V R I J E   U N I V E R S I T E I T   B R U S S E L

F A C U L T E I T   W E T E N S C H A P P E N
V A K G R O E P   I N F O R M A T I C A   E N   T O E G E P A S T E   I N F O R M A T I C A
S Y S T E M S   T E C H N O L O G Y   A N D   A P P L I C A T I O N S   R E S E A R C H   L A B

STAR Lab Technical Report

Data Modelling versus Ontology Engineering

Peter Spyns, Robert Meersman & Mustafa Jarrar

affiliation	
keywords	ontology engineering, data modelling
number	STAR-2002-04
date	21/10/2002
corresponding author	Peter Spyns
status	published
reference	Sheth A. & Meersman R. (ed.), SIGMOD Record Special Issue on Semantic Web, Database Management and Information Systems 2000 31(4)

Pleinlaan 2, gebouw G-10, B-1050 Brussel
Phone: +32-2-629.3308 • Fax: +32-2-629.3525

Data modelling versus Ontology engineering

Peter Spyns Robert Meersman Mustafa Jarrar
Vrije Universiteit Brussel – STARLab
Pleinlaan 2, Building G-10
B-1050, Brussel, Belgium
+32-2-629. 3753 +32-2-629. 3308 +32-2-629. 3487
Peter.Spyns @vub.ac.be meersman @vub.ac.be mjarrar @vub.ac.be

ABSTRACT
Ontologies in current computer science parlance are computer based resources that represent agreed domain semantics. Unlike data models, the fundamental asset of ontologies is their relative independence of particular applications, i.e. an ontology consists of relatively generic knowledge that can be reused by different kinds of applications/tasks. The first part of this paper concerns some aspects that help to understand the differences and similarities between ontologies and data models. In the second part we present an ontology engineering framework that supports and favours the genericity of an ontology. We introduce the DOGMA ontology engineering approach that separates "atomic" conceptual relations from "predicative" domain rules. A DOGMA ontology consists of an *ontology base* that holds sets of intuitive context-specific conceptual relations and a layer of "relatively generic" *ontological commitments* that hold the domain rules. This constitutes what we shall call *the double articulation* of a DOGMA ontology [1].

Categories and Subject Descriptors
H.1.1 [Systems and Information Theory]: general systems theory

General Terms
Design, Reliability, Standardization, Theory.

Keywords
Ontology and knowledge engineering, data modelling

1 INTRODUCTION
Although there exist many definitions of ontologies in the scientific literature, some elements are common to these definitions: a computer ontology is said to be an "agreement about a shared, formal, explicit and partial account of a conceptualisation" [5,19]. In addition, we retain that an ontology contains the vocabulary (terms or labels) and the definition of the concepts and their relationships for a given domain. In many cases, the instances of the application (domain) are included in the ontology as well as domain rules (e.g. identity, mandatoriness, rigidity, etc.) that are implied by the intended meanings of the concepts. Domain rules restrict the semantics of concepts and conceptual relationships in a specific conceptualisation of a particular application domain. These rules must be satisfied by all applications that want to use – or "commit to" [4] an interpretation of – an ontology.

A data model, on the contrary, represents the structure and integrity of the data elements of the, in principle "single", specific enterprise application(s) by which it will be used. Therefore, the conceptualisation and the vocabulary of a data model are not intended *a priori* to be shared by other applications [17]. E.g., consider a bookstore ontology with a rule that identifies a book by its (unique) ISBN. All applications that commit to this interpretation of this ontology [6] need to satisfy the identification rule. Library applications that do not foresee an ISBN for every book will not be able to commit to (or reuse) the bookstore ontology. Without such a bookstore ontology, two applications would even not be able to communicate (no sharing of vocabulary and domain rules by two applications). Modelling ontologies for a wide usage in an open environment, such as the Semantic Web, obviously is a challenging task. Providing more ontology rules, which are important for effective and meaningful interoperation between applications, may limit the genericity of an ontology. However, light ontologies, i.e. holding none or few domain rules, are not very effective for communication between autonomous software agents.

