VRIJE UNIVERSITEIT BRUSSEL

FACULTEIT WETENSCHAPPEN

VAKGROEP INFORMATICA EN TOEGEPASTE INFORMATICA
SEMANTICS TECHNOLOGY AND APPLICATIONS RESEARCH LAB

STAR Lab Technical Report

Semantically Unlocking Database Content through Ontology-based Mediation

Pieter Verheyden, Jan De Bo & Robert Meersman

affiliation:
keywords ontology engineering, commitment language
number STAR-2004-12
date 2/09/2004
corresponding author Pieter Verheyden
status final
reference Bussler C., Tanner V. & Fundulaki I., (eds.), Proceedings of the 2nd Workshop on Semantic Web and Databases (SWDB 2004), LNCS 3372, Springer Verlag, 2005, pp. 109 -126

Pleinlaan 2, Gebouw G-10, B-1050 Brussel
Phone: +32-2-629.1237; Fax: + 32-2-629.3819
http://www.starlab.vub.ac.be

Semantically Unlocking Database Content through Ontology-Based Mediation

Pieter Verheyden, Jan De Bo and Robert Meersman

Vrije Universiteit Brussel - STARLab
Pleinlaan 2, Gebouw G-10, B-1050 Brussels, Belgium
{pverheyd,jdebo,meersman}@vub.ac.be
http://www.starlab.vub.ac.be

Abstract. To make database content available via the internet, its intended shared meaning, i.e. an interpretation is required of the database (schema) symbols in terms of a so-called ontology. Such an ontology specifies not only concepts and their relationships in some language, but also includes the manner in which an application or service is permitted to make use of these concepts. Ontologies therefore also play a key role in making databases interoperate. The DOGMA approach to ontology engineering is specifically adapted to the classical model-theoretic view of (relational) databases. Noteably, it rigorously separates an ontology base of elementary lexical fact types called lexons, from the rules and constraints governing the concepts referred to by the lexons in the ontology base. These rules are reified in so-called ontological *commitments* of applications to the ontology base. In this paper we formalise and make precise the structure of this commitment layer by defining Ω-RIDL, a new type of so-called commitment language. Examples derived from its use in a non-trivial case study are provided. We illustrate how some of its key constructs, designed to specify mediators by mapping databases to an ontology base, can conveniently be reused in a conceptual query language, and report on its ongoing implementation.

1 Introduction

Suppose we want to make certain database content meaningfully available for applications on the World Wide Web. In such an open environment applications and application types in general are unknown a priori, including the manner in which they will want to refer to the data, or more precisely, to the concepts and attributes that take their values from the database. Therefore, elements of meaning for the database's underlying domain have to be agreed, and represented explicitly. They will need to be stored, accessed, and maintained externally to the database schema as well as to the intended applications. Computer resources that formally represent a domain's semantics in this external, application-independent way are called (domain-)*ontologies*. In a nutshell, an application system and in particular its database schema can be assigned a formal semantics, also known as (first order) *interpretation*. Such semantics in our

approach has two separate components, (a) a mapping from the schema's symbols and relationships to a suitable ontology base expressed in lexical terms, and (b) expressions, separate and "ontological", of how database constraints restrict the use of, or precisely *commit to*, the concepts referred by the terms in this ontology base.

In this paper we discuss how elements of a relational database are mapped on elements of an existing domain ontology. We investigate possible difficulties that can be encountered during this non-trivial task. Further, we describe how to translate domain constraints on the database level to semantic constraints on the ontology level. In order to impose these semantic constraints on the terms and relations of the ontology, we developed a new ontological commitment language called *Ω-RIDL*. The above mentioned principles are illustrated and clarified by a practical case study. In this case study we investigate how the relational database of the National Drug Code (NDC) Directory relates to the medical ontology LinKBase®.

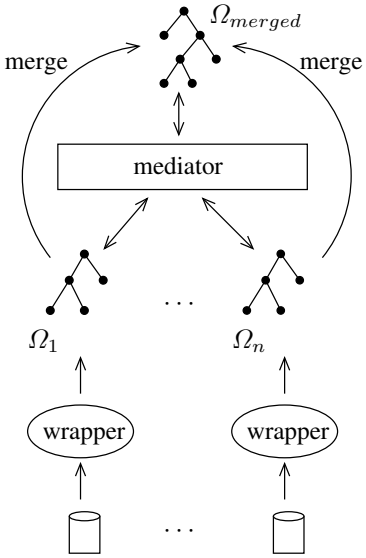

Fig. 1. Mediator approach for data integration.

The research in this paper fits in the broader context of data integration because it will be very unlikely that a user's information needs will be satisfied by accessing the data repositories accessible through mappings associated with a single ontology. To support this, ontologies are aligned with each other. The OBSERVER framework [20] proposes an approach to use the inter ontology relationships to translate the original query from terms of the source ontology into terms of another component, also referred to as a target ontology. This kind of query rewriting does not always occur without loss of information. The *Interon-*

tology Relationship Manager (IRM) in the OBSERVER system serves as a pool where all interontology relationships between the different ontologies are made available. For n ontologies involved one has to compute $\frac{n(n-1)}{2}$ sets of interontology relationships. To minimise this effort we have chosen for a *mediator inspired framework*. It is our goal to develop a framework for data integration that is easy to maintain and to extend. Therefore the source ontologies are merged into one global ontology. In a binary merging strategy this requires only $n-1$ alignments [2]. The only additional steps to be performed are to check for conflicts and to integrate the separate ontologies into a global ontology. The mediator then decomposes the global query into a union of queries on the underlying source ontologies and unifies all resultsets into a global result. The framework is depicted in Figure 1. Each time our framework is extended with a new ontology we only have to merge this ontology with the global ontology and adjust the mediator accordingly. It is obvious that this is less time consuming than having to perform alignments with all present ontologies.

