Vrije Universiteit Brussel

FACULTEIT WETENSCHAPPEN
Vakgroep Informatica en Toegepaste Informatica
Semantics Technology and Applications Research Lab

The use of lexicons and other computer-linguistic tools in semantics, design and cooperation of database systems

STAR Lab Technical Report

1999

Robert Meersman

affiliation:
corresponding author: Robert Meersman
keywords: ontology, lexicon, database modelling, semantics
number: STAR-1999-02
date: 16/06/2005
status: final
reference: Zhang, Y., Rusinkiewicz, M. & Kambayashi, Y., (eds.), Cooperative Database Systems and Applications '99, The Proceedings of the Second International Symposium on Cooperative Database Systems for Advanced Applications (CODAS'99), Springer Verlag, pp. 1-14

Pleinlaan 2, Gebouw G-10 B-1050 Brussel | Tel. +32 (0)2 629 12 37 | Fax. +32 (0)2 629 38 19 | http:// www.starlab.vub.ac.be

The use of lexicons and other computer-linguistic tools in semantics, design and cooperation of database systems

R.A. Meersman[1]

Vrije Universiteit Brussel (VUB)
STARLab
Building G/10, Pleinlaan 2
B-1050 Brussels Belgium
meersman@vub.ac.be

Abstract. Computerized lexicons and thesauri are kinds of so-called "ontologies" that enable new formalizations of the semantics of information systems. This semantics plays a crucial if sometimes hidden role in the analysis, design and implementation of such systems (and in fact general software systems). Explicitly adding (the access to) global, domain- or application-specific ontologies to systems makes it possible to define better agreements about the meaning of the represented information. We call the resulting computerized information systems CLASS for Computer-Lexicon Assisted Software Systems. When using such techniques to define and formalize cooperation of databases in larger information systems, the need for a "global" common ontology (lexicon, thesaurus) becomes apparent. We may readily identify some desirable properties and requirements. Extrapolating from an ongoing EU research project called TREVI and other projects elsewhere, avenues of future research in this area may be indicated. In particular, certain problems can be identified with well-known existing lexicons such as CYC and WordNet, as well as with sophisticated representation- and inference engines such as KIF or SHOE. We argue that large public lexicons should be very simple, i.e. their semantics become implicit by agreement among "all" users, and ideally completely application independent. An important challenge is therefore that architectures should become *modular* to allow the adding of application-dependent semantics (represented in thesauri). In particular, this requirement poses interesting research issues on methodology that need to be resolved. **Note.** Parts of this paper are based on [Mee99] which however did not go deeply into methodological issues.

1. Introduction

The semantics of an information system has always been an important but difficult issue, meriting but often also defeating good formal treatment. Yet this issue dominates -often tacitly- the analysis, design and implementation of such systems.

[1] This research was partially supported by the ESPRIT Project "TREVI", nr. 23311 under the European Union 4th Framework Programme

The relevant literature contains many different ways and attempts to represent the meaning of data for use in databases and their applications in computerized information systems. We claim that computer-based "ontologies" (which covers lexicons and thesauri, see below for a more precise definition) may provide a new and useful step in formalizing the semantics of represented information. In fact we shall argue that such ontologies, in principle, can *actually be* (or rather become in the near future) the semantic domain for an information system in a very concrete and useful manner.

Modern distributed object technologies on the market such as CORBA [OMG95], DCOM [Red97] etc. make it ever more likely, desirable, and even *necessary* that objects or agents performing interesting services "meet" interested consumers of those services without prior knowledge of each other's design and precise functionality. Mediators, wrappers and other devices have been proposed to help make this possible, but these are at present rarely full runtime solutions. The sheer multitude of such objects/agents in the future indeed dictates that the necessary agreements will have to be negotiated in real-time as the "meeting" occurs. Without a universal standard for programming and data modeling, and even in the hypothetical presence of such standard it is hard to imagine that these agreements can occur without the use of sophisticated and general lexicons or thesauri (now often called by the neologism "ontologies"). These will need to be (a) very large resource(s), possibly only one per language, but (b) with a rather simple, "semantics-less" structure that makes them nearly completely independent from any domain-specific applications.

To make ontologies useable, *a priori* and mostly *implicit* agreements must exist about their contents (i.e. their individual entries). While this is a steep requirement it is to be expected it will still be much easier and more stable than dealing with a very large number of individual and application context-specific local agreements about the "real world" itself. Even if one fails to reach a complete match between one's terms and those in the ontology, the benefits would likely outweigh the cost of the required compromise. At present, only a few and partial resources of a related kind exist, either as research prototypes (e.g. the very popular [WordNet], see also [Fel98]) or proprietary developments as e.g. [CYC], but they are as yet far removed from being a standard.

