and is able to identify objects responsible for inconsistencies. Such a solution is already implemented in several DL reasoners, e.g. Pellet. Once the knowledge base has been detected as inconsistent, we invite the end-user to refine the confidence value of each object responsible for the knowledge base inconsistency.

The next question to ask ourselves is: when to check the consistency of the (merged) knowledge base ? In fact, this knowledge base can only be detected inconsistent after the application of some alignments. This is due to the consistency of the merged ontology computed from our Galois connection based solution.

We will come back to this inconsistency aspect but first, we would like to make precise the definition of uncertainties on the alignments. We consider that alignments originating from some external knowledge or deduced by our FCA solution (e.g. $F_1 \equiv F_2$) are set with a default value of 1. This assumption is motivated by the following facts:

- the quality of the external ontology generally imported in specific ontologies. That is, we consider the import of an ontology fragment as a strong end-user commitment which ensures the adequacy and quality of this external ontology.

- in practice, our FCA solution only computes concept equivalence on large concept extensions which are likely to be correct.

Nevertheless, the end-user has the ability to refine confidence values on any alignment. Each alignment proposed by the end-user requires a confidence value which can only be defined manually.

Consider our running example and the alignments of (18), we can define the following set of possibilistic formulas for our alignments :
$\{(F_1 \equiv F2,\ 1),(M1 \sqsubseteq M_2,\ 0.5),(C_1 \equiv C_2,\ 0.9),(vegetation_1 \equiv vegetation_2,\ 1)\}$ That is, we are totally confident on the following alignments: $F_1 \equiv F_2$ and $vegetation1 \equiv vegetation_2$. But to a certain extent, we are not totally confident of the correctness of $C_1 \equiv C_2$ and $M_1 \sqsubseteq M_2$ since their degrees of certainty are respectively of 0.9 and 0.5.

The process of generating a consistent merged ontology with respect to a set of alignments and some certainty levels can be defined by the following algorithm.

Algorithm 2	createOntology (Ontology O, Alignment Al, Dataset D)
1	create an ontology O' from O and Al.
2	create an ABox A' from O' objects of D
3	WHILE ($< O', A' >$ is inconsistent)
4	$I(D)$ = inconsistent set of objects of $< O', A' >$
5	Ask end-user to set confidence values to entries of $I(D)$
6	END WHILE
7	classify O'
8	return O'

The understanding of this algorithm is relatively straightforward. Our FCA-based solution generates a merged ontology which is refined by a set of alignments (step 1) and the object matrix of our source ontologies is transformed into an ABox (step 2). All axioms are associated with a certainty degree which makes this knowledge a possibilistic one. We now need to clarify the notion of consistency checking of step 3 in the context a possibility logic theory. The notion of consistency of a possibilistic knowledge base (PKB) is related to its possibility distribution, denoted π_{PKB}. Adapted to the DL context, a PKB corresponds to $\langle PTBox, PABox\rangle$ where $PTBox$ and $PABox$ are respectively a possibilistic TBox and ABox. The classical DL axioms associated to $PTBox$ (resp. $PABox$) is $TBox$, i.e. $\{\phi_i \mid (\phi_i, \alpha_i) \in PTBox\}$ (resp. $ABox$ defined similarly) and $KB = \langle TBox, ABox\rangle$. With $\alpha \in [0, 1]$, the α-cut of $PTBox$ is (defined similarly for $PABox$):
$PTBox_{\geq\alpha} = \{\ \phi \in TBox \mid (\phi, \beta) \in PTBox$ and $\beta \geq \alpha\}$.

Thus, $PKB_{\geq\alpha} = \langle PTBox_{\geq\alpha}, PABox_{\geq\alpha}\rangle$.

The inconsistency degree of PKB, denoted $Inc(PKB)$, is defined as $Inc(PKB)$ = $\max\{\alpha_i | PKB_{\geq\alpha}$ is inconsistent$\}$.

Moreover, how can we integrate uncertainties at the object and alignment levels in a unifying manner? To this end, we introduce the notions of possibility of a concept and of an instance which respectively correspond to the possibility of a concept to be satisfiable and the possibility that an ABox axiom is consistent with the TBox. In order to assign a confidence value to an ontology concept C based on the confidence value of its extension, we propose a simple solution that sets the possibility value of C to the maximum of the possibility of all its instances. Note that several other solutions can be proposed to assign a possibility value to a concept, e.g. the average possibility value of its instances. Recall that due to the soundness assumption of the dataset, all objects have a default value of 1, thus initially all concepts have a possibility of 1.

In the context of our running example with alignments (18), the first two lines of the *createOntology* algorithm generates the O_{m2} ontology with an ABox containing the 6 objects of the Table 3.1. This classical knowledge base (CKB) is inconsistent since the intersection of MF_2 and BLF_1 is not empty. In fact a standard DL reasoner is able to identify object #6 as a source of this inconsistency. In step (5) of our our algorithm, the end-user is proposed to modify the certainty level associated to object #6. Suppose that the end-user is aware that some parcels have been erroneously classified or that some sensors were not really accurate during some field experiments and sets the certainty level of this object to a value of 0.3. We now concentrate on two concepts which have object #6 in their extensions: MF_2 and β. The possibility value of MF_2 (resp. β) is the maximum possibility value of objects #3 and #6, i.e. max{1,0.3}=1, (resp. maximum possibility value of object #6, i.e. max{0.3}=0.3).

Let α=0.3, we have $PKB_{\geq 0.3} = \{PTBox_{\geq 0.3}, PABox_{\geq 0.3}\}$ where the formulas $MF_2 \sqsubseteq \neg BLF_1$, $\beta \sqsubseteq BLF_1$ are contained in $PTBox_{\geq 0.3}$ and $PABox_{\geq 0.3}$ contains the assertions $MF_2(obj\#6)$ and $\beta(obj\#6)$ (respectively stating that object #6 is an instance of MF_2 and β). It is clear that $PKB_{\geq 0.3}$ is inconsistent. Now let α=0.5, then $PKB_{\geq 0.5} = \{PTBox_{\geq 0.5}, PABox_{\geq 0.5}\}$ where $PTBox_{\geq 0.5}$ contains the formula $MF_2 \sqsubseteq BLF_1$ and $PABox_{\geq 0.5}$ contains the assertions $MF_2(obj\#6)$ $PKB_{\geq 0.5}$ is clearly consistent. Therefore Inc(PKB)=0.3. Hence, our method enables us to compute several merged ontology based on a set of given alignments and interactions with end-users to specify possibility values of

certain objects.

3.2 Semantic integration in the DaltOn framework

This contribution discusses a method for developing scientific applications. One of the main message is that separation of concerns – a key paradigm known from software engineering – can help to ease handling complex application scenarios as they often occur in scientific domains. This is achieved by applying the Perspective Oriented Process Modeling (POPM) [JB96] process modeling method which already introduces a separation of concerns up to a certain extend and by further separating pure data integration tasks from domain related tasks. Doing so does not only increases the readability of a process but also allows for domain users to solely focus on their expertise – the scientific analysis.

The other main contribution is an ontology based data integration framework called DaltOn; with DaltOn handwritten integration components which are hard to maintain are not needed anymore. Instead of fixing transformation semantics in code, it is specified as a mapping between ontologies and thus directly incorporates the semantic of transformed data. Since data transformation is de facto specified on a conceptual level, changing and adjusting these transformations whenever schemata or ontologies evolve is rather easy. Thus DaltOn can be adapted to new scenarios very fast. This is also supported by the transparent system design DaltOn is based on; it is an open system which can be extended at any point whenever the current functionality is not sufficient. Also the transformation process itself, given as a normal workflow, can be customized. Therefore DaltOn and thus the semantic integration approach implemented by it, is available for a wide range of systems and is not restricted to – however very well supported by – POPM and its corresponding tool set.

3.2.1 DaltOn architecture

The architecture of the DaltOn Integration Framework (displayed in Figure 3.5) is following the approach of separating concerns into single and independent functions. Thus DaltOn has three major conceptual abstractions, namely Data Provision, Data Operation and Data Integration.

Data Provision bundles components which are used for enacting physical data exchange between data sources and data sinks – in the context of a process between a data producing step (source) and a data consuming step (sink). Each of the sub-components of the Data Provision bundle fulfills a specific task: Data Extraction and Selection cares about the extraction of a (sub-) set of data from a source based on user- and application-specific criteria, Data Transportation handles physical data transport and Data Insertion performs insertion of data.

Data Operation encompasses Format Conversion (FC) and Data Preparation (DP); the FC sub-component is carrying out syntactic transformations of data, for instance the conversion of data given in CSV (comma separated values) into an XML representation and back. DP contains functions which can be applied to data such as unit conversions or simple arithmetic operations but is not meant to replace scientific analysis steps. Both, DP and FC can be used "stand-alone"; their main use is to provide infrastructure services within DaltOn.

Data Integration is the most important abstraction of DaltOn, aiming at integrating data semantically as well as structurally. It also consists of two components: Semantic Integration (SeI) and Structural Integration (StI). The SeI component implements the functions that are purely related to the detection and resolution of semantic conflicts and provides a solution for a terminological transformation. On the other hand, StI provides the capability of integrating datasets based on their structures. It assumes that there is no semantic conflict between source and sink datasets, but datasets are incompatible in terms of their structures. For instance it may include the functions for merging, splitting, and concatenating datasets / data records.

Beside these three main abstractions, DaltOn is using wrappers for accessing data sources through a unified interface (but not for executing data integration tasks) and a RDF based (triple) data store as repository. Any information that is needed by the single components is put into the repository; for instance local and reference ontologies, document schemata, mappings between schemata and ontology elements, instance documents and ontology matching information. However, the repository also contains information about wrappers and format converters. All functionalities can be used stand-alone or in combination with each other; also not every component must be used necessarily. Which functionality is required only depends on the task. Thus, DaltOn allows specifying needed functionality at a very fine level instead of providing a monolithic system with all its negative implications. Any information that is needed by the single components is put into the repository; e.g. local and reference ontologies, document schemata, mappings between schemata and ontology elements, instance documents and ontology matching information. However, the repository also contains information about wrappers and format converters. It is worth to mention here that all DaltOn components built up a library for data integration. All functionality can be used standalone or in combination with each other; also not every component must be used necessarily. Which functionality is required only depends on the task. Thus, DaltOn allows specifying needed functionality at a very fine level instead of providing a monolithic system with all its negative implications.

3.2.2 Architecture of the SeI component

In the rest of this section, we focus on the SeI component which is the most relevant in the context of this dissertation. The main objective of SeI is to generate a valid input document for the target step of a workflow application. In order to illustrate our solution we present in Figure 3.6 the details of this component's architecture. This architecture is based on the set of documents each workflow application can access. This set comprises four kinds of documents:

- An instance document which corresponds to the output document of Application1 (produced by the source step), respectively the input document of Application2 (consumed by the sink step).
- A schema associated to each instance document.
- A mapping between elements of the schema to elements of a local ontology. This document enables us to provide semantics to all the elements used in an instance.

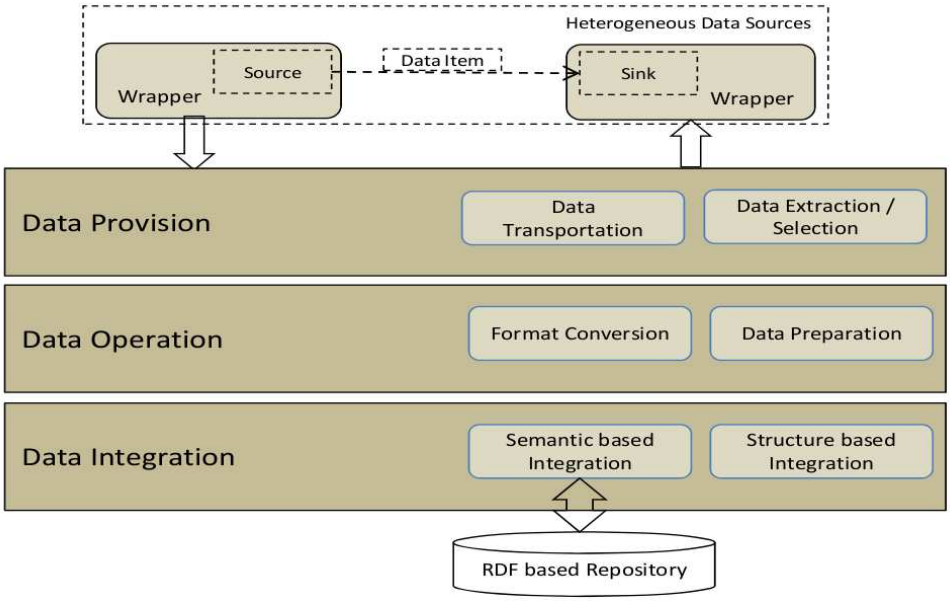


Figure 3.5: Architectural abstractions of the DaltOn framework

- A local ontology which supports the particular interpretation of each concept in an application.
- A reference ontology which provides a common vocabulary to the local ontology. This approach makes the local ontologies comparable and enables to process matches between concepts.
- The repository is responsible for the storage of the knowledge bases associated to the application domain. That is, it stores the TBoxes of the local and reference ontologies as well as an ABox for the reference ontology. The repository is provided with query facilities enabling SeI to retrieve information (using the SPARQL query language). Finally, it also stores the mappings that are being discovered by our matching solution.

The main motivation for this architecture is the ability to easily integrate changes that may occur to the documents previously presented, e.g. schemata, mappings and ontologies. For instance, in the context of scientific applications, the format and schema provided by external sensors are frequently modified. This is possibly due to the replacement of old sensors or the introduction of new ones. In such a situation, a hard coded (procedural) integration implementation would require adaptations in the code, i.e. a task that cannot be handled by a domain expert but requires a computer scientist. Using our DaltOn (declarative) approach and its SeI component, the user needs to introduce the schema associated to a new sensor (an XML document) and change the mapping to the local ontology. The local ontology is likely to have minor modifications, e.g. by introducing a specialization of a concept, to adapt to the new sensor. But in most situations, the reference ontology does not need to be modified as it represents a high level view of the domain of discourse. The only situation where the

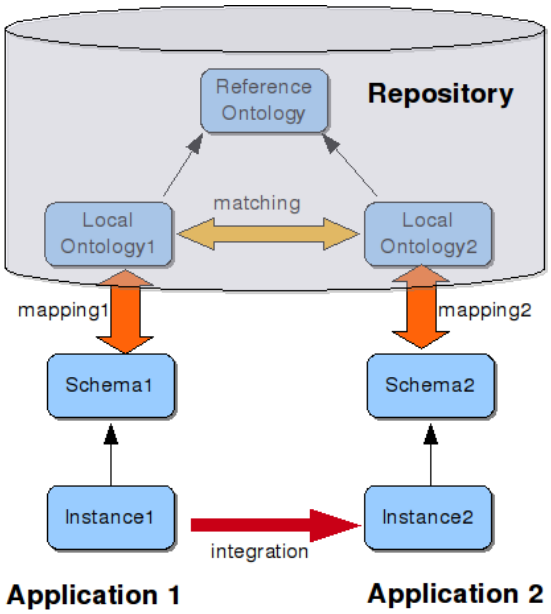


Figure 3.6: Abstract view of the Semantic Integration component

reference ontology should be modified concerns cases of non-monotonic changes of local ontologies. Moreover, the code implemented for the integration does not need to be modified as we have a totally declarative approach. This approach responds to a request of the domain experts we have been working with:

- Their domain of expertise enables them to easily understand and integrate the semantics of new sensors through modifications of mappings and local ontologies. A major constraint is to offer them a user-friendly graphical interface to perform these changes. So far, the GUI we have developed in collaboration with domain experts satisfy them and the learning phase of our tools is effective, e.g. the DBOM tool.

- They are not willing to spend time in learning a programming language and maintaining the application code (procedural approach).

We now motivate the choice of this architecture by providing some details for each component and by considering their associated design cost. The role of an application instance and schema document is obvious in the context of a workflow application. They are usually created by the application developer and come at no extra cost since they are needed by the workflow system. The mappings, ontologies and ABox assertions contained in the repository impose extra work from the application designers and require some expertise in the application domain as well as in knowledge engineering. Nevertheless, the task of developing these documents is limited due to the following: (i) generation of a single reference ontology is generally sufficient, (ii) reuse of local ontologies among several workflows is generally possible, (iii) low expressiveness of the reference ontology, (iv) use of adapted tools which simplify the creation of these documents.

Concerning aspects (i) and (ii), our experiences in using DaltOn in medicine, biological and ecological domains emphasize that usually one unique reference ontology is sufficient for all workflows of an application. The design of this reference ontology can be facilitated by exploiting existing domain ontologies, for instance in bioinformatics or ecology. Our reference ontology do not need all the expressiveness proposed by some well-known ontologies in scientific domains. In fact, we will see that a concept hierarchy with description of the domain and range of properties is sufficient.

Concerning (iii), the expressive power of the local and reference ontologies are not the same. The reference ontology provides a common vocabulary on the domain of discourse. This common vocabulary shared among pairs of ontologies, enables schema mapping to be generated. The reference ontologies should be restricted in the following way: (a) for atomic concepts C and D, terminological axioms are limited to concept inclusion ($C \sqsubseteq D$) and concept equality ($C \equiv D$). (b) for roles, considered as binary predicates, we limit ourselves to specify the parameters (domain and range). Due to its low expressive power, the reference ontology can be designed relatively rapidly by a person with a good knowledge, but not necessarily a domain expert, of the application domain. The design of this ontology is facilitated by graphical ontology editors such as Protégé.

