On Spatial Ontologies

STEFANO SPACCAPIETRA[1]
NADINE CULLOT[2], CHRISTINE PARENT[3], CHRISTELLE VANGENOT[1]

[1]Database Laboratory, Swiss Federal Institute of Technology, Lausanne, Switzerland
stefano.spaccapietra@epfl.ch, christelle.vangenot@epfl.ch

[2] LE2I Laboratory, University of Burgundy, Dijon, France
nadine.cullot@u-bourgogne.fr

[3]HEC-INFORGE, University of Lausanne, Switzerland
christine.parent@unil.ch

Abstract. The development of Internet technologies has prompted the emergence of the Semantic Web, whose aim is to deliver well-defined web resources and make them accessible by end-users and processes. Ontologies play a key-role to realize this ambitious goal. They are suitable means to model information with its meaning, thus enhancing chances of successful sharing. A large body of research has investigated ontology representation and reasoning, ontology engineering and the development of ontology applications. This has led to the specification of different models for ontology design and tools for ontology engineering. Spatial ontologies require specific models and tools to deal with the geographical aspects of the data. Unfortunately, little attention has been put up to now into spatial ontologies. This invited paper is intended as a brief introduction to the current state of art in ontologies, with a focus on spatial ontologies. It highlights strength and weakness of current research trends. It also proposes a specific approach based on our previous work on MADS, a spatio-temporal conceptual data model. The aim is to prepare the reader to enter this challenging domain.

1 Introduction[1]

Information sharing through semantic Web services is the new paradigm for the development of data intensive applications in the 21st century. Ontologies are expected to play a key role in this context. An ontology provides a representation of concepts of the real word, as agreed by a community of people. It serves as semantic reference for users or applications that accept to align their interpretation of the semantics of their data to the interpretation stored in the ontology. Different types of ontologies may exist, ranging from sophisticated dictionaries to rich conceptual and formal descriptions of concepts with their relationships and constraints.

A first generation of ontologies, and possibly the most frequently used at the moment, act as sophisticated dictionaries or thesauri. They focus on the definition of terms, and their organization into generalization/specialization hierarchies, enriched by semantic links commonly used in linguistics (e.g., synonymy, antonymy). We call these taxonomic ontologies. They define a reference vocabulary and are relatively easy to use. Wordnet[2] and currently EuroWordnet are representative taxonomic ontologies.

The current trend in ontological developments aims at supporting richer ontology models that enable sharing more complex information. Beyond the organization of terms to denote the concepts, these ontologies enrich the description of the semantics of concepts by associating to each concept a structured description of its properties. Users of these ontologies, that we call descriptive ontologies, can learn what are the appropriate terms to talk about the concepts and what are the kernel information structures to implement in their applications.

Descriptive ontologies share with conceptual database schemas the effort to model some domains or some activities. However, while ontologies have been traditionally considered as a means to explain the world, database schemas are also interested in supporting the management of data representing the world of interest. As a consequence, ontologies usually do not store instances of the concepts they describe (except for those instances that are useful in explaining the concepts), while databases are purposely designed to efficiently

[1] This work is supported, in the framework of the EPFL Centre for Global Computing, by the Swiss National Funding Agency OFES as part of the European projects KnowledgeWeb (FP6-507482) and DIP (FP6-507483). It is also supported by the MICS NCCR funded by FNRS in Switzerland, under grant number 5005-67322. UNIL contribution is additionally supported by OFES as part of the European project INTEROP.

[2] http://www.cogsci.princeton.edu/wn

handle huge amounts of instances. Ontologies with instances are often referred to as knowledge bases.

Geographical information is no exception to the above trend and search for effective interoperability. Geographical Web services have proliferated, most often to provide access to all sorts of maps covering all possible places on earth. While the move to semantic geographical Web services requires the same kind of ontological facilities as for other applications, research and development in ontology management has almost neglected to consider space and time as essential components of modern information systems. Thus, most of current ontologies do not take into account the spatial and temporal characteristics of information.