But it is unmistakable that with the advent of e-commerce, and the resulting natural language context of its related activities, that ontologies, lexicons and thesauri and research in their use for system design and interoperation will receive a major market-driven push. The availability of XML [BPS98] and the obvious platform for ontological knowledge it provides will also remold the landscape of added value supply and demand on the internet (for an example of this derived from the [SHOE] ontology project, see [HHL98]). Organizations that promote and commercially develop ontologies as resources for this purpose are springing up: [Ontology], [RosettaNet], [X-Act] as well as several "public" communities around XML, to name a few.

Next we try to illustrate a few basic concepts and requirements using an example inspired on [TREVI] (but not part of its current implementation).

2. Motivating Example (from [Mee99]).

A number of distributed software agents are playing around a blackboard communication facility. Some of these agents are **`Publisher`**, **`Categorizer`**, **`Intaker`**, **`Matcher`**, and **`User Agent`**. There may be several of each, and others, active at any given time. As we look, **`User Agent`** happens to be servicing a subscriber who wants to set up an on-line news website on basketball. With the help of **`User Agent`** he has just selected a suitable website template according to his user profile, and posted a request on the blackboard based on this. **`User Agent`** has a *domain-specific thesaurus* of such templates to choose from. The subscriber wishes to title the website "Hoops Today". For the sake of the example, assume the template (or the subscriber) is somewhat simplistic, and after instantiation by the subscriber for his profile looks like (in a fictitious and arbitrary but hopefully self-explaining syntax):

```
website
      is-a          place < virtual
      part_of       internet
      has           layout = [default]
      has           title = "Hoops Today"
      has           content = on-line_news < stream
      has           subject = basketball +
      has_not       subject = (basketball_player -, "sex" ~)
```

The intended meaning is that the desired website generalizes linguistically to a place, the hypernymy hierarchy is limited to virtual places, and is in part-of relationship to the notion "internet". It has on-line news content, limited to be a stream; the subject must be "basketball" or maybe something more general (the **+**); but not containing tidbits involving basketball players or more specific individuals (the **–**) mentioned together with their favorite pastime or a synonym for it (the **~**). Note that entries in such a thesaurus in reality must be the result of proper, "production quality" and domain-specific *data modeling*.

The tokens "is-a", part-of", "has", "has_not" are called *roles* and they determine how the term following them has to be interpreted; "has content" in our thesaurus implies that content is an *attribute* of website. Note the first two items would in general not be application-specific properties of a website; however the subscriber didn't find "web site" in his favorite lexicon (WordNet, a popular lexicon also we use in several examples further on), and so **`User Agent`** had to extend his domain thesaurus a little.

Clearly the interpretation of roles, and of operators like **+** when they are domain- or application-specific, is the internal responsibility of the **`User Agent`**. We say that they belong to its *ontology*. This is a general principle; agents may communicate with each other of course but must do so only through a simple "flat" lexicon devoid of domain specific semantics. The role "is-a" for instance describes in general a domain independent (meta)relationship and so is implemented in almost every known lexicon, thesaurus or "ontology", and its semantics should be inherited from and interpreted at the language (not domain) level.

After this syntactical intermezzo, let's turn back now to our little system where the agents are waiting, and quickly run through a possible scenario for it. For the sake of further simplicity we shall ignore the "not", "is-a" and "part-of" roles and just concentrate on content type and required subject. The reader may easily imagine scenarios that involve these properties as well.

A **`Publisher`** in the meantime has triggered on the posted request, knowing it can fulfill it provided it can find the right inputs. It will need to receive these as data coming from other agents; in this case it looks for NITF[2] formatted news text and, since it has experience with the media and news items, also for a relevance ranking with respect to the subject "basketball". Note that **`Publisher`** needs no specialized basketball knowledge; it only uses the fact that basketball is a subject.

`Publisher` now posts itself two messages on the blackboard, the first asking for NITF-formatted on-line news, the other asking for a relevance factor of news articles in relation to the subject basketball.

Two agents trigger on these requests. The **`Intaker`** agent can supply the NITF formatted text, so the agent commits to this. This commitment is accepted by the **`Publisher`** which removes this part of its request.

The second agent is a **`Matcher`** who can compute the relevance provided it can receive categorization data on the subject basketball. It posts a request for such categorization on the blackboard.