The local ontologies provide their own interpretation to the concepts of the reference ontology and also introduce new concepts described in terms of the concepts and roles of the reference ontology. This means that for acyclic local ontologies, the expansion [BCM+03] of concepts is described only in terms of concepts and roles of the reference ontology. The local ontologies enjoy the expressiveness of the OWL DL language or some of its OWL2 profiles, e.g. OWL2QL.

Concerning (iv), the design of the different ontologies (reference and local ones) as well as the generation of reference ontology concept and role assertions, stored in the repository, are facilitated by the use of the DBOM Protégé plug-in (see Chapter 2).

3.2.3 Schema to ontology mapping

We now concentrate on the mapping which relates elements from schema1 (respectively schema2) to concepts and roles of local ontology1 (resp. ontology2). A schema mapping is generally represented as a triple consisting in a source schema, i.e. the XSD schema in a workflow application, a target schema, i.e. a local ontology, and a mapping specifying relationships between the source and the target schemata. We exploit this representation and restrict the set of mapping relationships to:

- a mapping to an ontology concept (denoted 'mappedToConcept'),
- a mapping to an ontology role (denoted 'mappedToRole') and
- mapping to a concept instance and which is stored in the repository (denoted 'mappedToIndividual').

The syntax of our mapping solution is restricted such that not all combinations of the mapping relationships are accepted. We characterize these restrictions and their associated semantics in Table 3.2.

Table 3.2: Mapping possibilities in SeI

	MappedTo			
#	**Concept**	**Role**	**Individual**	**Semantics**
1				Not accepted
2	×			Empty XML element is mapped to an ontology concept
3		×		Not accepted
4			×	Equivalence to a concept instance
5	×	×		Non empty XML element is mapped to a concept and a role
6	×		×	Empty XML element is mapped to an ontology concept and an individual
7		×	×	Not empty XML element mapped to a role and a concept instance
8	×	×	×	Non empty XML element is mapped to a concept and role as well as a concept instance

The simplest abstraction of an XML document is a labeled ordered tree, possibly with data values associated to the leaves. But for our mapping approach, we take advantage of the object model view which can also be applied to an XML document. Starting from this view, we assume that any XML element is at least mapped to a DL concept or DL individual. This first assumption enables us to disallow the mapping #1 and #3 which do not inform about an associated DL concept nor DL individual. The purpose of mapping #4 is to inform the system about the absence of mapping for a given XML element. In fact, this is most effectively and rapidly performed by users by omitting such a mapping for this element.

We now consider the emptiness of an XML element. In cases where it is not empty, i.e. it contains a data value, it is necessary to map it to a DL property. This is the case of mapping #5, #7 and #8 in Table 3.2. Mapping #8 is a specialization of mapping #5 where extra information about an associated concept individual is provided. Mapping #7 can be viewed as being equivalent to #8 where the type instance is not specified. We will motivate this situation later in this section. In cases of an empty XML element, no DL property needs to be attached to the mapping. Hence, it corresponds to mappings #2 or #6. The latter being a specialization of the former where extra information about a DL concept instance is provided.

Finally mapping #4 is considered as a shortcut of mapping #6 where the DL concept is omitted. This kind of mapping is supported if the processing of the DL realization reasoning procedure, i.e. providing the most specific concept an individual is an instance of, returns a single concept. Thus there cannot be any ambiguities about the type of this individual.

3.2.4 Methodology and heuristics

In this section, we present the different steps necessary to generate the input document of Application2. These steps correspond to (i) matching the local ontologies, (ii) matching the (XML) schemata and (iii) generating the target instance document.

Matching local ontologies

This matching step searches for correspondences between the DL concepts of both local ontologies. This operation is supported by the existence of a common vocabulary, i.e. the reference ontology. In order to discover as many consistent matches as possible, we consider two methodologies to find correspondences: DL-based and navigation-based mappings.

The DL-based approach is performed using a DL reasoner and particularly its concept subsumption inference procedure.

In the navigation-based approach, we consider an ontology as a directed acyclic graph where nodes correspond to DL concepts and the edges correspond to DL properties. Basically, it searches for navigation paths between two concepts. This is performed by exploiting the (SPARQL) query facilities of the (triple store) repository. The navigation-based approach also exploits a DL reasoner with its concept subsumption, instance checking and realization inference procedures. This approach is non-deterministic and may return several different paths. So it is important for our algorithm to qualify paths and to select the most appropriate one. This qualification is based on several factors: the length (L) of each paths (i.e. the number of properties along a path) and the characteristics of the properties used along a path, i.e. functionality, inverse functionality.

As our implementation formalizes ontologies using decidable species of OWL, i.e. OWL Lite and OWL DL, it is possible to distinguish properties based on their functional characteristics.

The decidability issue of DL reasoning tasks is a main concern in our solution. For this reason, we are not considering inverse functional properties, which are supported in OWL, but are only associated to decidable inferences for object properties. Thus inverse functional properties on data type properties is allowed in OWL Full ontologies which is known to be undecidable ([BvHH$^+$04]). We now distinguish between several navigation approaches: (i) L=1 and the property is functional: 'functionalNavigation'. (ii) L=1 and the property is not functional: 'nonFunctionalNavigation'. (iii) L $>$ 1: 'pathNavigation'.

The match operator applied in our workflow context is able to find several correspondences, usually belonging to the two presented categories, between a given pair of DL concepts. In order to deal with this issue, we propose a heuristic to select a preferred correspondence. This heuristic is based on a total order of the DL-based and navigation-based categories.

Definition 11 *For a given pair of DL concepts C1 and C2, respectively from local ontologies 1 and 2, if a set of correspondences are found between these two concepts: We know that there must be at most one DL-based correspondence between C1 and C2 but several navigation-based mappings can coexist with it. For this reason, we rank the navigation-based correspondences according to*

a preference total order: functionalNavigation > nonFunctionalNavigation > pathNavigation.

Concerning navigation-based relationships, setting a property to be functional is an important commitment for the knowledge engineer. We thus consider that a functionalNavigation is preferred to a nonFunctionalNavigation. Finally we consider that navigation with a single edge is more trustworthy than a path made of several edges.

Thus we obtain a partial order on the total set of discovered correspondences. On the use cases we have implemented with DaltOn so far, we added a heuristic stating that functionalNavigation is preferred to concept generalization which is preferred to nonFunctionalNavigation, thus obtaining a total order on correspondence preferences: concept equivalence > concept specialization > functionalNavigation > concept generalization > nonFunctionalNavigation > pathNavigation. We consider that other heuristics could be applied, e.g. generalization > functionalNavigation, and SeI supports the definition of specific preference orders.

Matching schemata

The purpose of this step is to discover mappings between Schema1 and Schema2 from the mappings discovered in the previous step, i.e. between local ontology1 and Local ontology2. This step can be easily performed using the schema to ontology mapping, i.e. from schema 1 to local ontology 1, respectively schema 2 and local ontology 2.

It is interesting to note that this operation can be considered as a composition of mappings. This composition exploits the matching previously obtained from DL concepts. Thus several mappings apply between elements of the schemata. We can consider that our solution is also able to discover one-to-n mapping. Finally, due to the dual matching solution (logic-based and navigational-based), the accuracy of the data stored in the repository and the possibilities to adjust heuristics, we have not encountered problems with false matching. These false matchings generally occur when the local ontologies are modified due to replacement or configuration modifications at the sensors. In these cases, we consider that adjustments need to be performed on the local ontologies and, possibly in non-monotonicity situations of the local ontologies, to the reference ontology.

Target instance generation

Starting from these mappings, it is possible to consider the generation of data values for (non-empty) target elements. For navigation-based correspondences, the processing is relatively obvious as it is sufficient to follow the selected paths between two concepts stored in the repository. This navigation is performed starting from a specific node of the ontology graph. This node is identified by the information provided by source information.

For DL-based correspondences, it is required to inspect the DL properties associated to each mapping in order to detect possible transformations. A final step consists in enabling the integration of data from application1's instance, possibly with the help of the repository, onto application2's instance document. Different forms of mappings are available, e.g. relational queries, relational view

definitions, XQuery queries or XSLT transformations, to perform this task. We opted for XSLT transformation since we do not need the expressiveness and complexity of relation queries and views. Moreover implementations of XLST transformation are currently considered to be more reliable than XQuery's. By selecting XSLT, we also benefit from procedural attachment possibilities when performing transformations. That is SeI includes a set of procedures, developed in the Java language, to enable the retrieval of values stored in the repository at runtime. Most of these procedures generate, from predefined templates, SPARQL queries and execute them on the repository's ABox.

3.3 Presentation of papers

3.3.1 FCA papers

[CJ08] **Olivier Curé** and Robert Jeansoulin. An FCA-based solution for ontology mediation.
In ONISW, pages 39–46, 2008.

The paper presents a Formal Concept Analysis based methodology for ontology mediation. More specifically, the proposed methodology enables one to create concepts that are not in the source ontologies, to label the new concepts, and to optimize the resulting ontology by eliminating redundant or irrelevant concepts. Additionally, the paper presents an approach toward ontology refinement, which enables one to measure and rank the concepts in the mediated ontology and to define multiple forms of mediated ontology by relaxing some of the ranking criteria.

[CJ09] **Olivier Curé** and Robert Jeansoulin. An FCA-based solution for ontology merging.
Journal of Computing Science and Engineering, 3(2):90–108, 2009.

The program committee of the ONISW workshop has selected our paper for a long and self-contained version. A novel contribution of this paper is to propose a set of heuristics to measure the strength of generated concepts. These measures correspond to:

- support, i.e. a frequency measure based on the idea that values which co-occur together frequently have more evidence to justify that they are correlated and hence are more interesting.
- derivation, i.e. a measure based on the number of derivations performed in the process of generating the merged ontology.
- lattice position, i.e. we characterize the position of an empty set in the lattice as `leaf` or `non-leaf`.

Given these measures, we propose ranking solutions that can be adapted to end-users need.

[Cur09d] **Olivier Curé** Merging expressive ontologies using formal concept analysis.
In OTM Workshops, pages 49–58, 2009.

This paper extends the JCSE paper by providing description of emerging concepts in terms of elements of the source ontologies. The description language of the DL tackled $\mathcal{ALC}$, that is the minimal DL language that is of practical interest. Intuitively, a DL expansion operation is performed using the concept hierarchy defined in the merged lattice.

[Cur10b] **Olivier Curé** Merging Expressive Spatial Ontologies using Formal Concept Analysis with Uncertainty Considerations
Accepted and to be published in "Methods for Handling Imperfect Spatial Information",
Editors: Robert Jeansoulin, Odile Papini, Henri Prade, Steven Schockaert (21 pages).

This paper extends the work described in [Cur09d] with a logic-based approach to deal with uncertainties encountered at the object and alignment levels of our FCA-based merging solution. This approach is anchored into possibilistic logic which provides an efficient solution for handling uncertain or prioritized formulas and coping with inconsistency. In this work, we use a DL extension of possibility theory, and we are thus able to easily handle the creation of different consistent merged ontologies.

3.3.2 DaltOn papers

[CJ07a] **Olivier Curé** and Stefan Jablonski. Ontology-based data integration in data logistics workflows.
In ER Workshops, pages 34–43, 2007.

This paper is a first attempt at combining workflow aspects with ontology-based data transformations. The authors present an approach for data integration consisting of two components. A workflow component (Data Logistics) is dealing with the technical task of data transmission and exchange. And an ontology-based component is responsible for data transformation. The whole approach is illustrated using health care use cases.

[CJJ+08] **Olivier Curé**, Stefan Jablonski, Florent Jochaud, Abdul Rehman and Bernhard Volz.
Semantic data integration in the DaltOn system.
In ICDE Workshops, pages 234–241, 2008.

This paper improves and provides more functionalities to the semantic integration component of the DaltOn system compared to [CJ07a]. The DaltOn system is responsible for executing data transformations between adjacent steps in a workflow. Such data transformations involve format conversions as well as semantic transformations. The semantic integration component of DaltOn performs semantic transformations with the aid of a reference domain ontology and descriptions of the source and sink schemata in terms of the reference ontology.

[JCRV08] Stefan Jablonski, **Olivier Curé**, M. Abdul Rehman, and Bernhard Volz.
DaltOn: An infrastructure for scientific data management.
In ICCS (3), pages 520–529, 2008.

This paper focuses on the workflow management and Data Logistics aspects of the DaltOn framework. Its execution semantics is provided and examplified on a meteorological use case.

[JVR+09] Stefan Jablonski, Bernhard Volz, M. Abdul Rehman, Oliver Archner, and **Olivier Curé**. Data integration with the dalton framework a case study. In SSDBM, pages 255–263, 2009.

This paper presents the components of the DaltOn framework on a concrete meteorological example. It emphasizes on the different solution to integrate information (structural and semantic) and highlights how easily domain experts can update the different components to reply to sensors modifications.

3.4 Conclusion and perspectives

The ontology merging solutions based on FCA has been quite useful on several projects, e.g. geographical information. But it still has some drawbacks when dealing with large ABoxes, e.g. millions of parcels in a landcover application. So the main perspective around this project is to discover solutions to reduce the amount of processed objects and at the same time retaining all possible information. This solution necessarily needs to deal with uncertainty issues already taken into account in our work so far. Another future work concerns treating more expressive ontologies and to reach the expressiveness of OWL2 DL, i.e. $\mathcal{SROIQ}$(D) DL.

Concerning the DaltOn framework, we would like to pursue our research in two directions. The first one aims at proposing data and workflow provenance within the system. This is very important for end-users to understand why and how a set of data are the inputs of a given application. We believe that RDF is an ideal format to support such operations. The second directions concerns the storage of RDF triples. Up to now, we have used a standard triple store but we believe that we can improve retrieval performances and functionalities by introducing a novel storage solution (see the Conclusion chapter for more on this topic).

Chapter 4

Ontology-based data quality enhancement of relational databases

The quality of information stored in databases has a significant impact on the efficiency of organizations and businesses managing and exploiting them [BS06]. This is particularly relevant in the medical domain where inaccurate or false information can have dramatic effects on the health of patients. In this chapter, we again consider our self-medication application which has been designed and is being managed with a team of domain experts. At the core of this application is a relational database containing information on symptoms of the self-medication domain, drugs available on the French market and SEHR data. Concerning drugs, all the information present in the Summary of Product Characteristics (SPC, e.g. posology, contra-indications, side-effects) are stored in the database conjointly with additional information such as a rating based on an efficiency/tolerance ratio, price, opinion from domain experts, social security system reimbursement rate, etc. Our database also integrates several international standards in the pharmacology domain, e.g. the European pharmaceutical Market Research Association (EphMRA) classification and on symptoms and diseases, e.g. the tenth edition of the International Statistical Classification of Diseases and Related Health Problems (ICD 10). These information can be transformed into a knowledge base using the DBOM system (Chapter 2) and hence support different forms of reasoning in applications.

The main objective of introducing these terminologies is to support interoperability with other healthcare systems by integrating standard identifiers.

The XIMSA application targets the general public, i.e. end-users assumed to have no medical knowledge. Hence we cannot expect end-users to detect false information, as it could the case for an application designed for healthcare professionals. Consequently, the soundness and completeness of the stored information is a top priority since inaccurate or wrong information may be proposed to the patient and jeopardize his health.

This chapter presents several contributions to maintaining high data quality in databases based on operations performed at the ontology level. An obvious question is why do we maintain the data quality at the database level rather

than at the ABox level? The motivation is twofold. In the first hand, the kind of databases we are dealing with are generally used by several applications and some of them do not require ontologies and any form of associated reasoning. In the second hand, our ontology-based application aims to benefit from the robust and efficient storage system of relational databases. To this end, we consider that storing data in a RDBMS and representing the conceptual layer in an ad hoc ontology system is a good practice especially if the ontology can be maintained in main memory. This has the advantage of having a low main memory footprint (the ABox, stored in secondary memory, is usually way larger than the TBox), offering a persistence solution for the ABox, benefiting from transaction and fault tolerance already available in RDBMS. Note that a system like OBDA also adopts this approach. In the Conclusion Chapter, we briefly present some on-going work on an efficient storage solution of RDF datasets in RDBMS.

A second question concerns the level at which we process data quality operations, i.e. detecting incorrect information ? We argue that the ontology level is better suited than the database level since it is semantically richer, i.e. represents more constraints, and allows standard inference services to be executed by optimized reasoners.

The different approaches proposed in this chapter can be decomposed as follows:

- the presentation of an induction based reasoning solution to enrich and refine an ontology from relational databases and the use of this ontology to clean the underlying databases.
- some contributions on conditional dependencies in the relational model. Recently, some conditional dependencies, i.e. special forms of database dependencies, have been introduced for data quality and cleansing purposes.
- the representation of pertinent data quality conditional dependencies at the ontology level. Using concept hierarchies present in most ontologies, this approach proves its superiority to a standard SQL representation. This approach is investigated in both a standard ontology domain and in the context of Ontology-Based Data Access (OBDA).

4.1 Data quality enhancement using ontologies and inductive reasoning

In this section, we propose an approach which takes advantages of the large number of databases maintained in the world as well as the many available hierarchical classifications, thesauri and taxonomies they are storing. In the ontology research field, these notions are associated with a set of concepts that are more or less strictly organized in hierarchies. The fundamental work of [Bra83] analyzes the meaning of the taxonomic relationships and highlights that multiple types of taxonomic relationships exists. Also, as proposed in [HdB07], depending on the context, it is possible to interpret the hierarchical organizations of these terminologies as defining partial order relations on their concepts.

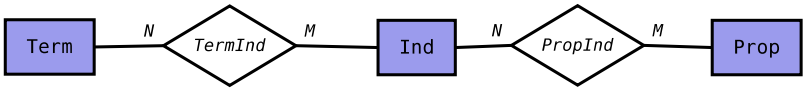


Figure 4.1: Typical ER diagram used in terminology enrichment

We concentrate on the contexts where such properties can be assumed and we therefore call them classifications.