Therefore, in addition to the discussion on how to differentiate ontology from data modelling, we want to state a fundamental principle (introduced in [13]) – now called the *double articulation* of an ontology – for modelling and engineering shareable and re-usable ontologies. As a result, the outline of this paper is as follows: in the subsequent section (2), the similarities and differences between modelling of ontologies versus data models are discussed. The principle of the double articulation for ontology modelling and engineering is explained in section 3 with the

[1] The inspiration for the expression comes from the double articulation of a natural language as defined by Martinet [11]. Also his original definition carries over to our ontology context.

introduction of the STAR Lab DOGMA approach followed by an extensive example (section 4). Finally, a summary (section 5) concludes this paper.

2 MODELLING DATA SCHEMAS VS. ONTOLOGY MODELS

Data models, such as database or XML-schemes, typically specify the structure and integrity of data sets. Thus, building data models for an enterprise usually depends on the specific needs and tasks that have to be performed within this enterprise. The semantics of data models often constitute an informal agreement between the developers and the users of the data model [13] and which finds its way only in application prog that use the datamodel. E.g., in many cases, the data model is updated on the fly as particular new functional requirements pop up. In the context of open environments (as is the Semantic Web), ontologies represent knowledge that formally specifies agreed logical theories for an application domain [6]. Ontological theories, i.e. a set of formulas intended to be always true according to a certain conceptualisation [18], consist of domain rules that specify – or more precisely, approximate – the intended meaning of a conceptualisation. Ontologies and data models, both being partial accounts (albeit in a varying degree) of conceptualisations [5], must consider the structure and the rules of the domain that one needs to model. But, unlike task-specific and implementation-oriented data models, ontologies, in principle and by definition – see above – should be as much generic and task-independent as possible. The more an ontology approximates the ideal of being a formal, agreed and shared resource, the more shareable and reusable it becomes As is mentioned by Ushold, reusabilility and reliability are system engineering benefits that derive from the use of ontologies [18]. To these, we also add shareability, portability and interoperability and for the remainder of this paper we consider them all covered by the notion of "genericity".

In what follows, we discuss how (formally expressed) domain rules influence the genericity of knowledge modelled. The items mentioned below – in a non exhaustive manner – do not (yet) lead to a numerical measure or function that unequivocally allows differentiating an ontology from a data model. Nevertheless, they are useful points of reference when making the comparison.

1. *Operation levels.* Domain rules can be expressed on a low, implementation-oriented level, such as data types, null value, primary key (e.g. to enforce uniqueness) etc. More abstract rules, such as totality, rigidity, identity [7], etc. operate on a higher level irrespective of particular ways of implementation. The more abstract the domain rules are, the more generic the rules will be.

2. *Expressive power.* Data engineering languages such as SQL aim to maintain the integrity of data sets and use typical language constructs to that aim – e.g. foreign keys. In general, domain rules must be able to express not only the integrity of the data but also of the domain conceptualisation. Therefore, the language for the domain rules should include constructs that express other kinds of meaningful constraints such as taxonomy or that support inferencing – as is the case for e.g. DAML+OIL and OWL [3]. Providing expressive domain rule languages can lead to a more correct and precise conceptualisation of a domain. However, the addition of too specific domain rules (introducing more details or a higher complexity) can lead to a decrease of the genericity of a conceptualisation.