Space and time can meet ontologies in three distinct ways. First, they can be the domains described by the ontology. Ontologies of space define the concepts that are used in specifying space, spatial elements (e.g., point, line, polygon), spatial relationships such as topological relationships, and continuous fields. The specifications by ISO and the Open Geospatial Consortium[3], used to define GML (Geography Markup Language), are an example of ontology of space. Similarly, ontologies of time define the concepts that are used in specifying time and temporal elements, such as instant, interval, chronon, etc., and temporal relationships such as precedes, after, etc. Temporal SQL is one example of specifications that include ontological definitions about time.

Second, space and time can be the implicit background to an application domain that relies on geographical data. In this case we speak about ontologies of geographical domain. Electricity networks, transport networks, pollution and water control systems are examples of possible ontological domains. These application-oriented ontologies are similar to domain ontologies for other non-geographical domains.

Finally, space and time can be used to enrich the description of the concepts in the ontology, to represent their spatial and temporal localization, in the same way spatio-temporal data models support the description of spatial and temporal features in spatio-temporal databases. We use the term spatio-temporal ontologies to refer to ontologies that allow specifying the spatio-temporal characteristics of their concepts.

Ontologies of space and time are already streamlined for standardization. Several ontologies of geographical domains are being elaborated by the corresponding communities, if not worldwide at least at some more or less global level. On the contrary, spatio-temporal ontologies are in their infancy, in particular because of the lack of an appropriate model, capable of dealing with space and time at the ontological level, and of a suitable reasoning engine. Such a model could be borrowed from research results on spatio-temporal conceptual data models, developed by the database community. Indeed, conceptual models share with ontology models the concern for expressiveness and real world orientation (i.e., both avoid addressing implementation aspects). They provide for accurate structural descriptions, which, in our opinion, will boost the trend to move from taxonomic to descriptive ontologies.

This paper advocates that the best response to requirements for spatio-temporal ontologies will most likely come from the convergence between the ontological and conceptual modeling approach. To support this view, we first recall the characteristics of traditional ontological approaches. Section 2 surveys the variety of logic-based languages, which currently form the mainstream approach to ontology modeling, highlighting their advantages and disadvantages in terms of expressivity and reasoning. Section 3 overviews some representative and concrete proposals, based either on logic models or on conceptual data models, for ontology management systems. Section 4 discusses the main research directions from the ontology community to complement description logics with spatial facilities, a first step towards spatial ontologies. Section 5 presents what the conceptual modeling approach can provide to support spatio-temporal data modeling. Section 6 concludes in favor of a hybrid approach that would combine advantages from both research streams.

2 Logic-based models

Research on ontologies as an information management facility originated in the artificial intelligence and knowledge representation communities. This research is founded in logic as the basic theory to support reasoning. Currently, description logics (DL) appear as the leading formalisms for the development of ontology models and languages. However, other logics such as F-Logic (Frame Logic) and Horn-Logic may become serious challengers, in particular for the specification of hybrid solutions. This section briefly surveys and compares these three families of logics.

2.1 Description Logics

Description logics (Baader et al., 2002) are a family of knowledge representation languages that seem to be specifically tailored for ontology description as they focus on describing the semantics of concepts, and using inference to automatically classify new concepts in the concept hierarchy (based on the subsumption relationship) and to check non-contradiction among specifications. D. McGuiness (McGuiness, 2001) makes a good case for description logics, explaining their current emergence as ontology languages. Analyses of the evolution of description logics (initially called terminological logics), can be found in (Nardi et al., 2002) and in (Baader et al., 2004). Baader et al. show

[3] http://www.opengeospatial.org

that evolution from the first proposals as semantic networks to the current family of DL languages is driven by the search of the best compromise between expressive power, decidability and complexity of the reasoning algorithms. Changes in the techniques for the algorithms have prompted higher expressivity of the languages. This evolution has been accompanied by the development of several systems, from the early KL-ONE system (Brachman et al., 1985) to modern implementations such as FaCT (Horrocks, 1998) and RACE (Haarslev, 1999[a]).

DL languages vary in expressive power, depending on the building-operators that are retained for the language. Languages are named after the facilities they support (Baader et al., 2001). Examples hereinafter use $\mathcal{SHIQ}$ (Horrocks, 2000), an expressive DL implemented in the FaCT and RACER (Haarslev, 2001) systems.

In DL systems, a knowledge base consists of a Terminological Box, which describes conceptual knowledge in terms of concepts, roles and restrictions on them, and an Assertional Box, which holds knowledge about the instances. $\mathcal{SHIQ}$ also allows for inverse roles (noted R^{-1}), transitive roles and sub-roles.