The focus of this paper is to present a new ontological commitment language called *Ω-RIDL*, and not to elaborate further on the mediator framework here proposed. The syntax of the language and its principles are introduced in section 4, and its usage is explained by means of a case study which we describe in section 3. In section 5 we illustrate how ontological commitments are deployed in the mediator framework. We finalise this paper with sections on related work (section 6) and future work (section 7), and present a conclusion in section 8. In section 2 we briefly discuss our DOGMA approach to ontology engineering.

2 The DOGMA Ontology Model

DOGMA[1] is a research initiative of VUB STARLab where various theories, methods, and tools for ontologies are studied and developed. A DOGMA inspired ontology is based on the classical model-theoretic perspective [21] and decomposes an ontology into an *ontology base* and a layer of *ontological commitments* [17, 18]. This is called the principle of *double articulation* [22].

An ontology base holds (multiple) intuitive conceptualisation(s) of a particular domain. Each conceptualisation is simplified to a "representation-less" set of context-specific binary fact types called *lexons*. A lexon is formally described as a 5-tuple $< \gamma \quad term_1 \quad role \quad co-role \quad term_2 >$, where γ is an abstract context identifier, lexically described by a string in some natural language, and is used to group lexons that are logically related to each other in the conceptualisation of the domain. Intuitively, a lexon may be read as: within the context γ, the $term_1$ (also denoted as the *header* term) may have a relation with $term_2$ (also denoted as the *tail* term) in which it plays a *role*, and conversely, in which $term_2$ plays a corresponding *co-role*. Each (context,term)-pair then lexically identifies a unique *concept*. An ontology base can hence be described as a set of plausible elementary fact types that are considered as being true. Any specific (application-dependent) interpretation is moved to a separate layer, i.e. the commitment layer.

[1] Developing Ontology-Guided Mediation for Agents

Ontology Base

Context	*Header Term*	*Role*	*Co−role*	*Tail Term*
MEDICINE	DENTAL DRUG	IS_A		MEDICINAL PRODUCT
MEDICINE	MEDICINAL PRODUCT	HAS–PATH		ROUTE OF ADMINISTRATION
MEDICINE	PER VAGINA	IS_A		ROUTE OF ADMINISTRATION
MEDICINE	MEDICINAL PRODUCT	HAS–INGREDIENT	IS–INGREDIENT–OF	INGREDIENT OF MEDICINAL SUBSTANCE
MEDICINE	MEDICINAL PRODUCT	HAS_ASSOC		MATERIAL ENTITY BY PRESENTATION SHAPE
MEDICINE	LOTION	IS_A		MATERIAL ENTITY BY PRESENTATION SHAPE
MEDICINE	MEDICINAL PRODUCT	HAS_ASSOC		ENTERPRISE
MEDICINE	ENTERPRISE	HAS_ASSOC		COUNTRY – STATE
MEDICINE	CANADA	IS_A		COUNTRY – STATE

Fig. 2. A small extract of the ontology base represented by a simple table format.

The commitment layer mediates between the ontology base and its applications. Each such ontological commitment defines *a partial semantic account of an intended conceptualisation* [13]. It consists of a finite set of axioms that specify which lexons of the ontology base are interpreted and how they are *visible* in the committing application, and (domain) rules that semantically constrain this interpretation. Experience shows that it is much harder to reach an agreement on domain rules than one on conceptualisation [19]. E.g., the rule stating that *each patient is a person who suffers from at least one disease* may hold in the Universe of Discourse (UoD) of some application, but may be too strong in the UoD of another application.

3 A Motivating Case Study

In the health care sector, access to correct and precise information in an efficient time frame is a necessity. A Hospital Information System (HIS) is a real-life example of an Information System consisting of several dispersed data sources containing specific information, though interrelated in some way. These data sources can vary from highly structured repositories (e.g., relational databases), structured documents (e.g., electronic patient records), or even free text (e.g., patient discharge notes written in some natural language). VUB STARLab joins hands with Language and Computing (L&C) N.V. [2] in the IWT R&D project SCOP[3] with the aim of finding a suitable solution to integrate such medical data sources through "semantic couplings" to an existing medical ontology. The initial focus was set on medical relational databases.

Throughout the years, L&C has built up, and still maintains, an extensive medical ontology called LinKBase® [12]. Further, The National Drug Code

[2] URL: `http://www.landcglobal.com`

[3] Semantic Connection of Ontologies to Patient data

(NDC) Directory of the U.S. Food And Drug Administration (FDA) was used as a case study. The ontological commitment to a DOGMA ontology base containing ontological knowledge from (a relevant part of) LinKBase® [4] was defined for the NDC Directory. Figure 2 presents a small extract of the ontology base represented by a simple table format.

In the following subsection we give some relevant background information on the NDC Directory and its relational database. Parts of its ontological commitment definition will be used for illustration purposes in section 4.

3.1 The NDC Directory

The National Drug Code (NDC) Directory was originally established as an essential part of an out-of-hospital drug reimbursement program under Medicare, and serves as a universal product identifier for human drugs. The current edition of the NDC is limited to prescription drugs and a few selected over-the-counter (OTC) drug products. The following information about the listed drug products are available: product trade name or catalogue name, National Drug Code (NDC), related firms, dosage form, routes of administration, active ingredient(s), strength, unit, package size and type, and the major drug class.

