

VRIJE UNIVERSITEIT BRUSSEL

FACULTEIT WETENSCHAPPEN VAKGROEP INFORMATICA EN TOEGEPASTE INFORMATICA SYSTEMS TECHNOLOGY AND APPLICATIONS RESEARCH LAB

STAR Lab Technical Report

An Architecture and Toolset for Practical Ontology Engineering and Deployment: the DOGMA Approach.

Robert Meersman & Mustafa Jarrar

affiliation keywords number date corresponding author status reference

ontology engineering, semantic web, architecture STAR-2002-06 22/10/2002 13:50 Mustafa Jarrar accepted in a modified form (see Technical Report STAR-2002-08) original version of the paper for ODBASE02

An Architecture and Toolset for Practical Ontology Engineering and Deployment: the DOGMA Approach.

Robert Meersman, Mustafa Jarrar

VUB STARLab Vrije Universiteit Brussel Pleinlaan 2 – B-1050 Brussels, Belgium {meersman, mjarrar}@vub.ac.be

Home page: http://www.starlab.vub.ac.be/

Abstract. This paper presents a specifically database-inspired approach (called DOGMA) for engineering formal ontologies, implemented as shared resources used to express agreed formal semantics for a real world domain. Our methodology aims to addresses several related issues, such as (a) the scalability of building and sharing ontologies; (b) the maximization of knowledge reusability; (c) the design and engineering process, that also simplifies building and managing ontologies; (d) the coexistence of several rule systems and ontology languages around a same ontology; and (e) the reconcile of the need to represent semantics independently from language with the need to create and use processes entirely rooted and described in (natural) language. We first define formal ontologies in a logic sense, i.e. as "representationless" mathematical objects that form the range of a classical interpretation mapping from a first order language (sometimes called a conceptual schema, and assumed to lexically represent an application), to a set of possible ("plausible") conceptualizations of the real world domain. We then give a database-inspired "view" on implementations of ontologies seen as resources. Following common model-theoretic database practice we decompose such resources into ontology bases and into of their explicit so-called ontological commitments. Such architecture allows to make the latter (crucial) notion explicit as a separate layer, with concrete and dedicated services, mediating between the ontology base and the application instances that commit to the ontology. We claim it also leads to methodological approaches that naturally extend database modeling theory and practice, and so may in turn lead to scalable solutions for ontology-based systems. We discuss examples of the DOGMA implementation of the ontology base server and commitment server.

1 Motivation, context and overview of related work

What are ontologies. Computer science (re-)defines ontology as a branch of knowledge engineering, where agreed semantics of a certain domain is represented formally in a computer resource, which then enables sharing and interoperation between information systems (IS). Representing the formal semantics for a certain domain implies conceptualizing the domain objects and their interrelationships in a declarative way. Ontologies should therefore represent formal and agreed so-called *ontological commitments* (for definitions, see below) needed for new open environments (e.g. electronic commerce, B2B, semantic web). In an open environment autonomous applications possibly developed without *a priori* knowledge about each other, need to communicate to exchange data in order to make transactions interoperate.

For the time being and for mental imagery's sake, picture such an ontology as a set of object (type-)s and their conceptual relationships expressing possible facts in a domain (an EER or ORM diagram labeled with natural language terms will do fine), plus first order theory expressing rules, constraints, ... involving the concepts over this domain. However, below we shall emphasize that a correct understanding of true ontologies is possible only if they are considered as language- and task-independent entities, and the interaction with their necessarily lexical representation forms an essential aspect of their effective use. Indeed, reconciling the requirement of representing semantics independently from language while necessarily all agreement must be entirely rooted and negotiated in natural language by itself turns ontology engineering into a compelling research subject. To a database engineer the following parallel may perhaps be enlightening: implementations of ontologies will in a sense achieve a form of "semantics independence" for information- and knowledge based systems. Just like database schemas achieved data independence by making the specification and management of stored data elements external to their application programs, ontologies now will allow to specify and manage domain semantics external to those programs as well.

Ontologies are shared computer-based resources. The fundamentally *a-priori*shared nature of an ontology however makes it important even essential for our understanding, to realize that ontology engineering is more than data modeling, even when taking business rules into account [DJM02]. For example, representing the formal semantics of the air travel domain is more than a set of data models for a number of airline reservation systems, which likely would have been autonomously specified for optimal use within an individual organization or company. Thus, an ontology is more generic than a data model. Furthermore, an ontology is more than a mere "is_a"- taxonomy of terms, as it often seems to suggest in the literature. It includes a much richer set of relationships, such as instance_of, part_of, ..., which all might deserve a "generic semantics". Not surprisingly this turns out to be an important methodological and tool support issue. Sharing concepts, often even just *identifying* them, "independently" of language and across representational paradigm boundaries obviously is a hard problem of semantics, extensively studied in various forms of schema and view integration, mostly in organization-specific contexts, within the database field ([Sa98] [AB01] [ZSC01] [PS98]).

Why ontologies: databases and the internet. Research on ontologies is not a new field as such. It was the subject of rather vigorous activity within AI, but has received a recent renewed impetus as a result of a number of relevant factors that incidentally are not specific to the AI domain (for an excellent survey of this evolution up to 1995, see [UG96]). Today's technology indeed creates a "push" effect by the availability of a large number of various relevant computerized terminological resources such as

lexicons, thesauri and glossaries. These are in general not yet real ontologies, but achieve at least shareability, sometimes across domains, though their use of natural language. Also, very large numbers of *database schemas* exist that due to their nature often epitomize high-quality knowledge representation (but not always an "agreed" one --see below) and so may serve as basis to be "ontologized" into more standardized representations of knowledge about many application domains. In this paper we study and make more precise the correspondences -as well as some differences- between database modeling and ontology engineering that was first mentioned and argued in [M99a] [M99b] [M01]. On the "pull" side, there is a pressing requirement to harvest the wealth of mostly unstructured knowledge that is present on the Web. Most notably, the so-called Semantic Web effort [B99] proposes to turn the web into a resource that allows to make more meaningful ("semantical") and therefore more productive use of that knowledge. In view of the Web's size, this conversion clearly needs to be automated as much as possible. In particular e-business requirements especially related to intelligent Web services provide serious economical "pull" justification; for a recent analysis of this see [F01].

