[JM02] Jarrar M. and Meersman R.: 'Formal Ontology Engineering in the DOGMA Approach'. In Proceedings of the International Conference on Ontologies, Databases and Applications of Semantics (ODBase 02). Springer Verlag, LNCS 2519, pp. 1238 - 1254. (2002).

Formal Ontology Engineering in the DOGMA Approach.

Mustafa Jarrar, Robert Meersman

VUB STARLab
Vrije Universiteit Brussel
Pleinlaan 2 – B-1050 Brussels, Belgium
{ mjarrar, meersman}@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. We address several related key issues, such as knowledge reusability and shareability, scalability of the ontology engineering process and methodology, efficient and effective ontology storage and management, and coexistence of heterogeneous rule systems that surround an ontology mediating between it and application agents. Ontologies should represent a domain's semantics *independently* from "language", while any process that *creates* elements of such an ontology must be entirely rooted in some (natural) language, and any *use* of it will necessarily be through a (in general an agent's computer) language.
To achieve the claims stated, we explicitly decompose ontological resources into *ontology bases* in the form of simple binary facts called *lexons* and into so-called *ontological commitments* in the form of description rules and constraints. Ontology bases in a logic sense, become "representationless" mathematical objects which constitute the range of a classical *interpretation mapping* from a first order language, assumed to lexically represent the commitment or binding of an application or task to such an ontology base. Implementations of ontologies become database-like on-line resources in the model-theoretic sense. The resulting architecture allows to materialize the (crucial) notion of commitment as a separate layer of (software agent) services, mediating between the ontology base and those application instances that *commit* to the ontology. We claim it also leads to methodological approaches that naturally extend key aspects of database modeling theory and practice. We discuss examples of the prototype 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 support formal and agreed so-called *ontological commitments* (for definitions, see below) needed for new open application 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. For an example, see fig. 2. A correct understanding of ontologies must however reconcile that they are repositories of (in principle) language- and task-independent knowledge, while any effective use by e.g. software agents naturally requires interaction with *some necessarily lexical* representation.. Also the creation of ontologies as (sets of) agreements about structure and semantics of a domain requires the use of —usually natural— language, leading to interesting research issues on methodology.

Information systems (in any broad sense, especially web-based ones) are expected to benefit substantially from the use of ontologies as externalized resources of agreed knowledge. To a database engineer the following parallel may perhaps be enlightening: implementations of ontologies will in a real sense permit a form of "*semantics independence*" for such 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 makes it important to understand that ontology engineering, while similar to data modeling, is substantially more than that, even when the data modeling methodology takes business rules into account [6]. Representing formal semantics in the domain of "air travel" is more than designing, or collecting, a set of data models for a number of airline reservation systems. Existing data models likely would have been autonomously specified for optimal use within an individual organization or company. Thus, an ontology needs to be even more *generic*, across tasks and even task types, than a data model is for a number of given applications. Just adding a mere "*is_a"*- taxonomy of terms is not sufficient, as the literature sometimes seems to suggest. An ontology needs to include (the meaning of) a much richer set of relationships, such as *instance_of*, *part_of*, …, which depending on the domain all might deserve a "generic semantics".

Ontologies must be Scalable Resources. As the main purpose of an ontology is to be a shared and agreed semantic resource over a wide range of agents, building scalable ontologies will effectively be a group effort, with ontologies growing over time [19]. In particular, they will need a form of consensus about the conceptualizations to be adopted. In [12] such a consensus is the result of a mental process, assisted by exemplifying, testifying, investigating etc, while [24] proposes a so-called Adequacy Search. Any such process will inevitably be oriented to *tasks* to be carried out, and are likely to be influenced also by personal taste and even may reflect fundamental disagreements [2]. Several conceptualizations could be adopted for the same domain [15], especially in large-scale and multi-domain ontologies, which may lead to potentially "locally" inconsistent (and incomplete) ontologies. Notice that difficulties and disagreements in the conceptualization process normally appear at a “deeper” level of abstraction, i.e. as a result of conceptual heterogeneity and difficulties in ontology integration [14]. (This level is dubbed the “Detail Level” by [31].) Rules constrain the structure and interrelationships of the concepts. More specifically other words, *constraints, rules and procedures* are essential to achieve an understanding about a domain’s semantics, but agreement about them in general is difficult and nearly always specific to a context of application. Note that from an ontology’s *application point of view* constraints will likely be 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 disagreement might 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 value is a mandatory property for the book to have, or that “person has age”, but disagree on the value range. In general database design methodology has shown that people agree fairly easily about the basic facts in a domain than about the "lower level" details of and constraints on these facts.