The aim of our approach is twofold: (1) enrich classifications using database sources and (2) detect and repair database inconsistencies using these enriched classifications. We now provide more details about these two steps.

Terminology enrichment using inductive reasoning

The enrichment of classifications integrated in databases is performed using inductive reasoning. A first step of this operation consists in creating an ontology from tuples stored in some database relations. This is performed using the DBOM system (presented in Chapter 2) via the definition of schema mappings that transform tuple values into concepts/roles and organize their hierarchy accordingly. The output of this operation is henceforth called an ontology and can be serialized into RDFS or OWL documents.

The purpose of the enrichment step is to enable the aggregation of database tuples in a coherent way such that common properties can be discovered and associated to ontology concepts. Our approach exploits the structure of a star schema (aka dimensional model) which typically consists of a large table of facts (known as a fact table), with a number of other tables surrounding it that contain descriptive data, called dimensions. More precisely, we consider that a relation, or a set of relations, `Term` in the database schema R stores a given terminology, e.g. the EphMRA classification. Let consider that a relation `Ind` stores individuals of the database domain, e.g. drug products. Then it is most likely that a one-to-many or many-to-many relationship, or a chain of relationships, `TermInd` relates facts between `Term` and `Ind`. We can also assume that properties, e.g. contra-indications or side-effects in the pharmaceutical domain, about these individuals are either directly stored in `Ind` or stored in a relation `Prop` in which case a possibly many-to-many relation `PropInd` relates facts between `Prop` and `Ind`. Thus the `Ind` relation plays a central role in our solution as it enables us to join elements from `Term` to elements of `Prop`. Figure 4.1 presents a Entity Relationship diagram of the kind of star schema we are using in this approach.

Given a fact in `Term`, it is possible to aggregate tuples from `Ind`, via `TermInd`, in a sufficiently coherent manner and to extract valuable properties from these groups. The aggregation of tuples is performed using automatically generated SQL queries. Each query calculates for each possible domain value the ratio of this value occurrences on the total number of elements of the group. Only tuples with a ratio superior or equal to a user-defined threshold are retrieved and associated, via an automatically generated property, to a concept.

In some situations, it may be necessary to revise the implantation of properties in the concept hierarchy. The purpose of this refinement is to associate similar property/value couples common to sibling concepts to their most general super concept. This is performed using standard DL subsumption inferences.

Starting from these OWL ontologies, we can now perform the enrichment by

inductive reasoning. This task exploits the drug database instance for which we present an extract, Figure 4.2, which is related to drug contra-indications. Figure 4.2a proposes an extract, not all columns are displayed, of the *Drug* relation with two drugs and with CIP identifiers. Figure 4.2b presents an incomplete list of the *ContraIndication* relation which stores all terms related to drug contra-indications. Now the *ProductContraIndication* relation enables one to relate products identified by CIPs with their contra-Indications (Figure 4.2c). The *TherapeuticClass* relation regroups all therapeutic classes encountered in the drug domain and identifies them with integer values(Figure 4.2d). This identifier is related to CIP codes in the *ProductTherapeutic* relation (Figure 4.2e). Finally, two relations relate EphMRA and ATC codes to CIPs, respectively *ProductEphMRA* (Figure 4.2f) and *ProductATC* (figure 4.2g) relations.

In the following, we present the inductive reasoning method on the EphMRA ontology, also named AC-ontology, and stress that an adaptation for the ATC ontology is obvious. The method used to enrich the AC-ontology is based on induction reasoning on relevant groups of products, generated using the AC hierarchy. Intuitively, we navigate in the hierarchy of AC concepts and create groups of products for each level, using the ProductEphMRA relation (Figure 4.2f). Then, for each group we study some specific domains which correspond to fields in SPCs, e.g. contra-indications, and for each possible value in these domains we calculate the ratio of this value occurrences on the total number of elements of the group. Table 4.1 proposes an extract of the results for the concepts of the respiratory system and the contra-indication domain. This table highlights that our self medication database contains 56 antitussives (identified by AC code *R05D*), which are divided into 44 plain antitussives products (*R05D1*) and 12 antitussives in combinations (*R05D2*). For the contra-indication identified by the number 76, i.e. allergy to one of the constituents of the product, we can see that a ratio of 1 has been calculated for the group composed of the *R* AC code. This means that all 152 products (100 %) of this group present this contra-indication. We can also stress that for this same group, the `breast-feeding` contra-indication (#9) has a ratio of 48 %, this means that only 72 products out the 152 of this group present this constraints.

We now consider this ratio as a confidence value for a given AC-concept on the membership of a given domain's value. This membership is materialized in the ontology with the association of an AC-concept to a property, e.g. the hasContraIndication property, that has the value of the given contra-indication, e.g. breast-feeding (#9). In our approach, we only materialize memberships when the confidence values are superior to a predefined threshold θ, in the contra-indication example we set θ to 60%.

This membership is only related to the highest concept in the AC hierarchy and inherited by its sub-concepts. For instance, the breast feeding contra-indication (#9) is associated to the *R05* AC-concept as its confidence value (83%) is the first column on line with *contraId* 9 that presents a θ superior to 60% in the *R* hierarchy. Also, the pregnancy contra-indication (#21) is related to the *R05D2* AC concept since its value is (73%).

Using this simple approach, we are able to enrich the AC-ontology with axioms related to several fields of SPCs. At the end of this enrichment phase, the expressiveness of the newly generated ontology still corresponds to an OWL DL ontology. The following code proposes an extract of the AC-ontology, in RDF/XML syntax, where we can see the definition of *R05D2* concept (line #1

(a) Drug relation

cip	productName
3272615	Hexapneumine
3572151	Biocalyptol

(b) ContraIndication relation

contraID	contraName
9	BreastFeeding
21	Pregnancy
108	Productive cough
110	Respiratory insuffisancy

(c) ProductContraIndication relation

cip	contraId
3272615	9
3272615	21
3272615	108
3272615	110
...	
3572151	9
3572151	108
3572151	110

(d) TherapeuticClass relation

classId	className
295	Antitussive

(e) ProductTherapeuticClass relation

cip	classId
3272615	295
3572151	295

(f) ProductEphMRA relation

ephMRA	cip
R05D1	3572151
R05D2	3272615

(g) ProductATC relation

atc	cip
R05DA20	3272615
R05DA08	3572151

Figure 4.2: Extract of the self-medication database

Table 4.1: Analysis of contra-indications for the respiratory system

	R	R05	R05D	R05D1	R05D2
occurrences	152	71	56	44	12
ContraId					
9	.48	.83	.86	.82	1
21	.26	.39	.3	.2	.73
76	1	1	1	1	1
108	.34	.69	.84	.84	.82
109	.35	.66	.8	.8	.82
110	.34	.73	.89	.86	1
112	.34	.71	.88	.86	.91

to #12). This description states that the concept :

- has the contra-indication identified by *CI_21* (line #2 to #7) which corresponds to pregnancy (line #13 to #16).
- is a subconcept of the *R05D* concept (line #8)
- is disjoint with the concept identified by the *R05D1* code
- has a comment, expressed in the french language (line #10).

```
1. <owl:Class rdf:about="&j.0;R05D2">
2.  <rdfs:subClassOf>
3.   <owl:Restriction>
4.    <owl:onProperty
         rdf:resource="&j.0;hascontraIndication"/>
5.    <owl:hasValue rdf:resource="&j.0;CI_21"/>
6.   </owl:Restriction>
7.  </rdfs:subClassOf>
8.  <rdfs:subClassOf rdf:resource="&j.0;R05D"/>
9.    <owl:disjointWith rdf:resource="&p1;R05D1"/>
10.  <rdfs:comment
         xml:lang="fr">ANTITUSSIFS EN ASSOCIATION
11. </rdfs:comment>
12.</owl:Class>
13.<j.0:contraIndication rdf:about="&j.0;CI_21">
14. <rdfs:comment xml:lang="fr">grossesse
15. </rdfs:comment>
16.</j.0:contraIndication>
```

This method can easily be applied to the ATC ontology or other drug related ontologies as soon as we consider that the ontology is presented in a DL formalism and a relation relates CIPs to identifiers of this ontology.

Ontology-based detection and repairing

In a second phase, we use the ontology defined in the first step to detect data quality violations. This detection is performed on-demand after a set of modifications have been performed on the databases. A main idea of our approach is to consider that the axioms added to the ontologies correspond to additional source database data dependencies. These data dependencies are strongly related to instances of the database and their representation are generally not supported in most standard RDBMS. So this approach enables us to store these data dependencies as valuable properties of expressive ontologies.

Starting from these resulting ontologies, we check if the source database violates some of these data dependencies and propose a data cleaning solution. By repairing violations, we aim to enhance the data quality of the source databases. We have been influenced by ISO 9000 Quality standards by considering two data quality elements: completeness and correctness. In terms of completeness, we are interested in the presence or absence of data in the dataset. Considering the second quality element, we assume the correctness of the datasets used to generate the ontology. Anyhow, correctness of data still needs to be ensured when new information is inserted or updated.

Due to possible exceptions in the dataset, the automatic repairing of data may not be pertinent. Thus a semi-automatic approach is proposed where the end-user validates modifications proposed by the system. We assist the end-user by automatically detecting possible violations and by offering an effective and user-friendly graphical user interface (GUI) to semi-automatically repair the data. This GUI takes the form of a matrix which emphasizes possible violations of ontology constraints. Moreover, end-users can interact with this matrix to repair the data. Repairing is driven using a concept-centric or an attribute-centric approach.

In the former one, groups of tuples are formed according to concepts of the ontology. Figure 4.3 presents an example of a concept-centric matrix for the *R05D2* EphMRA code where entries of the first row correspond to contra-indications identifiers and entries of the first column are drug identifiers. At a row x and a column y, the matrix can be filled with 3 different values: (1) a 'x' symbol indicates that the tuple of `Ind` identified by value x has the property identified by value y for that concept, (2) a '?' highlights that the tuple does not have this property and that according to the ontology, it should be the case, (3) an empty cell indicates that the drug does not have this property and this state holds with the ontology's knowledge. In case (2), the end-user can click on the interrogation mark to automatically correct the database by inserting a new tuple in `PropInd` and thus stating that the database individual has this property.

CIP	110	76	9	112	109	108	21	103	165	880	40	493	111	217	342	453	191	28	913
3656706	×	×	×	×	×	×	?				×		×						
3464473	×	×	×	×	×	×	×	×	×	×		×							
3464496	×	×	×	×	×	×	×	×	×	×		×							
3464467	×	×	×	×	×	×	×	×	×	×				×					
3032035	×	×	×	×	?	?	?				×						×		×
3418154	×	×	×	×	?	?	?												
3049455	×	×	×	×	×	×	×							×		×		×	
3071638	×	×	×	?	×	×	×	×			×								
3117429	×	×	×	×	×	×	×	×	×	×		×							
3109660	×	×	×	×	×	×	×						×		×				
3281057	×	×	×	×	×	×	×						×		×				

Figure 4.3: Extract from an ontology concept-centric view: contra-indication matrix for the *R05D2* EphMRA concept

In the attribute-centric approach an attribute of the database, possibly not associated to an ontology concept, a concept created from a relation *Prop* and a set of available ontologies are selected. The selected attribute serves to design groups of database tuples (via the creation of GROUP BY SQL queries) and the set of ontologies enables to analyze this group according to the information stored in these ontologies. The results are displayed in a matrix similar to the one presented previously: columns are individuals of a given concept and rows are tuples from the created group. The cells of the matrix can again be empty or filled with 'x' with the same interpretation as the previous approach. But the cells can also be filled with integer values that range from $2^n - 1$ with n the

number of ontologies in the set. These values identify the elements of the power-set of n minus the empty set which is being dealt with the 'x' symbol. Figure 4.4 proposes an extract for the contra-indication SPC field for the *antitussive* therapeutic class using two ontologies, namely the EphMRA and ATC ontologies generated with DBOM. Hence the three distinct values that can appear in the matrix are being given the following meaning:

- A value of 1 in a cell highlights a proposition made from the first ontology, i.e. EphMRA ontology in our example. This is the case for the contra-indication with value 109 and products identified with CIP 3481537 and 3371903.
- A value of 2 in a cell highlights a detection made from inferences using the second ontology, i.e. ATC ontology.
- A value of 3 in a cell highlights that both ontologies (EphMRA and ATC) have detected this cell as a candidate for violation.

	111	110	76	112	113	9	109	108	40	107	1367
3447693	x	x	x	x	x	x	x	x	x		
3366227	x	x	x	2	3	x	x	x	x	x	
3282358	x	x	x	2	3	x	x	x	x		
3481537	x	x	x	x	x	x	1	x	x		
3296018	x	x	x	x	3	x	x	x	x		
3296024	x	x	x	x	x	x	x	x			x
3371903	x	x	x	x	x	x	1	x	x		

Figure 4.4: Extract from a database attribute-centric view: contra-indication matrix for antitussive therapeutic class

This attribute-centric approach works for n ontologies but requires that a mapping is defined between concepts of these ontologies.

4.2 Condition Inclusion dependencies (CINDs)

Among the data quality solutions that have been investigated recently, data dependencies is a promising one. Some of them are based on extensions of the notion of data dependency which has been thoroughly investigated and exploited in the domain of relational databases, e.g. functional and inclusion dependencies, henceforth denoted FDs and INDs [AHV95]. For instance, in [Fan08] the authors highlights an attempt to improve data quality using conditional dependencies.

So far, two forms have been investigated: conditional functional dependencies (CFDs) [BFG+07] and conditional inclusion dependencies (CINDs) [BFM07], corresponding respectively to conditional counterparts of FDs and INDs. That is a pattern tableau potentially containing variables and constant values is associated to each conditional dependencies. Intuitively, they hold only for the

tuples that satisfy some conditions and are supposed to capture more of the inconsistencies of real-life data.

Example 4.2.1 *Consider the following database schema:*
drug (CIP, PRODUCTNAME, PRODUCTRATE, PRODUCTYPE)
atc (ATCCODE, ATCNAME)
atcDrug (CIP, ATCCODE)
drugContra (CIP, ATCCODE)
where the drug relation contains a (french) product identifier (CIP), a product name (PRODUCTNAME), the reimbursement rate (PRODUCTRATE) and the type of the product (PRODUCTYPE) corresponding to allopathy, homeotherapy, etc. The atc relation stores all ATC codes (ATCCODE) and their names (ATCNAME). Another relation, atcDrug, relates drug products to their ATC codes. Finally, the relation drugContra stores the ATC codes a drug is contraindicated to. Some of the FDs and INDs that hold for this database include:

fd1 : $drug\ (CIP \rightarrow PRODUCTNAME)$
ind1: $drugContra(ATCCODE) \subseteq atc(ATCCODE)$
ind2: $atcDrug(CIP) \subseteq drug(CIP)$

Figure 4.5 presents a portion of a drug database instance. Starting from this small instance, we observe that standard FDs and INDs are not sufficient to represent the set of constraints associated to the pharmacology domain. For instance, we would like to state that drug reimbursement at a 65% rate must be of the type 'allopathy' and that only allopathic drugs are present in the *atcDrug* relation. Moreover, we would like to represent the fact that all drug products contra-indicated to *Iproniazid* must be either composed of the molecules *Zolmitriptan* or *Dextromethorphan*. These constraints require a notion of condition, a feature not supported by traditional FDs and INDs. But their conditional counterparts, resp. CFDs and CINDs have been introduced to support these constraints:

cfd2: *drug* (PRODUCTRATE=65 → PRODUCTYPE='allopathy')
cind3: *atcDrug*(CIP) ⊆ *drug*(CIP, TY='allopathy')
cind4: *drugContra*(CIP, ATCCODE='N06AF05') ⊆
atcDrug(CIP,ATCCODE='N02CC03' || 'R05DA09')

The database instance of Figure 4.5 does not satisfy these 3 conditional dependencies. For instance, cfd2 is violated by tuple t6 since the *Cephyl* drug is a homeopathic drug but is reimbursed at the 65% rate. Moreover cind3 is violated by tuple t12 because as a homeopathic drug, *Cephyl* should not have an entry in the *atcDrug* relation. The treatment of cind4 is more involved since it implies a form of disjunction. That is a drug contraindicated with *Iproniazid* must be either composed of the *Dextromethorphan* or the *Zolmitriptan* molecules. This is not the case for tuple t16 since the drug identified by (CIP) value 3786018 is composed of *Phloroglucinol*. Interestingly, a similar form of disjunction has recently been introduced in the context of CFDs, yielding the notion of eCFDs [BFGM08]. Besides, we would also like to detect that the *Tuxium 30mg* drug (t4), which contains the *Dextromethorphan* molecule (t10), is not contraindicated to the *Iproniazid* molecule.

Most of the investigations conducted on conditional dependencies have focused on their syntax, semantics and properties on precise problems, e.g. consistency, implication and existence of finite axiomatization. The understanding

Figure 4.5: Records from the drug database

Tuple	CIP	PRODUCTNAME	PRODUCTRATE	PRODUCTYPE
t1	3533665	Zomigoro 2.5mg	65	Allopathy
t2	3282358	Capsyl 15mg	0	Allopathy
t3	3544309	Zomig 2.5mg	65	Allopathy
t4	3311692	Tuxium 30mg	35	Allopathy
t5	3786018	Phloroglucinol Arrow	35	Allopathy
t6	3187559	Cephyl	65	Homeotherapy

(a) *drug* relation

Tuple	CIP	ATCCODE
t7	3533665	N02CC03
t8	3282358	R05DA09
t9	3544309	N02CC03
t10	3311692	R05DA09
t11	3786018	A03AX12
t12	3187559	N02BA01

(b) *atcDrug* relation

Tuple	CIP	ATCCODE
t13	3533665	N06AF05
t14	3282358	N06AF05
t15	3544309	N06AF05
t16	3786018	N06AF05

(c) *drugContra* relation

Tuple	ATCCODE	ATCNAME
t17	R05DA09	Dextromethorphan
t18	N06AF05	Iproniazid
t19	N02CC03	Zolmitriptan
t20	A03AX12	Phloroglucinol
t21	N02BA01	Acetylsalicilc acid

(d) *atc* relation

of these issues now offers an ideal environment for the design of efficient data cleansing techniques and tools.