3. *User, purpose and goal relatedness.* Almost inevitably, users, goals and purposes [2] influence the modelling decisions during a conceptualisation of an application domain, see e.g. [18] – in the worst case an encoding bias could occur [4]. E.g., the granularity of the modelling process, the decision to model something as a class or an attribute, a lexical or non-lexical object type (see section 4), … all depend directly on the intended use of the conceptualisation. Domain rules operate on the constructed domain model, and therefore are also under the spell of the "intended use bias". A data model, in principle, nicely and tightly fits the specified goals and users of an application. It is clear that the genericity of a conceptualisation suffers from being linked too tightly to a specific purpose, goal or user group. Many (monolithical) ontologies, e.g. represented by means of DAML+OIL [3], are limited to one specific purpose due to the limited expressive power of the domain rule language! Clashes between different intended uses of such monolithical ontologies can occur and manifest themselves mostly at the level of domain rules.

4. *Extendibility.* Unlike data models, where modelling choices only have to take the particular universe of discourse of a specific application into account, a conceptualisation of a domain ontology is supposed to "consider the subjects separately from the problems or tasks that may arise or are relevant for the subject" [18]. It concerns the ease with which non-foreseen uses of the shared vocabulary can be anticipated [4]. We include in this notion also the domain rules as they determine how the vocabulary is used – which is in line with the definition of an ontological commitment [4]. E.g., a lot of attention might be paid to the question of what "exactly" identifies a concept, for instance, when

[2] The modeler's influence should be counterbalanced by the collaborative way of working during the modeling process.

modelling the identity of a person. The more relevant basic – almost philosophical – issues of concepts are discussed during the modelling stage, the more extensive [3] a conceptualisation (including the domain rules) will be. It is doubtful if monolithic ontologies can score well on this aspect. E.g. how graceful does performance degrade when the ontology-size multiplies or goes to a different order of magnitude [10].

The criteria mentioned above help to understand the differences between data models and ontologies and can serve to evaluate conceptualisations in general (including ontologies). The problem is that there doesn't exist a strict line between generic and specific knowledge [1]. And, there is a conflict between the genericity of the knowledge - as a fundamental asset of an ontology - and a high number of domain rules that are needed for effective interoperability. Monolithic ontologies are particularly sensitive to this problem as has been explained in item 3 above. Therefore, we want to introduce in the next section a fundamental ontology engineering principle, which builds on existing database modelling expertise, to resolve this conflict.

3 ONTOLOGY MODELLING IN THE DOGMA APPROACH: Ontology base, Commitments and Lexons

In this section we present the DOGMA[4] initiative for a formal ontology engineering framework – more details in [10]. The double articulation of an ontology is introduced: we decompose an ontology into an *ontology base*, which holds (multiple) intuitive conceptualisation(s) of a domain, and a layer of *ontological commitments*, where each commitment holds a set of domain rules. We adopt a classical database model-theoretic view [12,16] in which conceptual relationships are separated from domain rules. They are moved – conceptually – to the application "realm". This distinction may be exploited effectively by allowing the explicit and formal semantical interpretation of the domain rules in terms of the ontology. Experience shows that agreement on the domain rules is much harder to reach than one on the conceptualisation [15].

The *ontology base* consists of sets of intuitively "plausible" domain fact types, represented and organised as sets of context-specific binary conceptual relations, called *lexons*. They are formally described as <γ: Term1, Role, Term2>, where γ is a *context identifier*, used to group lexons that are intuitively "related" in an intended conceptualisation of a domain. Therefore, the ontology base will consist of *contextual components*. For each context γ and term T, the pair (γ, T) is assumed to refer to a unique concept. E.g., Table 1 shows an ontology base (for 'libraries' and 'bookstores') in a table format – taken from DogmaModeler [5] – that assures simplicity in storing, retrieving, and administrating the lexons. The ontology base in this example consists of two contexts: `'Books'` and `'Categories'`. Notice that the term `'Product'` that appears within both contexts refers to two different concepts: the intended meaning of `'Product'` within the context `'Categories'` refers to a topic of a book, while within `'Books'`, it refers to a "sellable entity".