Knowledge reusability is another important goal of building ontologies ([18] [34] [23] [13] [11]). 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. Ontologies may only need to be reused partially: for example, when building a “Manufacturing” ontology, one may wish to reuse the “Customers” aspects from an existing “Shopping” ontology, if they are assumed to share a same conceptualization about a certain set of axioms. The ability to share a partial conceptualization (as a result of partial agreement) across two ontologies depends on the degree of abstraction that can be applied by ontology engineers to their respective concepts. To improve knowledge reusability, several researchers from the problem-solving area (e.g. Chandrasekaran and Johnson [3], Clancey [4], or Swartout and Moore [32]) have proposed the idea of structuring the knowledge into different levels of abstractions, where Steels in [30] 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 [27] such as the importance of context, the need for more knowledge, etc.

It seems plausible that building large knowledge bases will only be possible if efforts are combined (Neches et al in [26]). This translates into a requirement for a unified framework that enables and maximizes knowledge reusability. Such a framework must be scalable and allow connecting of ontological theories in spite of the diversity of ontology languages and their representation models.

The above aspects and considerations translate within DOGMA into a model and associated 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.

Methodology by Transition and Growth. *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 ([9]). 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 ontologies, and introduce and discuss these in our "DOGMA" framework. By examples, Section 3 illustrates this framework for building, (re)using, 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

According to Gruber [11] an ontology is "an explicit specification of a conceptualization", referring to an *extensional* ("Tarski-like") notion of a conceptualization as found e.g. in [15]. Guarino and Giaretta [12] pointed out that this definition *per se* does not adequately fit the purposes of an ontology. They argue correctly that a conceptualization benefits from invariance under changes that occur at the instance level by transitions between merely different "states of affairs" in a domain, and thus should not be extensional. Instead, they propose a conceptualization as an *intensional* semantic structure i.e. abstracting from the instance level, which encodes implicit rules constraining the structure of a piece of reality. In other words an ontology becomes a logical theory which possesses a conceptualization as an explicit, partial model.

While we arrived at it independently from a database-inspired perspective [21], in the DOGMA framework we embrace this viewpoint but unlike [12] 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 limited 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 will permit us to make hopefully explicit most of the key issues in ontology organization, engineering, scalability and methodology listed above, starting from familiar database design principles.

2.1 Model Theoretic Database Inspiration for Ontologies: the Ontology Base

By adopting agreement as pragmatic basis for the formal semantics of information systems (see [20] for an early position on this) we claim that classical, i.e. model-theoretic database technology and methodologies become suitable for "reuse" in an ontology context, and therefore perhaps is an interesting new research subject in its own right.

Suppose we want to build a system to support the running of scientific conferences such as ODBASE'02, but in such a way that its domain knowledge (its *ontology* of course) is *a priori* maximally accessible, reusable, and "understood" by —as yet— unidentified software agents. The openness of this environment prohibits us from prescribing a single definitive set of concepts, but instead we need to provide for an extensible set of alternative plausible worlds from which agents can "choose" and to which they can "commit". In DOGMA we will split these knowledge components into a set of *lexons*, grouped into abstract contexts, and into a layer of commitments. For the Scientific Conferences Domain, some lexons could be

```
(Organization-ContextID)
    Person    IsMemberOf Committee
    Person    Chairs     Committee
    Committee ChairedBy  Person
    Reviewer  SubtypesOf Person
    Author    SubtypesOf Person
    Reviewer  Reviews    Paper
    Paper     ReviewedBy Reviewer
    Paper     WrittenBy  Author
```

```
    Author    Presents   Paper
    Paper     Has   PaperTitle
    Paper     Has   PaperNumber
  {…}
(ResearchAreas-ContextID)
    Representation_and_Storage SuperAreaOf Ontology_Languages
    Representation_and_Storage SuperAreaOf Semi-Structured_Data
    Applications_and_Evaluation SuperAreaOf Semantic_Web
    Applications_and_Evaluation SuperAreaOf Media_Archives
   {…}
```