Note that these factors were not in play in the 1980s, which may account in part for the fact that at that time ontology research failed to make substantial progress in achieving "product quality" computerized support for common sense reasoning, yet originally seen as one of the main applications for artificial intelligence [H85]. Clearly however these are now critical success factors for ontologies, and note, they all implicitly involve the notions of *size* and *scalability*. As should be obvious (and argued in more detail further in this paper), certain methods and techniques for conceptual modeling from the database field, and for the efficient organization, running and maintenance of very large datasets may therefore be of use.

One important issue in this respect, but not one studied in depth in this paper, will be the efficiency also of emerging tools, such as the DOGMAModeler presented in Section 5 supporting (a) the above conceptual ontologization process, where performance is measured by *human-machine interface* factors (e.g. CASE) and by the time-complexity of e.g. "alignment" and "merging" algorithms, and (b) the actual running of the ontology service, with performance measured by information retrieval factors and by classical trade-offs between space and time common to database transaction processing.

Ontologies must be scalable resources. As the main purpose of an ontology is to be a shared and agreed semantic resource in a wide range of agents, building scalable ontologies is to be carried out by groups of people, taking into account that ontologies grow over time [KF01]. Before building up an ontological theory, builders should first generate a consensus about one conceptualization. In [GG95] generating such a consensus is a mental process and done by exemplifying, testifying, investigating etc, or by a so-called Adequacy Search as proposed in [NCM+00]. Such a process will inevitably be oriented to the tasks to be carried out, and are likely to be influenced also by personal tastes and may even reflect fundamental disagreements [BM99]. Several conceptualizations could be chosen for the same domain [GN87], especially in large-

scale and multi-domain ontologies, which may lead to potentially inconsistent (and incomplete) ontologies. In addition, we believe that this slows down the construction of an ontology and increases the costs. Notice that the difficulties and disagreements in the conceptualization process normally appear at a "deeper" level of abstraction. An experience about conceptual heterogeneity and ontology integration [GPS98] outlined that "disagreement persists at a deep, 'ontological' level ...". This level is also known as "Detail Level" by [SGP97]. The details and the rules that constrain the structure and interrelationships of the concepts. In other words, constraints, rules and procedures are essential to achieve an understanding about a domain's semantics, but agreement about them in general is very difficult and nearly always specific to a context of application. Furthermore, from an ontology's application point of view constraints are likely there to limit updates of data stores that exist entirely within that application's realm, the actual consistency of which will not be the ontology's responsibility. For example it is easy to agree that "person has a blood-pressure", while a disagreement might be on whether the actual value of this pressure is (too) high in a given context. People could agree on "a book has ISBN" but might disagree whether for a given application that ISBN is a mandatory property for the book to have, or "person has age", but might disagree on the range. As a result, people can agree easily about the basic facts in a domain that are at a higher level of abstraction, while disagreements mostly appear at a lower level of abstraction, i.e. the details and constraints of these facts.

Knowledge reusability is another important goal of building ontologies ([IFFJ97] [UG96] [MFGB99] [GPB99] [G95]). As a result of a conceptualization process, an ontological theory will stand as a formal resource of knowledge, reusing such resources means sharing the same conceptualization. In the activity of knowledge reuse, ontologies may only need to be reused partially, for example, when building a "Carrental" ontology, one may need to reuse the "Payment" aspects from an existing "Shopping" ontology, where they share the same conceptualization about a certain set of axioms. Sharing a partial conceptualization (as a result of partial agreement) across two ontologies depends on the level of abstraction, i.e. less shareability appears in the deeper knowledge (as was discussed above). To improve knowledge reusability, several researchers from the problem-solving area (e.g. Chandrasekaran and Johnson [CJ93], Clancey [C92], or Swartout and Moore [SM93]) have proposed the idea of structuring the knowledge into different levels of abstractions, where Steels in [S93] proposed a componential framework that decomposes a knowledge level into reusable components. In addition to the level of abstraction, several issues related to the reusability of knowledge are outlined and discussed in [R00] such as the importance of context, the need for more knowledge, etc.

Many believe that building large knowledge bases will only be possible if efforts are combined (Neches et al in [PFP+92]). Therefore, we argue that a unified framework to enable and maximize knowledge reusability should be followed. Such a framework must be scalable and allow for connecting of ontological theories regardless of the diversity of ontology languages and their representation models.

All of the above considerations will translate within DOGMA in an architecture that explicitly separates "base" facts in a domain from constraints, rules, identification, derivation etc that occur to support an application's *use* of an ontology.

Matching ontology methodology to architecture. Knowledge management is the corporate control of an organization's business data and metadata and of their use in applications that are increasingly connected to "external" business domain knowledge. From the above it should not surprise that effective corporate knowledge management is becoming dependent on the availability of semantic information resources. Most likely the most immediate business applications of ontologies will lie in this area ([F01]). As an organization's information typically resides in its (large) databases, data dictionaries, websites, documents, and in its people, this implies not just scalability and knowledge reusability but also a methodological approach to the "ontologization" of information resources at the individual organization level, one that is geared towards current information paradigms. Methodology implies teachability and repeatability, in general will be aimed at the involvement of non-computer experts, and therefore must be based on sound, easy to understand and broadly accepted principles. Naturally, any good methodology will closely reflect the architecture of the resulting system. For instance, the separation of facts and constraints indicated above allows a "database-style" architecture for ontologies and their use in information systems, which in turn leads to familiar techniques for the creation, deployment and maintenance phases in their lifecycles.

Structure of this paper: in section 2 we discuss fundamental challenges and goals for engineering and deploying ontologies, and introduce and discuss these in our "DOGMA" framework. By examples, Section 3 illustrates this framework for building, (re)using, and mining ontologies. Section 4 briefly discusses aspects of the important issue of ontological consistency and versioning that emerge while engineering an ontology. Section 5 overviews design and implementation consequences for ontology tools (in particular the ontology base and commitment servers) under development as part of the DOGMA System at VUB STARLab. Section 6 then lists early conclusions and maps ongoing and future work.

2. The DOGMA Approach to Ontology Engineering

The definition of ontology (as a computer resource) has been presented in the literature from different points of view. According to Gruber an ontology is defined as "an explicit specification of a conceptualization", referring to an *extensional* notion of a conceptualization as defined e.g. in [GN87]; Guarino and Giaretta [GG95] pointed out that this definition does not adequately fit the purposes of an ontology. They argue that a conceptualization benefits from invariance under changes that occur at the instance level between different "states of affairs" in a domain, and thus should not be extensional. Instead, they define a conceptualization as an *intensional* semantic structure, which encodes implicit rules constraining the structure of a piece of reality. In

other words an ontology becomes a logical theory which gives an explicit, partial account of a conceptualization.