A rampant **`Categorizer`** passes by and decides it could partially fulfill this request, given the right inputs. It cannot fulfill the exact categorization required, but after consulting a general-purpose lexicon (such as WordNet for example) it finds it can satisfy a more general categorization on the subject of "sports". Since fulfillment is only partial the posting agents will leave their request on the blackboard until, maybe, in the end the **`User Agent`** commits if he decides relevance is high enough. For instance, look at Fig. **1** which shows the (partially flattened and edited) entry for the specialization hierarchy containing sports and basketball in WordNet. If **`Categorizer`** would generalize to (another) "court game", the high fan-out under that term might reduce the relevance noticeably.

More requests will be posted until finally **`Publisher`** can make a partial commitment that is accepted by the **`User Agent`**, and now a series of parameterized steps is sent to the system for execution. Note that all the steps in the process made use of domain-specific and general "data" knowledge residing in thesauri, while the “operations" knowledge necessary for the use, application or rewriting of the rules was kept inside the agents. This is of course intentional and makes the system flexible and robust, e.g. to allow run-time reconfigurations caused by new (types of) agents appearing at the table.

sport, athletics -- (an active diversion requiring physical exertion and competition)
=> contact sport -- (a sport that necessarily involves body contact between opposing players)
=> outdoor sport, field sport -- (a sport that is played outdoors)
=> gymnastics --
(a sport that involves exercises intended to display strength and balance and agility)
=> track and field --
(participating in athletic sports performed on a running track or on the field associated with it)

[2] NITF = News Industry Text Format, electronic standard used by international news agencies

Fig. 2. Sports hierarchy including "basketball" obtained from the [Wordnet] lexicon (partly edited and collapsed for simplification)

Certain aspects of the above system, organization and scenario of operation are currently under investigation as part of a research effort derived from the [TREVI] Project at VUB STARLab. The current implementation of TREVI itself (in EU Esprit Project #23311) is a personalized electronic news server system that is designed to operate on a continuous basis (the news wire is provided by Reuters plc., a partner in the project), and using a lexicon to identify the subject of and "enrich" on-line news items with links to databases, supporting a high volume of subscribers. On the basis of a user profile the system provides the subscriber with personalized news on a website, through an email message service, or via a custom-made client application. See also [BDP99] for details on the parsing and subject identification. The system is unusually modular and a limited ability to process meta-data presently already allows adding new functionality (module types) to the system at runtime, i.e. while it is actually running. User profiles are mapped onto series of processing steps, each corresponding to a parameterized stateless operation of a module type. The system performs merging and parallelization of the series behind the scenes to achieve an optimized execution scenario. The current system prototype uses a Java/CORBA backbone architecture over a distributed set of (NT) servers.

[TREVI] is an example of a project that uses lexicon technology in several ways. We may call such systems *CLASS*-type for Computer Lexicon Assisted Software

Systems. The design implied in the example above would make such technology fundamental to its operation. As should be immediately obvious, because of the heterogeneous nature of agent technology and the "openness" of possible request, formal agreement on the meaning of the terms (and if possible processes) is indeed primordial. We believe that thesauri (domain-specific ontologies) and the availability of global lexicons will make this possible, by providing a pragmatically usable substitute for a formal, "reductionist" definition of information system semantics. We explain terminology and these principles in the next section.

3. The role of Ontology in the Semantics of Information Systems

The term "ontology", or rather as the neologism "*an* ontology" is in use already for some time in the AI community and usually denotes an collection of linguistic objects organized in a variety of ways. In fact, within the information systems community most of these ontologies would be called nothing else but *data models* for an underlying, usually fairly narrow, (application) domain. Incidentally, a data model in modern IS terminology is a much richer structure than merely a set of relational tables or records, and may involve subtypes, constraints ("business rules") and other constructs that help in modeling a domain or system more "semantically". To make this more precise, we need to define the concepts of information system, semantics and ontology a little more formally.

We shall assume here for simplicity (but without loss of generality) that an information system is defined in a strict model-theoretic paradigm by a pair (S, P) where S denotes a *conceptual schema* and P denotes a *population* of instances that "satisfies" S in an intuitive but well-defined sense (viz. logically P is a *model* for S). Informally speaking, S typically contains "sentence patterns" in terms of types or classes, including taxonomies, naming conventions, and (some) constraints or "business domain rules" while P contains instances of these sentences –ground atomic formulas– in terms of proper nouns, numbers and other lexical elements of the universe of discourse. P is typically implemented as a database, for instance. In this generic architecture there is also an active component, the *information processor* which manipulates the instances of P such that consistency, i.e. satisfaction of S, is maintained. A customary implementation of an information processor is by a DBMS, nearly always supplemented by a set of *application programs* that either implement user-desired functionality, or consistency requirements defined in S, or both.

Note this view is simplified since it does not –at least not explicitly– treat dynamic aspects of the domain such as events, processes, … It does however to see an IS expressed as a set of well-formed formulas in a *language* describing both S and P (the latter as ground atomic formulas).