In $\mathcal{SHIQ}$, concepts can be expressed using Boolean constructors (conjunction ($\cap$), disjunction ($\cup$) and negation ($\neg$)). $\mathcal{SHIQ}$ supports the existential and universal restriction constructors ($\exists$R.C and $\forall$R.C) that define the set of instances that are linked by role R to at least (or only to) instances of the C concept. $\mathcal{SHIQ}$ also supports the number restriction constructors ($\leq$nR, $\geq$nR, =nR), and the qualified number restriction constructors ($\leq$nR.C, $\geq$nR.C, =nR.C) that define the set of instances that are linked by at least (at most, exactly) n instances of the R role (to C instances).

For example, assuming knowledge about the organization of a scientific conference is to be defined, knowledge such as:

"A Committee has at least 10 members, which are persons."

is stated using a qualified number restriction constructor:

≥ 10 isCommitteeMember^{-1}.Person

where isCommitteeMember^{-1} is the inverse role of isCommitteeMember and Person is a concept.

Concepts are organized into a subsumption hierarchy, which includes two implicitly defined concepts: $\top$, the root concept in the hierarchy of concepts, and $\bot$, the sub-concept of all concepts. The concept hierarchy stems from two kinds of axioms: concept inclusion ($\sqsubseteq$), and concept equivalence ($\equiv$). For example, knowledge such as:

"An author is a person who has written at least one paper."

is defined in $\mathcal{SHIQ}$ by the following concept inclusion axiom:

Author $\sqsubseteq$ Person $\cap$ $\exists$ writes.Paper

where Paper and Person are either primitive or defined concepts, and writes is a role. The axiom implicitly defines Author as a sub-concept of Person.

Knowledge such as:

"The committee consists of the organization committee and the scientific committee."

is defined by the following concept equivalence axiom:

Committee $\equiv$ OrganizationCommittee $\cup$ ScientificCommittee

Important features that are not provided by $\mathcal{SHIQ}$ but are available in other DL languages are Concrete Domains and Nominals. Concrete Domains are a means to introduce concrete sets, such as the set of real numbers, integers or strings. For example, RACER adds to its $\mathcal{SHIQ}$ implementation two concrete domains: rationals and integers.

Nominals are a way to define singleton sets. For example, one could express:

"The most significant conferences are the conferences which have been chaired by Robinson."

SignificantConference $\equiv$ Conference $\cap$ $\exists$ chairedBy.Robinson

where Robinson is a nominal.

However, additional features, such as Concrete Domains and Nominals, significantly increase the reasoning complexity for DL languages.

2.2 F-Logic

F-Logic (Kifer, 1995) is an object-oriented logical rule language. An overview of the language may be found in (Angele et al., 2004), together with an excerpt of a case study for the German Telecoms and a discussion on the use of F-Logic for ontology modeling. F-Logic has been implemented, with some extensions, in systems such as Florid (Ludäscher et al., 1998), OntoBroker (Decker, 1999) and Flora (Yang and Kifer, 2000).

F-Logic allows defining classes of objects with mono or multi-valued methods. Notice that the term "method" is used in F-Logic to describe both the "attribute" and "method" of the object-oriented vocabulary. Methods can have parameters. Object classes are organized in a generalization/specialization hierarchy.

The following examples illustrate possible F-Logic definitions for the previous scientific conference example. They use the syntax specify in the Florid (F-Logic Reasoning in Databases) system which implements the basic features of F-Logic. The last version of Florid, called FloXML, provides access to XML documents. More details on the Florid system can found in (May, 2000).

"Definition of an object type person with a name, several first-names, and several addresses identified by their type (office, home, etc.); A person may be member of and may chair committees."

```
person   [ name => personName,
```

```
firstNames =>> personFirstname,
address@(type) => personAddress,
isMemberOf =>> committee,
chairs =>> committee ]
```

where => denotes a mono-valued method, =>> denotes a multi-valued method, and @ describes the parameter of the method. Methods return values (e.g. name, firstNames, and address) or object identifiers (e.g. isMemberOf and chairs)

"Definition of an is-a link between two object types."

chairman :: person

Rules are a crucial feature of F-Logic. They offer the possibility to derive information extending the object base intentionally. Rules are of the form "whenever the precondition is satisfied, the conclusion also is."