The layer of *ontological commitments* mediates between the ontology base and its applications. Each ontological commitment corresponds to an explicit instance of an (intensional) first order interpretation of a task in terms of the ontology base. Each commitment consists of rules that specify which lexons from the ontology base are visible for usage in this commitment (see rules 1 & 7 prefixed with 'DOGMA.' in Table 2), and the rules that constrain this view (= commits it ontologically). E.g., 'library' applications that need to exchange data between each other, will need to agree on the semantics of the interchanged data messages, i.e. share an ontological commitment.

In DOGMA, ontological commitments do not a priori have to be expressed in one specific ontology language – see items 1 and 2. In accordance with the aspects mentioned in section 2, we emphasise that modelling ontological commitments in general will not be too specific to a limited number of applications – see item 3. Instead, they should be extendible – see item 4. As a result, (re-)usability, shareability, interoperability and reliability of the knowledge will be enhanced. Ontological commitments also become reusable knowledge components. An elaborated example on the commitment layer will be presented in the following section.

Note that an Object Role Modelling Mark-up Language [2] has been developed at STAR Lab to represent ORM [8] models in an XML-based syntax to facilitate exchanges of ontology models between networked systems. ORM, being a semantically rich modelling language has been selected as the basis for an ontology language that is to be extended within the DOGMA approach. As DOGMA native commitment language we similarly develop Ω–RIDL as an ontological extension of the RIDL language (e.g. [20]).

---

[3] Extensiveness is not always the same as a high granularity, but the latter can sometimes be the result of the former. The differentiating factor here is the user, purpose or goal relatedness.

[4] Developing Ontology-Guided Mediation for Agents.

[5] DogmaModeler is a research prototype of a graphical workbench, developed internally at STAR Lab, that serves as modelling tool for ontologies on basis of the ORM graphical notation.

Table 1: “BibliOntology Base”

Ontology Base (Lexons)				
N	ContextID	$Term_1$	Role	$Term_2$
1	Books	Book	Is_A	Product
2	Books	Book	Has	ISBN
3	Books	Book	Has	Title
4	Books	Book	WrittenBy	Author
5	Books	Book	ValuedBy	Price
6	Books	Author	Has	First_Name
7	Books	Author	Has	Last_Name
8	Books	Price	Has	Value
9	Books	Price	Has	Currency
10	Categories	Topic	SuperTopicOf	Computers
11	Categories	Topic	SuperTopicOf	Sports
12	Categories	Topic	SuperTopicOf	Arts
13	Categories	Computers	SuperTopicOf	Computer_Science
14	Categories	Computers	SuperTopicOf	Programming
15	Categories	Computers	SuperTopicOf	Product
16	Categories	Product	SuperTopicOf	CASE_Tools
17	Categories	Product	SuperTopicOf	Word_Processors
18	Categories	Product	SuperTopicOf	DBMS

We conclude this section by summarising that the DOGMA approach takes agreed semantical knowledge out of an IT application that makes use of an external ontology. This is done in much the same way that “classical” databases take data structures out of these applications. Likewise, ontologies built in accordance with the principle of the double articulation achieve a form of semantical independence for IT applications [14].

4 EXAMPLE

We take again the BibliOntology Base provided in Table 1 and present two different *kinds* of applications: ‘Library’ applications that need to interoperate with other libraries, and ‘Bookstore’ applications that additionally need to interoperate with other bookstores, customers, publishers, etc. Suppose that each kind of application has different domain rules that do not necessarily agree with the other’s rules, i.e. perform ‘slightly’ different tasks. E.g., unlike bookstores, library applications don’t exchange pricing information. Likewise, bookstores identify a book by its ISBN, while in library systems, ISBN is not a mandatory property for every book. They identify a book by combining its title and authors.

Figure 2 and Figure 1 show a graphical representation (taken from the DogmaModeler tool) of the ontological commitments for ‘bookstore’ and ‘library’ applications respectively. Both commitments share the same BibliOntology Base (see Table 1). Each commitment consists of a set of domain rules that define the semantics of exchanged data messages. Note that applications that commit to an ontology may retain their internal data models. E.g., Figure 3 and Figure 4 show valid XML data messages that comply with the ontological commitments that are defined in Figure 2 and Figure 1 respectively.