In order to design such tools, a first step consists in discovering conditional dependencies. This effort is necessarily automated since manual discoveries are too expensive, due to large information volume, and require the involvement of domain experts. Automated methods based on the analysis of sample database instances have already been published in [CM08], [GKK+08] and [FGLX09]. But these papers only propose solutions for the discovery of CFDs. Unfortunately, it is considered that the full potential of data cleansing tools based on conditional dependencies reside in methods exploiting both CFDs and CINDs.

In [Cur09a], we have studied different forms of CINDs, mainly differentiated by restrictions on appearance of constant values, and proposed methods to automatically discover CINDs for two of the most relevant CIND forms: patterns with constants only and patterns with constants and variables. Obviously, the processing of the former is more efficient than the latter. Thus an "on-demand" approach offers end-users to select the most pertinent and efficient discovery solution.

Moreover, we extended the standard CIND approach by introducing disjuncts in the right hand side of the embedded dependency (e.g. $cind_2$) and emphasized that this comes at no extra cost in the discovery and interpretation of these dependencies.

A second issue in designing data quality tools is related to the detection of (conditional) data dependencies violations. An appropriate method for the domain of relational databases consists in generating sets of SQL queries from CFDs and CINDs. This is mainly motivated by an ease of maintenance of the dependencies and of integration in RDBMS. The execution of these queries on a database instance permits to identify data items that are violating these constraints. Again, this issue has only been addressed for CFDs [BFG+07] and to the best of our knowledge, no methods have been proposed for CINDs.

This issue was addressed for our two forms of CINDs with methods that automatically generate SQL queries for the discovered CINDs. Furthermore, we propose methods generating a single SQL query for a set of CINDs that hold other the same couple of relations. This query is generated from a merge of all the pattern tableaux of this set of CINDs.

Finally, we tackle two main issues: the classical detection of CIND violation (i.e. correctness of underlying dataset) and the less frequently addressed issue, at least using conditional data dependencies, of missing values (i.e. completeness of dataset). We have designed a solution based on generating SQL queries to detect missing values. Again this query generation exploits both the embedded IND and the pattern tableau of a CIND.

The main difference lies in the direction of the embedded IND of a CIND. Until now, the SQL queries we have generated present the following pattern: (1) the relation of the nesting select query consists of the left hand side relation of the embedded dependency and (2) the relation of the nested select query corresponds to the right hand side relation of the embedded dependency. That is the query is generated by exploiting the traditional direction of an IND. Our detection of missing data solution generates queries based on the inverse of the embedded IND of a CIND. Investigations on exploiting inverse of rules have already been conducted. For instance, [LMSS95] presents an algorithm to answering queries using views based on the inversion of rules.

$cind_2^-$: (atcDrug(cip,atcCode) $\subseteq$ drugContra(cip, atcCode), T_2^-)

T_2^- :

Cip	AtcCode	Cip	AtcCode
_	N02CC03	_	N06AF05
_	R05DA09	_	N06AF05

Figure 4.6: CFD and CIND examples in the medical domain

The embedded INDs of our CINDs are restricted syntactically and this supports a simple inversion plan. The inversion is performed in two steps: (1) inversion of the embedded IND and (2) inversion of the pattern tableau T_p. The first step requires to move the relation and attributes of the right hand side of the embedded IND to the left hand side and vice versa. The second step is also a simple inversion to the relations, attributes and tuples of T_p from the right hand side to left hand side and vice versa. Anyhow, a special attention is given on the attributes of Y_p which are containing disjunctions: each disjunct generates a new pattern tuple and all other constants and variables of the original tuple are copied. Figure 4.6 illustrates this approach with a presentation of the inverse $cind_2$.

We have also proposed a solution to optimize the SQL-based methods by generating a single SQL query for a set of CINDs that hold other the same couple of relations. The method we have proposed enables us to detect violations of a set of CINDs, i.e. it identifies tuples of R1 that do not have counter part in R2. But this solution does not enable to explain the violation since many patterns are bundled in a given SQL query. The solution consisting of multiple SQL queries, i.e. one for each CIND, enables us to explain the violation and thus may be used to guide the end-user in cleaning the database. It is up to the creator of a data quality tool to select one of these 2 approaches, that is whether she wants to provide an explanation solution or not.

In the next section, we introduce an extension of our data quality work on conditional dependencies to the field of ontologies.

4.3 Conditional dependencies in the context of ontologies

The quality of drug databases can be considerably improved with the usage of conditional dependencies. Considering the results of [FGLX09] on CFDs and [Cur09a] on CINDs, it is now possible to extend existing data cleansing tools.

Moreover, in several application domains, e.g. medicine, geography and biology to name a few, information is inherently hierarchical and many standards propose graph structures to represent them, e.g. the Anatomical Therapeutic Chemical classification (ATC) or the European PHarmaceutical Market Research Association (EphMRA)'s terminology in pharmacology. Integrating a hierarchical structure at the core of conditional dependencies representation enables us to reduce the number of stored dependencies and to accelerate the detection of their violations. Although many attempts to add constraints to ontologies have been proposed, we do not know of any relying on conditional dependencies.

The main idea behind our approach consists in improving the data quality of

the relational databases using ontologies by representing conditional dependencies with queries executed over the ontology. A main issue consists in serializing the conditional dependencies in a query language compliant with the ontology formalism we are dealing with, e.g. RDFS and OWL. We propose two different approaches for representing conditional dependencies: (i) using SPARQL in a standard RDFS/OWL context and (ii) using SparSQL, an epistemic query language developed in the context of OBDA. A main advantage of both approaches is to minimize the set of queries needed to detect violations by relying on standard DL reasoning services and ontology hierarchies.

4.3.1 Conditional queries as SPARQL queries

The ontologies of the Semantic Web are generally exchanged using an RDF format. In a nutshell, RDF (Resource Description Framework) is a directed, labeled graph data format which is composed of triples, i.e. subject, predicate and object. For the purpose of querying RDF triples, the W3C has published a graph-matching query language called SPARQL (Sparql Protocol And RDF Query Language). In this context, a SPARQL query consists of a pattern which is matched against an RDF graph and the values obtained from this matching are processed to give the answer. Such a query has three parts: (i) a pattern matching part which includes several interesting features, e.g. filtering, (ii) solution modifiers like distinct, order, limit, etc. (iii) output of the query. The execution of these queries will highlight inconsistent individuals of the knowledge base and which can be easily transposed to tuples in the database (using DBOM).

The setting of violation detection is the following: a knowledge base K and sets Σ of conditional dependencies where $\Sigma = \Phi \cup \Psi$, with Ψ and Φ sets of respectively CFDs and CINDs. Intuitively, we want to identify all inconsistent ontology concepts which are violating Σ. To do so, we consider the concept and individual generation approach used in DBOM consists in using the *dbom:id* property to map primary key to certain objects of the ABox. We are using this representation to identify an object causing an inconsistency with respect to our conditional dependencies. Remember that DBOM supports compound primary keys, i.e. primary keys composed of several attributes, and our tuple identification system takes benefit of it.

The mapping between database and ontology entities is exploited by our query generation solution. We consider a mapping function `M` which relates elements from the relational database to elements of the ontology. For instance, for a relation `R`, the corresponding ontology concept is `M(R)` and similarly for both types of properties.

For both CFDs and CINDs, we tackle the issues of detecting of CFD/CIND violations as well as identifying object/tuples with missing values. We propose SPARQL query generations for both of these approaches. The SPARQL queries generated by our system returns a single identifier in the SELECT clause and can contain in the WHERE clause the following possibilities: (i) triples with variables (starting with a '?' symbol'), predicate and concept names, either starting with a 'rdf:' namespace or a ':' symbol to indicate that it tackles the current ontology; (ii) FILTER operations which enable to test the equivalence of a variable with a data value and (iii) OPTIONAL with FILTER and bound patterns to support a negation as failure approach and do not force all the

query pattern to hold. Finally, different triple patterns are generated whether a database relation is mapped to a concept or to an object property. Typically, a mapping to a concept will generate a triple of the form: $?x\ rdf:type\ M(R)$, while a mapping to an object property will generate: $?x\ :M(R)\ ?y$.

4.3.2 Conditional queries in an OBDA context

Moreover, we considered a similar approach in the context of the Ontology-Based Data Access (OBDA) approach [PLC+08]. OBDA tackles the issue of relating databases to ontologies by providing (1) a conceptual view over data repositories and (2) inference enabled query answering solutions. It is based on the DL_Lite family of Description Logics (DL) [BCM+03] which contributed to the definition of the OWL 2 QL profile [MGH+08].

We consider that OBDA proposes a nice test bed for conducting ontology-based data quality experiments on relational databases. To the best of our knowledge, this work is a first attempt to use the reasoning services and query answering solutions of an OBDA system in order to improve the data quality of relational databases. This can be performed by checking the satisfaction of a set of constraints over database instances. Concerning the representation issue of conditional dependencies, since OWL 2 QL does not support the introduction of individual names in concept descriptions, we adopted a query representation. But since most OBDA systems do not support negation as failure patterns in SPARQL (i.e. through the introduction of negation and bound operators), we have selected the SparSQL epistemic query language [CPP+08]. Hence discovered conditional dependencies need to be translated into SparSQL.

Our translation technique works as follows: for each CFD (resp. CIND) we use the mappings to search for the property mapped to each attribute in embedded dependency. This is performed by selecting the mapping assertions whose SQL queries contain these attributes as distinguished variables, hence they must be mapped to ontology elements. Then we search for the concepts associated to these properties using the identification functions provided by the OBDA system. Given this concept/relation binding, we can compact conditional dependencies that have common attribute sets and then search for the most common specific super concept. This is performed by assuming a covering constraint over sub concepts and using a standard DL reasoner. A straightforward syntactical generation of *SparSQL* queries terminates the translation.

4.4 Presentation of papers

[CJ07b] **Olivier Curé** and Robert Jeansoulin. Data quality enhancement of databases using ontologies and inductive reasoning. In OTM Conferences (1), pages 1117–1134, 2007.

This paper present the induction-based approach of enriching a classification into an expressive ontology. The main algorithm presented in this paper, namely IBOE (Induction-Based Ontology Enrichment), operates in a top-down manner on concept hierarchy and searches to enrich each concept by retrieving values from automatically generated aggregate queries. Methods to clean the database data are also proposed (concept and attribute centric approaches)

and evaluations have been performed on pharmaceutical information. These database repairing solutions are currently used in production on the databases we are maintaining for our self-medication application.

[Cur09a] **Olivier Curé**. Conditional inclusion dependencies for data cleansing: Discovery and violation detection issues.
In QDB workshop at VLDB, 2009.

In this paper, several forms of CINDs are presented and two particular forms, denoted `pConst` and `fullConst`, are studied. They essentially differ on the areas where constants are allowed in the dependencies. Then two techniques, one for each CIND form, are proposed and evaluated. Given a set of such CINDs, a method to discover their violations in a database instance is presented. They take the form SQL queries which are automatically generated by the system. The queries created aim to detect inconsistent data as well as missing values. Finally, a method is proposed to generate a single query for a set of CINDs that hold over the same couple of relations.

[Cur09b] **Olivier Curé**. Improving the data quality of relational databases using OBDA and OWL2QL.
In OWLED, 2009.

In this paper, we consider a novel aspect of OBDA systems: their reasoning services and query answering solutions could be used to improve the data quality of the source databases. This can be performed by checking the satisfaction of a set of constraints over database instances. The constraints considered in this work are CFDs and CINDs. Concerning discovery, we claim that it is more efficient to discover conditional dependencies directly from the database sources and thus enjoying existing optimized implementations. Concerning representation, since OWL 2 QL does not support the introduction of individual names in concept descriptions, we also adopted a query representation. Since most OBDA systems do not support negation as failure patterns in SPARQL (i.e. through the introduction of negation and bound operators), we have selected the SparSQL epistemic query language [CPP$^+$08]. Thus conditional dependencies are provided by an external solution and we need to translate them in SparSQL. In order to obtain a minimal set of these queries, this translation exploits standard DL reasoning services, i.e. concept subsumption.

[Cur10a] **Olivier Curé**. Improving the Data Quality of Drug Databases using Conditional Dependencies and Ontologies
Accepted and to be published in 2010
in ACM Journal of Data and Information Quality

This paper details my contributions on enhancing data quality of databases using an ontology with conditional dependencies (CFDs and CINDs). Since this paper was a response to a special issue call on healthcare systems, it focuses on drug databases and associated ontologies (designed using DBOM) already encountered in this dissertation: EphMRA and ATC.

4.5 Conclusion and perspectives

The solutions presented in the chapter enhance the data quality of relational databases using operations at the ontology level. The differences between these solutions are mainly related to the representation of the constraints used to detect incoherent and missing information. In the inductive approach, we store dependencies at the ontology level and this may interact with standard inferences. In the conditional dependency based approach, the constraints are represented at the query level. Its main advantage consists in the ability to define a trigger based approach when an update operation is performed at the database level, then a set of queries are executed at the ontology level to validate the new information. This approach is similar to the solution presented in the synchronization trigger based solution of Chapter 2. Another advantage is to propose a more efficient maintenance solution of the set of constraints. This is due to the localization of the dependencies in a set of queries rather than being spread across all the ontology.

Future work on the data quality aspect concerns studying Conditional Exclusion Dependencies (CEDs). Their discovery is a hard problem since false information are generally neither stored in databases or knowledge bases. Hence their discovery must be performed manually by a team of domain experts or require external knowledge. Anyhow, together with CFDs and CINDs, they can be very useful in achieving a high degree of quality in databases and are needed in various domains such as earth science and medicine.

Chapter 5

Other works involving ontologies

In this chapter, I present two research works performed in the context of a FP6 European project (VENUS) and an ANR project (STAMP). They both consider ontology modeling and reasoning within specific contexts, respectively archaeology and dynamic landscape modeling.

5.1 Modeling an application ontology of underwater archaeological surveys of amphorae

This section describes knowledge representation issues encountered on the VENUS European Project (Virtual ExploratioN of Underwater Sites, IST-034924). This project is a collaborative venture aiming to bring together archaeological and scientific methodologies with technological tools for virtual exploration of deep underwater archaeological sites.

The overall objective of archaeology is to improve our knowledge of the past. Today, underwater archeology opens, from the deep past of the sea, a direct route to shipwrecks, complex works testifying on the wealth and the diversity of exchanges between human beings. Although underwater archaeology shares common techniques and standards with its ground counterpart, it has some specificities which are related to the technical conditions of operation, i.e. environment, weather conditions and logistics, nature of discovered sites and the less significant influence of stratigraphy. A direct consequence of this situation is that the knowledge of the studied items is both provided by underwater archaeology and photogrammetry measures. In order to enable a virtual exploration of the studied underwater sites, we need to represent and archive digital artifacts corresponding to the studied items. Moreover, the information about these items are usually uncertain, inaccurate or imprecise. Therefore, this aspect has to be carefully tackled within a computational point of view.

The first task performed by the archaeologists is the interpretation of the observations acquired within the surveying process. The archaeologists provide a meaning to the observations partially based on the photogrammetry measures. During this interpretation phase, the archaeologists handle two important as-

pects: by nature, archaeological information is incomplete, and in case of well known artifacts, like amphorae, some features are not observable and require inference rules, often vague for deducing their value.

Since objects, e.g. amphorae, belong to typologies that are structured within hierarchies, underwater archaeological knowledge is structured. From these hierarchies, partial pre-orders or total pre-orders may be defined on the features of the objects. Moreover when dealing with underwater archaeological surveys, the acquisition process, i.e. photogrammetry, has to be taken into account in order to define a suitable knowledge representation. Archaeological information's structured nature leads to represent the generic knowledge by means of ontologies. In order to represent the underwater archaeological surveys, we construct a knowledge base consisting of generic knowledge and observations. The generic knowledge consists of an application ontology stemming from both underwater archaeology knowledge and photogrammetry as well as of constraints, e.g. integrity constraints, domain constraints. Photogrammetry measures provide the observations.

In order to provide a consistent representation of the underwater archaeological surveys a special attention has to be paid to consistency checking. Focusing on reasoning processes, we have selected the DL formalism to represent our knowledge base. This formalism responds to our need in terms of expressiveness, sound and complete inference procedures and the availability of efficient and reliable tools. Interoperability is an important issue for our ontology. The application developed within this project should require several forms of interoperability with existing cultural heritage related systems. To this end, we mapped our ontology to ISO CIDOC Conceptual Reference Model (CRM). And as this mapping was not sufficient to express all the notions of our ontology, we extended CIDOC CRM with new concepts and properties. This mapping and extension is processed on the OWL DL version of CIDOC CRM and we thus enjoy a DL with all its associated reasoning facilities.