While we arrived at it independently from a database-inspired perspective [M99], in the DOGMA framework we embrace this viewpoint but unlike [GG95] and subsequent work by Guarino et al, we also pursue this idea to arrive at concrete software architectural and engineering conclusions.

In the following sections we treat the fundamental issues for engineering and deploying ontologies that follow from this in more detail.

While the scope of this paper does not allow a fully detailed exposition of DOGMA's formalism, in what follows we will refer to existing related literature and illustrate largely by example its —somewhat simplified— formal structure model for ontology engineering. The illustrations derive from a prototype ontology modeler-/server-/mining-/alignment environment currently under development in the authors' lab. It shall permit us to make hopefully explicit most of the key issues however in ontology organization, engineering, scalability and methodology listed above, starting from familiar database design principles.

The baseline of the approach, and of this paper, is that ontologies constitute a fundamental resource in any meaningful IS infrastructure and that tried and tested database technologies and methodologies can provide a real productivity benefit to ontology development. Conversely, by definition, database schemas and other components of IS specification (from program code to user interfaces to documentation) will benefit by acquiring a real formal semantic dimension.

2.1 Database inspiration for ontologies: the ontology base

It is precisely the above mentioned agreement aspect as basis for the formal semantics of information systems (see [M94] for an early position on this) that we claim makes classical, i.e. model-theoretic database technology and methodologies suitable for "reuse" in an ontology context, and therefore perhaps an interesting new research subject in its own right.

Most recent ontology research, and the resulting formalisms and languages [On-toWeb] indeed are based on versions of earlier *description logics* [BHP+99] [FHvH00] and so correspond more closely with the proof-theoretic view of database [R88] having natural implementations with Datalog and deductive databases in general. Although the proof-theoretic paradigm (arguably) is the more elegant and "general" one, and although the relationship between the model- and proof-theoretic views is well-understood since [R88] ff., it is undeniably that the model-theoretic view of databases gave rise to the eminently scalable technology and successful industry of high-performance DBMS, tools and applications. By providing to ontology engineering a precisely defined analogue to the model-theoretic paradigm of databases we find

that important methodological and productivity advantages are obtained as well as technological ones, such as scalability, performance and a "familiar" transition path from existing database environments. For the latter statement, evidence emerges in that even the prototypical DOGMA approach, while limited in other respects, is perceived by database practitioners and domain experts as fairly intuitive.

According to this well-tried model-theoretic database methodological principle, in the DOGMA framework we therefore decompose an ontology formally into an *ontology base*, or *ontobase* for short, a set of context-specific binary fact types which we call *lexons* (see example below), and instances of their explicit *ontological commitments*; the latter in our architecture become reified as a separate layer mediating between the ontology base and the instances of applications that commit to the ontology, see Fig.1.

Ontological commitments are then considered to be *interpreted* in terms of the contents of (possibly a selected context within) this ontology base.

Any computer representation of an ontology, albeit *by definition different* from the ontology itself, obviously must be lexically rendered (see Sowa's discussion about ontologies and semiotics [S00a]). It must also at least provide correct contextual identification of its concepts (possibly to be negotiated by its application instances) through some language. To maximize the "conceptual gain" of the interpretation, the formalism for specifying an ontology(-base) should be as simple as possible. Thus the ontology base is a set of conceptual relationships, i.e. while intuitive knowledge level and its formal semantics will be "approximately" specified in the commitment layer. To accommodate alternative "models" of reality, or even versions as knowledge about the world evolves e.g. through observations, the ontology base may contain many different conceptualizations, even about the "same" real world domain. In summary an ontology base is:

- A set of possible ("plausible") conceptualizations of the real world domain;
- Each is a set of context-specific binary facts types, called *lexons*. Notation: <γ: Term₁, role, Term₂>. Here γ∈ Γ is just an abstract *context identifier* chosen from a set, (more about this below). The lexical terms (Term₁, role, Term₂) express a binary conceptual relationship in some given agreed language. (Only one of its two roles is used below.)

Example. The following ontology base contains a single —obviously very incomplete— ontology-base with lexons in a hopefully self-evident syntax, the *-ID* suffix denotes abstract identifiers to assumed and agreed *contexts*:

```
(company-ContextID)
```

```
employee is_a person
employee is_a contract_party
employee has first_name
employee has last_name
employee has empl_id
employee has birth_date
employee has start_date
employee has salary
employee works_in department
{...}
```

(employment-ContextID)

salary	has amount_in-\$
salary	has amount_in-€
salary	synonym_of remuneration
salary	is_a remuneration
salary	expressed_in currency
salary	converted_to currency
{}	

A Note on Contexts. Contexts in the DOGMA lexon structure (for the purpose of this paper) appear just as abstract (in principle semantics-less, i.e. *uninterpreted* within this formalism) identifiers that intuitively relate to a "grouping" of "contextually related" lexons. They provide an ability for internal organization of an ontology base; each context identifier maps to a *source* (for example a document) containing an intended conceptualization with its implicit assumptions Somewhat informally, lexons are assumed (by an outside cognitive agent such as a human) to be "true" in that context, i.e. they are a partial model for the intended conceptualization seen as a logical "theory". In our initial implementation of the DOGMA framework, each context identifier is mapped to a documented source (e.g. in natural language, or a database, or a lexicon, ...). This assists in a methodological sense ontology engineers who build, maintain, or (re)use lexons in "knowing the intended meaning of a lexon".

The extraction of lexons from a context's source is a research topic in its own right, of course. For this paper however we assume that these extractions "are done" and merely provide an architecture with a repository that allows to store and manages the result of this process.

Contexts have been and are the subject of occasionally intense study notably in AI; examples are [McC93], specifying them as higher-order theories [S00a]. Work on contexts that is related to our approach is outlined in [R00] reporting on research effort under way for adding contexts into KIF in order to facilitate the translation of facts from one context to another. Also, large KBS such as CYC require context to be captured in order to applying knowledge for different domains.