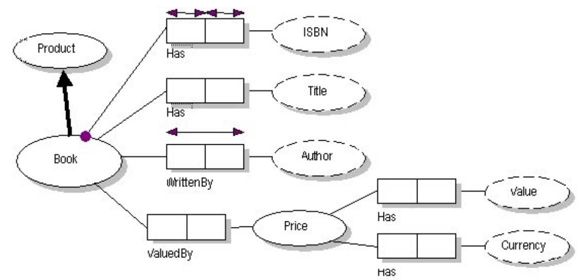

Figure 2: bookstore commitment (OC_A)

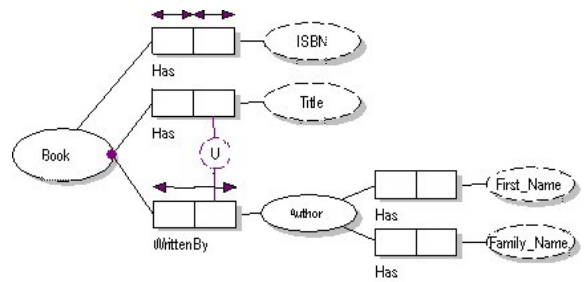

Figure 1: library commitment (OC_B)

Table 2 shows a declarative textual representation of the two ontological commitments OC_A and OC_B. We adopt a notational convention to denote the ontology language by a prefix – c.q. "ORM.”([8]) – for rules that are intended to be interpreted as "standard" ORM. For simplicity of reading, we present the ORM rules as verbalised fixed-syntax English sentences (i.e. generated from agreed templates parameterised over the ontology base content). Notice that the ontological commitments in this example are supposed to be specified at the knowledge level [4], i.e. they are more than data models and integrity constraints.

Rules 1 & 4 are visibility rules that determine which lexons from the ontology base are “committable” for that particular commitment. More precisely, these rules determine which lexons are part of the model (first order interpretation) for that particular commitment seen as a theory. The visibility rules make sure that updates in the ontology base do not necessarily affect every commitment. As a result, the commitments have a certain stability and the ontology base can be updated whenever suited. The double articulation of a DOGMA ontology resolves the clashes referred to in item 3 of section 2.

```
<Book Sub-type-of='Product'>
  <ISBN> 0805317554 </ISBN>
  <Title>Fundamentals of Database Systems</Title>
  <Author> Ramez A. Elmasri</Author>
  <Author>Shamkant B. Navathe </Author>
  <Price Value='95' Currency='USD' />
</Book>
<Book Sub-type-of='Product'>
  <ISBN>1558606726</ISBN>
  <Title>Information Modeling and Relational...</Title>
  <Author>T. Halpin</Author>
  <Price Value='60' Currency='USD' />
</Book>
```

Figure 3: message compliant with OC_A

```
<Book>
  <ISBN>0444700048</ISBN>
  <Title>Database Semantics</Title>
  <Author First_Name='Robert' Last_Name='Meersman'/>
  <Author First_Name='# ? #' Last_Name='Steel'/>
</Book>
<Book>
  <Title>Knowledge Representation:...</Title>
  <Author First_Name='John' Last_Name='Sowa'/>
  <Author First_Name='David' Last_Name='Dietz'/>
</Book>
```

Figure 4: message compliant with OC_B

For example, OC_B does not commit to the BibliOntology Base (see Table 1) to use information about Price (lexon IDs 5, 8 & 9 of Table 1 as defined in the visibility rule 1), and likewise OC_A does not even see the lexons about an Author having a First_Name or a Family_Name (lexon IDs 6 & 7 of Table 1 as defined in the visibility rule 4). Rules 2 and 5 define the lexical object types (LOTs), which are dotted circles in ORM-style. Lexical objects refer to individual "utterable" entities; while the non-lexical objects (NOLOTs), the non-dotted circles in ORM-style, refer to "non-utterable" entities [9].