As an example of one *commitment* (-fragment), for instance by an application that wishes to access, or register submitted papers, consider

```
(ConferenceAdmin Commitment)
  <Each Committee ChairedBy at most one Person>
  <Each Person who chairs a Committee must also IsMemberOf that Committee>
  <Each Reviewer Reviews at least one Paper>
  <Each Paper which is WrittenBy a Person must not ReviewedBy with that Person>
  {…}      (Rules are verbalized in a suitable pseudo-NL syntax)
```

Commitment[1] implies the choice of, and/or adherence to, a set of rules, constraints, derivations that will in general depend on the task to be performed: rules that hold in one commitment need not do so in another, but will nevertheless need to be formally *interpreted* (in a first-order logic sense) in terms of the lexons in the same or "related" contexts.

It could be noted at this point that most recent ontology research, and the resulting formalisms and languages [25] are based on versions of earlier *description logics* [1] [10] and in general correspond more closely with the proof-theoretic view of database [28] with its 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 [28] ff., it is undeniably so that the model-theoretic view of databases gave rise to a technology, scalable *par excellence* and a successful industry of high-performance DBMS, tools and applications. By bringing 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, early 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*, 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

[1] We will return to this example in more detail in Section 3.

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.

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 [29]). 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 mapping, the formalism for specifying a conceptualization, as an ontology(-base), should be as simple as possible, e.g. just objects and relationships in the mathematical sense as intended by Tarski [15]. Thus our ontology base is a set of (binary, even) conceptual relationships, while other domain knowledge and its formal semantics will be "approximately" specified in a *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 (defined in DOGMA in terms of *ontological contexts*, see below, and we in fact shall use the terms interchangeably) even about the same real world domain. In summary we have

Definition. An *ontology base* is

- A set of context-specific binary fact types, called *lexons*. Notation: $<\gamma$: $Term_1$, Role, $Term_2>$. Here $\gamma \in \Gamma$ is just an abstract *context identifier* chosen from a set, (more about this below). The lexical terms ($Term_1$, Role, $Term_2$) are constructed from a given alphabet;
- For each $\gamma \in \Gamma$, and each term T occurring in a lexon, the pair (γ, T) specifies exactly a *unique concept*.

Remarks. Lexons are thus assumed to express a binary conceptual relationship that is agreed to hold within a given context (among "all" the parties involved in the ontology, using some given metalanguage). Note only one of its two roles is used. The requirement of uniqueness for the specified concepts translates into a strong condition on the notion of contexts. Contexts may also be used to accommodate different alternative, "plausible" conceptualizations in one ontology base. See the Note on Contexts below.

A Note on Contexts. Contexts have been and are the subject of occasionally intense study notably in AI; examples are [22], specifying them as higher-order theories [29]. [27] reports 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.

In DOGMA contexts only provide internal organization of an ontology into contextual knowledge components, i.e. context identifiers are used, intuitively and informally, to "group" lexons that are "related" in an intended conceptualization of a domain. In the DOGMA lexon structure (for the purpose of this paper) therefore they appear merely as abstract identifiers. At this stage their only formal "semantics" or interpretation within DOGMA is defined as a *mapping* from Γ to a *collection of sources* (not further defined, but for example a corpus of documents) each assumed to contain an intended conceptualization together with its implicit assumptions. Turning again to intuition, lexons are assumed (by an outside cognitive agent such as a human understanding that document) to be "true within that context's source". In the Note on Formal Semantics below we shall return to this informal ":interpretation" of an ontology base as a set of "true facts".