By federal regulation, NDCs are 10-digit numbers that identify the labeller/ vendor, product, and trade package size. NDCs follow one of three different formats: 4-4-2, 5-4-1, or 5-3-2. The first set of digits, the labeller code assigned by the FDA, identifies the labeller (i.e. any firm that manufactures, repacks, or distributes a drug product). The second set of digits, the product code assigned by the firm, identifies a specific strength, dosage form, and formulation for that particular firm. The third set of digits, the package code assigned by the firm, identifies package sizes. Because of the variability of the length of the subcodes within an NDC, almost all governmental and commercial organisations other than the FDA use 11-digit NDCs. In particular, the Centers for Medicare & Medicaid Services (CMS)[5] uses and distributes 11-digit NDCs. These non-standard 11-digit NDCs are created by a system of zero-filling so that each NDC follows a 5-4-2 format (e.g., 00006-4677-00). NDCs may be reused and reassigned to different drugs. So, a given NDC cannot be assumed to be constant over time. If a manufacturer is acquired by another firm, or if a manufacturer sells the production rights of a drug to another entity, there is a good chance that the new manufacturer or re-distributor will change all the NDCs assigned to a particular drug (even though the drug product remains exactly the same in terms of its formulation, preparation, packaging, etc.).

The relational database schema of the NDC Directory is presented by Figure 3. The freely available ASCII data files from which this relational database has

[4] Due to some significant differences between both ontology approaches, the exchange of ontological knowledge was not so straightforward. We will not elaborate on this issue because it is less relevant here, but, we can mention that this exchange could be done semi-automatically by using RDFS as communication language between both ontology frameworks.

[5] URL: `http://www.cms.hhs.gov`

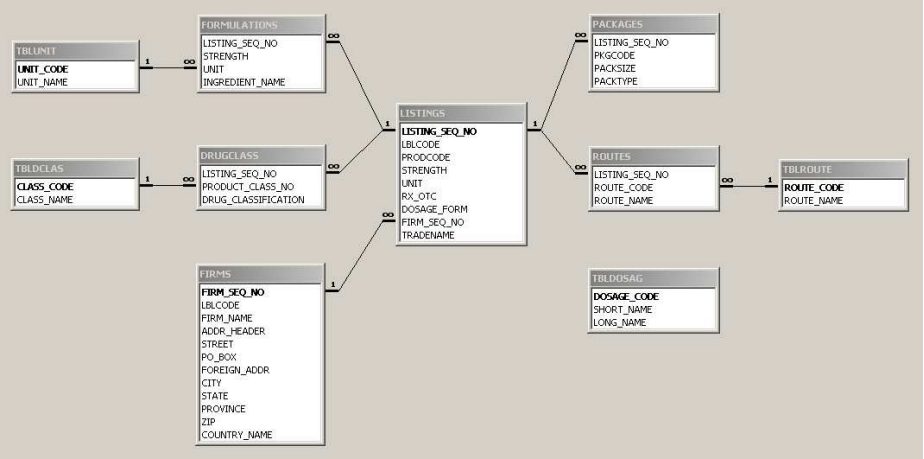

Fig. 3. The relational database schema of the NDC Directory.

been constructed, together with detailled descriptions, can be found on the official website of the NDC Directory [6]. We mention following issues that clearly indicate a poor design of the relational database regarding its provided schema and population:

- **Referential integrity**. We had to manually update the population of some relations to enable a correct linking with other relations (e.g., the linking of the relation "ROUTES" with "TBLROUTE").
- **Normalisation**. Some attributes of the relation "LISTINGS" allow multiple entries as one value. As a result, the relational database schema is not in first normal form (1NF).
- **Data redundancy**. Some attributes appear in more than one relation, which causes update anomalies (e.g., the attribute "LBLCODE" can be found in the relation "LISTINGS" as well as in the relation "FIRMS").

4 Defining Ontological Commitments in Ω-RIDL

4.1 Historical background

The main syntactic principles of Ω-RIDL are adopted from RIDL[7], an old conceptual language developed in 1979 by R. Meersman at the Database Management Research Lab (Brussels) of Control Data. It was developed as an integrated formal syntactic support for information and process analysis, semantic specification, constraint definition, and a query/update language at a conceptual level

[6] URL: `http://www.fda.gov/cder/ndc/`

[7] Reference and IDea Language

rather than at the logical "flat data" level. The conceptual support for RIDL was provided by (the "binary subset" of) the so-called *idea/bridge* model for conceptual schemata developed by Falkenberg and Nijssen. Problem specifications in this model were obtained through a methodology commonly known as NIAM[8] [24], which is the predecessor of ORM[9] [14]. A result of this analysis methodology was (partially) represented by a conceptual data schema graphically depicted by a dedicated diagram notation. In the idea/bridge philosophy, such a conceptual data schema was also denoted as an idea/bridge view of a world (i.e. the UoD on which the analysis is done). A fundamental characteristic was the strict separation between *non-lexical object types* (NOLOTs; "things" that cannot be uttered or written down, e.g., "patient") and *lexical object types* (LOTs; "things" that can be uttered, written down, or otherwise represented, e.g., "date of birth") [25]. A relation (consisting of a role and co-role) between two NOLOTs was called an *idea*; a relation between a NOLOT and a LOT was called a *bridge*. Such relationships are commonly called *fact types*. Further, subtype relations between NOLOTs were also supported. This strict separation between NOLOTs and LOTs was also explicitly respected by RIDL. Since the idea/bridge philosophy was very close the user's understanding of a problem, RIDL also had to be close to a natural formulation of the information description and manipulation [16].

RIDL can be roughly divided into two parts: the constraint definition part (RIDL\cns) and the query/update part (RIDL\qu). These two parts were used by two, in general disjunctive, kind of users: database engineers and end-users. Database engineers used RIDL\cns to formally and naturally express a conceptual data schema and its constraints. At compile time, such a conceptual data schema was (semi-)automatically transformed into a relational database schema, satisfying some normal form which was controlled by the database engineer [10]. The end-user used RIDL\qu, after the generated relational database was populated, to retrieve/update data at runtime through (possibly interactive) conceptual queries on the conceptual data schema, instead of constructing SQL queries on the underlying relational database [16]. During the eighties, dedicated tools were developed and enhanced for the transformation of a conceptual data schema into a relational database schema (RIDL* graphical workbench [10, 11]), and the translation of RIDL queries/updates into correct SQL queries/updates (RIDL Shell).