Table 2: some commitments for the BibliOntology Base [6]

RuleID	Rule Definition	Commit-ment_ID
1	DOGMA.Visible-Lexons to this commitment are {$$L1 .. $$L4, $$L7, $$L8,};	OC_A
2	ORM. Lexical Object Types are {ISBN, Title, Author, Value, Currency};	OC_A
3	ORM.Mandatory(Each Book Has at least one ISBN);	OC_A
8	ORM.InternalUniqueness(Each Book Has at most one ISBN);	OC_A
9	ORM.InternalUniqueness(Each ISBN IsOf at most one Book);	OC_A
10	ORM. InternalUniqueness(Each Book maybe WrittenBy many different Author (s), and each Author maybe Writes many different Book(s));	OC_A
4	DOGMA.Visible-Lexons to this commitment are {$$L2 .. $$L4, $$L6, $$L7,};	OC_B
5	ORM. Lexical Object Types are {ISBN, Title, First_Name, Family_Name};	OC_B
6	ORM.Mandatory(Each Book Has at least one Title and WrittenBy at least one Author, at the same time);	OC_B
7	ORM.ExternalUniqueness(Both (Title, Author) as a combination refers to at most one Book);	OC_B
8	ORM.InternalUniqueness(Each Book Has at most one ISBN);	OC_B
9	ORM.InternalUniqueness(Each ISBN IsOf at most one Book);	OC_B
10	ORM. InternalUniqueness(Each Book maybe WrittenBy many different Author (s), and each Author maybe Writes many different Book(s));	OC_B

Notice that deciding what is a LOT and what is a NOLOT is goal or purpose related (see item 3 of section 2). E.g. the author's name in OC_A is defined as a LOT while in OC_B it is defined as a NOLOT since the library applications use the first and family names as a combined identifier with the title. In addition, multiple commitments can be defined on (a selection of) the same (large) ontology base. Both applications commit to use the ISBN concept represented by the lexon with ID 2 (see Table 1). However, OC_A has a different commitment on it than OC_B. Rules 3 & 8 and rules 5 & 6 respectively define the identification rule as already mentioned in section 1.

5 CONCLUSION

In this paper, we have described some aspects that help to understand the distinction between data models and ontologies. As a result, a mismatch between the genericity of ontologies and the specificity of domain rules has been

[6] Physically, this table is stored in a non-redundant form – for more details we refer to [10].

detected. In order to resolve this mismatch, we have proposed the DOGMA framework for ontological engineering that introduces a double articulation for ontologies. An extensive example has illustrated the advantages of this double articulation of a DOGMA ontology.

6 ACKNOWLEDGMENTS

Parts of this research have been funded by the OntoBasis (IWT-GBOU 2001 010069) and CC FORM projects (EU-IST-2001-34908). We also thank our STAR Lab colleagues.

7 REFERENCES

[1] Chandrasekaran B. & Johnson T., (1993), Generic Tasks and Task Structures: History, Critique and New Directions, in David J., Krivine J. & Simmons R., (eds.), Second Generation Expert Systems, Springer, pp. 233 – 272.

[2] Demey J., Jarrar M. & Meersman R., (2002), A Conceptual Markup Language that supports interoperability between Business Rule modeling systems, in Pu C. & Spaccapietra S. (eds.), Proc. of the Tenth Internat. Conf. on Cooperative Information Systems (CoopIS 02), LNCS 2519, Springer

[3] Fensel D., I. Horrocks I., van Harmelen F., Decker S., Erdmann M. & Klein M., (2000), OIL in a nutshell, in Dieng R. et al. (eds.), Knowledge Acquisition, Modeling, and Management, Proc. of the European Knowledge Acquisition Conf. (EKAW-2000), LNAI 1937, Springer Verlag, Heidelberg

[4] Gruber T., (1995), Towards Principles for the Design of Ontologies Used for Knowledge Sharing, International Journal of Human-Computer studies, 43 (5/6): 907 – 928.