Clearly there is a lot more than meets the eye here; in particular expressing relationships between concepts (as is needed when *aligning* or integrating ontologies) from different contexts cannot be done without e.g. a notion of *context calculus* in which to define the relationships (predicate formulas) that are permitted or assumed to hold between contexts. This notion is not explored further in this paper. Also, the extraction of lexons from a context's source is a research topic in its own right of course, involving NL parsing and understanding in the case of ontology mining from documents [8]. 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.

2.2 The Commitment Layer

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

The ontological commitments may be seen as a set of reusable knowledge components. Such components may interoperate since they share the same ontology base. In practice "similar" applications reuse or inherit commitments from each other, which should facilitate new applications to commit to and use the ontology. (Also, successful commitments in certain domains and applications likely will become "popular" (i.e. serve a more general purpose) and a *de facto* trusted resource in their own right for achieving interoperability, or just compatibility between applications.

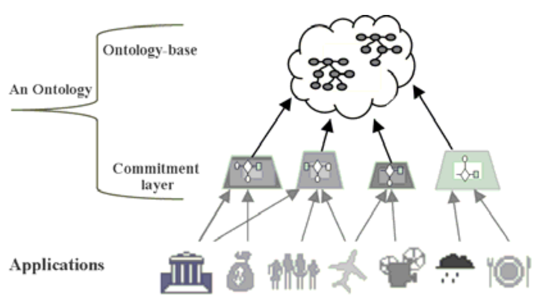

Fig. 1. Knowledge Organization in DOGMA Framework

A Note on Ontology as a Formal Semantics. An ontology base in DOGMA is the range of the (first-order) commitments (seen as interpretation mappings) of the application software agents, which for formal 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. It is fundamental to realize that this formalism implies that to the application agents, the ontology (i.e. the ontology base plus the agent's commitment to a part of it) ***is*** the real world, nothing more nor less. Lexons in a DOGMA ontology base are always "true", i.e. free of further "interpretation". Alternative truths, or partial ones as typically emerge during the engineering process have to be provided in separate conceptualizations or contexts (see the Note on Contexts above). 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. Incidentally 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 customized interpretations of an "is_instance_of" relationship in the relevant commitments… The ontological commitments above are merely part of the specification of this mapping, namely they specify the intensional interpretations of an application in terms of the ontology base.

Naturally there is a trade-off between complexity and size that lies in the requirements to (a) manage the (huge) size and (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 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 [17] and NIAM, aN Information Analysis Method [35], also “Nijssen’s-” or “Natural”-IAM). This indeed allows identifying and analyzing some of the essential differences between database- and ontology modeling. While we stated that formal ontologies are best thought of as abstract, mathematical entities, 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/NIAM, 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 studied and understood (it forms part of the ontological commitment) and is the subject of ongoing research.

3 Example : A simple Ontology in the DOGMA Framework

The following example, with its necessary simplicity, shows part of a Trivial Conference ontology, used by two different *kind of* conference applications. Fig. 2 shows the graphical representation of this ontology in an ORM diagram. Notice that the ontology in this example is not the aim of the paper itself, and is *supposed* to be specified at the knowledge level[1], i.e. it is more than a data model for the application instances. Applications that commit to this ontology may retain their internal data models[2].