RIDL was developed at the same time when the first SQL systems appeared on the market and was therefore far ahead of its time. Although none of the commercial RIDL* prototypes found their way to the market, the RIDL* fun-

[8] aN/Natural/Nijssen's Information Analysis Method

[9] Object-Role Modeling (URL: `http://www.orm.net`)

[10] This control included the choice of how a subtype relation from the conceptual data schema should be translated in the relational database schema to be generated. This can be done by, e.g., an "indicator" attribute (e.g., a relation "Person" having an attribute "Sex" only allowing the values "M" or "F"), or by a foreign key (resulting in a decomposition, e.g., the relations "Male" and "Female" with foreign keys to the relation "Person").

damentals still live in today's ORM-based modelling and database design CASE tools. RIDL's conceptual querying part got the attention of Halpin and resulted in ConQuer[11][4], and a successor ConQuer-II [5], a language for building conceptual queries within the ORM context.

Although RIDL was intended for data(base) modelling, its main syntactic principles have been reconsidered to be adopted for the development of an ontological commitment language, simply called Ω-RIDL ("Ω" refers to "ontology base").

4.2 Defining an ontological commitment

An ontological commitment defined in Ω-RIDL consists of four distinct parts:

1. a commitment declaration,
2. a lexical interpretation layer,
3. a lexical association layer,
4. a semantic constraint layer.

In the following subsections we will focus on each part separately. It will also be clear that these four parts together define an ontological commitment; they are closely linked with each other and therefore are not to be seen as independent of each other.

To give the reader already an idea of how an ontological commitment definition looks like, a highly trimmed version of the ontological commitment definition corresponding to our case study is therefore given below:

```
define commitment in context MEDICINE with subsumption IS_A/[]
 lexical interpretations
   map FIRMS.COUNTRY_NAME=CANADA
   on  CANADA IS_A "COUNTRY - STATE" [] [HAS_ASSOC] ENTERPRISE
 lexical associations
   assoc FIRMS.COUNTRY_NAME=CHINA with "COUNTRY - STATE"
 semantic constraints
   each ENTERPRISE HAS_ASSOC exactly one "COUNTRY - STATE"
end
```

In this example, words in upper case are elements of either the committing relational database, either the committed ontology base; words in lower case are keywords of Ω-RIDL. Double quotes are used in the language to denote a terminal consisting of more than one string which are separated from each other by blank spaces. Note how the language aims at defining an ontological commitment close to its natural formulation. As a result, most syntactic expressions can be naturally read and understood by humans.

[11] CONceptual QUERy

4.3 Commitment declaration

The commitment declaration states the context in which the commitment will be defined, referenced by its name from the ontology base, and the ontological relation(s) that will be interpreted in the commitment as subsumption relation(s). Such an ontological relation, referenced by resp. its role and co-role labels, must be described by at least one lexon within the declared context. As a result, the specialisation of a "super"-term will play a declared role (e.g., "is a") in the commitment, and the generalisation of a "sub"-term will play a corresponding declared co-role (e.g., "subsumes").

A commitment declaration is syntactically simplified to one sentence, e.g.:

```
define commitment in context MEDICINE with subsumption IS_A/[]
```

Note that this example introduces a so-called *syntactic placeholder*, expressed with "`[]`", which denotes a non-existing co-role in the ontology base. Such placeholders were introduced in the language because most co-roles are not modelled in LinKBase® [12]. They serve as null values which can be replaced if a corresponding co-role is eventually modelled in the ontology base by an authorised ontology engineer.

4.4 Lexical interpretation layer

The lexical interpretation layer contains lexical mappings. A lexical mapping defines a mapping of a formula expressing a path of the relational database (e.g., the attribute expressed by the formula "FIRMS.CITY") on a path in the ontology base.

An ontological path is recursively defined as an ordered sequence of lexons from the ontology base, within the declared context. A minimal ontological path is constructed from one lexon, e.g.:

```
"MEDICINAL PRODUCT" HAS-INGREDIENT "INGREDIENT OF MEDICINAL SUBSTANCE"
```

For reading convenience we do not include the corresponding co-role here. However, in some cases the co-role must be explicitly specified to disambiguate which lexon is interpreted. Let us clarify this with an example. Imagine following lexons being modelled in the ontology base [13]:

```
<MEDICINE,PHYSICIAN,HAS_ASSOC,XXX,PATIENT>
<MEDICINE,PHYSICIAN,HAS_ASSOC,YYY,PATIENT>
```

If the first lexon has to be interpreted, we have to express a (minimal) ontological path as follows:

[12] An ontology engineer is not allowed to model a relation between a $concept_x$ and a $concept_y$ in LinKBase®, if, according to the real world, that relation does not hold for *each possible instance of* $concept_x$.

[13] Note that these lexons cannot be modelled in LinKBase®.

```
PHYSICIAN HAS_ASSOC [XXX] PATIENT
```

where the co-role of the lexon to be interpreted is explicitly specified between square brackets. In the case of a non-existing co-role, we use the same syntactic placeholder we already introduced earlier, e.g.:

```
"COUNTRY - STATE" [] [HAS_ASSOC] ENTERPRISE
```

For understanding convenience we explicitly specify the corresponding role between square brackets.