The International Committee for Documentation's Conceptual Reference Model (CIDOC-CRM) is an object-oriented conceptual model which has been developed by the ICOM/CIDOC documentation Standards Group to provide an ontology for cultural heritage. The CIDOC-CRM, initially created for museums domain by ICOM, found applications across the broader cultural heritage field. However, we do believe that it is able to represent not only the objects, but also the way they are collected, identified and assigned with various more or less imprecise measurements, and hypothetical attributes concerning their age and origin. When these assignments are made concurrently by different observers, mixing human people and artificial devices, it becomes very important to track and mark the data collection process all along. After a decade of standards development work by the CIDOC, it became an international ISO standard (ISO 21127:2006) that establishes guidelines for exchange of information between cultural heritage institutions. CIDOC CRM distinguishes Entities, which are classes of physical or immaterial instances, and Properties, which are binary relations between classes, possibly quantified by cardinalities.

5.1.1 Application ontology

In this section, we only present an extract of our application ontology of underwater archaeological surveys of amphorae. This extract details the kind of

items encountered on the first underwater site at Pianosa, i.e. amphorae which are characterized by numerous features, each feature belonging to a domain of several values.

Some of the knowledge we represent in our ontology is provided by photogrammetry measures. They come from the ARPENTEUR (ARchitectural PhotogrammEtry Network Tool for Education and Research)[DG98] tool which is a set of software applications for photogrammetric measures targeting archaeologist and architect end-users. The main idea of this tool lies in a measure process guided by the application domain's knowledge and thus does not require specific knowledge about the concepts of photogrammetry. For example, the description of the shape of a model of an amphorae gives the part of the amphorae that are measured. The results of this step are produced as XML documents and files dedicated to 2D/3D visualization (SVG, VRML).

Following Guarino's classification [Gua95], the proposed ontology corresponds to an application ontology where the underwater archaeological knowledge is captured with a domain ontology that describes the vocabulary relating to amphorae while the knowledge relating to photogrammetric data acquisition process is captured with a task ontology. We take inspiration from the method described in [NM] to define concepts, attributes of concepts and relations between concepts.

A first set of the main concepts encountered in the ARPENTEUR tool are:

- ID_ITEM defines an identified object studied during an exploration.
- CONNECTABLE_OBJECT defines an object which is able to communicate with other objects.
- ITEM is an object which is identified and able to communicate with other objects. It is a sub-class of both previous classes.
- SPATIAL_ITEM defines an object which has spatial properties.
- MEASURABLE_ITEM defines an object which can be measured using photogrammetry, in particular it has measurable areas.
- ARCHAEOLOGICAL_ITEM defines an object which has archaeological properties.
- AMPHORA_ITEM defines an object which is an amphora.
- AMPHORA defines a whole amphora according to Arpenteur's point of view. An amphora is considered as a whole amphora if it can be completely rebuilt.
- AMPHORA_FRAGMENT defines an object which is a fragment according to Arpenteur's point of view, that is to say it cannot be completely rebuilt.
- DRESSEL20, BELTRAN2B,.. are amphorae which can be distinguished by the set of their respective property values. Thus we represent them as sub-concepts of the AMPHORA concept.

In order to characterize the previously described concepts, we define another set of concepts. These concepts are considered like attributes of the concepts defined in the previous list:

- METROLOGY defines some morphological attributes common to all measurable items.
- AMPHORA_METROLOGY defines some morphological attributes specific to amphora items.
- SURVEY_SESSION_DATA defines some information about the survey session in Arpenteur.
- DOCUMENTATION defines some archaeological information about an item.

The first set of concepts can be organized following a concepts hierarchy which is displayed in Fig. 5.1. This figure also contains some properties which can be distinguished into data type properties, i.e. properties with a concrete co-domain, and object properties, i.e. individuals of some ontology concepts as co-domain. For instance, the "has_amphora_metrology" property is an object one as it enables one to relate AMPHORA_ITEM individuals to AMPHORA_METROLOGY individuals. Accordingly, the "name" property of the ID_ITEM concept is a data type property whose co-domain is a string of characters corresponding to the name of the item.

Now that we have defined the organization of concepts and properties of our application ontology, we can define some constraints. These constraints introduced by terminological axioms make statements about how concepts and properties are related to each other. Their goal is to provide consistency restrictions on admissible KBs. We would like to distinguish some particularly important constraints in the context of the VENUS project:

- real world constraints, e.g. a distance and a volume can not have negative values,
- conditional constraints which are applied on some (data type) properties of the METROLOGY concept and its sub-concept AMPHORA_METROLOGY, e.g. height, etc.. According to the typology of the amphora, the values of these attributes should be included in an interval of 20 percent around the theoretical value. For example, the height of a short Dressel 2-4 amphora should be included between 0,68 m and 1,02 m[1].
- extrinsic constraints which correspond to the fact that two objects cannot have the same identifier that is to say the `idn` attributes of two concepts ID_ITEM has to be different for two distinct objects, and that the amphora should respect the following spatial constraint: the 3D representation of different objects cannot intersect.

5.1.2 Mapping and extension to CIDOC CRM

Our goal is now to map the DL TBox of the previous section onto the CIDOC CRM. This mapping supports the generation of a merged ontology. A detailed study of both ontologies emphasize an epistemological overlap as well as some

[1]The theoretical value used to determine this interval was given by an archaeologist who took part to the Pianosa mission.

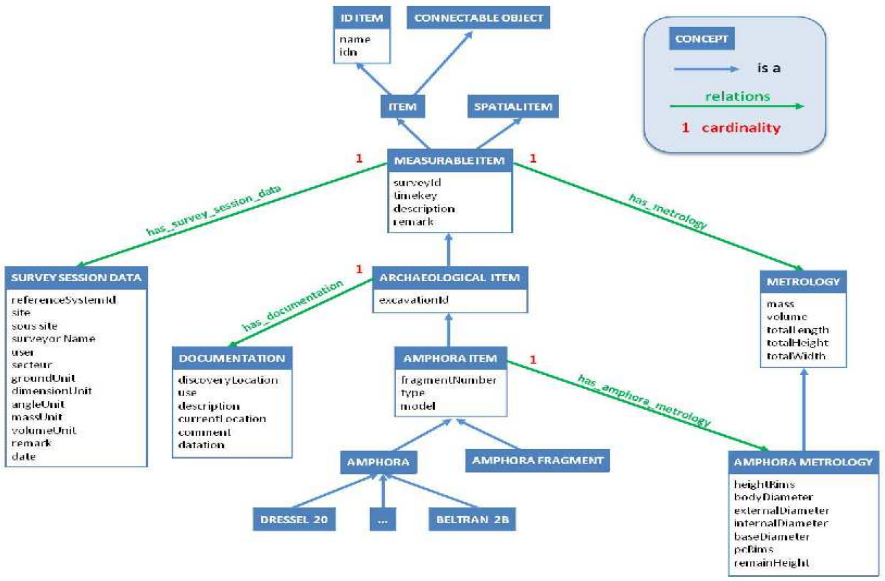


Figure 5.1: Application ontology adapted from Arpenteur

discrepancies. This means that some of the concepts and roles of our application ontology can be represented using CIDOC CRM's TBox elements. These correspondences are represented by mapping rules relating elements of both ontologies with equivalence ($\equiv$) and subsumption ($\sqsubseteq$) relationships. For the remaining concepts and roles of our application ontology, those that do not have any correspondence with CIDOC CRM, it is necessary to extend the TBox of the latter. Toward this extension, a special attention is given on retaining the concept and role hierarchies of CIDOC CRM.

To clarify our mapping and extension methodology, we denote by $TBox_a$ the application ontology's TBox and state that $TBox_a = TBox_c \cup TBox_{nc}$ where $TBox_c$ denotes the subset of $TBox_a$ that has at least one correspondence with an element of CIDOC CRM's TBox and $TBox_{nc}$ the subset of $TBox_a$ that has no correspondence with any elements of CIDOC CRM. Additionally, we can state that $TBox_c \cap TBox_{nc} = \emptyset$. The method is necessarily manual since the consideration of the meaning of concepts and roles of both ontologies is paramount to the design of a suitable mapping. For instance, the `rdfs:comment` annotation of CIDOC CRM were actively used to understand the precise meaning of each ontology elements. Anyhow, our method ensures an efficient and correct mapping generation.

The objective of our method is to maximize the size of $TBox_c$, and thus minimize the size of $TBox_{nc}$, in order to enable an increased interoperability with other ontologies mapped to CIDOC CRM. The method starts with the concept hierarchy of $TBox_a$. We proceed with a top-down approach on the concept hierarchy. For each concept or group of concepts, we search for a correspondence to an entity of CIDOC CRM. If a correspondence is found then we move this source concept to the set $TBox_c$ otherwise we move it to $TBox_{nc}$. Ideally, if a $TBox_a$ concept C_0 corresponds to a CIDOC CRM entity E_0, then

its sub-concepts should be mapped to sub-concepts E_0. Once all the concepts of $TBox_a$ have been studied, we can extend CIDOC CRM with the concepts in $TBox_{cn}$.

At the step of identifying a correspondence, we also compare the set of properties needed by our application ontology and the proposed set of properties of the studied CIDOC CRM concept. In many cases, correspondences between properties where found and exploited in the mapping. For properties of our application ontology not matched to any CIDOC CRM properties, we decided to extend its set of properties. Properties which needed to be extended on to CIDOC CRM where related to identifiers and descriptive contents for data type properties. Concerning object properties, extensions where needed for extended concepts as well as between CIDOC CRM concepts.

We now present several mapping rules resulting from the application of this method on Fig. 5.1's ontology:

1. $E_19 \equiv ID_ITEM \sqcap CONNECTABLE_ITEM \sqcap ITEM \sqcap SPATIAL_ITEM$
2. $E_22 \equiv MEASURABLE_ITEM$
3. $E_84 \equiv ARCHAEOLOGICAL_ITEM$
4. $E_16 \sqsupseteq METROLOGY$
5. $E_54 \sqsupseteq TOTALWIDTH$
6. $P_48 \equiv idn$
7. $\top \sqsubseteq \forall hasSurveySessionData.E_22$
8. $\top \sqsubseteq \forall hasSurveySessionData^{-}.E_7$

Axiom (1) states that the CIDOC CRM's E_19 (`Physical Object`) concept is equivalent to the Item block of our ontology since it is defined as class containing items of a material nature that are units for documentation and have physical boundaries that separate them completely in an objective way from other objects. Axioms (2)-(5) provide equivalence and subsumption relationships between `Man made object` (E_22), `Information carrier` (E_84), `Measurement` (E_16 - actions measuring physical properties) and `Dimension` (E_54 - measured quantifiable properties) CRM concepts and respectively MEASURABLE_ITEM, ARCHAEOLOGICAL_ITEM, METROLOGY and TOTALWIDTH of our ontology. Note that the TOTALWIDTH which was an Amphora data type property in our application ontology was transformed into a concept in CRM (and this is the case for all attributes of METROLOGY and AMPHORA METROLOGY). This transformation is motivated to benefit from the quantifiable properties of the `Dimension` concept, e.g. value and unit storage. Being defined as a sub concept of E_84, all indirect sub concepts of ARCHAEOLOGICAL_ITEM, namely AMPHORA and AMPHORA_FRAGMENT, DRESSEL20, etc., also belong to E_84's concept hierarchy. Moreover, they are also in the concept hierarchy of E_22, E_19 since CRM states that $E_84 \sqsubseteq E_22 \sqsubseteq E_19$. Considering property mappings, axiom (6) provides an example where CRM's P_48 (`has_preferred_identifier`) is defined to be equivalent to the `idn` attribute of the ID_ITEM concept.

Axioms (7) and (8) present a CRM extension at the property level. That is a property defined in our application ontology, namely $hasSurveySessionData$, now relates concepts E_22 (as domain) and E_7 (activity - as co-domain).

5.1.3 Reasoning with the application ontology

In this section, we highlight on the reasoning-based features implemented for the VENUS project. An overall architecture of the project is proposed in Fig. 5.2 (arrows are numbered to understand processing flow). Basically, the `Conceptual description` (whose processing is described in Section 5.1.2), `Measure` and `Data` boxes support the generation of a KB whose TBox corresponds to an ontology merged from the application ontology and CIDOC CRM based on our mapping and ABox is instantiated from measured items.

The main reasoning procedure consists of knowledge base consistency checking which is programmed using the APIs provided by standard OWL reasoner, e.g. Pellet [SPG+07]. Modern reasoners provide for a so-called concrete domain support, such that constraints over values from concrete domains (e.g. integers and reals) referred to by multiple individuals can be postulated. For instance, consider that during an exploration an archaeologist states with the help of ARPENTEUR that a given amphora A_1 has type 'DRESSEL20'. The values associated to this amphora instance are stored in our knowledge base and relate to the properties of the concepts METROLOGY, e.g. height, and AMPHORA_METROLOGY, e.g. rim height. Our knowledge base states that for a DRESSEL20, some constraints must hold, e.g. the 'height' value must be between 0.53 m and 0.79 m. If the value associated to `height` for the individual A_1 does not belong to this interval, the knowledge base is considered inconsistent. Using the Pellet reasoner, our system can easily identify instances responsible for the knowledge base inconsistency, pinpoint the properties of a particular instance causing the inconsistency and also propose knowledge base repairing facilities by proposing a coherent amphora type for an inconsistent instance or providing correct interval values for certain properties of an instance.

Fig. 5.2 also presents the final module of the VENUS system: Virtual reality. Intuitively, it proposes an augmented reality environment which enables end-users to visualize and navigate along the underwater archaeological site. Note the relationship between the reasoning and virtual reality boxes which enables users to highlight on site the inconsistent instances.

5.1.4 Future works and conclusion

In this work, we have presented the construction of a knowledge base that represent the domain of underwater archaeology surveys of amphorae. This enables us to support reasoning, among which consistency checking is one of the most important. For interoperability reason we mapped our application ontology to the CIDOC CRM and extended it for some peculiar aspects.

Since archaeological information is by nature incomplete, uncertain, inaccurate and evolutive, non-monotonic reasoning has to be performed : revision when new evidences contradict previous hypothesis, update when the archaeological site evolves according to weather conditions or the evolution of the excavation process, fusion in case of different sources of information. The CIDOC CRM choice provides an easy way to represent "positive knowledge", but we

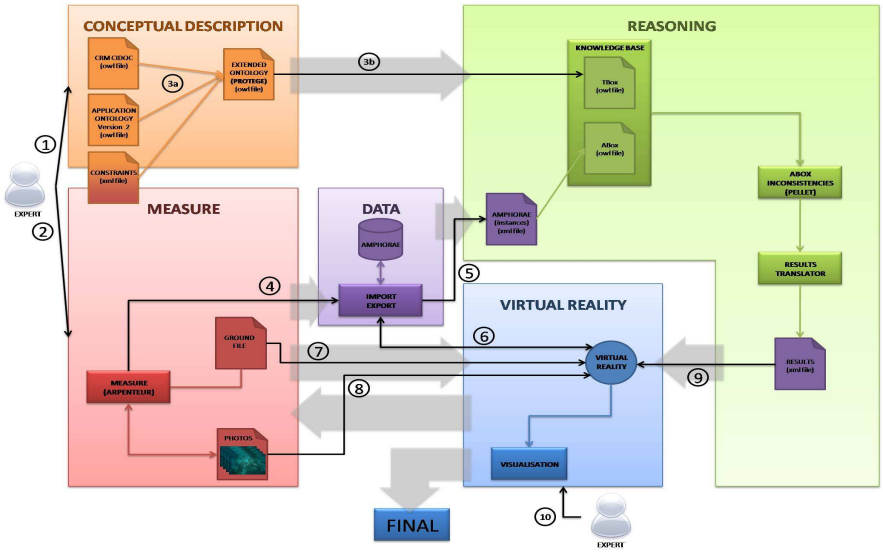


Figure 5.2: Synthetic view of reasoning with application ontology

have also mentioned the need to represent constraints between features, and any negative knowledge that can be explicit: forbidden values, forbidden relations, etc. We must also ask: How to manage the information about data quality? How to manage the uncertainty in this overall context? Hence, the representation of constraints, of differing data quality, of preferences, etc. is left to the user responsibility. Restoring consistency by accepting or refusing parts of the registered knowledge, is left to further reasoning processes.

5.2 Ontologies to design a Domain Specific Language

Model Driven Engineering (MDE) is a popular trend in software development where the primary artifact is not a program, but a model which can potentially be instantiated. MDE can be used to design simulation applications for complex domains such as life sciences, health care, biodiversity and ecosystems. In these fields, running simulations enable to understand how domains evolve in particular situations as well as to predict their behavior in some given scenarios. For this research, we focus on Dynamic Landscape Modeling (DLM) but our approach can easily be generalized to other domains. DLM is a challenging field because space and time in multiple scales need to be represented. Although many modeling formalisms could be used to model this domain (e.g. agent-based, cellular automata), we propose a novel approach based on Domain Specific Languages (henceforth DSL). In [DSP+09], we motivated this approach and presented the main concepts of a prototype language named Ocelet.

A main advantage provided by Ocelet is its modeling flexibility since new models can either be built with new DSL primitives or from selecting existing primitives stored in repositories. The design of primitives is facilitated by the

reduced number of Ocelet's language elements: entity, relation, service, scenario and datafacer. These five core elements have been identified and defined to enable, through their interactions, the expression of most scenarios encountered in dynamic landscape modeling and other complex domains requiring the representation of spatial and temporal information.

The modeling framework surrounding Ocelet consists of a model building environment, a code generator and compiler, and a program execution platform. The program execution platform is an adapted environment for applications created with Ocelet in the landscape modeling domain. The main purpose of this environment is to guarantee the ability to build applications with the capacity to themselves dynamically adapt to their evolution and change in its execution context. It also takes into account the aspects of a distributed execution. The approach used to realize this environment is based on a component model to better separate the functional and non-functional aspects, as well as on the Service oriented computing paradigm. Some of the early experiments conducted with this environment enabled us to define a road map for the following three Ocelet extensions.