[5] Guarino N. & Giaretta P., (1995), Ontologies and Knowledge Bases: Towards a Terminological Clarification, in Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing, N. Mars (ed.), IOS Press, Amsterdam, pp 25 – 32.

[6] Guarino N., (1998), Formal Ontologies and Information Systems, in Guarino N. (ed.), Proc. of FOIS98, IOS Press, pp. 3 – 15.

[7] Guarino N. & Welty C., (2002), Evaluating Ontological Decisions with OntoClean, in Communications of the ACM, 45 (2): 61 – 65.

[8] Halpin T., (2001), Information Modeling and Relational Databases: from conceptual analysis to logical design, Morgan-Kaufmann, San Francisco

[9] ISO, (1982), Terminology for the Conceptual Schema and Information Base, ISO Technical Report TR9007

[10] Jarrar M. & Meersman R., (2002), Formal Ontology Engineering in the DOGMA Approach, in Liu Ling & Aberer K. (eds.), Proc. of the Internat. Conf. on Ontologies, Databases and Applications of Semantics (ODBase 02), LNCS 2519, Springer Verlag

[11] Martinet A., (1955), Economie des changements phonétiques, Berne: Francke, pp. 157-158.

[12] Meersman R., (1994), Some Methodology and Representation Problems for the Semantics of Prosaic Application Domains, in Ras Z., Zemankova M., (eds.), Methodologies for Intelligent Systems (ISMIS 94), LNAI 869, Springer Verlag, Heidelberg

[13] Meersman R., (1999), The Use of Lexicons and Other Computer-Linguistic Tools in Zhang Y., Rusinkiewicz M, & Kambayashi Y. (eds.), Semantics, Design and Cooperation of Database Systems, in The International Symposium on Cooperative Database Systems for Advanced Applications (CODAS 99), Springer Verlag, Heidelberg, pp. 1 – 14.

[14] Meersman R., (2001), Ontologies and Databases: More than a Fleeting Resemblance, in d'Atri A. and Missikoff M. (eds), OES/SEO 2001 Rome Workshop, Luiss Publications

[15] Meersman R., (2002), Semantic Web and Ontologies: Playtime or Business at the Last Frontier in Computing ?, in NSF-EU Workshop on Database and Information Systems Research for Semantic Web and Enterprises, pp.61 – 67.

[16] Reiter R., (1988), Towards a Logical Reconstruction of Relational DB Theory, in Mylopoulos J. & Brodie M., Readings in AI and Databases, Morgan Kaufman

[17] Sheth A. & Kashyap V., (1992), So far (schematically) yet so near (semantically), in Hsiao D., Neuhold E. & Sacks-Davis R. (eds.), Proc. of the IFIP WG2.6 Database Semantics Conf. on Interoperable Database Systems (DS-5), Lorne, Victoria, Australis. North-Holland, pp. 283 – 312.

[18] Ushold M. & King M., (1995), Towards a Methodology for Building Ontologies, in Proc. of the Workshop on Basic Ontological Issues in Knowledge Sharing (IJCAI95 workshop), also available as AIAI-TR-183 [ftp.aiai.ed.ac.uk/pub/documents/1995/95-ont-ijcai95-ont-method.ps.gz]

[19] Ushold M. & Gruninger M., (1996), Ontologies: Principles, methods and applications, in The Knowledge Engineering Review, 11 (2): 93 – 155.

[20] Verheyen G. & van Bekkum P., (1982), NIAM, aN Information Analysis method, in Olle T., Sol H. & Verrijn-Stuart A. (eds.), IFIP Conference on Comparative Review of Information System Methodologies, North Holland