The next step is then to add a lexon with a common term to a minimal ontological path, e.g.:

```
CANADA IS_A "COUNTRY - STATE" [] [HAS_ASSOC] ENTERPRISE
```

is constructed from the following two lexons:

```
<MEDICINE,CANADA,IS_A, ,COUNTRY - STATE>
<MEDICINE,ENTERPRISE,HAS_ASSOC, ,COUNTRY - STATE>
```

We distinguish two kinds of lexical mappings: *reference* mappings and *relation* mappings. A reference mapping expresses a mapping involving a reference path from the committing relational database. Such a reference path is an attribute or an attribute value, and is expressed by an intuitive formula, e.g., "FIRMS.CITY". The following reference mapping involves an attribute being mapped:

```
map LISTINGS.DOSAGE_FORM
on  "MATERIAL ENTITY BY PRESENTATION SHAPE" [] [HAS_ASSOC] "MEDICINAL PRODUCT"
```

and must be read and interpreted as follows: the relation "LISTINGS" contains an attribute "DOSAGE_FORM" that semantically corresponds with "MATERIAL ENTITY BY PRESENTATION SHAPE" that has a relation with role "HAS_ASSOC" with "MEDICINAL PRODUCT" to which "LISTINGS" semantically corresponds. In other words, "LISTINGS" is mapped on "MEDICINAL PRODUCT", the "." is mapped on the relation with role "HAS_ASSOC", and "DOSAGE_FORM" is mapped on "MATERIAL ENTITY BY PRESENTATION SHAPE". Attribute values can reflect ontological knowledge as well, and therefore it is sometimes necessary to define reference mappings at the level of attribute values, e.g:

```
map FIRMS.COUNTRY_NAME=CANADA
on  CANADA IS_A "COUNTRY - STATE" [] [HAS_ASSOC] ENTERPRISE
```

In this example, the subsumption relation is to be found between the attribute "COUNTRY_NAME" and its values (e.g., "CANADA").

Some attributes are merely added to the relational database schema as unique tuple identifiers, and therefore reflect no semantics in the application's UoD. Next to that, they are often the result of a decomposition during normalisation,

and function as foreign keys. However, a foreign key often semantically corresponds with a (direct or indirect) relation between two terms in the ontology base. Let us clarify this with some examples. In the following relation mapping, a foreign key (expressed by a formula) is mapped on the direct relation between two terms:

```
map (LISTINGS.FIRM_SEQ_NO = FIRMS.FIRM_SEQ_NO)
on  "MEDICINAL PRODUCT" (HAS_ASSOC) ENTERPRISE
```

For understanding convenience we use parenthesis to delimite which element is mapped on which element. In some cases, a combination of foreign keys needs to be mapped. In the following example, the combination of two foreign keys is mapped on the direct relation between two terms:

```
map (LISTINGS.LISTING_SEQ_NO = ROUTES.LISTING_SEQ_NO,
     ROUTES.ROUTE_CODE = TBLROUTE.ROUTE_CODE)
on  "MEDICINAL PRODUCT" (HAS-PATH) "ROUTE OF ADMINISTRATION"
```

4.5 Lexical association layer

The lexical association layer contains (possible) lexical associations. A lexical association defines an association between a reference path of the relational database, which is meaningful in the considered UoD, with a term of the ontology base. A reference path is a formula expressing an attribute or attribute value which has not already been mapped by a reference mapping defined in the lexical interpretation layer.

Lexical associations are also to be seen as syntactic placeholders. Let us clarify this by following example:

```
lexical interpretations
  map FIRMS.COUNTRY_NAME=CANADA
  on  CANADA IS_A "COUNTRY - STATE" [] [HAS_ASSOC] ENTERPRISE
lexical associations
  assoc FIRMS.COUNTRY_NAME=CHINA with "COUNTRY - STATE"
```

The attribute value “FIRMS.COUNTRY_NAME=CHINA” could not be mapped because the ontology base does not contain a semantically corresponding term, e.g., “CHINA”. Therefore, it is lexically associated with the term “COUNTRY - STATE” in expectation from a corresponding lexon involving the associated term, e.g., the lexon <MEDICINE,CHINA,IS_A, ,COUNTRY - STATE>. If this lexon is eventually modelled in the ontology base by an authorised ontology engineer, the above association can be transformed to a reference mapping, i.e.:

```
map FIRMS.COUNTRY_NAME=CHINA
on  CHINA IS_A "COUNTRY - STATE" [] [HAS_ASSOC] ENTERPRISE
```

4.6 Semantic constraint layer

The semantic constraint layer accounts for the intended meaning of the conceptualisation by defining one or more constraint rules on interpreted lexons. These rules reflect (as good as possible) the rules intended by the UoD of the application, e.g., the integrity constraints of the committing relational database. The syntax in which these constraint rules are expressed is adopted from the old RIDL, e.g.:

```
each ENTERPRISE HAS_ASSOC exactly one "COUNTRY - STATE"
```

expresses the rule that each application instance of "ENTERPRISE" must play the role "HAS_ASSOC" with "COUNTRY - STATE" exactly once. This rule constrains a lexon interpreted through a following reference mapping:

```
map FIRMS.COUNTRY_NAME=CANADA
on  CANADA IS_A "COUNTRY - STATE" [] [HAS_ASSOC] ENTERPRISE
```

and reflects the attribute "COUNTRY_NAME" not allowing null values, i.e. each particular firm is located in exactly one country or state (according to the considered UoD).

5 Deploying Ontological Commitments for Mediation

Defining ontological commitments for relational databases (or applications in general) must aim for some practical use. In this section we demonstrate how an ontological commitment (defined in Ω-RIDL) can be deployed for mediation, i.e. the translation of a conceptual query (query on ontology level) into a correct logical query (query on database level).

By adopting the ORM diagram notation we graphically represent an ontological commitment by a tree. Figure 4 presents a part of the ontological commitment of the NDC Directory represented by such a tree. An ontological commitment tree is constructed by connecting the ontological paths from the the lexical interpretation layer of the ontological commitment definition. A dashed ellipse (a LOT in the original NIAM context) represents the start term of an ontological path. Terms other than the start term involved in an ontological path are represented by solid ellipses (NOLOTs in the original NIAM context). Subsumption relations are represented by arrows; other ontological relations are represented by boxes. Boxes highlightened in bold indicate that relation mappings are involved. The combination of the dot and box arrow graphically represents the constraint rule: *each ENTERPRISE HAS_ASSOC exactly one "COUNTRY - STATE"*.