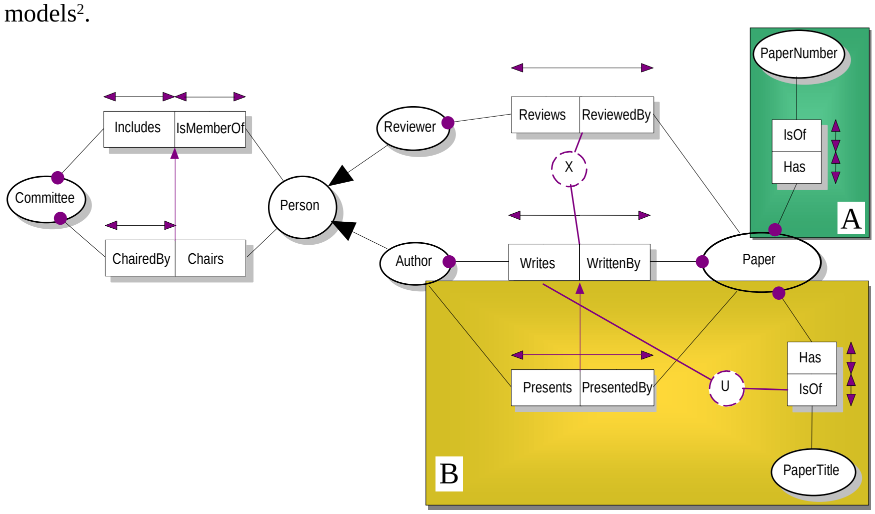

Fig. 2. Trivial Scientific Conference Ontology3

Each kind of conference application in general will have certain rules that do not necessarily agree with those of other kinds; 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 instead 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 hold.

[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 [11]

[2] 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.

[3] If the reader is not familiar with reading ORM schemas, he can find its representation in Table 1 and Table 2.

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 as Semantic Web: (1) the completeness of an ontology should be considered and managed, and (2) applications might not commit to an ontology because they do not agree about the ontology's interpretation. For the sake of reusability we believe that such issues should not be ignored —as they cannot be avoided— but instead be *managed*.

In Fig 2 and Table 1 below we represent the Scientific Conference ontology base both as link types in an ORM-style diagram and as lexons in a "database" format. Next, in Table 2 we define the ontological commitments. 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 rule system/standard is used, in the form of a rule prefix. For example, the prefix "ORM." is used in Table 2 for rules which are intended to be interpreted as "standard" ORM ([17]) by "standard ORM" tools. Furthermore, each ontological commitment should define an ontological *view*, i.e. state which lexons are used and constrained in that particular commitment. For simplicity we allow the use of rule numbers 1, 5, and 12 to show that the symbolic representation of those lexons is constrained and is visible as they are defined in the ontology base.

For methodological reasons of organization and management that ruses knowledge 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.

Table 1. The Ontology Base

Ontology Base (Lexons)				
LNo	**Context**	**Term1**	**Role**	**Term2**
1	Organization	Person	IsMemberOf	Committee
2	Organization	Committee	Includes	Person
3	Organization	Person	Chairs	Committee
4	Organization	Committee	ChairedBy	Person
9	Organization	Reviewer	SubtypesOf	Person
10	Organization	Person	Types	Reviewer
11	Organization	Author	SubtypesOf	Person
12	Organization	Person	Types	Author
13	Organization	Reviewer	Reviews	Paper
14	Organization	Paper	ReviewedBy	Reviewer
15	Organization	Author	Writes	Paper
16	Organization	Paper	WrittenBy	Author

17	Organization	Author	Presents	Paper
18	Organization	Paper	PresentedBy	Author
19	Organization	Paper	Has	PaperTitle
20	Organization	PaperTitle	IsOf	Paper
21	Organization	Paper	Has	PaperNumber
22	Organization	PaperNumber	IsOf	Paper

Notice that we present the ORM rules in Table 2 by *verbalizing* them into fixed-syntax English sentences (i.e. generated from agreed templates parameterized over the ontology base content). We believe that this allows 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 ([35], [7]) or expressed in another formalism as ORM Markup Language [6].

Fig. 3 shows that the application "Conference A" using 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.

Table 2. The Commitment Layer

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 {$$L17 .. $$L20}	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 Author Presents the same paper more than once, and it is disallowed that the same Paper PresentedBy the same Author more than once)	V3
11	ORM.SubSet(Each Author who Presents a Paper must also Writes that Paper)	V3
12	DOGMA. Visible Lexons to this commitment are {$$L1 .. $$L16}	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 paper which is WrittenBy a Person must not ReviewedBy with that Person)	V1
22	ORM.SubSet(Each Person who chairs a Committee must also IsMemberOf that Committee)	V1
24	ORM.Mandatory(Each Reviewer Reviews at least one Paper)	V1
25	ORM.InternalUniqueness(It is disallowed that the same Reviewer Reviews the same paper more than once, and it is disallowed that the same Paper ReviewedBy the same Reviewer more than once)	V1
26	ORM.Mandatory(Each Author Writes at least one Paper)	V1