The semantics of an information system may then classically –and conveniently– be defined (e.g. following [G&F87]) as a mapping (interpretation) of the terms of this language into the "real world" or rather into a substituting conceptualization. It is absolutely fundamental to realize the importance and consequences of the choice of this conceptualization; in fact Gruber [Gru93] precisely defines an ontology as the specification of a conceptualization.

In information systems design and cooperation, an obvious but poorly recognized fact is that any non-trivial semantics must be the result of an *agreement* (most likely qualified by a *group of agents* and a *moment in time*) among all present and future users, designers and implementers of the IS. It is essential to realize that these agreements cannot in general be proven from axioms by reductionist argument, simply because of the required, and wholly unproductive, complexity. It is therefore especially here where IS semantics is most clearly distinguished from e.g. denotational semantics of programming languages [Sch86] and where the usefulness of an explicit, common, even global ontology (lexicon, thesaurus) becomes most apparent. Ontologies may indeed conveniently *replace* the actual real-world domain of the semantics interpretation mapping, i.e. become concrete and practical (albeit quite large) conceptualizations of the real world in a limited, well-defined sense.

To help avoid confusion we will at this point and for the purpose of this paper adopt the terminology ([Mee99]):

- ***an ontology*** is any formally specified conceptualization, replacing the real world in a semantics interpretation mapping as described above;
- ***a thesaurus*** is a domain-specific ontology (examples of domains are Manufacturing, Laptop_Manufacturing, Naïve_Physics, Corporate_Law, Ontology_Theory, …) or an application(s)-specific ontology, such as for Inventory_Control, Airline_Reservations, Conference_Organization, …;
- ***a lexicon*** is a (natural) language-specific ontology, e.g. for English, Esperanto, Polish, …

The application, domain or set of linguistic concepts that constitute the subjects under consideration are called the *universe of discourse* of the ontology. Admittedly for a thesaurus the distinction between application-specific and domain-specific can be vague and must often be left to intuition, especially since ontologies may cover a wide *set* of applications. This may be even more so for the finer levels of ontology that exist within an application; Qantas' *instantiation* of an Airline_Reservations application is likely to use different terms, jargon and coding conventions than Sabena's, and within an instantiated application it may be useful to decompose it into its individual *functions*. Nevertheless this level-like distinction is a convenient and customary one in many IS modeling methodologies where applications are developed within a shared domain; it is sometimes known as the *onion model* [ISO90]. Next we take a closer look at some of these methodological aspects.

4. Methodological Issues in Building and Applying Ontologies

As indicated earlier, the construction of an ontology and especially a "complete" lexicon involves numerous agreements on the meaning of the terms (entries) represented and their relationships to other terms, usually within a given context. It is therefore crucial that the process of creating an ontological resource follows a –preferably formal– methodology. Such a methodology needs to be much more than a representation technique, and ideally requires a stepwise procedure starting with perception and cognitive observation of an application, a domain, or even the universe as a whole. Many such methodologies have been proposed, developed and defended

over the years in the area of databases and information systems. Not all are equally directly applicable to creating ontologies, however.

4.1. Similarities between ontologies and conceptual schemas for information systems –some examples

Any IS methodology may in a sense be considered as a set of techniques and procedures, usually fairly informal, for reaching the semantics agreement discussed above among designers, domain experts and users of the IS. Most work in ontology construction today is in fact a form of data modeling with techniques similar to those pioneered for database development, but for two fundamental differences discussed below. A few examples are well-known ontologies such as [CYC], a vast and interesting resource which is rather a true (extensional) lexicon, Ontolingua [KIF], which is closer to a tool for incrementally constructing (intensional) *theory ontologies* –for terminology see [SPK96]–, both of which are domain-independent ontologies, and [RosettaNet] which is a wide-ranging (and growing) thesaurus on the domain of supply chains, being constructed by a partnership of industries. An embryonic thesaurus for an Enterprise Domain may be studied in [UKM98]. Likewise the use of ontology in wider domains such as Law is emerging [B&V97].

To illustrate the flavor of such ontologies, Fig. 2 shows the CYC definition of the constant Skin as part of the CYC *microtheory* (i.e. context) on Physiology. It shows an example of the ubiquitous taxonomy relationship classifying Skin as a kind of Animal_Body_Part. Fig.3 displays a part of a KIF (Ontolingua) theory for Binary Relations, and likewise implies that it is a kind of constrained (specialized) Relation.