A conceptual query can now be formulated by constructing a subtree of our ontological commitment tree. Let us demonstrate this with an example. A naturally formulated query can be: *list all cities in Germany in which enterprises are located that are related to medicinal products having a nasal route of administration*. By adopting RIDL\qu (the query/update part of the old RIDL) and the syntactic placeholder mechanism of Ω-RIDL, this query can be formally written down as:

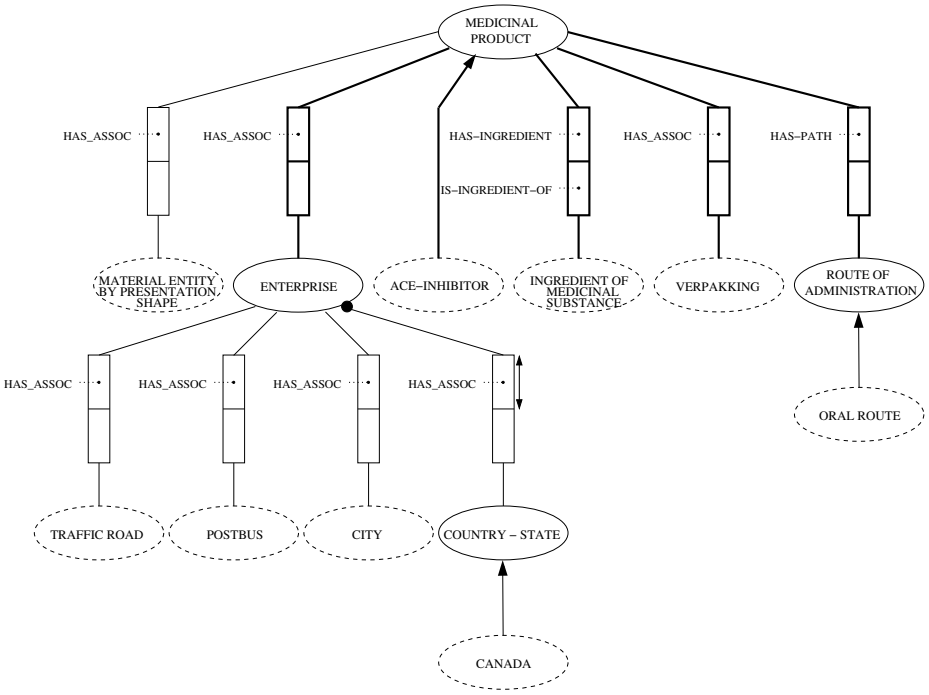

Fig. 4. Part of the ontological commitment of the NDC Directory represented by a tree graphically depicted by adopting the ORM notation.

```
list CITY [] [HAS_ASSOC] ENTERPRISE
                         (HAS_ASSOC GERMANY
                          and
                          [] [HAS_ASSOC] "MEDICINAL PRODUCT"
                                         HAS-PATH "ORAL ROUTE")
```

Figure 5 presents the graphical representation of this query as a subtree of the ontological commitment tree of Figure 4. The translation of this conceptual query into a correct logical query is done by a tree traversal:

- the left "selection" branch is traversed buttom-up;
- the middle and right "condition" branches are traversed top-down, connecting them with the logical "and"-operator (as specified by our formulated conceptual query).

During this traversal we deploy the reference and relation mappings defined in the corresponding ontological commitment to decide whether (part of) a branch of the conceptual query tree is visible in the committed relational database and, if so, in what we have to translate it. Figure 6 presents the resulting SQL query. Boxes denote elements from reference mappings; boxes highlightened in

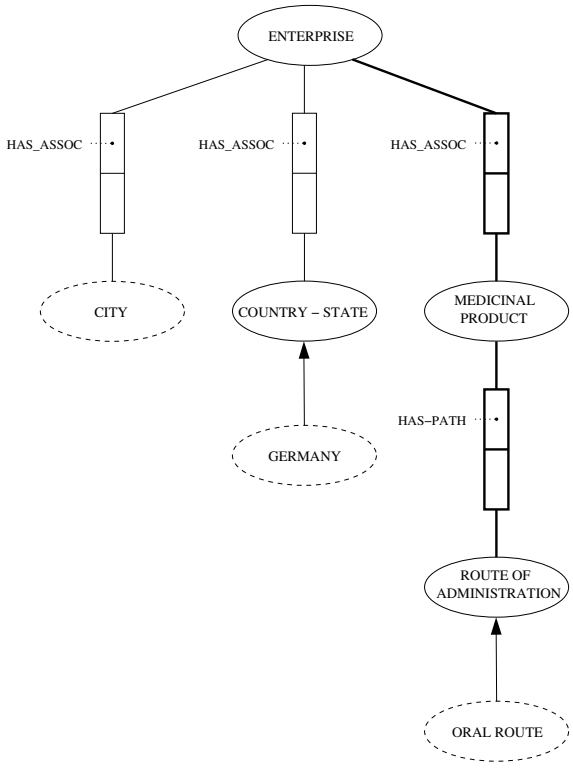

Fig. 5. Example of a conceptual query represented as a subtree of an ontological commitment tree.

```
SELECT FIRMS.CITY                                              → left selection branch
FROM FIRMS, LISTINGS, ROUTES, TBLROUTE
WHERE ( FIRMS.COUNTRY_NAME = "GERMANY"                         → middle condition branch
        AND
        ( FIRMS.FIRM_SEQ_NO = LISTINGS.FIRM_SEQ_NO
          AND
          LISTINGS.LISTING_SEQ_NO = ROUTES.LISTING_SEQ_NO      → right condition branch
          AND
          ROUTES.ROUTE_CODE = TBLROUTE.ROUTE_CODE
          AND
          TBLROUTE.ROUTE_NAME = "ORAL" )
```

Fig. 6. The SQL query as a result of the conceptual query translation.

bold denote elements from relation mappings. The execution of this SQL query on the relational database of the NDC Directory finally returns us the desired instance data.