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

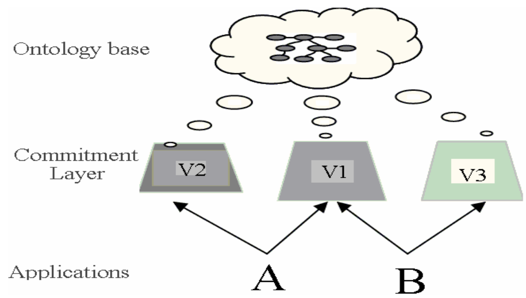

Fig. 3. Organization of the Interpretation Layer

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 in a meaningful way. 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). 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 [34]. For example, the two 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 [19]. 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. The meaning in such case is that all commitments together form one *complete* interpretation [11] 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, i.e. adopting a particular description logic, helps analyzing the consistency and evaluating the ontology. To give two examples, a formal toolkit for ontological analysis is introduced in [16] to help check the ontological consistency of taxonomies, and in [7] RIDL-A was defined as consistency analyzer for the well-circumscribed NIAM/ORM rules system [17], 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 briefly outlines 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 up the ontology base and the commitment layer. The most recent active version of the prototype implementation design for both commitment layer and ontology base may be downloaded[4]. 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.

[4] http://www.starlab.vub.ac.be/research/dogma/OntologyServer.htm

DOGMAModeler is a suite of ontology engineering tools, including ontology browser, editor, manager, and mining 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 [6] that is easy to exchange, as well as the verbalizations of ontological commitments into pseudo natural language[5].

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 DOGMA ontology server to develop WordNet-based ontological support [33]. 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 [5]. 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 views[6].

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 network[7]. In *OntoBasis*, 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 Web[8].

6. Conclusion

In this paper we have presented a architecture for ontologies that includes an ontology base and a commitment layer to mediate between the ontology base and applications. The ontology base 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 discussed the benefits that could be achieved. The DOGMA project aims at implementing a proof

[5] http://www.starlab.vub.ac.be/research/dogma/dogmamodeler/
[6] www.hltcentral.org/projects/namic
[7] http://www.ontoweb.org
[8] http://www.starlab.vub.ac.be/research/ontobasis

of concept for this approach, in order to simplify building, deployment and (re)use of ontologies for semantics in a multi- domain environment.

Acknowledgements. The authors are grateful to the other members of the STARLab team, for stimulating discussions and criticism. 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

1. Bechhofer S., Horrocks I., Patel-Schneider P.F., Tessaris S.: A Proposal for a Description Logic Interface. In: Lambrix, P.: Proceedings of the International Workshop on Description Logics, et al (eds.) (1999) 33-36
2. Bench-Capon T.J.M., Malcolm G.: Formalizing Ontologies and Their Relations. In: Proceedings of DEXA'99 (1999) 250-259
3. Chandrasekaran B., and Johnson T.R.: Generic Tasks and Task Structures: History, Critique and New Directions. In David J.M., Krivine J.P., and Simmons R. (Eds.): Second Generation Expert Systems, Springer-Verlag, London (1993) 233-272
4. Clancey W.J.: Model construction operators. *Artificial Intelligence* 53(1) (1992) 1-115
5. De Bo J., Jarrar M., Majer Ben., Meersman R.: Ontology-based author profiling of documents. In: Towards Improved Evaluation Measures for Parsing Systems Workshop, Third International Conference on Language Resources and valuation, (2002)
6. Demey, J., Jarrar, M. & Meersman, R.: A Conceptual Markup Language that supports interoperability between Business Rule modeling systems, Proceedings of the Tenth International Conference on Cooperative Information Systems (CoopIS 2002).
7. De Troyer, O.M.F., Meersman, R., Verlinden, P.: 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 Cycle, North-Holland/IFIP Amsterdam (1988) 375—459
8. Daelemans W., Buchholz S., Veenstra J.: Memory-based shallow parsing. In: Proceedings of CoNLL-99, Bergen (1999)
9. Fensel, D.: Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce, Springer Verlag (2001)
10. Fensel D., Horrocks I., van Harmelen F. et al.: OIL in a Nutshell. In: Proceedings of EKAW 2000, Springer-Verlag (2000)
11. Gruber T.R.: Toward principles for the design of ontologies used for knowledge sharing, International Journal of Human-Computer Studies, 43(5/6) (1995)
12. Guarino, N. and Giaretta, P.: 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 (1995) 25-32
13. 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)
14. Gangemi A., Pisanelli D., Steve G.: Ontology Integration: Experiences with Medical Terminologies. In: Guarino, N.: Formal Ontology in Information Systems, (ed.), IOS Press (1998)
15. Genesereth, M.R., Nilsson, N.J.: Logical Foundation of Artificial Intelligence. Morgan Kaufmann, Los Altos, California (1987)