"A person who chairs a committee is a chairman."

The rule is stated as:

P:chairman :– P:person[chairs–>>C].

This rule expresses that if P is a person who chairs some committees C, then this person is a chairman.

"A person who chairs a committee of a conference which is co-located with another conference, also chairs this latter conference."

P:person[chairs–>>C2] :– P:person[chairs–>>C1],
C1:conference[isColocated–>>C2].

This rule derives new values for the method chairs. It expresses that if a person P chairs a conference C1 and if this conference is co-located with another one C2, then the person P also chairs C2.

2.3 Horn-Logic

Horn-Logic languages, such as Datalog and its extensions, are being considered as potential contributors to future ontology management systems. These languages are restrictions of first order logic to definite clauses. They allow the description of knowledge with predicates. Extensional knowledge is expressed as facts, while intentional knowledge is defined by rules. Rules can be recursive, which takes the expressive power of these languages beyond the limits of traditional database languages (relational algebra and calculus). Datalog (Abiteboul et al., 1995) has provided the theoretical foundation for most deductive database systems. Datalog is initially a Horn-Logic rule language without function symbol or negation. Later extensions include introducing complex objects and a restricted negation.

For example, let us assume the following predicates: pcMember (P,C) stating that the person P is a PC member of the conference C, and delegates (P1,P2) stating that the reviewer P1 has delegated a review to the person P2. Then the reviewers of a conference can be defined by the following predicates:

reviewer(P,C) :– pcMember(P,C).

reviewer(P,C) :– delegates(Px,P), reviewer(Px,C).

This latter rule illustrates the recursive feature of the rules defining the transitive closure of the predicate reviewer.

2.4 Suitability of Logic-based Approaches for Ontology Management

The three families of logic-based languages we just surveyed offer important features for building an ontology management system, but, in our opinion, none alone possesses all desirable features.

Description logics languages focus on supporting description of semantics as openly as possible. They allow writing specifications that may be inconsistent. In any real situation, such inconsistencies are likely to appear. Defining an ontology is a distributed task. First, it results from collaborative work, with input from many designers whose work is asynchronous and autonomous. Second, it is a never-ending task, as new specifications can arrive anytime. In such a context, relying on the designers to always provide correct specifications would be unsafe, and the only reasonable solution is to have specifications checked by the system. Inference engines associated to DL languages are tailored to check the consistency of the specifications. They do so applying the open world assumption (i.e., they assume that what is explicitly stated in the ontology is only a subset of what holds in the real world). This is perfectly consistent with the openness goal, recognizing that the available knowledge is an incomplete picture of the real world. For the same reasons, it is also very important that DL inference tools be able to automatically determine where new concepts are to be inserted into the subsumption hierarchy. Moreover, a number of theoretical results are available regarding decidability issues and the complexity of the reasoning algorithms (depending on supported functionality). This work is a preliminary to providing industrial strength systems.

Thanks to the above features, DL appears as an excellent candidate to support ontology systems. Unfortunately, DL strength is on the theoretical level. From a practical, ontology-engineering viewpoint, DL show important weaknesses. First, ontology modeling in DL is not at all an intuitive task. Representation of each single real world object is split into many axioms about concepts and roles, leading to an overall design that is very difficult to apprehend. This limits the possibility to have ontologies defined by humans. A second disadvantage of description logics concerns their limited power in terms of querying functionality. More expressive ontology query languages are needed. Finally, DL reasoners have important scalability problems, and it remains to be demonstrated that they can afford dealing with very large knowledge bases.

Although nowadays proposed as support to ontology management, the initial objective of both F-Logic and Horn-Logics is different. They target the ability to deduce new knowledge from existing knowledge. To that extent, they support rules that intentionally describe the new knowledge to be computed (as instances of derived concepts). Their inference tools are optimized for this knowledge deduction task. The possibility to state recursive rules enhances their expressive power. F-Logic adopts an object-oriented model, which allows defining objects in a much more intuitive form than description logics. Most traditional Horn-Logics, such as Datalog, adopt a relational data model.