First, reasoning facilities should be supported in the framework. This is important since modeling a domain results from a collaborative work where each team of domain experts proposes a model related to its own domain of expertise. Once all expert teams have provided their (sub)models, the complete model of the problem can be built by their merge. This approach raises the issue of consistency of the local models as well as of the (merged) global model. For instance, a merge of consistent local models can generate an inconsistent global model. In order to perform these consistency checking of models, we favor a declarative, logic-based approach since it generally provides better explanations of inferences and offers re-usability possibilities.

A second aspect concerns the implementation of a Graphical User Interface (GUI). This software component should ease the design of models much like a UML class diagram helps in the definition of classes and methods in object-oriented programming. An important requirement for this GUI is to support interactions with reasoners.

Finally, we believe that the adoption of our approach depends on the availability of implemented, ready to use DSL primitives. This means that domain experts are not compelled to systematically design their models from scratch. Instead, they should have the opportunity to integrate existing models in a user-friendly way. This raises the following issues: what pivot model formalism do we select for our framework and is there a set of transformation rules available to this formalism? We have selected the OWL since it is a logic-based formalism with a large repository of available models with existing reasoners and APIs.

5.2.1 The Ocelet DSL

The Ocelet DSL consists of 5 basic elements of the language which are meant to interact with one another:

- An **entity** can either be an atomic building block of the model or a container containing other entities. In the latter case, we call such an entity a composite. For instance, a landscape can be modeled by a composite entity containing (atomic) pond entities. An entity is able to store in-

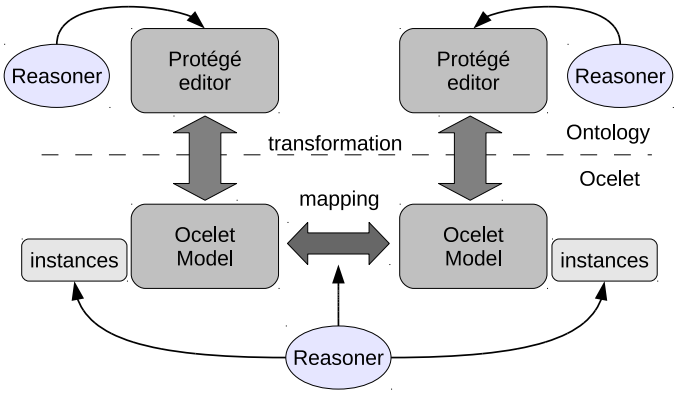


Figure 5.3: Architecture of Ocelet model management

formation using attributes, e.g. a pond has a certain depth. An entity communicates with its environment through input an output ports named services.

- A **service** is a communication port of an entity. It is an input service when it accepts input values or events from other entities and it is an output service when it exports values or events to other entities. A Pond entity can have an *evaporation* service in and *pondState* service out. Input services link to output services by means of relations.
- A **relation** is the expression of how a set of entities relate to each other at a given moment. Hence a relation definition describes which entities can be related and also contains a functional part which describes what is happening when they interact.
- A **scenario** expresses the spatial and temporal internal behavior of a composite entity. This is performed by managing the entities and relations it contains. For instance, a scenario can activate all relations at every given time step.
- A **datafacer** is a component through which entities access data. The aim of datafacers is to abstract access methods to the entities. In the domain of geographical information, a datafacer can take the form of an external database, a satellite image repository or an internal log file.

In [DSP+09], we have shown that these five elements are trade off between ease of modeling and expressive power to represent situations encountered in dynamic landscape modeling.

5.2.2 Ontology-based Ocelet modeling architecture

This section concentrates on Ocelet's model management and its interactions with the ontology development environment (Protégé editor and Pellet reasoner). Figure 5.3 proposes a general overview of its components. The dashed line separates these two environments where the upper part is concerned with ontologies and lower one with DSL primitives.

At the ontology level, OWL ontologies are designed by experts using the Protégé editor. During this process, it is possible at any moment to check the consistency of the current ontology and classify the concepts defined. These operations are performed by an external DIG (DL Implementation Group) compliant reasoner. In fact, the communication between Protégé and the reasoner is supported by the DIG http service. We consider that ontologies validated by domain experts are necessarily consistent.

Once the experts are satisfied with their ontologies, it is time to transform them into corresponding Ocelet models. This step is represented by the transformation arrows crossing the dashed lines. Note that the arrows are bidirectional meaning that it is also possible to define an OWL ontology from a model designed directly with Ocelet. This process is completely automated, meaning that it does not require interactions with end-users, and solely rests upon the mapping defined between OWL and Ocelet elements. For each OWL document, an Ocelet model is created. In order to run the desired simulations, a merge of all these Ocelet models is needed. The main component supporting this operation is a set of correspondences/associations defined by domain experts between Ocelet primitives. These associations correspond to (partial) order relations between entities, attributes and relations. This means that one can state that an entity in one model is equivalent, a specialization or a generalization of an entity in another model.

Although each Ocelet model is supposed to be locally consistent (since their corresponding OWL ontologies are consistent), the merged Ocelet model, i.e. union of all Ocelet models, can be inconsistent. This is generally caused by the order relations defined between Ocelet elements and their respective semantics. Hence it is necessary to check the consistency of the merged model. For this task, we are using the same reasoner as exploited at the ontology level, namely Pellet. This can be performed by exploiting the OWL API [2] which enables to generate internal structures handled by Pellet corresponding to OWL concepts and properties from any formalism. Thus, the global Ocelet model is transformed into Pellet structures and a consistency checking inference is processed. One particularly important aspect about the selection of Pellet is its ability to provide explanations about discovered inconsistencies. In general, they are sufficiently detailed to enable the ontology designer to perform a repair on Ocelet primitives or equivalence relations. But this process is very hard to automate since repairing a model is usually nondeterministic, i.e. several operations on the structure of the models can lead to a new consistent model but only a few of these repairs satisfy the semantics required by the domain experts.

Finally, other forms of inconsistencies can occur at runtime when the different steps of a scenario are processed. Each scenario step potentially generates or modifies the set of individuals of the universe of a Ocelet simulation. This is for example the case in the Lokta Volterra use case we presented in [DSP+09] when the set of predators and preys is modified by differential equations at each time step of the scenario or in the Rift Valley Fever example also introduced in [DSP+09]. The temporal and spatial considerations implied by each step of a scenario provide a dynamic aspects to the universe of individuals. Hence, in time and space a given individual may change its type, i.e. Ocelet entity. Lets consider a simple example where a function generates an individual a with

[2] http://owlapi.sourceforge.net/

age attribute equal to 10 years old, hence with type *Child*. After successive scenario steps, *a* will turn 18 years old and due to definitions of the *Child* and *Adult* concepts/entities, it will be inconsistent to consider him a child anymore. These inconsistencies need to be considered by Ocelet's model management but they are different from the other kind of inconsistency introduced previously since they involve information about individuals. In analogy with DL, this corresponds to the difference between consistency checking given $\mathcal{T}$ (TBox only) and given $\mathcal{K}$ (TBox together with ABox). The Pellet reasoner handles both kinds of inconsistency checking, hence we can use it through translations via the OWLAPI.

Ontology generation An end-user may generate an OWL ontology using different approaches. They all make intensive use of the Protégé editor, hence requiring a minimum of training on this software, as well as an associated reasoner (via HTTP communication). All approaches enable us to run standard reasoning services at any moment.

The first obvious manner is to create the ontology from scratch. The end-user generally defines a set of concepts and (data type and object) properties and then refines the concept definitions using operators from the selected DL (e.g. $\forall, \exists, \sqcap, \sqcup, \neg$).

The second solution consists in importing a set of existing ontologies and to fine-tune it by modifying, inserting or deleting some concepts and properties. This approach implies that the domain expert is aware that an ontology covering its domain of expertise is available and valuable. The number of existing ontologies in scientific domains is increasing at an important rate and allows knowledge sharing.

Another solution involves using DBOM and hence creating knowledge bases from data not originally stored in OWL (see Chapter 2)

Presentation of the mapping Given the description of Ocelet's elements, a mapping from some elements of Ocelet is straightforward to elements of an ontology. For instance, it is clear from the definition of an Ocelet entity that it can be mapped to an Ontology concept since they both correspond to unary predicate which constitute a type for a set of individuals. Similarly, an Ocelet's entity attribute correspond to a data type property in OWL. They both correspond to binary predicates where the domain of the property is the OWL concept/Ocelet entity and its range is a value with a given type (numerical, string of characters, date, etc.). In the OWL language, an object property correspond to a binary relation where both domain and range are concepts. This notion corresponds to binary forms of Ocelet's relation. Considering the case of n-ary Ocelet relations (with $n > 2$), a reification mechanism in the ontology, i.e. this relation is reified by means of a new concept and n functional object properties, is sufficient to support the mapping without loss of information.

Concerning datafacers, ontologies defined using DBOM may serve to establish connections between Ocelet entities and databases but this topic will be detailed in an upcoming paper. Finally, services and scenarios do not have obvious counterparts with OWL builder blocks.

5.2.3 Conclusion and future works

The approach we have proposed presents ontological entities as *templates* for DSL primitives. That is, an end-user manipulates a GUI knowledge base edi-

tor, benefiting from interactions with reasoners to classify concept hierarchies, detect unsatisfiable concepts and check consistency, to generate an OWL ontology. Since OWL is not a programming language, these ontologies only serves as a modeling paradigm and need to be merged and translated into our dynamic landscape tailored DSL. The merge step corresponds to defining correspondences between ontology concepts and properties. Then the generated primitives can be enriched with codes written in a programming language. Some of these procedures are mathematical functions supporting the creation of concept individuals. At this stage, interactions with a DL based reasoner can check the consistency of the merged knowledge.

Concerning future works, we would like to run more experiments with domain experts and in particular to get detailed feedbacks from their use of the GUI as well as their understanding of the explanations. This is currently being done in the context of the cooperation with researchers of CIRAD team in Montpellier.

5.3 Presentation of papers

[DSP+09] Pascal Degenne,Danny Lo Seen, Didier Parigot, Rémi Forax, Anne Tran, Ayoub Ait Lahcen, **Olivier Curé**, Robert Jeansoulin
Design of a Domain Specific Language for modeling processes in landscapes
In Ecological Modeling, 2009.

This paper presents and motivates the main elements of the Ocelet DSL (i.e. Entity, Service, Relation, Datafacer and Scenario) in the context of dynamic landscape modeling. The use of Ocelet is illustrated on two distinct examples: Lokta Volterra and a mosquito borne disease use cas in Senegal.

[OC10b] **Olivier Curé**, Rémi Forax, Pascal Degenne, Danny Lo Seen, Didier Parigot, Ayoub Ait Lahcen.
Design of a Domain Specific Language for modeling processes in landscapes
In IARIA MOPAS conference, 2010. To appear

This paper presents a mapping approach between elements of the Ocelet DSL and OWL ontologies. It highlights that available ontology editors and existing ontologies can be used to define primitives of Ocelets. Moreover, an emphasis on the use of reasoning functionalities associated to OWL ontologies is given at both the ontology and instance levels.

[OC10a] **Olivier Curé**, Mariette Serayet, Odile Papini, Pierre Drap
Toward a Novel Application of CIDOC CRM to Underwater Archaeological Surveys
In SWARCH-DL, 2010. To appear

This paper summarizes the work produced within the VENUS project on mapping an application ontology of the domain of amphorae to the CIDOC-CRM standard. It also proposes an inference-based solution to detect and repair inconsistencies of the knowledge base. This repairing solution is related to virtual reality module designed in the context of VENUS.

5.4 Conclusion and perspectives

Funding for both the VENUS and STAMP projects ended recently. Anyhow, we consider that there still is some work to do on both of these projects.

Concerning the VENUS project, we defined several reasoning and revision solutions (DL-based, Removed Sets Fusion and Answer Set Programming) to handle inconsistent knowledge bases. We believe that in-depth comparison of the performances of each of these solutions may help us to define an efficient hybrid solution.

We are currently asking for an extension of the ANR STAMP and are also considering other funding forms. During the 2010 summer, two master students were working at CIRAD to enrich the implemented framework. The paper we presented at the MOPAS conference received the best paper award and we are invited to submit a self-contained, long version of this research for the International Journal On Advances in Life Sciences. In this paper, we aim to present a solution that permits to perform parallel reasoning on sub sets of the knowledge bases.

Chapter 6

Curriculum Vitae

`Olivier Curé` 41 years old, French citizenship

IGM LabInfo CNRS UMR 8049, Team GTMC (Géomatique, Télédétection et Modélisation des Connaissances - Geomatics, Remote sensing, Knowledge modeling)

5bd Descartes 77454

Marne la Vallée Cedex 2 France

phone : 01 60 95 77 21

fax : 01 60 95 77 57

email : ocure@univ-mlv.fr

Web site: http://perso.univ-mlv.fr/ocure

6.1 Education

1999 Ph.D. University of Paris V in Computer Science Artificial Intelligence

Title: IMSA: An Interactive Multimedia System for Auto-medication

Chair: JP. Giroud (U of Paris V), Referees: D. Laurent (U. of Marne la Vallée), C. Carrez (CNAM), Invited: M. Philippon (U. of Paris V), JC. Le Pen (U. of Dauphine)

Supervisor: N.Cot (U of Paris V)

1994 Magistère (postgraduate diploma) from EHEI and University of Paris V in Computer Science

DEA (M.Sc.) from University Paris V in Computer Science and Artificial Intelligence

Research Master thesis: "Programming by Demonstration", supervisor N.Cot (U. of Paris V)

6.2 Positions

Since sept 2002: Associate Professor at University Paris-Est, France

Sept. 1999 to 2002: Part-time teaching at University Paris-Est and freelance application designer/developer for several companies in the medical domain.

6.3 Languages

English (fluent)
French (native language)

6.4 Research activities

My research activities are focused on Knowledge Representation (KR) and are centered on the following directions:

- Integration and exchange of data/knowledge formalized with technologies of the Semantic Web (RDFS and OWL)
- Ontology mediation with the issues of alignment, merging, mapping and matching.
- Efficient storage of RDF triples

6.4.1 Integration and exchange of data

I am interested in integration and exchange of data between structured (e.g. relational databases), semi-structured (e.g. XML) documents and ontologies. A special attention is given to the formalisms of the Semantic Web, i.e. RDFS and OWL. I have tackled some issues of the relational database case with the implementation of the DBOM (DataBase Ontology Mapping) tool: impedance mismatch, synchronization issues between the sources and the target, inconsistency handling with preferences, reasoning aspects involved in instantiation of ABoxes. The approach adopted by DBOM consists of a hybrid solution between data exchange and integration approaches, that is the data available at the sources are copied in the ABox of the knowledge base and the TBox is generated from mapping with the sources. I am currently working on an extension of DBOM where end-users can decide whether they want the ABox data to be materialized or virtualized (i.e. the data remain in the sources and are accessed via queries expressed over the TBox).

DaltOn (Data Logistic and Ontologies based integration) is a project conducted with the team of Prof. Stefan Jablonski at University of Bayreuth, Germany. The main goal of this project is to propose a framework based on process modeling which has three major conceptual architectural abstractions, namely Data Provision, Data Integration and the internal Repository. I was responsible for the design and implementation of the data integration and in particular with the semantic based data integration. We tested our DaltOn implementation with several scientific domains such as medicine, biology and ecology. We are currently extending DaltOn with a data provenance solution which is highly needed by end-users.

6.4.2 Ontology mediation

One of the first aspects we are considering in the Ontology-Based Data Access (OBDA) setting is the automatic generation of mappings from previously defined mappings and constraints defined on the sources and the target.

Lately, I started working on a Formal Concept Analysis (FCA) based solution to merge ontologies. The results I have obtained so far enable to introduce new concepts in the merged ontology and to axiomatize its concepts using the axioms of the source ontologies. The ontologies I have studied are expressed in Description Logics (DL), in particular $\mathcal{ALC}$ but I am aiming at studying more expressive ones.

6.4.3 Knowledge representation

I am working on several projects (VENUS, STAMP, XIMSA) that require to represent and reason with knowledge from several domains, i.e. archeology, spatial and medicine.

In the VENUS project, I worked with researchers at the LSIS laboratory in Marseilles on extending the CRM CIDOC ontology for archeology and participated in the design of several ontology designs in order to reason with them. An important issue in this project consists in handling and revising inconsistencies. During this collaboration, between January 2008 and august 2009, I participated in the redaction of 7 deliverables (D1 to D7).

The core of the STAMP project consists in modeling dynamic landscapes with Spatial, Temporal And Multi scale Primitives. Apart from the knowledge representation challenge, I also collaborate on the design of DSL (Document Specific Language). Within this project, I have proposed an approach that generates the primitives of a DSL from the definition of an ontology. Among different features, this approach enables to check the consistency of the DSL.

As a member and director, of the GTMC research team, I'm working on the representation and fusion of geographical information and knowledge.

Finally, my work on the XIMSA project, and the production version of SantéClair, requires to continuously work on the integration of classification, terminologies of the medical and pharmaceutical domain. On this project, data and meta data represented in the knowledge base are used to enhance data quality

6.4.4 RDF triple storage

The vision of the Semantic Web is becoming a reality with billions of RDF triples being distributed over multiple query able endpoints (e.g. Linked Data). Although there has been a body of work on RDF triples persistent storage, it seems that the problem of providing an efficient, in terms of query performance and data redundancy, inference enabled approach is still open.

Recently, in a joint work with David Faye and Guillaume Blin, we have proposed several storage solutions to effectively retrieve and update information from graphs stored in a column oriented database approach. Considering novel and efficient approaches, we are also studying solutions emerging from the NoSQL community.