2.2 The Commitment Layer

An *interpretation* is the mapping (semantics) from individual application instances (conceptual schema) syntactically described in some language, into the ontology base, which is assumed to contain conceptualizations of all relevant elementary facts. Logically speaking we —somewhat pragmatically— assume at present that "natural" interpretations behave like first order logical theories for which at least one context provides a logical model. In less precise other words, the applications implement semantics by correctly "satisfying", or in the words of Guarino and Giaretta [GG95], *committing to* (a subset of) the interpretations in terms of such a formally agreed ontology base. They see every relationship in an *ontological commitment* as an ordinary mathematical relation, and (in our terminology) mapable to a conceptual relationship in the ontology base.

In the DOGMA approach we give a new and concrete software engineering interpretation to this notion and in doing so, we *uncouple* (in the classical database and software engineering sense) ontological commitments from both the possible diversity of their representations and from the ontology base in which they are to be interpreted This improves scalability and reusability.

Ontological commitments in DOGMA furthermore become executable by *interpreters*; these can be a computer programs if commitments is expressed in a suitably computable formal language, or they can be humans or other intelligent agents who are able to agree on the commitment process. (Note the latter could be less obvious than it sounds; see the influential paper by De Millo et al. [MLP79]).

More formally speaking, the commitment layer is organized as a set of *ontological commitments*, each is an explicit *instance* of an (intensional) first-order *interpretation*; each commitment is a consistent set of rules (/axioms) in a given syntax that constrains an application (or also: *commits it ontologically*) to a particular aspect of reality (which is assumed to be conceptualized in the ontology base).

Example (verbalized in a suitable pseudo-NL syntax): <Each Manager who Heads a Company must also Works_For that Company>

For improving knowledge reusability, in a commitment layer the set of ontological commitments will be seen as a set of reusable knowledge components. Such components are connected since they share the same ontology base. In practice, similar applications reuse/inherit commitments from each other, which on the one hand will facilitate new applications to commit to and use the ontology, and on the other hand, successful commitments in certain domains and applications will likely become "popular" and therefore a *de facto* trusted resource in their own right for achieving interoperability, or just compatibility between applications.



Fig. 1: Knowledge organization in DOGMA Framework

A note on ontology as a formal semantics. As indicated earlier, an ontology(-base) is the range of the (first-order) commitments (seen as interpretation mappings) of the application (or rather its conceptual schema, which for convenience we shall assume to be expressed in a first order language). "Real" interpretations, which thus actually

are the definition of semantics, are truth-preserving mappings from the application to the "real world domain", usually called models. Lexons in a DOGMA ontology base are always "true", i.e. free of further interpretation. "Alternative truths" have to be provided in separate conceptualizations or contexts. Contexts that specify improbable or impossible (contradictory) worlds are possible, especially in the early stages of engineering an ontology, but in practice will have few or no applications that can commit to them. Note also that (some of) the actual instances of a real world may or may not be part of a given conceptualization. For instance, the notion described by the term "November" may refer to an instance in some conceptualizations, and to an ontological concept in others. This yields another reason why ontologies behave not quite the same as data models, although it suffices in this particular case to formally specify a "custom" interpretation of the "is_instance_of" relationship... The ontological commitments above are merely part of the specification of this mapping, namely they constitute the (intensional, abstracting from instances) interpretations of an application in terms of the ontobase.

Furthermore, this way of building and structuring ontologies helps to prevent application-specific rules and encodings to enter a shared ontology base. As an obvious result, building ontology bases and their commitments becomes easier and more scalable, because the rules and constraints (which mostly are the difficult part to agree) are moved to the commitment layer and the agreement about them within an ontological commitment is easier than it is within the whole ontology (as discussed before). Moreover, the possibilities for ontology mining are also easier, for example when mining ontological knowledge (lexons) as well as their rules (often as constraints) from certain high quality data sources such as corporate databases (see Example 2 in the next Section).

Naturally there is a trade-off between complexity and size that lies in the requirements to (a) manage the (huge) size and relative organizational complexity of the lexon base, (b) map nearly all application assumptions to the terms and relations of the lexons in the ontology, and (c) develop, link and manage (even index) the application- and domain-specific commitment packages (e.g. in the form of sets of constraints and functions). With the design of the DOGMA commitment Server discussed further in this paper we attempt to provide at least an initial solution to some of these problems.

The alert reader may have noted incidentally that our approach appears motivated —at least in part— by earlier experience with successful "semantical" database (-schema) modeling methodologies used in practice (ORM, Object-Role Modeling [H01] and NIAM, an Information Analysis Method [VB82], also "Nijssen's-" or "Natural"-IAM). This indeed allows to identify and analyze some of the essential differences between database- and ontology modeling. While we find that formal ontologies are best thought of as abstract, mathematical entities, as stated above any use of them must be through a (lexical, application) language. ORM and especially NIAM have strong methodological roots for handling this distinction. However, the principal modeling feature of ORM, the adoption of an explicit separation between lexical (term-) and non-lexical (concept-) knowledge, partly disappears in an ontology context (all

knowledge being lexical). In fact the precise ontological relevance of the "bridge" between the lexical and non-lexical knowledge base for the "ontology proper" is as yet not fully understood (it forms part of the ontological commitment) and is the subject of ongoing research.

3. Examples

3.1 Example 1: A simple ontology in the DOGMA framework

In the following example, with its necessary simplicity, we show part of a Trivial Conference ontology, used by two different conference application instances. Fig. 2 shows (using ORM in the VisioModeler tool) the graphical representation of this ontology. Notice that the ontology in this example is *supposed* to be specified at the knowledge level¹, i.e. is more than a data model for the application instances. Applications that commit to this ontology may retain their internal data models¹. For simplicity we only assume that the terms in the lexons are aligned (not necessarily identically, they can be constructed) with an *a priori* agreed lexicon (WordNet [WN] would be a good example).

¹ The Knowledge Level is a level of description of the knowledge of an agent that is independent of the symbol-level representation used internally by the agent [G95]

¹ Note that the commitments may be more than *integrity constraints* (to be committed by an application), such as derivation or reasoning rules that may help to enrich or filter queries.



Fig. 2: Trivial Scientific Conference ontology²

Each conference application instance in general will have certain rules that do not necessarily agree with those of other instances; application B for example agrees with application A on all lexons and rules, except those grouped as "A" in Fig. 2, likewise application A agrees with everything except those rules grouped as "B". For instance, application A identifies a Paper by Paper_Number, while application B identifies the same paper by the combination of Paper_Title and a reference to its Author. Also in application B, the Person who presents a Paper must be the Author of this Paper, while in application A this rule does not exist.