#$Skin

A (piece of) skin serves as outer protective and tactile sensory covering for (part of) an animal's body. This is the collection of all pieces of skin. Some examples include TheGoldenFleece (an entire skin) and YulBrynnersScalp (a small portion of his skin).
isa: #$AnimalBodyPartType
genls: #$AnimalBodyPart #$SheetOfSomeStuff #$VibrationThroughAMediumSensor #$TactileSensor #$BiologicalLivingObject #$SolidTangibleThing
some subsets: (4 unpublished subsets)

#$AnimalBodyPart

The collection of all the anatomical parts and physical regions of all living animals; a subset of #$OrganismPart. Each element of #$AnimalBodyPart is a piece of some live animal and thus is itself an instance of #$BiologicalLivingObject. #$AnimalBodyPart includes both highly localized organs (e.g., hearts) and physical systems composed of parts distributed throughout an animal's body (such as its circulatory system and nervous system). Note: Severed limbs and other parts of dead animals are NOT included in this collection; see #$DeadFn.
isa: #$ExistingObjectType
genls: #$OrganismPart #$OrganicStuff #$AnimalBLO #$AnimalBodyRegion
some subsets: #$Ear #$ReproductiveSystem #$Joint-AnimalBodyPart #$Organ #$MuscularSystem #$Nose #$SkeletalSystem #$Eye #$RespiratorySystem #$Appendage-AnimalBodyPart #$Torso #$Mouth #$Skin #$DigestiveSystem #$Head-AnimalBodyPart (plus 16 more public subsets, 1533 unpublished subsets)

Fig. 2. CYC® constants: #$Skin generalizes to #$AnimalBodyPart. ©Cycorp, Inc.

Class BINARY-RELATION
Defined in theory: Kif-relations
Source code: frame-ontology.lisp

Slots on this class:
Documentation**:**
A binary relation maps instances of a class to instances of another class.
Its arity is 2. Binary relations are often shown as slots in frame systems.
Subclass-Of**:** Relation

Slots on instances of this class:
Arity**:** 2

Axioms:

```
(<=> (Binary-Relation ?Relation)
    (And (Relation ?Relation)
        (Not (Empty ?Relation))
        (Forall (?Tuple)
                (=> (Member ?Tuple ?Relation) (Double ?Tuple)))))
```

Fig. 3. A KIF built-in theory for Binary Relations (a subtype of Relations)

Examples of more modest projects applying existing data modeling methods to building ontologies (actually application-dependent thesauri) are [ECS98] who use

OSM [Emb98] to build a schema for an application domain that is used interestingly to generate a parser and SQL database generator for WWW documents over this domain, and the similarly inspired and targeted [OntoBroker] project, reported in [DEF99], with its emphasis on information retrieval.

4.2. Fundamental differences between ontologies and conceptual schemas for IS

Ontologies and conceptual schemas of course are formally distinct, the former mathematically speaking being the domain and the latter the range of the semantic interpretation mapping. However as explained earlier, both can be seen as representations of a commonly perceived reality, and as such are intimately related. It is thus to be expected that designing either one would make use of related approaches.

Not surprisingly, a number of the ontology approaches reuse –or even rediscover– principles and methods for constructing conceptual schemas (data models plus constraints, events, derivation rules,…) for information systems that were in use for nearly three decades [ISO90]. There are however two fundamental –and interrelated– differences between ontologies and "classical" information system schemas which necessitate careful consideration when applying a data modeling method to ontology construction:

(1) **ontologies, especially lexicons, need to be *application independent***. In IS design the underlying notion (indeed the very purpose of databases) is *data independence*, which means that one prefers a conceptual schema rich in semantics such that IS applications ideally become trivial and can be written purely in function of application requirements, and thus independently of data representation and management. Global ontologies, being domain or even language models, on the other hand need to be as "neutral" as their purpose allows, leading to almost *semantics-less* representations of domains. [CYC] and [WordNet] may be seen as exponents of such application independence (but suffer from other problems). Note how this fits with a view of semantics as agreements among user groups: the wider the required applicability, the more difficult overall agreement will be, ultimately –in the theoretical worst case– degenerating into a concordance of lexical terms from an agreed natural language lexicon and their syntactical use… Naturally, this in turn requires that application- and domain semantics either be delegated to specific thesauri, or preferably –in order to keep thesauri simple as well– to interpreters of these.