F-Logic and Horn-Logics reasoners do not have to address consistency checking as the Horn-restriction guarantees that contradictory knowledge cannot be inferred. They do not deal with classification of concepts. Most of them use the closed world assumption (i.e., they assume that only what is explicitly stated holds). This is consistent with their knowledge management approach (similar to data management in DBMS, which routinely apply the closed world assumption). Being closer to the database approach, they have less scalability problems.

To sum up, DL approaches cope very well with the idea of supporting collaborative and asynchronous development of ontologies. F-Logic and Horn-Logics provide operational solutions that cope with the expected development of large knowledge bases. It is therefore not surprising that new research efforts are being undertaken to combine both approaches into a hybrid architecture that would benefit from the advantages of the different solutions.

3 Current Alternative Trends

As stated, logic-based languages currently form the mainstream approach to ontologies. Other approaches have emerged as realistic alternatives. To complete the reader's information, this section provides a short introduction to a few proposals that are in our opinion good representatives of the different trends. For a more detailed comparison of these trends, on the basis of identified ontology requirements, we refer the reader to (Cullot, 2003).

OWL[4] (Ontology Web Language) is the emerging standard for ontology languages. It stems from a synthesis of two previous standardization efforts, DAML, developed by the American DARPA project, and OIL, developed within an European research project, and is now supported by the W3 Consortium. OWL (Horrocks, 2002) consists of three languages: OWL lite, the simplest one, OWL DL and OWL Full. OWL-DL is a DL language with a markup syntax, whose semantics and foundation are an extension of $\mathcal{SHIQ}$, presented in the previous section, with nominals and concrete datatypes. Concrete datatypes available in OWL are a very restricted form of concrete domains. OWL Full has been developed to totally include the semantics of the previous RDFS (RDF Schema) standard.

A major alternative is offered by KAON (KArslruhe ONtology) (Motik et al, 2002), a representative proposal transferring a knowledge representation know-how into the ontology domain. KAON is an ontology and semantic web framework allowing the design and management of ontologies. It includes an ontology modeling language based on RDFS, with some proprietary extension and a conceptual query language. KAON supports modularization through the recursive definition of sub-models. Each sub-model has two components:

- An ontology structure, holding definitions of concepts, oriented binary relationships between concepts, and attributes. Relationships may be symmetric, transitive, and have an inverse. Minimum and maximum cardinality constraints for relationships and attributes may be specified. Concepts and relationships are organized as two distinct generalization hierarchies.

- An instance pool, holding instances of concepts and relationships, and attribute values. Specific to KAON is the possibility to have spanning objects, i.e. a real world entity being represented both as a concept and as an instance.

The third alternative trend relies on experience and results from the conceptual data modeling community. A major example is the DOGMA proposal (Jarrar et al., 2003), an ontology-engineering framework based on ORM (Object-Role-Modeling), a binary relationship conceptual data model. DOGMA splits the ontology into two parts:

- The ontology base, holding the data structure. Its definitions may be contextualized using a context name.

- A set of ontological commitments. A commitment is a set of integrity constraints (e.g., definition of identifiers, cardinalities) that govern the ontology for its use in a specific application.

The idea is that the ontology base holds generic knowledge about a domain, while its association to a commitment set specializes the ontology for a given application within the domain.

The use of a binary data model keeps DOGMA data structures quite similar to those of DL languages. To make ontologies more human-friendly, we have proposed an ontology approach based on MADS (Spaccapietra et al, 1999), an extended Entity-Relationship model. MADS offers another advantage; it supports specification of spatial and temporal features, which are ignored by OWL, KAON, and DOGMA.

[4] www.W3.org/TR/owl-features/

4 Spatial Ontologies

We focus hereinafter on spatial ontologies, as defined in the introduction. First, this section discusses the two research directions that have already addressed the issue of dealing with space in a DL environment.

4.1 Extending DL to Space

The first research direction represents a typical DL extension approach. The aim is to enrich DL with a concrete domain for the spatial dimension. The work by Haarslev (Haarslev et al., 1998; Haarslev, 1999[b]) represents this direction.