6.5 Teaching activities and student supervision

Since 1999, I have taught every year between 220 and 300 hours (for a total of 2800 hours over 10 years). My teaching covers the following domains and levels:

- 1nd year at University: algorithms, C and Java programming languages, Web design, Database.
- 2nd year at University: algorithms, C and Java programming languages, Database.
- 3rd year at University: Database, Logic programming, Java programming language, functional programming.
- 4th year at University: Advanced database, Artificial Intelligence, UML, Java programming language, Web 2.0.
- 5th year at University: Advanced database, Knowledge Representation, Semantic Web, Logics (Propositional, First Order and Description Logics) and Reasoning, Java programming language.

I have used the following tools and technologies during my lectures and assignments:

- Database : Oracle, PostgreSQL, MySQL, Column oriented databases (MonetDB), document oriented databases (CouchDB), Access, SQL, Datalog
- Programming languages: C, Java, PHP, JavaScript, Lisp, Prolog
- Semantic Web : RDF, RDFS, OWL, Pellet, Jena, SPARQL

In 2008, I also took part in a teaching project in Morocco (Casablanca and Rabat) where I gave two lectures of 20 hours (in french) on database administration. Finally, in 2007 and 2008 I gave lectures (in english) at the University of Bayreuth on Semantic Web technologies (3 hours).

6.5.1 Supervision of MsC Theses

I supervised the work of many Masters (DEA) trainees and 5th year of an engineer school (ESIEE):

- 2004: Hervé Schoenenberger, "Conception d'une interface graphique du type QBE (Query By Example) permettant de définir des requêtes depuis des ontologies d'un environement Web" . Master 1 at U. of Paris-Est.
- 2004: Christelle Montcho, "Développement d'une interface Web de stockage dans une base de données d'un dossier médical au format du Web Sémantique" , Master 1 at U. of Paris-Est.
- 2005: Johann Vallée, "Recherche et développement d'une interface incorporant des éléments du Web Sémantique dans une application dédiée à l'automédication ". Master 2 at U. of Paris-Est.
- 2005: Raphael Squelbut, "Intégration de données et ontologies". Master 2 at U. of Paris-Est.
- 2005: Tristan Moreaux "L'intelligence Artificielle dans les jeux vidéos". Master 2 et ESIEE.

- 2006: Florent Jochaud: "Intégration de règles dans les services web sémantiques". Master 2 at U. of Paris-Est.
- 2007: Jean-David Bensaïd: "Développement d'un plugin Protégé pour le mapping bases de données / ontologies". Master 1 at U. of Paris-Est.
- 2007: Julien Morali, "Eyes World, Un monde à partager". Master 2 at ESIEE.

6.5.2 Ph.D. supervision

I supervise the scientific work of the the following Ph.D. Students:

- Chahnez Zackaria at U. of Paris-Est, Full supervision (defended in December 15th 2009)
- Abdelbasset Gueimeida at U. of Paris-Est, supervision with G. Salzano (defended October 16th 2009)
- Michel Treins at U. of Paris-Est, supervision with G. Salzano (to be defended in 2010)

6.5.3 Ph.D. committee

I was one of the examiners for Ph.D. defense of :

- Amar Zerdazi with C.Pelachaud (co-supervisor), M. Lamolle (supervisor), P. Bazex (referee), G. Salzano (referee) and J.F. Degremont (chairman). The defense took place on the 2nd of July 2007 at U. of Paris 8 and the title of the thesis was : "Cadre formel de l'appariement de schémas XML pour l'intégration de données".
- Chahnez Zackaria with C.Pelachaud (referee), Jean-Claude Martin (referee), K. Smaïli (referee and chairman), G.Salzano (co-supervisor), O.Curé (supervisor). The defense took place on the 15th of December 2009 at U. of Paris Est and the title of the thesis was : "Contributions á la détection de conflits relationnels dans les ćhanges d'e-mails entre personnes en situation de travail coopératif. Une approche fondée sur les modèles statistiques et les ontologies".
- Ludovic Menet with C.Pelachaud (co-supervisor), M.Lamolle (supervisor), C.Percebois (referee), G. Salzano (referee), A. El Mhamedi (chairman) The defense took place on the of 24th June 2010 at U. of Paris 8. and the title was: 'Formalisation d'une approche d'Ingénierie Dirigée par les Modèles appliquée au domaine de la Gestion des Données de Référence'.
- Mariette Serrayet with O.Papini (supervisor), P.Drap (co-supervisor), J.Lang (referee), MO. Cordier (referee), N. Creignou, A. Hesnard, S.Lagrue The defense took place on the 6th of May 2010 at U. of Marseille and the title was:'Raisonnement à partir d'informations structurées et hierarchisées : Application à l'information archélogique'.

6.6 Scientific collaborations

6.6.1 National collaborations

The French scientific community in computer science is structured around national working groups in various research fields. The exact administrative structure varies in time and has been called AS, GDR, PRC, etc.

I am member of GDR MAGIS (Methods and Applications for Geomatic and Spatial Information).

I was a member of the SCDD (Systemes Complexes, Decision Distribuée) working group.

I was a member of CNRS specific action TOPIK (RTP147) : TOPIK : Transformation des Organisations, Projets, Production, Ingénierie, Innovation : Knowledge Management.

I am the UPEMLV coordinator on the ANR STAMP (Modelling dynamic landscapes with Spatial, Temporal And Multiscale Primitives). This project was funded for three years and involves researchers from CIRAD (Montpellier, France) and INRIA (Sofia, France).

6.6.2 International collaborations

I am a member of AGILE (Association Geographic Information Laboratories Europe)

I was the coordinator of the VENUS European project (FP6) (Virtual Exploration of Underwater Sites) for the University of Paris Est. I am a PC-co chair (with C. Bussler, now replaced by D. Thau, and S. Jablonski) and PC member for the ADI workshop (Ambient Data Integration) at the OTM 2008, 2009 and 2010 Conferences.

I served as a referee for the special issue on Logic Programming in Databases (from Datalog to Semantic Web rules) of the Theory and Practice of Logic Programming Journal.

I also served as a reviewer fro several issues of the journal Transactions on Information Technology in BioMedicine

Since 2006, I am a PC member of the PhD workshop at EDBT.

Prof. S. Jablonski and I got funded in 2007,2008 and 2010 by the Centre de Coopération Universitaire Franco-Bavarois (Bayerisch-Französisches hochschulzentrum) for our scientific collaboration on the DaltOn system.

6.7 Administrative tasks

I am currently director of the GTMC (Géomatique, Télédétection, Modélisation des Connaissances - Geomatics, Remote sensing, Knowledge modeling) research team which is a new team of the UMR CNRS IGM LabInfo.

I was a member of the council of IFIS (engineering institute of University of Paris Est).

I was director of the education for an undergraduate diploma in Computer Science at University of Paris Est from 2000 to 2002. During that period, I managed a staff of 25 teachers and 150 students.

I managed the computer resources of the IFIS institute from 2003 to 2008. Approximately, 1000 students study at the institute each year.

6.8 Software Developments, Publications and Communications

6.8.1 Software Developments

I implemented and supervised the following softwares:

- DBOM (DataBase Ontology Mapping) is a Protégé plug-in, written in Java, which enables to design ontologies from relational databases. DBOM presents several features such as tackling the impedance mismatch problem between the relational and the object models, providing a preference-based solution to deal with inconsistencies, population of an ABox using some inferences, etc..
- IMSA (Interactive Multmedia System for Auto-medication) and its extension XIMSA (eXtended IMSA) are web applications enabling the general public to self-medicate efficiently and safely. A light version of IMSA was sold to the french health service company SantéClair. The application is now targeting 5 million clients of three important insurance companies. We are working on a version 2 of this project with the team at SantéClair. This version will be based on XIMSA and will provide advanced reasoning facilities. Both of this systems are implemented using JEE technologies.

I took part in the implementation of DaltOn (Data Logistic with Ontologies). This project is the result of the collaboration with the team of Prof. Stefan Jablonski at University of Bayreuth. I designed and implemented the semantic integration module of DaltOn. It is written in Java and exploits technologies such as XML, RDF, RDFS, OWL and a DIG reasoner.

6.8.2 International Journals

[J6] O. Curé. Improving the Data Quality of Drug Databases using Conditional Dependencies and Ontologies. Accepted to the ACM International Journal of Data Quality. 2010.

[J5] O. Curé, R. Jeansoulin. An FCA-based solution for Ontology Merging. Journal of Computing Science and Engineering, 2009

[J4] P. Degenne, D. Lo Seen, D. Parigot, R. Forax, A. Tran, A. Ait Lahcen, O. Curé, R. Jeansoulin. Design of a Domain Specific Language for modelling processes in landscapes Ecological Modelling. Volume 220, Issue 24, 24 December 2009, Pages 3527-3535.

[J3] O. Curé. Mapping Databases To Ontologies To Design And Maintain Data In A Semantic Web Environment. Journal of Systemics, Cybernetics and Informatics, Volume 4, Number 4, 2006, pp 52-57.

[J2] O. Curé . Evaluation methodology for a medical e-education patient oriented information system. Journal of Medical Informatics and the Internet in Medicine, volume 28, issue 1, March 2003.

[J1] O. Curé . Overview of the IMSA project, a patient-oriented information system. Data Science Journal. Volume 1, Issue 2, August 2002.

6.8.3 Book Chapters

[B3] O. Curé. Merging Expressive Spatial Ontologies using Formal Concept Analysis. In Methods for Handling Imperfect Spatial Information (Springer). Editors: R. Jeansoulin, O. Papini, H. Prade, S. Schockaert. Accepted, to be published in 2010.

[B2] Manuel Zacklad, Aurélien Bénel, Flore Barcellini, Catherine Barry-Gréboval, Valérie Bénard, Jean-François Boujut, Sandra Bringay, Jean-Marie Burkhardt, Mathilde de Saint Leger, Françoise Détienne, Françoise Darses, Sylvie Guibert, Myriam Lewkowicz, Gaëlle Lortal, Michel Treins, O. Curé, Marie-Josèphe Pierrat , Warren Sack, Gabriella Salzano, Amalia Todirascu-Courtier, William A. Turner. La redocumentarisation du monde., sous la direction de Roger T. Pedauque Chapitre 3 :Processus d'annotation dans les documents pour l'action : textualité et médiation de la coopération . Toulouse : Cépaduès. 2007.

[B1] M. Treins, G. Salzano, O. Curé . Annotations dans les documents pour l'action, sous la direction de Pascal Salembier et Manuel Zacklad Chapitre 3 :Gestion des annotations dans le dossier médical informatisé - Analyse des apports des normes et standards et propositions pour la conception de solutions . Edition Hermès-Lavoisier.

6.8.4 Publications in Conferences (with review)

[C42] O.Curé, M. Serayet, O. Papini, P. Drap. Toward a Novel Application of CIDOC CRM to Underwater Archaeological Surveys. SWARCH-DL at IEEE ISCS 2010.

[C41] P. Degenne, A. Ait Lahcen, O. Curé, R. Forax, D. Parigot,D. Lo Seen. Modelling the environment using graphs with behaviour: do you speak Ocelet? IEMSS 2010.

[C40] O. Curé, R. Forax, P. Degenne, D. Lo Seen, D. Parigot, A. Ait Lahcen. Ocelet: An Ontology-based Domain Specific Language to Model Complex Domains. MOPAS 2010.

[C39] O. Curé.Conditional Inclusion Dependencies for Data Cleansing: Discovery and Violation Detection Issues. QDB workshop at VLDB 2009

[C38] C. Zakaria, O. Curé, G. Salzano, K. Smaili. Formalized Conflicts Detection Based on the Analysis of Multiple Emails: An Approach Combining Statistics and Ontologies. COOPIS at OTM Conferences 2009: 94-111

[C37] O. Curé. Incremental Generation of Mappings in an Ontology-Based Data Access Context. ODBASE at OTM Conferences 2009: 1025-1032

[C36] O. Curé. Merging Expressive Ontologies Using Formal Concept Analysis. OTM Workshops 2009: 49-58

[C35] O. Curé. Improving the Data Quality of Relational Databases using OBDA and OWL 2 QL. OWLED 2009

[C34] S. Jablonski, B. Volz, M. A. Rehman, O. Archner, O. Curé. Data Integration with the DaltOn Framework - A Case Study. SSDBM 2009: 255-263

[C33] O. Curé , Incremental Generation of Mappings for Ontology-based Data Access. ODBASE 2009

[C32] R. Jeansoulin, O. Curé, M. Serayet, A. Gademer, JP. Rudant, Geographical information is an act, not a fact. Poster at AGILE 2009.

[C31] F. Alcala et al. VENUS (Virtual ExploratioN of Underwater Sites) Two years of interdisciplinary collaboration. Accepted at VSMM08

[C30] C. Zakaria, O. Curé, K. Smaïli , Conflict ontology enrichment based on triggers . Accepted at CIKM ONISW workshop 2008

[C29] O. Curé, Robert Jeansoulin , An FCA-based solution to Ontology Mediation . Accepted at CIKM ONISW workshop 2008

[C28] O. Curé , Preference-enabled Information Integration for the Semantic Web . Accepted at CIKM ONISW workshop 2008. [C27] S. Jablonski, O. Curé, M. A. Rehman, B. Volz , Architecture of the DaltOn Data Integration System for Scientific Applications . Accepted at ICCS 2008

[C26] C. Zakaria, O. Curé , Vers un système de détection de conflits dans les échanges d'emails . Accepted at SIIE 2008

[C25] S. Jablonski, O. Curé, M. A. Rehman, B. Volz Architecture of the DaltOn Data Integration System for Scientific Applications . Accepted at WSES 2008 workshop at CCGRID

[C24] O. Curé, JD. Bensaid . Integration of relational databases into OWL knowledge bases: demonstration of the DBOM system . Accepted at IIMAS workshop located at ICDE 2008

[C23] O. Curé, S. Jablonski, M. A. Rehman, B. Volz. Semantic Integration in the DaltOn system . Accepted at IIMAS workshop located at ICDE 2008

[C22] O. Curé, R. Jeansoulin . Data Quality Enhancement of Databases using Ontologies and Inductive Reasoning. Accepted at ODBASE 2007

[C21] O. Curé, S. Jablonski . Ontology-based Data Integration in Data Logistics Workflows . Accepted to CMLSA workshop at ER 2007.

[C20] O. Curé, F. Jochaud . Preference-based Integration of Relational Databases into a Description Logic . Accepted to DEXA 2007

[C19] O. Curé, JP. Giroud . Ontology-based Data Quality enhancement for Drug Databases . Accepted at the Health Care and Life Sciences Data Integration for the Semantic Web Workshop at the WWW 2007 conference

[C18] O. Curé, R. Squelbut . Integrating data into an OWL Knowledge Base via the DBOM Protégé plug-in 9th International Protégé conference 2006

[C17] O. Curé, R. Squelbut . Semantic mapping to synchronize data and knowledge bases at the instance level ESWC 2006 poster session

[C16] M. Treins, O. Curé, G. Salzano . On the interest of using the HL7 CDA standard for the exchange of annotated medical documents . IEEE CBMS 2006, Salt Lake City, USA.

[C15] O. Curé, R. Squelbut Data integration targeting a drug knowledge base . Proceedings of EDBT Workshop 2006 Information Integration in Healthcare Applications, Munich, Germany. LNCS 4254

[C14] O. Curé, R. Squelbut . A database trigger strategy to maintain knowledge bases developped via data migration. Proceedings of EPIA 2005 - december 2005, Covilha, Portugal. LNAI 3808.

[C13] O. Curé . Semi-automatic data migration in a self-medication Knowledge-based system (extended version) . Proceedings of WM 2005 - april 2005, Kaiserlautern, Germany. LNAI 3782.

[C12] O. Curé . Mapping Databases to Ontologies to Design and Maintain Data in a Semantic Web Environment . Proceedings of CITSA 2005 (International Conference on Cybernetics and Information Technologies, Systems and Applications). July 2005.Orlando, USA. Volume II

[C11] O. Curé . Ontology interaction with a patient electronic health record Proceedings of IEEE CBMS 2005 (Computer-Based Medical Systems) June 2005, Dublin, Ireland. [C10] O. Curé . Semi-automatic data migration in a self-medication Knowledge-based system. Proceedings KMM 2005 (Current Aspects of Knowledge Management in Medicine) - april 2005, Kaiserlautern, Germany. [C9] O. Curé . QBEO : Query By Example over Ontologies framework Proceedings of CCCT 2004 (Computing, Communications and Control Technologies) - august 2004 - Austin, USA [C8] O. Curé . XIMSA : eXtended Interactive Multimedia System for Auto- medication . Proceedings of IEEE CBMS 2004, Bethesda, USA [C7] O. Curé . Designing patient-oriented Systems with Semantic Web technologies . Proceedings of IEEE CBMS 2003 - june 2003 - New York, USA. [C6] O. Curé, M. Levacher, JP. Giroud . Medical e-education for the patient. Proceedings of Mednet 2001 in Technology and Health Care Volume 9, Number 6 / 2001. Mednet 2001 - december 2001 - Udine. Italy.

[C5] O. Curé, M. Levacher, JP. Giroud. Interdisciplinary collaboration for a patient oriented medical information system . Proceedings of the IEEE MTAC 2001 - november 2001. UCI, Los Angeles.USA

[C4] O. Curé . IMSA : Interactive Multimedia System for Automedication Proceedings of Codata 2000 - october 2000 Baveno. Italy.

[C3] J. Fruitet, O. Curé . Tourisme et système d'information : Internet, un état de lieux. La recherche en tourisme . Actes du colloque de Foix. mai 2000. Foix. France.