(2) **ontologies, being conceptualizations of domains, unlike for information systems attempt to model a part of reality**, so to speak. A conceptual schema for an IS on the other hand ultimately describes a storage and application program model, i.e. specifies an implementation of structure and behavior according to stated functionality requirements. In other words, one models a *system*, usually with a priori known application requirements, rather than a *domain*; it is this principle which in fact makes data independence possible, as discussed above. This means in practice that while an ontology may also be seen as a database of sentence types (and sentences, statements about the real world) many of these need not have themselves explicit *instances*, the description, storage and manipulation of which is of course the (sole) purpose of an IS. In

fact these instances *do* exist, at the application level: as entries in an application-specific thesaurus they would constitute the semantic interpretations of the *population* of the information system (database). In the terms of the WordNet example in Fig. 1, one would not in general expect the LA Lakers basketball team to figure as entry in the lexicon, but the fact that they play a match on the Ostend Municipal Basketball Court may be a fact stored in the IS that must then fit a pattern in the lexicon if the IS and its conceptual schema are to be considered "semantically correct".

4.3. Methodological implications and requirements for ontology construction

Semantic simplicity at the lexicon level in order to achieve application independence requires that lexicon entries be as elementary as possible, for instance of the form called *lexons* in [Mee99], namely

$\gamma(t_1\ r\ t_2)$, where t_1 and t_2 are lexicon terms, r is a role in a (binary) relationship and γ is a label (again a lexicon term) for a context

For example, for the WordNet **news_item** hypernymy hierarchy ("superset chain") in Fig.4, we may define the lexons γ_1(news_item is-a item) and γ_2(news_item is-a relation) but while the first lexon will be acceptable in most contexts γ_1, clearly it is going to depend very much on the context γ_2 to make the second lexon sound "meaningful".

news item IS A KIND OF ...
1 sense of news item

Sense 1
news item -- (an item in a newspaper)
=> **item, point** -- (a distinct part that can be specified separately in a group of things that could be enumerated on a list; "he noticed an item in the New York Times"; "she had several items on her shopping list"; "the main point on the agenda was taken up first")
  => **part, portion, component part, component** -- (something determined in relation to something that includes it; "he wanted to feel a part of something bigger than himself"; "I read a portion of the manuscript"; "the smaller component is hard to reach")
    => **relation** -- (an abstraction belonging to or characteristic of two entities or parts together)
      => **abstraction** -- (a general concept formed by extracting common features from specific examples)

Fig. 4. "Superset" taxonomy for news item in the WordNet lexicon

It should be noted at once that except for trivial or generic cases roles do not appear in extensional lexicons such as WordNet or CYC (the only ones being is-a, generalizes, part-of, and very few others). And those that occur are not always defined in a semantically uniform way: inspecting even the one entry on **sports** in Fig.1

readily produces puzzling classification questions (why is rowing not an athletic game? Is it not also a water sport? Is water polo not a water sport, etc.). The same observation is true for nearly every thesaurus we have come across. These may not disturb a human user too much because of our additional cognitive and reasoning equipment, but seriously hamper the automation of semantics. Explicitly listing roles (for each entry, and often several, depending on the context γ) will at the very least assist classification in large lexon databases according to application or domain, but also document in a formally registered way the implicitly agreed meaning of the individual lexons seen as sentences about the ontology's universe of discourse.

We conjecture that so-called object-role models such as ORM [Hal95], a derivative of NIAM [VvB82] and originally devised as methodologies for defining database schemas from natural language sentences, will prove to be very useful to help construct ontologies. ORM and other models having roles as first class citizens also have a number of CASE tools supporting them such as InfoModeler™, now part of Visio Enterprise™ [VISIO]. The project reported in [ECS98] seems to corroborate this for the OSM method. The SORE experiment (for Simplistic Ontology with Role Extensions) to represent meaningful (small) domain thesauri, e.g. those released by [RosettaNet], using lexons in ORM has just been undertaken at the author's STARLab and will be the subject of a forthcoming report. Incidentally, the RosettaNet ontologies (actually thesauri) *are* developed according to a methodology, but fail at present to map their standardized term definitions (Property Specifications) to a "global" lexicon such as e.g. WordNet.

Very few IS methodologies that model domains rather than systems exist (a consequence of the reasons given in 4.2). One of those that do is KISS [Kri94], an elaborate approach involving processes, subjects, objects and actions. Some interesting lexical CASE tools have been under development for it [HHH97], also in passing in [BvR95], and it would be interesting to explore also the suitability of KISS for ontology construction. Another (representation) method is that of Sowa's Conceptual Graphs (e.g. [Sow99]) that seems particularly suitable for representing more complex domain thesauri to be built as logical theories (allowing inferencing). Perhaps the earliest concrete work in CLASS related topics is to be found in [BvR92], and in several related contributions in [MvR91].