Haarslev defines a concrete domain for polygons, for use with the $\mathcal{ALCRRP(D)}$ description logics, thus allowing to combine knowledge representation and spatial reasoning into a unique paradigm. The Polygon concrete domain is defined with a set of predicates representing topological relationships between two objects: equal, disjoint, touching, strictly_overlapping, tangentially_contains, tangentially_inside, strictly_contains, and strictly_inside predicates. Their formal definition can be found in (Haaslev, 1997). These predicates are organised in a hierarchy from the most generic (a predicate named spatially_related) to the most specific ones. For example, the generally_inside (g_inside, in short) predicate is defined by:

g_inside ≡ t_inside ∪ s_inside ∪ equal

where t_inside and s_inside stand for tangentially_inside and strictly_inside. Similarily, the predicates g_contains and g_overlapping are defined.

Haarslev assumes that all objects are spatial and have an attribute hasArea defining its geometry, which is a polygon.

The topological predicates can be used to:

- define concept restrictions based on the hasArea attribute, using the generic concept-forming predicate operator (∃ f.P). For example, assuming Switzerland is an instance of the spatial concept Country, and Lake is another concept, the expression:

SwissLake ≡ Lake ∩ ∃ hasArea.g_inside(Switzerland)

specifies the defined concept SwissLake as the set of Lake instances whose area is inside the area of the Switzerland instance.

- define topological roles using the role-forming predicate restriction (∃(f)(f).P). For example, it is possible to define the role isTInside as:

isTInside ≡ ∃ (hasArea)(hasArea).t_inside

which means that any couple of instances whose extents satisfy the topological predicate t_inside are linked by the role isTInside. At this point, the expression:

EULake ≡ Lake ∩ ∃ isTInside.EUCountry

defines EULake as the set of Lake instances whose area is inside the area of an instance of EUCountry. Notice that this definition results in the fact that lakes overlapping two European countries will not be instances of EULake.

The reasoning mechanism is correspondingly extended to take into account the topological predicates. It allows reasoning on both terminological and topological specifications, taking into account the hierarchy of the topological predicates:

- The verification of the consistency of the specifications is extended to verify, for example, the satisfiability of the concepts defined using topological roles,

- The insertion of the spatial concepts in the subsumption hierarchy is performed according to their terminological and spatial definitions.

A complete presentation of this work (which includes similar extensions for time) can be found in (Haarslev et al., 1998; Haarslev, 1999[b]).

To sum up, Haarslev's proposal is well in line with other works offering DL extensions based on the definition of a concrete domain. The ability to define topological roles makes easier the specification of the spatial concepts and objects. The concrete domain provides access to spatial reasoning algorithms that allow the extension of the terminological reasoning to the spatial dimension.

However, further extensions would be necessary to make DL really practicable for GIS application. The description of the geometry of the objects cannot be restricted to polygons; a richer range of spatial types is necessary (point, line...). For knowledge bases (i.e., with instances) it would be also necessary to be able to compute spatial relationships from the geometry of the objects. Unfortunately, the extended DL still does not support values for the geometry of objects (hasArea values), hence does not support any geometry-related computation.

4.2 Combining a DL system with a GIS

The second research direction stems from the premise that DL-based systems are unlikely to scale up to efficiently handle large geographical knowledge bases. Wessel (Wessel 2003), for example, reports on a real experiment done with the DL reasoner RACER, using data (digital vector maps) from the local government of Hamburg. The experiment resulted in a huge ABox (containing all the topological relationships of the map), which proved to be quite prohibitive to query without the help of appropriate index structures. After different attempts to optimize the time to evaluate a query, for example by computing not all the topological relationships but only the necessary ones, Wessel comes to the conclusion that a practicable framework for geographical objects management would rather require a combination of different paradigms. He proposes a hybrid deductive Geographic Information System, with a DL-component (Wessel, 2002).

Wessel's proposal relies on a family of description logics called $\mathcal{ALCI}_{RCC}$ (discussed in (Wessel, 2002)), which are suitable for qualitative spatial reasoning and include the possibility to define topological roles (as in Harsleev's work). Based on this DL family, he proposes (Wessel, 2003) a hybrid representation and reasoning framework to support spatial and thematic representation of geographic data. This framework consists of three main components:

- An extensional component, which handles the instances composing the spatial database. Spatial objects are seen as having a geometric part, managed by a geometry-handling software (in this case, a proprietary GIS) based on a spatial closed domain assumption, and a thematic part, handled by RACER. A mapping managed by the framework links the two parts. The split into two parts, however, is not inherent to the approach and other representation alternatives are being considered.