Building such ontologies by allowing only partial agreement about the conceptualization of a domain obviously is difficult and complex, but realistic. As discussed before, in such cases (which are common in open environments): (1) the completeness of an ontology [as an important ontological principle] should be considered and managed, while (2) applications might not commit to an ontology because they do not agree (or at most partially in typical cases) about the ontology's interpretation (for example rule consistency aspects). For the sake of reusability we believe that such issues should not be ignored —as they can not be avoided— but instead be *managed*.

In Fig 2 and Table 1 below we represent the Trivial Scientific Conference ontology base both as link types in an ORM-style diagram and as lexons in a "database" format. Next, we define the ontological commitments, see table 2. The representation of the rules in the commitment layer is not restricted to a particular ontology language or standard, but we adopt a notational convention to specify which rules system/standard is used, in the form of a prefix of the rule. For example, the prefix "ORM." is used in

² If the reader is not familiar with reading ORM schemas, he can find its representation in Table1 and Table2.

Table 2 for rules which are intended to be interpreted as "standard" ORM ([H01]) by "standard ORM" tools. Furthermore, each ontological commitment should define which lexons are used and constrained in that particular commitment. E.g., for simplicity we allow the use of rules number e.g. 1, 5, and 12 to show that the symbolic representation of those lexons will be constrained and will be visible as they are defined in the ontology base.

For methodological reasons of organization and management that maximize the knowledge reuse of these commitments, new applications must be able to easily commit to (selected contexts of) the ontology. We therefore group the rules into commitments, as illustrated in Table 2. Notice that any rule can be used within more than one commitment, but for simplicity we have not exploited this in this particular example.

Ontology base (Lexons)				
LNo	Context	Term1	Role	Term2
1	Conference	Person	IsMemberOf	Committee
2	Conference	Committee	Includes	Person
3	Conference	Person	Chairs	Committee
4	Conference	Committee	ChairedBy	Person
9	Conference	Reviewer	SubtypesOf	Person
10	Conference	Person	Types	Reviewer
11	Conference	Author	SubtypesOf	Person
12	Conference	Person	Types	Author
13	Conference	Reviewer	Reviews	Paper
14	Conference	Paper	ReviewedBy	Reviewer
15	Conference	Author	Writes	Paper
16	Conference	Paper	WrittenBy	Author
17	Conference	Author	Presents	Paper
18	Conference	Paper	PresentedBy	Author
19	Conference	Paper	Has	PaperTitle
20	Conference	PaperTitle	lsOf	Paper
21	Conference	Paper	Has	PaperNumber
22	Conference	PaperNumber	lsOf	Paper

Table 1: The TSC Ontology Base

Notice that we present the ORM rules in Table 2 by verbalizing them into fixedsyntax English sentences (i.e. generated from agreed templates parameterized over the ontology base content). We believe that this should allow non-experts to (help to) check, validate or build the commitment rules and will simplify the commitment modeling process. For ORM, verbalizations may eventually be replaced by RIDL Constraint Language expressions ([VB82], [DMV88]) or expressed in another formalism, and in such case we may compile them (RIDL-A, [DMV88]).

Fig. 3 shows that the application "Conference A" uses two commitments (V1, V2), while application "Conference B" uses commitments (V1, V3). This implies that each of the commitments (V1, V2) and (V1, V3) must be consistent, as will be discussed in section 4.

Incidentally, in preliminary tests involving users in a digital archive application, we found users are comfortable with the analogy with database systems, and this speeded up the commitment building and user scenario specifications. This effect is expected to encourage organizations to adopt ontologies; for instance, companies that have many databases and/or DTDs can build an ontology base for their business knowledge to enable or improve interoperability, or to support consistency and validation checks.

RuleID	Rule Definition	CID
1	DOGMA. Visible Lexons to this commitment are {\$\$L21 \$\$L22}	V2
2	ORM.Mandatory(Each Paper Has at least one PaperNumber)	V2
3	ORM.InternalUniqueness(Each Paper Has at most one PaperNumber)	V2
4	ORM.InternalUniqueness(Each PaperNumber IsOf at most one Paper)	V2
5	DOGMA. Visible Lexons to this commitment are { <u>\$\$L17</u> <u>\$\$L20</u> }	V3
6	ORM.Mandatory(Each Paper Has at least one PaperTitle)	V3
7	ORM.InternalUniqueness(Each Paper Has at most one PaperTitle)	V3
8	ORM.InternalUniqueness(Each PaperTitle IsOf at most one PaperTitle)	V3
9	ORM.ExternalUniqueness(Each (Author, PaperTitle) as a combination refers to at most one Paper)	V3
10	ORM.InternalUniqueness(It is disallowed that the same <u>Author</u> <u>Presents</u> the same <u>paper</u> more than once, and it is disallowed that the same <u>Paper</u> <u>PresentedBy</u> the same <u>Author</u> more than once)	V3
11	ORM.SubSet(Each <u>Author</u> who <u>Presents</u> a <u>Paper</u> must also <u>Writes</u> that <u>Paper</u>)	V3
12	DOGMA. Visible Lexons to this commitment are { <u>\$\$L1</u> <u>\$\$L16</u> }	V1
16	ORM.InternalUniqueness(Each Person Chairs at most one Committee)	V1
17	ORM.Mandatory(Each Committee Includes at least one Person)	V1
18	ORM.InternalUniqueness(Each Committee Includes at most one Person)	V1
19	ORM.InternalUniqueness(Each Committee ChairedBy at most one Person)	V1
20	ORM.Mandatory(Each Committee ChairedBy at least one Person)	V1
21	ORM.Exclusion (Each <u>paper</u> which is <u>WrittenBy</u> a <u>Person</u> must not <u>ReviewedBy</u> with that <u>Person</u>)	V1
22	ORM.SubSet(Each Person who chairs a Committee must also IsMemberOf that Committee)	V1
24	ORM.Mandatory(Each <u>Reviewer Reviews</u> at least one <u>Paper</u>)	V1
25	ORM.InternalUniqueness(It is disallowed that the same <u>Reviewer</u> <u>Reviews</u> the same <u>paper</u> more than once, and it is disallowed that the same <u>Paper</u> <u>ReviewedBy</u> the same <u>Reviewer</u> more than once)	V1

26	ORM.Mandatory(Each <u>Author Writes</u> at least one <u>Paper</u>)	V1
27	ORM.Mandatory(Each Paper WrittenBy at least one Author)	V1
28	ORM.InternalUniqueness(It is disallowed that the same <u>Author</u> <u>Writes</u> the same <u>paper</u> more than once, and it is disallowed that the same <u>Paper</u> <u>WrittenBy</u> the same <u>Author</u> more than once)	V1