Whichever the approach, one must devise a solution for the modularity problem: lexicons are quite large, and yet one must be able to "easily" add domain- and application-specific thesauri to them as the needs arise. Doing this manually in general would be foolhardy, and as e.g. using KIF quickly shows, it is non-trivial even with tools because of the many possible interactions if inferencing is required – another argument to keep ontologies simple. Even in terms of elementary lexons above, it is necessary to develop an "algebra of contexts" (union, "join", except, …) that allows the "plug-in" of a domain- or application-specific ontology.

5. Conclusion

The push by technological developments in distributed objects, by standardization of business objects in systems like SAP [K&T98] and by the emergence of large

extensional lexicons ([CYC], [WordNet]) combines with the pull of an XML-enhanced Internet and the needs of e-commerce to create a real need for ontology-assisted systems (CLASS). Such systems (when driven by suitably structured lexicons or thesauri) make possible in principle a clean treatment of information system semantics by substituting the ontological resources as "standard" conceptualizations of the real world.

Methodologies for cooperatively constructing large ontologies need to be designed; some techniques are already used experimentally in the [KIF] system. We argue however that a very simple architecture for lexicons will be essential for their success, combined with a simple way of implementing agreements among all types of designers, experts and users. Ontologies benefit from a layered architecture that allows specific domain-, application- and even application instance-specific thesauri to be added in such a way that the resulting linguistic resource remains as *application independent*, as possible, leaving the actual interpretation of IS concepts to the software agents that use it..

Literature References

[B&V97] Bench-Capon, T.J.M. and Vissers, P.R.S.: Ontologies in legal information systems; the need for explicit specifications of domain conceptualisations. In: Proceedings of the International Conference on AI and Law (ICAIL'97), Publ. ACM (1997).

[BDP99] Basili, R., Di Nanni, M., Pazienza, M.T.: Representing Document Content via an Object-Oriented Paradigm. In: Proceedings of the Eleventh International Symposium on Methodologies for Intelligent Systems ISMIS'99, Z.W. Ras and A. Skowron (eds.), Springer Verlag, Berlin (1999).

[BPS98] Bray, T., Paoli, J. and Sperberg-McQueen, C.M.: Extensible Markup Language (XML). World Wide Web Consortium (W3C). Available at www.w3.org/TR/1998/rec-xml-19980210.html (1998).

[Bro93] Brown, L. (ed.): The New Shorter Oxford Dictionary of the English Language. Clarendon Press, Oxford (1993).

[BvR92] Buitelaar, P., van de Riet, R.P.: The Use of a Lexicon to Interpret ER Diagrams. In: Proceedings of the 11th International Conference on the Entity-Relationship Approach, G. Pernul and A. Min Tjoa (eds.), IEEE Press (1992).

[BvR95] Burg, J.F.M. and van de Riet, R.P.: Color-X: Validating Linguistically Based Conceptual Models. In: Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing (KB&KS'95), N.J.I. Mars (ed.), IOS Press, Amsterdam (1995).

[DEF99] Decker, S., Erdmann, M., Fensel, D. and Studer, R.: ONTOBROKER: Ontology Based Access to Distributed and Semi-Structured Information. In: Database Semantics (DS-8), Semantic Issues in Multimedia Systems, R. Meersman, Z. Tari and S. Stevens (eds.), Kluwer Academic Publishers (1999).

[DMV88] DeTroyer, O., Meersman, R.A. and Verlinden, P.: RIDL* on the CRIS Case, a Workbench for NIAM. In: Computer Assistance during the Information Systems Life Cycle, , T.W. Olle, A. Verrijn-Stuart, and L. Bhabuta (eds), North-Holland, Amsterdam (1988).

[ECS98] Embley, D.W., Campbell, D.M., Smith, R.D.: Ontology-based Extraction and Structuring of Information from Data-Rich Unstructured Documents. In: Proceedings of the 7th ACM CIKM'98 Conference, Publ. ACM (1998).

[Emb98] Embley, D.W.: Object Database Development. Addison-Wesley (1998).