- An intensional component, which offers intensional reasoning capabilities. This component is used to model the ontology, in a typical DL-like formalism whose inference tools classify the specified concepts according to the subsumption relationship.

- A query component, support queries that address both the geometric and the thematic part of objects. Queries may address either the extensional or the intensional components. Although Wessel proposes a hybrid language to support such queries, no specific query language is imposed by the framework, leaving open the question to find the most appropriate language for each component of the framework.

While Harsleev focuses on ontologies per se (without instances), Wessel addresses practical issues and adopts a software engineering perspective to develop a running prototype for a knowledge base management framework.

5 The MADS Approach

This section presents our own proposal for future ontology and knowledge base management systems. Our first interest is in providing a human-oriented approach for ontology modeling. We believe this is an essential pre-requisite for the expected development of semantic Web services. Conceptual data models have been purposely designed for efficient human-friendly interactions. They have proven they offer the best solution to the problem. We easily conclude that this know-how should be transferred to the ontology world.

To deal with spatio-temporal ontology modeling, we propose our MADS data model. This conceptual data model supports most of the desired functionality to deal with space and time. MADS modeling concepts support both the discrete and the continuous field view of space (and time). For the discrete view, MADS relies on a rich hierarchy of spatial data types (point, line, area, etc.), similar to the one proposed by the Open Geospatial Consortium, and temporal data types (instant, interval, etc.). Spatial and temporal extents can be associated to object types, relationship types, and attributes, as needed for an accurate design of the data structure. For the continuous field view, MADS supports the definition of attributes whose value is a function of space, time, or both space and time. Such attributes may be defined for any known extent (e.g., an extent associated to the whole database, or to an object, or to an attribute value). MADS also supports the definition of spatial and temporal relationships, such as topological and synchronization relationships. Finally, MADS has a unique feature: It supports multiple perceptions and multiple representation for the same real world objects. This allows simultaneous and integrated support of heterogeneous application requirements (e.g., applications having different thematic focus or different spatial resolution).

MADS, however, offers limited support for reasoning. It supports derived attributes, derived relationships, and to some extent derived objects. More features (e.g., symmetric and transitive cyclic relationships) have to be added to make MADS better prepared for ontologies.

Finally, MADS data modeling is supported by a CASE tool that automatically translates MADS specifications into GIS (e.g., ArcView, MapInfo) and DBMS (e.g., Oracle) compliant specifications. A complementary CASE tool supports visual querying for a MADS database.

6 Conclusion

Semantic Web services are the new and promising paradigm for information management. To be semantically efficient, they need ontology management services that are easy to use and capable of handling large ontologies and associated knowledge bases. For the former goal, we propose to reuse the conceptual modeling techniques already developed by the database community. For the latter goal, we propose to target a combination of a description logics inference tool with a data management tool such as a GIS or DBMS. In such an architecture, reasoning services will be performed by the inference tool, while management services will be left to well-established data handlers.

This paper discusses the rational for our proposal. It shows why we feel that none of the currently proposed approaches to ontology management, be they based on logic theories (namely, description logics, F-Logic, and Horn-Logics) or on knowledge representation formalisms, can alone cope with all the requirements for ontology modeling and management.

The paper presented in some more detail existing work to provide support for spatial ontologies. We have shown that two research directions are being investigated in this context, where one is more theory-oriented and the other one is more practice-oriented. Our proposal is in line with the latter.

Advances in spatio-temporal ontology techniques are definitely needed and will benefit all application domains. We hope this paper helps readers with little familiarity with this research area in understanding the basic issues and the corresponding body of research.

References

S. Abiteboul, R. Hull, V. Vianu: Foundation of Databases. Addison-Wesley, 1995

J. Angele, G. Lausen, Ontologies in F-Logic, in Handbook on ontologies, Int. Handbooks on Information Systems, S. Staab and R. Studer Editors, Springer, pp 29-50, 2004

F. Baader, D. Calvanese, D. McGuinness, D. Nardi, D., P. Patel-Schneider editors, The Description Logic Handbook: Theory, Implementation and Applications. Cambridge University Press, 2002

F. Baader, I. Horrocks, U. Sattler, Description Logics in Handbook on ontologies, Int. Handbooks on Information Systems, S. Staab and R. Studer Editors, Springer, pp 3-28, 2004