Ontology base Commitment Layer
V1
V3
Applications

Table 2: The Commitment layer

Fig. 3: Organization of the interpretation layer

3.2 Example 2: Mining ontology base and commitments from existing resources

As indicated earlier, ontologies are very expensive resources to engineer (even taking into account that a "good" ontology should be a one-time effort...). Also ontologizing existing information systems is time-consuming and is a difficult task. In this example, we briefly demonstrate the possibilities and the simplicity of our approach for learning/mining ontological data from four commonly existing resources (it might even be seen as a simplification and encouragement for ontologizing such information models). In Fig. 4a we represent taxonomic knowledge, in Fig. 4b SQL syntax, in Fig. 4c XML DTD syntax and in Fig. 4d. an Extended Entity Relationship EER schema. Table 3 and Table 4 show respectively the derived ontology base and the commitment layer. This process might be carried out semi/automatically, and to guarantee the quality of the mined ontological data, we may need to align the mined terms with an existing ontology base (a company lexicon, a business area thesaurus, WordNet...), provided it is possible to uniquely determine a concept within a given context.



Fig. 4: Deriving/Mining ontology-base and interpretations from four common
existing resources

Lexons				
LexonNo	Context	Term1	Role	Term2
1	С	DataBase	IsSubsubjectOf	Computer Science
2	С	AI	IsSubsubjectOf	Computer Science
3	С	Internet	IsSubsubjectOf	Computer Science
4	С	Book	IsSubTypeOf	Publication
5	С	Book	Attributes	BookID
6	С	Book	Attributes	Title
7	С	Book	Attributes	Author
8	С	Book	Attributes	Price
9	С	Book	Attributes	ISBN

Table 3: Ontology base for Example 2

RuleID	Rule	Commitment ID
1	EER.#(Each Book Has identity by BookID)	C1
2	EER.*(Each Book Has at least one Title)	C1
3	DTD. (Each Book must Has one and only one Title)	C1
4	DTD.+(Each Book Has at least one Author)	C1
5	DTD.?(Each <u>Book</u> may <u>Has</u> one P <u>rice</u>)	C1

6	SQL. Primary Key(Each Book Has ISBN as a primary identity)	C1
7	SQL.NotNull(Each Book Has at least one Title)	C1

Table 4: Commitment layer

4. Establishing Ontological Consistency

What is consistent for one application may be inconsistent for another, this depends on the interpretation of reality, but of course applications that do not share a common consistent commitment cannot communicate or interoperate with each other. By definition, the ontology base as a "substitute for a plausible real world" must *always* be assumed to be consistent, although multiple *seemingly* incompatible alternatives may simultaneously coexist in it (but not within the same context, though). As indicated earlier, we presently assume that ontological commitments behave like logical theories for which at least one context provides a model. It is quite literally "a matter of interpretation" which model an application commits to. It is indeed the responsibility of this application's interpretation, *not* that of the ontology base, to maintain its own internal consistency. Note however that by working in this way we tend to maximize the independence between the ontology and the applications, which consequently increases the reusability of the knowledge involved. Applications can safely interoperate among each other and exchange data and transactions where they share "the same" ontological commitments [UG96]. For example, the two Trivial Scientific Conference applications A and B in Example 1 can interoperate over the commitment V1, the intersection of (V1, V2) and (V1, V3).

A note on ontology versioning. Ontologies are not static; at least while they are being engineered they grow (and are modified) over time or domain. Therefore versioning mechanisms normally adopted to deal with changes may cause consistency problems for the applications that commit to the ontology, as noted already in [KF01]. Adopting our approach, the need for an ontology versioning mechanism is simplified: (a) lexons can be added to the ontology base without any effect to the ontological commitments; and (b) lexons cannot be deleted or modified if they are in use (see rules 1, 5 and 12 in Table2). Adding or modifying rules in the ontological commitments also becomes easier to manage for a versioning mechanism, as the number of applications committing to a given ontological commitment in general is less than those committing to the whole ontology, therefore reducing the impact of changes to be controlled.

In the DOGMA architecture (see the note on semantics in Section 2.2) each ontological commitment necessarily must be a consistent theory, as it is a possible interpretation of a domain, i.e. forms a set of rules that constrain, interpret, or rather commit to a particular aspect of reality as specified in a conceptualization. On the other hand, it is allowed in our approach that an application can commit to more than one commitment, therefore we must require that a set of ontological commitments that are used by one application must be consistent with each other. Obviously, the meaning in such case is that all commitments together form one *complete* interpretation [G95] for such applications.

The complexity of establishing consistency strongly depends on the language that is used to explicitly express the commitments. Adopting a given well-defined set of rule types helps analyzing the consistency and evaluating the ontology. To give two examples, a formal toolkit for ontological analysis is introduced in [GW00] to help check the ontological consistency of taxonomies, and in [DMV88] RIDL-A was defined as consistency analyzer for the well-circumscribed NIAM/ORM rules system [H01], easily mapable to a subset of first order logic.

Nothing in the definition prevents different ontological commitments even on the same ontology base to be expressed in a mix of languages (e.g. in different rule systems). Of course this implies that a consistency analyzer must be able to map between them.

5. Implementation and tools: the DOGMAModeler for ontology engineering.

This section outlines briefly the tools and projects that are implemented and based on the approach described in this paper.

The kernel of the system is formed by the *DOGMA Server* which stores and serves the ontology base and the commitment layer. The most recent active version of the implementation design for both commitment layer and ontology base may be downloaded³. The main components in the prototype implementation design are the storage module and the API. Storage is in a vanilla database system, currently Microsoft SQL Server that just implements efficient serving of the ontology base and interpretations. The API (JAVA JDK 1.3) provides a unified access to the basic functionality of the ontology server, and is designed to be accessible from any high level programming language.

DOGMAModeler is a suite of ontology engineering tools, including ontology browser, editor, manager, and some simple annotators, mining, and aligning tools. It supports functionality for modeling both ontology base and commitments. It supports derivative of ORM as graphical notation, and its cross-bonding ORM-ML [DJM02] that is easy to exchange, as well as the verbalizations of ontological commitments into pseudo natural language⁴.