Apart from their use in mediation, conceptual queries are also important, as argued in [23], as a convenient way to formally define and specify end-user profiles, intended to customise an individual's interaction with the system. Intuitively, the result of an ontology query, or user profile, is a set of (concept) terms, together with the query formulation itself that implies the intended relationships between the concepts as seen and expected by the end-user.

6 Related Work

Efforts on integration of heterogeneous datasources can be divided into two main categories. A first category of approaches have in common that they build a global conceptual datamodel from different datasources. The second category follows a fundamental different approach in that datasources are mapped to *existing* domain ontologies. The methodology for data integration proposed in this paper is classified under the second category. We will now give a classification of various approaches in the first category.

1. *Schema integration:* In this case, the input of the integration process is a set of source schemata, and the output is a single (target) schema representing the reconciled intentional representation of all input schemata (i.e. a global conceptual schema). The output includes also the specification of how to map source data schemata into portions of the target schema. This kind of schema integration is often referred to as *view integration* in the database research community. View integration is considered as an essential step in database design. A stepwise methodology for schema integration is given in [2].
2. *Virtual data integration:* The input is a set of source data sets, and the output is a specification of how to provide a global and unified access to the sources in order to satisfy certain information needs. The data are kept only in the sources. These sources also remain autonomous throughout the whole process and are queried using views. Database integration [2] appears in distributed databases environments and has as main goal the design of an integrated global schema (often called a virtual view) from local schemata. Virtual data integration is not only restricted to databases but may as well be extended to other kinds of datasources (structured, semi-structured, or not structured at all). In [3] Bergamaschi illustrates how the MOMIS system is built. Briefly summarised, wrappers are responsible for translating the original description languages of any particular source into a common data language and to add all information needed by the mediator, such as the source name and the type. Above the wrappers there is the mediator, which is a software module that has as most important task to build a global conceptual schema. Queries are then formulated against the global schema

and are translated into local queries. The query result is then combined by the mediator and presented to the user. The TSIMMIS project [7] is primarily focused on the semi-automatic generation of wrappers, translators, and mediators.

3. *Materialised data integration:* As in the previous case, the input is a set of source data sets, but here the output is a data set representing a reconciled view of the input sources, both at the intentional and the extensional level. The field of data integration with materialised views is the one most closely related to data warehousing.
4. *Data Warehousing:* With the aid of wrappers and mediators a datawarehouse schema is formed of the local source schemata. The datawarehouse itself is responsible for storing the data of the local sources. Source integration in data warehousing identifies three perspectives: *a conceptual perspective*, *a logical perspective* and a *physical perspective* [15].

The OBSERVER system [20], which belongs to the second category, has already been discussed in section 1. We have argued the differences between our framework and that of OBSERVER. Another important difference which has been described in this paper is the ability we provide to impose semantic domain constraints at the ontology level. This aspect is completely absent in the OBSERVER project. Another project, comparable with OBSERVER, is SIMS [1]. In this system the different information sources are accessed using a system based on Description Logics, Loom. The CARNOT project [8] used the global upper ontology Cyc to describe the whole information system. A key shortcoming with this approach is the difficulty and complexity of managing a large global ontology (more than 50.000 entities and relationships). For this reason we have focused on an approach that involves the use of multiple ontologies as stated in the introduction.

7 Future Work

A prototype of a compiler, called *omegaridlc*, is developed to support the language in the current DOGMA ontology framework. It enables an automatic verification of an ontological commitment definition on syntax and semantics, and the translation to a more machine processable form. This form is currently a markup version of the language, expressed in the popular XML, which enables a more convenient adaptation by existing ontology-based mediation technology, e.g., the MaDBoks[14] system [9] developed by L&C N.V. as an extension to their LinKFactory® Ontology Management System [6]. This adaptation and its implementation is currently being investigated as part of the SCOP project.

8 Conclusion

In this paper we have focused on a new ontological commitment language called Ω-RIDL. This language is developed to naturally describe how elements of a

[14] MApping DataBases Onto Knowledge Systems

relational database semantically correspond to a (given) domain ontology. One of the novel aspects of this language is the support of imposing semantic domain constraints at the ontology level.

By means of a real-life case study we have explained the principles of the language, and demonstrated how a syntactic placeholder mechanism was introduced to overcome assumed incompleteness of the given domain ontology. Further, we have illustrated how some of its key constructs can conveniently be reused in a conceptual query language. We demonstrated this with an example of how an ontological commitment defined in Ω-RIDL can be deployed for mediation, i.e. the process of translating a conceptual query (query on ontology level) to a correct logical query (query on database level).

Acknowledgments: This work has been funded by the IWT (Institute for the Promotion of Innovation by Science and Technology in Flanders): Pieter Verheyden is supported in the context of the SCOP project ("Semantische Connectie van Ontologieën aan Patiëntgegevens"; IWT O&O #020020/L&C), while Jan De Bo has received an IWT PhD grant (IWT SB 2002 #21304). We also like to thank Carlo Wouters (La Trobe University; VUB STARLab) and Tom Deray (L&C N.V.) for their valuable comments on earlier drafts of this paper.