R.J. Brachman, J.G. Schmolze, An Overview of the KL-ONE knowledge representation system, Cognitive Science, Vol 9, pp 171-216, 1985

N. Cullot, C.Parent, S. Spaccapietra, C. Vangenot Ontologies: A contribution to the DL/DB debate, Proceedings of the 1st International Workshop on Semantic Web and Database (SWDB'2003), Co-located VLBD'2003, pp 109-130, 2003,

S. Decker, M. Erdmann, D. Fensel, R. Studer, OntoBroker™: Ontology based access to distributed and semi-distributed information. In R. Meersman at al., Database Semantics: Semantic Issues in Multimedia Systems, Kluwer Academic, 1999

V. Haarslev, R. Möller, SBox: A qualitative spatial reasoner – progress report. Proceeding of the 11th Int. Workshop on Qualitative Reasoning, L. Ironi Editor, pp 105-113, 1997

V. Haarslev, C. Lutz, R. Möller, Foundations of spatioterminological reasoning with description logics, Proceedings of the sixth Int. Conf. on Principles of Knowledge Representation and reasoning (KR'98), A.G. Cohn et al. eds, pp 112-123,1998

V. Haarslev, R. Möller, RACE System Description, Proceedings of Description Logic Workshop (DL'99), pp 130-132, 1999[a]

V. Haarslev V., A Logic-based Formalism for Reasoning about Visual Representations, Journal of Visual Languages and Computing, Vol.10, n°4, August 1999 [b]

V. Haarslev, R. Möller, RACER System Description, Proceedings of International Joint Conference on Automated Reasoning (IJCAR'2001), LNAI 2083, Springer-Verlag, pp 701-705, 2001

I. Horrocks, U. Sattler, S. Tobies, Practical Reasoning for expressive description logics. Journal of Interest Group in Pure and Applied Logic, 8(3), pp 239-264, 2000

I. Horrocks,"DAML+OIL : A reason-able Web Ontology Language, *EDBT 2002*, Jensen C., et al., Eds., Springer-Verlag, LNCS 2287, pp. 2-13, 2002.

M. Jarrar, J. Demey, R. Meersman, On Using Conceptual Modeling for Ontology Engineering, Journal On Data Semantics, S. Spaccapietra, Ed., Springer, LNCS 2800, pp. 185-207, 2003

M. Kifer, G. Lausen, J. Wu, Logical foundations od object-oriented and frame-based languages, Journal of ACM, 42, pp 741-843, 1995

W. May: How to write F-Logic Programs In Florid. A tutorial for the Database Language F-Logic, Version 3.0 FloXML, Institut für Informatik, Universität Freiburg, http://www.informatik.uni-freiburg.de/~dbis/florid/

D. McGuinness, Description Logics Emerge from Ivory Towers, Proceedings of the International Workshop on Description Logics, Stanford, CA, 2001.

B. Motik, A. Maedche, R. Volz., A Conceptual Modeling Approach for Semantic-Driven Enterprise Applications, Proceedings CooPIS/DOA/ODBASE, Meersman R., Tari Z., et al., Eds., Springer-Verlag, LNCS 2519, pp. 1082-1099, 2002

D. Nardi, J.B. Bachman, An Introduction to Description Logics, in Description Logic Handbook, Cambridge University Press, Baader F., et al., Eds., 2002

S. Spaccapietra, C. Parent, E. Zimanyi, Spatio-Temporal Conceptual Models: Data Structures + Space + Time, 7th ACM Symposium on Advances in Geographic Information Systems (ACM GIS'99), pp. 26-33, 1999

M. Wessel, On spatial reasoning with description logics – Position paper, Proceedings of the Int. Workshop on Description Logics (DL'2002), N° 53 in CEUR-WS, S. Tessaris et al. Eds, pp 156-163, 2002

M. Wessel, Some Practical Issues in Building a Hybrid Deductive Geographic Information System with a DL-Component, Proceedings of the 10th International Workshop on Knowledge Representation meets Databases 2003 (KRDB 2003), pp 15-16, 2003

G. Yang, M. Kifer, FLORA: Implementing an Efficient DOOD System Using a Tabling Logic Engine. Proceedings of Computational Logic, LNAI 1861, pp 1078-1093, 2000