Some of the principles underlying the DOGMA approach are and were illustrated (not to say refined or even developed as desirable side effects) in a number of projects such as HyperMuseum (EU Telematics-3088), where simplified ontologies in a digital-library-type query application were deployed, using an earlier version of the

³ http://www.starlab.vub.ac.be/Research/dogma/OntologyServer.htm

⁴ http://www.starlab.vub.ac.be/Research/dogma/DogmaModeler/

DOGMA ontology server to develop WordNet-based ontological support [SMD01]. In NAMIC (IST-1999-12392) it is intended to assist news agencies and journalists in authoring news items. The DOGMA ontology base model is used for storage of the ontology, which is then provided as a service to a query module. A commitment layer built on top of this ontology base as a JAVA API provides support for NAMIC-specific features such as profiles [DJBM02]. These profiles are in fact defined as query specifications on the ontology; for instance, the user profile of sports journalists would be based around a commitment that contains sports-related lexons in the ontology. Annotation of the incoming news stream could then be used to match the news content with the different users' preferences or views⁵.

OntoWeb is an EU thematic network (IST-2000-29243) for the support of semantic web and related research. A DOGMA-based ontology (among others) and its ontology-based query system are being developed as part of the server infrastructure underlying the semantically annotated web portal and websites of the network⁶. In *Onto-Basis*, a Flemish government funded long-term project, we explore the development and use of "practical" ontologies stored in the DOGMA Server for the knowledge management and advanced applications in a variety of business environments, as part of the future semantic Web⁷.

6. Conclusion and future work

In this paper we have presented an commitment layer to mediate between ontology base and applications, that separates the deeper semantics from the ontology base which is intended to be a computer-rendering of sets of simple, easy to agree on facts about possible "domains", to be accessed though an application's language. We have tried to analyse the dependency between the applications and the ontology, inspired by related research in database semantics, and next have identified and illustrated this commitment layer and discussed the benefits that could be achieved. The DOGMA project aims at implementing a proof of concept for this approach, in order to simplify building, deployment and (re)use of ontologies for semantics in a multi- domain environment.

In spite of the size and number of applications that will use an ontology, and the very long expected life-cycle of an ontology, almost none of the ontological research as yet fundamentally addresses the issues of scalability and design methodology. In a new long-term research project OntoBasis, some of these important issues will be explicitly studied and investigated by using the described approach in a number of concrete large-scale business applications from different domains.

⁵ www.hltcentral.org/projects/namic

⁶ http://www.ontoweb.org

⁷ http://www.starlab.vub.ac.be/research/ontobasis

Acknowledgements. The authors are grateful to the other members of the STAR-Lab team, especially Ben Majer, Peter Spyns and Jan Demey, for stimulating discussions and criticism. The ideas and presentation have also very much benefited from feedback at earlier seminars on the topic given by the first author at CMU (Pittsburgh) and EPFL (Lausanne), and by the second author at a stimulating Dagstuhl seminar on Rule-ML. Both authors gratefully acknowledge the beneficial environment for this type of research created by the OntoWeb Thematic Network!

Partial support for the reported work from t EC FP5 IST project NAMIC (IST-1999-12392) and the EC FP5 Thematic Network OntoWeb (IST-2000-29243) is hereby also gratefully acknowledged.

References

- [AB01] Alagic, S. and Bernstein, P.A.: "A Model Theory for Generic Schema Management". In Proceedings of DBPL'01, G. Ghelli and G. Grahne (eds), Springer-Verlag, (2001).
- [B99] Berners-Lee, T. et al.: Weaving the Web : The Original Design and Ultimate Destiny of the World Wide Web by its Inventor. Harper, San Francisco, (1999)
- [BHP+99] Bechhofer S., Horrocks I., Patel-Schneider P.F., Tessaris S., "A Proposal for a Description Logic Interface", in: Proceedings of the International Workshop on Description Logics, P. Lambrix et al (eds.), pp. 33-36, (1999).
- [BCM99] Bench-Capon T.J.M., Malcolm G., "Formalising Ontologies and Their Relations". In Proceedings of DEXA'99: 250-259, (1999).
- [CJ93] Chandrasekaran B., and Johnson T.R., "Generic Tasks and Task Structures: History, Critique and New Directions," in Second Generation Expert Systems, David J.M., Krivine J.P., and Simmons R. (Eds.), Springer-Verlag, London, pp. 233-272. (1993).
- [C92] Clancey W.J., "Model construction operators". Artificial Intelligence, 53(1):1-115, (1992).
- [DJBM02] Jan De Bo, Mustafa Jarrar, Ben Majer, Robert Meersman: "Ontology-based author profiling of documents", In Towards Improved Evaluation Measures for Parsing Systems Workshop, Third International Conference on Language Resources and valuation, LREC, (2002).
- [DJM02] Jan Demey, Mustafa Jarrar, Robert Meersman: "Markup Language for ORM Business Rules", In International Workshop on Rule Markup Languages for Business Rules on the Semantic Web, First International Semantic Web Conference, ISWC, (2002).
- [DMV88] De Troyer, O.M.F., Meersman, R. and Verlinden, P., "RIDL* on the CRIS Case: A Workbench for NIAM". In Computerized Assistance during the Information Systems Life Cycle, T.W. Olle, A.A. Verrijn-Stuart, and L. Bhabuta (eds.), pp 375--459, North-Holland/IFIP Amsterdam, (1988).
- [F01] Fensel, D. Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce, Springer Verlag, (2001).
- [FHvH00] Fensel D., Horrocks I., van Harmelen F. et al.: "OIL in a Nutshell". In "Proceedings of EKAW 2000", Springer-Verlag, (2000).
- [G95] Gruber T.R., "Toward principles for the design of ontologies used for knowledge sharing", International Journal of Human-Computer Studies, 43(5/6), (1995).
- [GG95] Guarino, N. and Giaretta, P., "Ontologies and Knowledge Bases: Towards a Terminological Clarification" in: Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing, N. Mars (ed.), pp 25-32, IOS Press, Amsterdam (1995).
- [GPB99] Gomez-Perez, A., & Benjamins, R., "Overview of Knowledge Sharing and Reuse Components: Ontologies and Problem-Solving Methods", in: Proceedings of the IJCAI-99 Workshop on Ontologies and Problem-Solving Methods (KRR5), Morgan-Kaufmann (1999).
- [GPS98] Gangemi A., Pisanelli D., and Steve G., "Ontology Integration: Experiences with Medical Terminologies", in: Formal Ontology in Information Systems, N. Guarino (ed.), IOS Press (1998).