References

1. Arens, Y., Knoblock, C. and Shen, W. (1996) Query Reformulation for Dynamic Information Integration. In *Journal of Intelligent Information Systems, 1996 6(2-3), pp. 99-130.*
2. Batini, C., Lenzerini, M. and Navathe, S. (1986) A Comparative Analysis of Methodologies for Database Schema Integration. In *ACM Computing Surveys, 1986 18(4) Dec, pp. 323-364.*
3. Bergamaschi, S., Castano, S., De Capitani di Vimercati, S., Montanari, S. and Vincini, M. (1998) An Intelligent Approach to Information Integration. In *Guarino, N. (ed.), Formal Ontology in Information Systems, Proceedings of the First International Conference (FOIS'98), IOS Press, pp. 253-267.*
4. Bloesch, A., and Halpin, T. (1996) ConQuer: a Conceptual Query Language. In *Proc. ER'96: 15th International Conference on Conceptual Modeling, Springer LNCS, no. 1157, pp. 121-33.*
5. Bloesch, A., and Halpin, T. (1997) Conceptual Queries using ConQuer-II. In *Proc. ER'97: 16th International Conference on Conceptual Modeling, Springer LNCS, no. 1331, pp. 113-26.*
6. Ceusters, W., Martens, P., Dhaen, C. and Terzic, B. (2001) LinKFactory® : an Advanced Formal Ontology Management System. K-CAP 2001, Victoria, Canada, October 2001.
7. Chawathe, S., Garcia-Molina, H., Hammer, J., Ireland, K., Papakonstantinou, Y., Ullman, J. and Widom, J. (1994) The TSIMMIS project: Integration of heterogeneous information sources. In *Proceedings of the* 10^{th} *Anniversary Meeting of the Information Processing Society of Japan, pp. 7-18.*
8. Collet, C., Huhns, M. and Shen W. (1991) Resource Integration Using a Large Knowledge Base in CARNOT. In *IEEE Computer, 24(12), pp. 55-62.*

9. Deray, T. and Verheyden, P. (2003) Towards a Semantic Integration of Medical Relational Databases by Using Ontologies: a Case Study. In *Meersman, R., Tari, Z. et al. (eds.), On the Move to Meaningful Internet Systems 2003 (OTM 2003) Workshops, LNCS 2889, Springer-Verlag, pp. 137-150.*
10. De Troyer, O., Meersman, R. and Verlinden, P. (1988) RIDL* on the CRIS Case: a Workbench for NIAM. In *Olle, T.W., Verrijn-Stuart, A.A., Bhabuta, L. (eds.), Computerized Assistance during the Information Systems Life Cyle, Elsevier Science Publishers B.V. (North-Holland), pp. 375-459.*
11. De Troyer, O. (1989) RIDL*: A tool for the Computer-Assisted Engineering of Large Databases in the Presence of Integrity Constraints. In *Clifford, J., Lindsay, B., Maier, D. (eds.), Proceedings of the ACM-SIGMOD International Conference on Management of Data, ACM Press, pp. 418-430.*
12. Flett, A., Casella dos Santos, M. and Ceusters, W. (2002) Some Ontology Engineering Processes and Their Supporting Technologies. In *Gómez-Pérez, A., Richard Benjamins, V. (eds.), Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web, EKAW 2002, LNCS, Springer-Verlag, pp. 154-165.*
13. Guarino, N., and Giaretta, P. (1995) Ontologies and Knowledge Bases: Towards a Terminological Clarification. In *Mars, N. (ed.), Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing, IOS Press, Amsterdam, pp. 25-32.*
14. Halpin, T. (2001) Information Modeling and Relational Databases (From Conceptual Analysis to Logical Design). Morgan Kauffman, 2001.
15. Jarke M., Lenzerini, M., Vassiliou, Y. and Vassiliadis, Y. (1999) Fundamentals of Data Warehouses. Springer-Verlag, 1999.
16. Meersman, R. (1982) The High Level End User. In *Data Base: The 2nd Generation, Infotech State of the Art Report (vol. 10, no. 7), Pergamonn Press, U.K., 1982.*
17. Meersman, R. (1999) The Use of Lexicons and Other Computer-Linguistic Tools in Semantics, Design and Cooperation of Database Systems. In *Zhang, Y., Rusinkiewicz, M., Kambayashi, Y. (eds.), Proceedings of the Conference on Cooperative Database Systems (CODAS 99), Springer-Verlag, pp. 1-14.*
18. Meersman, R. (2001) Ontologies and Databases: More than a Fleeting Resemblance. In *d'Atri, A., Missikoff, M. (eds.), OES/SEO 2001 Rome Workshop, Luiss Publications.*
19. Meersman, R. (2002) Semantic Web and Ontologies: Playtime or Business at the Last Frontier in Computing? In *NSF-EU Workshop on Database and Information Systems Research for Semantic Web and Enterprises, pp. 61-67.*
20. Mena, E., Kashyap, V., Illaramendi, A. and Sheth, A. (1998) Domain Specific Ontologies for Semantic Information Brokering on the Global Information Infrastructure. In *Guarino, N. (ed.), Formal Ontology in Information Systems, Proceedings of the First International Conference (FOIS'98), IOS Press, pp. 269-283.*
21. Reiter, R. (1984) Towards a Logical Reconstruction of Relational Database Theory. In *Brodie, M., Mylopoulos, J., Schmidt, J. (eds.), On Conceptual Modelling, Springer-Verlag, pp. 191-233.*
22. Spyns, P., Meersman, R. and Jarrar, M. (2002) Data Modelling versus Ontology Engineering. *SIGMOD Record: Special Issue on Semantic Web and Data Management, 2002, 31(4), pp. 12-17.*

23. Stuer, P., Meersman, R. and De Bruyne, S. (2001) The HyperMuseum Theme Generator System: Ontology-based Internet support for the active use of digital museum data for teaching and presentation. In *Bearman, D., Trant, J. (eds.), Museums and the web 2001: Selected Papers, pp. 127-137. Archives & Museum Informatics, Pittsburgh, PA, 2001.*
Available at: `http://www.archimuse.com/mw2001/papers/stuer/stuer.html`
24. Verheijen, G. and Van Bekkum, J. (1982) NIAM, aN Information Analysis Method. In *Olle, T., Sol, H., Verrijn-Stuart, A. (eds.), IFIP TC-8 Conference on Comparative Review of Information System Methodologies (CRIS-1), North-Holland.*
25. Wintraecken, J.J.V.R. (1985) Informatie-analyse Volgens NIAM. Academic Service, 1985. (English version published by Kluwer Academic Publishers, 1990).