[C2] O. Curé, Norbert Cot, M. Levacher, JP. Giroud Cognitive Science for the IMSA project . Proceedings of the Human Centered Processes Conference - September 1999, Brest, France - pp23-28.

[C1] O. Curé . A textual journal for telecommunications services 3rd ERCIM Workshop on User Interfaces for All - november 1997 Strasbourg. France.

6.8.5 Dissertation

[T1] O. Curé, SIAM: Système Interactif d'Automédication Multimédia. Ph.D. Thesis. Université Paris V. June 1999, (in french).

6.8.6 Deliverables for EU project

[D7] WP3R8 Reasoning with the application ontology for achaeological information (August 2009)

[D6] WP3R7 Using Pellet for reasoning with the application ontology for archaelogical information: Specification (August 2009)

[D5] WP3R4 Mapping of the application ontology for archaeological information onto CIDOC CRM: Specification (August 2009)

[D4] WP3R3 Application ontology for archaeological information: specification (August 2009)

[D3] D3.6 Reasoning with archaeological ontologies - Technical report and Prototype of software for the reversible fusion operations (July 2009)

[D2] D3.5 CNRS Knowledge based photogrammetric software interface 3. (January 2009)

[D1] D3.4 Representation of archaeological ontologies 1, Technical Report. (July 2008)

6.8.7 Others

"WebMed - guide médical" was a french magazine available on subscriptions for health care professionals. I was the only author on all papers which were 4 to 5 pages long.

[08] La maison intelligente . WebMed issue #11 oct 2001
[O7] Les sites Web intelligents . WebMed issue #10 Mai 2001
[O6] Un état des lieux sur l'internet mobile . WebMed issue #9 déc. 2000
[O5] Présentation des langages à balises . WebMed issue #8 oct. 2000
[O4] Les algorithmes génétiques . WebMed issue #6-7 Jui 2000
[O3] Les réseaux de neurones . WebMed issue #5 Avr 2000
[O2] Les systèmes experts (suite). WebMed issue #4 Jan 2000
[O1] Les systèmes experts . WebMed issue #3 Nov 1999

Chapter 7

Conclusion

This dissertation emphasizes some of my work on the relationships between relational databases and ontologies of the Semantic Web. It is an expression of a will to design applications in the context of the Web of data. As shown in the introduction, most of the methods and algorithms presented in this document have been applied in the field of medicine and in particular self-medication, a market with a high rate growth in most industrial countries. Anyhow, the domain independence of our different contributions have been highlighted in several publications and participations in projects in the geographical, ecological, biological domains.

Considering a complete and practical application like XIMSA forced me to tackle several aspects of integrating Web of data technologies: creating ontologies and knowledge bases from available database instances, merging these ontologies for interoperability reasons and ensuring data quality of source databases via operations performed at the ontology level. A common aspects arising from these different contributions is the will to handle uncertainties using different solutions: preferences associated to mapping assertions in the DBOM application, use of possibilistic logic and preferences in the FCA based merging solution, notion of confidence values and support in our data quality systems. Moreover, the data provenance solution currently being designed in the DaltOn framework integrates preferences at its core.

Anyhow, there is still a lot of work to do in the domain of designing 'smart' Web applications. Among the various trails I am aiming to investigate is the efficient storage of ABoxes. This is very important from an application development point of view since it can easily be a performance bottleneck. Considering the storage of RDF data sets in a relational databases, there is wide spectrum of solutions between two extremes: on the one hand a single relation, with three columns (i.e. subject, predicate and object) stores all triples and on the other hand the vertical partitioning of [AMH09] where triples are distributed over relations mapped to predicates of the ontology. Recently, with Guillaume Blin and David Faye, we have proposed an efficient storage solutions for RDF triples based on a column oriented relational database and which integrates reasoning at its core. That is the hierarchies of properties and concepts are taken into account when generating the relations and rewriting the SQL queries. Moreover, the so-called `RoStore` system is currently extended to support updates, i.e. deletions, insertions and modifications, of relational tuples. This is an impor-

tant aspect considering the release of the SPARQL 1.1 recommendation which supports triple updates. This solution is only one solution to store RDF data sets in a RDBMS and we are aiming to emphasize over solutions relevant in different contexts, e.g. emerging indexing solutions.

Finally, another goal is to study the emerging solutions tagged under the term NoSQL (i.e. "Not only SQL"). This movement is quite popular in the Web domain and proposes alternatives to the relational model. This trend emphasizes different approaches which are named document oriented, graph, key values stores and column families databases. Few academic papers have been published on the database solutions but some of these systems are already successful in many large Web companies, e.g. Google with Big Table, Amazon with Dynamo, Yahoo! with Pnuts, Facebook and twitter with Cassandra, SourceForge with MongoDB, etc. We believe that since NoSQL databases are considered particularly well adapted to the context of Web applications, it seems obvious to investigates their use in the storage of RDF, the main data model of the Web of data. In the next coming years, my plan is to pursue my investigations in this direction and to provide a comprehensive road map of the use of row and column oriented relational databases and NoSQL solutions for RDF storage, to provide proof of concepts and performance evaluations on several practical applications.

Bibliography

[AHV95] Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases*. Addison-Wesley, 1995.

[AMH09] Daniel J. Abadi, Adam Marcus 0002, Samuel Madden, and Kate Hollenbach. Sw-store: a vertically partitioned dbms for semantic web data management. *VLDB J.*, 18(2):385–406, 2009.

[BCM+03] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

[BFG+07] Philip Bohannon, Wenfei Fan, Floris Geerts, Xibei Jia, and Anastasios Kementsietsidis. Conditional functional dependencies for data cleaning. In *ICDE*, pages 746–755. IEEE, 2007.

[BFGM08] Loreto Bravo, Wenfei Fan, Floris Geerts, and Shuai Ma. Increasing the expressivity of conditional functional dependencies without extra complexity. In *ICDE*, pages 516–525. IEEE, 2008.

[BFM07] Loreto Bravo, Wenfei Fan, and Shuai Ma. Extending dependencies with conditions. In *VLDB*, pages 243–254, 2007.

[BGG+03] Sean Bechhofer, Aldo Gangemi, Nicola Guarino, Frank van Harmelen, Ian Horrocks, Michel Klein, Claudio Masolo, Daniel Oberle, Steffen Staab, Heiner Stuckenschmidt, and Raphael Volz. Tackling the ontology acquisition bottleneck: An experiment in ontology reengineering, 2003.

[Bir73] G. Birkhoff. *Lattice Theory*. American Mathematical Society, Colloquium Publications, 3rd edition, 1973.

[BLHL01] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, 284(5):34–43, 2001.

[Bra83] R. J. Brachman. What is-a is and isn't: An analysis of taxonomic links in semantic networks. *Computer*, 16(10):30–36, 1983.

[BS06] Carlo Batini and Monica Scannapieco. *Data Quality: Concepts, Methodologies and Techniques (Data-Centric Systems and Applications)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[BvHH+04] Sean Bechhofer, Frank van Harmelen, Jim Hendler, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, and Lynn Andrea Stein. OWL Web Ontology Language Reference. Technical report, W3C, http://www.w3.org/TR/owl-ref/, February 2004.

[CB08] Olivier Curé and Jean-David Bensaid. Integration of relational databases into owl knowledge bases: demonstration of the dbom system. In *ICDE Workshops*, pages 230–233, 2008.

[Cho06] Vicky Choi. Faster algorithms for constructing a concept (galois) lattice. *CoRR*, abs/cs/0602069, 2006.

[CJ07a] Olivier Curé and Stefan Jablonski. Ontology-based data integration in data logistics workflows. In *ER Workshops*, pages 34–43, 2007.

[CJ07b] Olivier Curé and Robert Jeansoulin. Data quality enhancement of databases using ontologies and inductive reasoning. In *OTM Conferences (1)*, pages 1117–1134, 2007.

[CJ07c] Olivier Curé and Florent Jochaud. Preference-based integration of relational databases into a description logic. In *DEXA*, pages 854–863, 2007.

[CJ08] Olivier Curé and Robert Jeansoulin. An fca-based solution for ontology mediation. In *ONISW*, pages 39–46, 2008.

[CJ09] Olivier Curé and Robert Jeansoulin. An fca-based solution for ontology merging. *Journal of Computing Science and Engineering*, 3(2):90–108, 2009.

[CJJ+08] Olivier Curé, Stefan Jablonski, Florent Jochaud, M. Abdul Rehman, and Bernhard Volz. Semantic data integration in the dalton system. In *ICDE Workshops*, pages 234–241, 2008.

[CM08] Fei Chiang and Renée J. Miller. Discovering data quality rules. *PVLDB*, 1(1):1166–1177, 2008.

[CPP+08] Claudio Corona, Emma Di Pasquale, Antonella Poggi, Marco Ruzzi, and Domenico Fabio Savo. When dl-lite met owl.... In Catherine Dolbear, Alan Ruttenberg, and Ulrike Sattler, editors, *OWLED*, volume 432 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008.

[CS05] Olivier Curé and Raphaël Squelbut. A database trigger strategy to maintain knowledge bases developed via data migration. In *EPIA*, pages 206–217, 2005.

[CS06a] Olivier Curé and Raphaël Squelbut. Data integration targeting a drug related knowledge base. In *EDBT Workshops*, pages 411–422, 2006.

[CS06b] Olivier Curé and Raphaël Squelbut. Integrating data into an owl knowledge base via the dbom protégé plug-in. In *9th International Protégé conference*, 2006.

[Cur02] Olivier Curé. Overview of the IMSA project, a patient-oriented information system. *Data Science Journal*, 1(2):66–75, 2002.

[Cur03] Olivier Curé. Designing patient-oriented systems with semantic web technologies. In *CBMS*, pages 195–200, 2003.

[Cur04] Olivier Curé. Ximsa : extended interactive multimedia system for auto-medication. In *CBMS*, pages 570–575, 2004.

[Cur05a] Olivier Curé. Ontology interaction with a patient electronic health record. In *CBMS*, pages 185–190, 2005.

[Cur05b] Olivier Curé. Semi-automatic data migration in a self-medication knowledge-based system. In *Wissensmanagement (LNCS Volume)*, pages 373–383, 2005.

[Cur08] Olivier Curé. Data integration for the semantic web with full preferences. In *ONISW*, pages 61–68, 2008.

[Cur09a] Olivier Curé. Conditional inclusion dependencies for data cleansing: Discovery and violation detection issues. In *QDB workshop at VLDB*, 2009.

[Cur09b] Olivier Curé. Improving the data quality of relational databases using obda and owl2ql. In *OWLED*, 2009.

[Cur09c] Olivier Curé. Incremental generation of mappings in an ontology-based data access context. In *OTM Conferences (2)*, pages 1025–1032, 2009.

[Cur09d] Olivier Curé. Merging expressive ontologies using formal concept analysis. In *OTM Workshops*, pages 49–58, 2009.

[Cur10a] Olivier Curé. Improving the data quality of drug databases using conditional dependencies and ontologies. *ACM Journal of Data and Information Quality*, Special issue on Healthcare(Information quality: The Challenges and Opportunities in Healthcare Systems and Services):Accepted, to be published, 21 pages, 2010.

[Cur10b] Olivier Curé. Merging expressive spatial ontologies using formal concept analysis with uncertainty considerations. In Henri Prade Steven Schockaert Robert Jeansoulin, Odile Papinin, editor, *Methods for Handling Imperfect Spatial Information*, page 21 pages. Springer, 2010.

[DD04] H. Prade D. Dubois, J. Lang. Possibilistic logic. pages 439–513. Oxford University Press, 2004.

[DG98] P. Drap and P. Grussenmeyer. Arpenteur, an architectural photogrammetry network for education and research. In *ISPRS Comm. V Symposium, number XXXII part 5, 1998*, pages 537 – 542, 1998.

[DHM05] Xin Dong, Alon Halevy, and Jayant Madhavan. Reference reconciliation in complex information spaces. In *SIGMOD '05: Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 85–96, New York, NY, USA, 2005. ACM Press.

[DP90] B. A. Davey and H. A. Priestly. *Introduction to Lattices and Order*. Cambridge University Press, 1990.

[DSP+09] P. Degenne, D. Lo Seen, D. Parigot, R. Forax, A. Tran, A. Ait Lahcen, O. Curé, and R. Jeansoulin. Design of a domain specific language for modelling processes in landscapes. *Ecological Modelling*, 220(24):3527–3535, 2009.

[Ehr07] Marc Ehrig. *Ontology Alignment: Bridging the Semantic Gap*, volume 4 of *Semantic Web And Beyond Computing for Human Experience*. Springer, 2007.

[EN06] Ramez Elmasri and Shamkant B. Navathe. *Fundamentals of Database Systems (5th Edition)*. Addison Wesley, March 2006.

[ES07] Jérôme Euzenat and Pavel Shvaiko. *Ontology matching*. Springer-Verlag, Heidelberg (DE), 2007.

[Fan08] Wenfei Fan. Dependencies revisited for improving data quality. In Lenzerini and Lembo [LL08], pages 159–170.

[FGLX09] Wenfei Fan, Floris Geerts, Laks V.S. Laksmanan, and Ming Xiong. Discovering conditional functional dependencies. In *ICDE*, page To appear, 2009.

[Fis70] Peter C. Fishburn. *Utility Theory for Decision Making*. Wiley, New York, 1970.

[GKK+08] Lukasz Golab, Howard J. Karloff, Flip Korn, Divesh Srivastava, and Bei Yu. On generating near-optimal tableaux for conditional functional dependencies. *PVLDB*, 1(1):376–390, 2008.

[Gru93] Thomas R. Gruber. A translation approach to portable ontology specifications. *Knowl. Acquis.*, 5(2):199–220, 1993.

[Gua95] Nicola Guarino. Formal ontology, conceptual analysis and knowledge representation. *Int. J. Hum.-Comput. Stud.*, 43(5-6):625–640, 1995.

[GW99] Bernhard Ganter and Rudolph Wille. *Formal concept analysis: Mathematical foundations*. Springer, Berlin-Heidelberg, 1999.

[HdB07] Martin Hepp and Jos de Bruijn. Gentax: A generic methodology for deriving owl and rdf-s ontologies from hierarchical classifications, thesauri, and inconsistent taxonomies. In *ESWC*, pages 129–144, 2007.

[JB96] Stefan Jablonski and Christoph Bussler. *Workflow Management: Modeling Concepts, Architecture and Implementation*. 1996.

[JCRV08] Stefan Jablonski, Olivier Curé, M. Abdul Rehman, and Bernhard Volz. Dalton: An infrastructure for scientific data management. In *ICCS (3)*, pages 520–529, 2008.

[JVR+09] Stefan Jablonski, Bernhard Volz, M. Abdul Rehman, Oliver Archner, and Olivier Curé. Data integration with the dalton framework - a case study. In *SSDBM*, pages 255–263, 2009.

[Kie02] Werner Kießling. Foundations of preferences in database systems. In *VLDB*, pages 311–322. Morgan Kaufmann, 2002.

[Kol05] Phokion G. Kolaitis. Schema mappings, data exchange, and metadata management. In *PODS*, pages 61–75, 2005.

[KS03] Yannis Kalfoglou and Marco Schorlemmer. Ontology mapping: the state of the art. *Knowl. Eng. Rev.*, 18(1):1–31, 2003.

[Len02] Maurizio Lenzerini. Data integration: A theoretical perspective. In *PODS*, pages 233–246, 2002.

[LL87] M. Lacroix and Pierre Lavency. Preferences; putting more knowledge into queries. In *VLDB*, pages 217–225, 1987.

[LL08] Maurizio Lenzerini and Domenico Lembo, editors. *Proceedings of the Twenty-Seventh ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2008, June 9-11, 2008, Vancouver, BC, Canada*. ACM, 2008.

[LMSS95] Alon Y. Levy, Alberto O. Mendelzon, Yehoshua Sagiv, and Divesh Srivastava. Answering queries using views. In *PODS*, pages 95–104. ACM Press, 1995.

[McG03] Deborah L. McGuiness. Ontologies come of age. In Dieter Fensel, James Hendler, Henry Lieberman, and Wolfgang Wahlster, editors, *Spinning the Semantic Web*, chapter 6, pages 171–196. MIT Press, Cambridge, MA, 2003.

[MGH+08] Boris Motik, Bernardo Cuenca Grau, Ian Horrocks, Zhe Wu, Achille Fokoue, and Carsten Lutz. Owl 2 web ontology language: Profiles. World Wide Web Consortium, Working Draft WD-owl2-profiles-20081202, December 2008.

[MHS07] Boris Motik, Ian Horrocks, and Ulrike Sattler. Bridging the gap between owl and relational databases. In *WWW*, pages 807–816, 2007.

[NM] N. F. Noy and D. Mcguinness. *Stanford KSL Technical Report KSL-01-05*.

[OC10a] Odile Papini Pierre Drap Olivier Curé, Mariette Serayet. Toward a novel application of cidoc crm to underwater archaeological surveys. In *SWARCH-DL*, page To appear, 2010.

[OC10b] Pascal Degenne Danny Lo Seen Didier Parigot Ayoub Ait Lahcen. Olivier Curé, Rémi Forax. Design of a domain specific language for modelling processes in landscapes. In *IARIA MOPAS*, page To appear, 2010.

[PLC+08] Antonella Poggi, Domenico Lembo, Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati. Linking data to ontologies. *J. Data Semantics*, 10:133–173, 2008.

[SM01] Gerd Stumme and Alexander Maedche. Fca-merge: Bottom-up merging of ontologies. In *IJCAI*, pages 225–234, 2001.

[SPG+07] Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. Pellet: A practical owl-dl reasoner. *J. Web Sem.*, 5(2):51–53, 2007.