[Fel98] Fellbaum, C.(ed.): WordNet: An Electronic Lexical Database. MIT Press (1998).
[Fow97] Fowler, M.: Analysis Patterns: Reusable Object Models. Addison-Wesley, Reading MA (1997).
[G&F92] Genesereth, M.R. and Fikes, R.E.: Knowledge Interface Format Reference Manual. Stanford Computer Science Department Report (1992).
[G&N87] Genesereth, M.R. and Nilsson, N.J.: Logical Foundations of Artificial Intelligence. Morgan Kaufmann Publishers, Palo Alto CA (1987).
[Gru93] Gruber, T.R.: A Translation Approach to Portable Ontologies. J. on Knowledge Acquisition, Vol. 5(2), 199-220 (1993).
[K&T98] Keller, G. and Teufel, T.: SAP R/3 Process Oriented Implementation. Addison-Wesley, Reading MA (1998).
[Hal95] Halpin, T.: Conceptual Schema and Relational Database Design. Prentice-Hall (1995)
[HHL98] Heflin, J., Hendler, J. and Luke, S.: Reading Between the Lines: Using SHOE to Discover Implicit Knowledge from the Web. In: Proceedings of the AAAI-98 Workshop on AI and Information Integration (1998); from webpage accessed Feb.1999 www.cs.umd.edu/projects/plus/SHOE/shoe-aaai98.ps
[HHH97] Hoppenbrouwers, J., van den Heuvel, W.-J., Hoppenbrouwers, S., Weigand, H., De Troyer, O.: The Grammalizer: A CASE Tool based on Textual Analysis. Infolab Report, Tilburg University, from webpage accessed March 1999 at http://infolab.kub.nl/prj/past/gram/ (1997).
[ISO90] NN.: Concepts and Terminology of the Conceptual Schema and the Information Base. ISO Technical Report TR9007, ISO Geneva (1990).
[Kri94] Kristen, G.: The KISS Method for Object Orientation. Academic Service (1994).
[L&G94] Lenat, D.B. and Guha, R.V.: Ideas fr Applying CYC. Unpublished. Accessed Feb. 99 from www.cyc.com/tech-reports/ (1994)
[Mee97] Meersman, R.: An Essay on the Role and Evolution of Data(base) Semantics. In: Database Application Semantics, R. Meersman and L. Mark (eds.), Chapman & Hall, London (1997).
[Mee99] Meersman, R.: Semantic Ontology Tools in IS Design. In: Proceedings of the Eleventh International Symposium on Methodologies for Intelligent Systems (ISMIS'99), Z.W. Ras and A. Skowron (eds.), Springer Verlag, Berlin (1999).
[MvR91] Meersman, R. and van de Riet, R.P. (eds.): Linguistic Instruments in Knowledge Engineering: the LIKE Project", North-Holland (1991).
[NBN99] Nodine, M., Bohrer, W., Ngu, A.H.H.: Semantic Brokering over Dynamic Heterogeneous Data Sources in InfoSleuth. In: Proceedings of the International Conference on Data Engineering (ICDE'99), L. Maciaszek (ed.), IEEE Press (1999).
[OMG95] CORBA: Architecture and Specification v.2.0. OMG Publication (1995).
[Red97] Redmond, F.E. III: DCOM: Microsoft Distributed Component Object Model. IDG Books, Foster City CA (1997).
[Rei84] Reiter, R.: Towards a Logical Reconstruction of Relational Database Theory. In: On Conceptual Modeling: Perspectives from AI, Databases, and Programming Languages, M.L. Brodie, J. Mylopoulos, and J.W. Schmidt (eds.), Springer-Verlag, New York (1984).
[Sch86] Schmidt, D.A.,: Denotational Semantics: A Methodology for Language Development. Allyn & Bacon (1986)
[S&P98] Smith, H. and Poulter, K.: Share the Ontology in XML-Based Trading Architectures. In: CACM Vol.42(3), 110-111 (1998).
[Sow99] Sowa, J.F.: Knowledge Representation. Logical, Philosophical and Computational Foundations. PWS Publishing, Boston MA (1999) [in preparation].

[SPK96] Swartout, W., Patil, R., Knight K., Russ, T.: Towards Distributed Use of Large-Scale Ontologies. In: Proceedings of the 10th Knowledge Acquisition for Knowledge-Based Systems Workshop (KAW'96), (1996).

[UKM98] Uschold, M., King, M., Moralee, S. and Zorgios, Y.: The Enterprise Ontology. In: The Knowledge Engineering Review Vol.13(1), 31-89, Cambridge University Press, Cambridge (1998).

[VISIO] Infomodeler™, now part of Visio Enterprise™ Modeler, Visio Corp. <www.visio.com/>

[VvB82] Verheijen, 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).

Internet References to Cited Projects

[TREVI] The EU 4th Framework Esprit project TREVI, http://trevi.vub.ac.be/

[CYC] The CYC® Project and products, www.cyc.com/

[KIF] The Knowledge Interface Format (also as KSL, Ontolingua), http://ontolingua.stanford.edu

[OntoBroker] http://www.aifb.uni-karlsruhe.de/WBS/broker

[Ontology] http://www.ontology.org/

[RosettaNet] http://www.rosettanet.org/

[SHOE] The Simple HTML Ontology Extensions Project, www.cs.umd.edu/SHOE/

[WordNet] The WordNet Project, www.cogsci.princeton.edu/~wn/

[X-Act] http://x-act.org/