16. Guarino, N., Welty, C.: Identity, Unity, and Individuality: Towards a Formal Toolkit for Ontological Analysis. In: Werner, H. (ed.): Proceedings of ECAI-2000, IOS Press, Berlin (2000) 219-223
17. Halpin, T.: Information Modelling and Relational Databases. 3rd edn. Morgan-Kaufmann (2001)
18. Iwasaki I., Farquhar A., Fikes, R., Rice, J.: A Web-Based Compositional Modeling System for Sharing of Physical Knowledge. In: Proceedings of IJCAI'97 Conference, Morgan-Kaufmann (1997) 494-500
19. Klein, M., Fensel, D.: Ontology Versioning on the Semantic Web. In: First International Semantic Web Working Symposium (SWWS-1) (2001)
20. Meersman, R.A.: Some Methodology and Representation Problems for the Semantics of Prosaic Application Domains. In: Ras, Z.W., Zemankova, M. (eds.): Methodologies for Intelligent Systems (ISMIS-94), Springer LNAI, Heidelberg (1994)
21. Meersman, R.: Semantic Ontology Tools in Information Systems Design. In: Ras, Z., Zemankova, M. (eds.): Proceedings of the ISMIS'99 Conference. LNCS, Springer Verlag, (1999)
22. McMacrthy, J. "Notes on Formalizing Context". In: Proceedings of IJCAI'93, Morgan-Kaufmann, (1993).
23. 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. KSI Press, Kaiserslautern Germany (1999) 36-43
24. Da Nobrega G.M., Castro E., Malbos P., Sallantin J., Cerri S.A.: A framework for supervised conceptualizing. In: ECAI'00. (2000) 17.1-17.4
25. http://www.ontoweb.org
26. 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 (1992) 777-788
27. Richards. D.: The Reuse of Knowledge: A User-Centered Approach, *International Journal of Human Computer Studies* (2000)
28. Reiter, R.: Towards a Logical Reconstruction of Relational Database Theory. In: Mylopoulos, J., Brodie, M.L. (eds.): Readings in AI and Databases. Morgan Kaufman (1988)
29. 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), Springer-Verlag Berlin (2000) 55-81
30. Steels, L.: The componential framework and its role in reusability. In: David, J.-M., Krivine, J.-P., Simmons,R. (eds.): Second Generation Expert Systems. Springer-Verlag Berlin (1993) 273-298
31. Steve G., Gangemi A., Pisanelli D.M.: Integrating Medical Terminologies with the ONIONS Methodology. In: Kangassalo H., Charrel J.P. (eds.): Information Modelling and Knowledge Bases VIII. IOS Press Amsterdam (1998)
32. Swartout, W.R., Moore, J.D.: Explanation in Second Generation Expert Systems. In:David, J.-M., Krivine, J.-P., Simmons, R. (eds): Second Generation Expert Systems. Springer-Verlag Berlin (1993)
33. Stuer, P., Meersman, R., 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 Bearman, D. & Trant, J. (eds.): Museums and the Web 2001. Selected Papers, Archives & Museum Informatics, pp.127-137, (2001)
34. Uschold, M. and Gruninger, M.: Ontologies: principles, methods and applications, The Knowledge Engineering Review, vol. 11, no. 2, June (1996).

35. Verheyen, G., van Bekkum, P.: NIAM, An Information Analysis Method. In: Olle, T.W., Sol, H., Verrijn-Stuart,A. (eds.): IFIP Conference on Comparative Review of Information Systems Methodologies. North-Holland (1982).