- [GN87] Genesereth, M.R. and Nilsson, N.J., Logical Foundation of Artificial Intelligence. Morgan Kaufmann, Los Altos, California (1987).
- [GW00] Guarino, N. and Welty, C.: "Identity, Unity, and Individuality: Towards a Formal Toolkit for Ontological Analysis", in: Proceedings of ECAI-2000, H. Werner (ed.), IOS Press, Berlin, pp 219-223, (2000).
- [H01] Halpin, T. "Information Modeling and Relational Databases", 3rd ed., Morgan-Kaufmann, (2001).
- [H85] Hayes, P., "The Second Naive Physics Manifesto," in Formal Theories of the Common Sense World (Vol. 1), J.R. Hobbs and R.C. Moore (eds.), Ablex Publishing Company, Norwood, NJ (1985).
- [IFFJ97] Iwasaki I., Farquhar A., Fikes, R., and Rice J., "A Web-Based Compositional Modeling System for Sharing of Physical Knowledge", in: Proceedings of IJCAI'97 Conference, pp. 494-500. Morgan-Kaufmann (1997)
- [KF01] Klein, M. and Fensel, D. "Ontology Versioning on the Semantic Web", in: First International Semantic Web Working Symposium (SWWS-1) (2001).
- [M94] Meersman, R.A., "Some Methodology and Representation Problems for the Semantics of Prosaic Application Domains". In Methodologies for Intelligent Systems (ISMIS-94), Z.W. Ras, M. Zemankova (eds.), Springer LNAI, Heidelberg, (1994).
- [M99a] Meersman, R. "Semantic Ontology Tools in Information Systems Design", in: Proceedings of the ISMIS'99 Conference, Z. Ras and M. Zemankova (eds.), LNCS, Springer Verlag, (1999).
- [M99b] Meersman, R. "The use of lexicons and other computer-linguistic tools in semantics, design and cooperation of database systems", in: Proceedings of the 2nd Conference on Database Applications and Semantics (CODAS'99)", Marek Rusinkiewicz, Y. Zhang (eds.), Springer Verlag LNAI (1999).
- [M01] Meersman, R., "Ontologies and Databases: More than a Fleeting Resemblance", in: "Proceedings of the 2001 OES-SEO Workshop, Rome", A. D'Atri, A. Sølvberg, L. Willcocks (eds), LUISS Edizioni, . ISBN 88-88047-21-1. (2001).
- [MFGB99]Motta, E., Fensel, D., Gaspari, M. and Benjamins, R. Specifications of Knowledge Components for Reuse, In Proceedings of the 11th International Conference on Software Engineering and Knowledge Engineering, Kaiserslautern, Germany, KSI Press, pp. 36-43 (1999).
- [MLP79] De Millo, R.A., Lipton, R.J., and Perlis, A.J., "Social Processes and Proofs of Theorems", CACM Vol. 22(5), ACM Publications, (1979).
- [NCM+00]Da Nobrega G.M., Castro E., Malbos P., Sallantin J., Cerri S.A., "A framework for supervised conceptualizing", In ECAI'00: pp. 17.1-17.4. (2000).
- [Ontoweb] http://www.ontoweb.org
- [PFP+92] Patil R.S., Fikes R.E., Patel-Schneider P.F., McKay D., Finin T., Gruber T., Neches R., "The DARPA Knowledge Sharing Effort: Progress Report", in: Proc. of Knowledge Representation and Reasoning, 777-788, (1992).
- [PS98] Parent C., Spaccapietra, S.: "Database Integration: an Overview of Issues and Approaches", CACM, vol. 41, no 5, pp. 166-178, May (1998).
- [R00] Richards. D. "The Reuse of Knowledge: A User-Centered Approach", International Journal of Human Computer Studies, (2000).
- [R88] Reiter, R. "Towards a Logical Reconstruction of Relational Database Theory", in: "Readings in AI and Databases", J. Mylopoulos and M.L. Brodie (eds.), Morgan Kaufman, (1988).
- [S00a] Sowa, J.F., "Ontology, metadata, and semiotics" in: Ganter, B, Mineau, G.W (eds), Conceptual structures: logical, linguistic and computational issues: 8th international conference on conceptual structures, ICCS 2000, Darmstadt, Germany, August 2000 (Lecture Notes in Artificial Intelligence, 1867), Berlin: Springer-Verlag, pp55-81 (2000).
- [S93] Steels, L., "The componential framework and its role in reusability", in: Second Generation Expert Systems, J.-M. David, J.-P. Krivine & R.Simmons, (eds.), pp. 273-298. Berlin: Springer-Verlag (1993).
- [Sa98] Santucci, G., "Semantic Schema Refinements for Multilevel Schema Integration", Data and Knowledge Engineering 25 (1998), pp. 301-326, North-Holland.
- [SGP98] Steve G., Gangemi A., Pisanelli D.M., "Integrating Medical Terminologies with the ONIONS Methodology", in: Information Modeling and Knowledge Bases VIII, Kangassalo H., Charrel J.P. (eds.), IOS Press Amsterdam (1998).

- [SM93] Swartout, W.R. and Moore, J.D., "Explanation in Second Generation Expert Systems", in: Second Generation Expert Systems, J.-M. David, J.-P. Krivine and R. Simmons (eds),. Berlin: Springer-Verlag (1993).
- [SMD01] Stuer, P., Meersman, R., and De Bruyne, S. "The HyperMuseum Theme Generator System: Ontology-based Internet support for the active use of digital museum data for teaching and presentation". In D. Bearman & J. Trant (eds.) Museums and the Web 2001: Selected Papers., Archives & Museum Informatics, (2001).
- [UG96] Uschold, M. and Gruninger, M, "Ontologies: principles, methods and applications", *Knowledge Engineering Review*, vol. 11, no. 2 (1996).
- [VB82] Verheyen, G. and van Bekkum, P.: NIAM, aN Information Analysis Method. In: IFIP Conference on Comparative Review of Information Systems Methodologies, T.W. Olle, H. Sol, and A. Verrijn-Stuart (eds.), North-Holland (1982).
- [ZSC01] Yannis Tzitzikas, Nicolas Spyratos and Panos Constantopoulos, "Mediators over Ontologybased Information Sources", in: Second International Conference on Web Information Systems Engineering (WISE 2001), Kyoto, Japan (2001).
- [WN] http://www.cogsci.princeton.edu/~wn/