Multirepresentation in ontologies

Djamal Benslimane[(1)], Christelle Vangenot[(2)], Catherine Roussey[(1)], and
Ahmed Arara[(1)]

[1] LIRIS, Université Lyon 1, IUT A 69622 Villeurbanne, France
{djamal.benslimane, catherine.roussey, ahmed.arara}@iuta.univ-lyon1.fr
[2] Swiss Federal Institute of Technology,
Database Laboratory, Lausanne, Switzerland
christelle.vangenot@epfl.ch

Abstract. The objective of this paper is to define an ontology language
to support multiple representations of ontologies. In our research, we fo-
cus on the logic-based ontology languages. As a matter of fact, we will
consider only languages that are based on description logics (DLs). At
first, we propose a sub-language of DL as an ontology language. Fur-
thermore we achieve multiple representations of ontological concepts by
extending such sub-language through the use of stamping mechanism
proposed in the context of multiple representation of spatial databases.
The proposed language should offer a modest solution to the problem of
multirepresentation ontologies.

1 Introduction

1.1 Motivation

Ontologies, as shared, common, representational vocabulary play a key role in
many artificial intelligence applications and information integration. They are
used as the basis for communication among multiple, heterogeneous systems.
Ontology representation is an important issue that has been investigated and
several languages are commonly used such as Ontolingua [10], KARL [3], OIL
[11], OCML [15].

There is no natural unique way to look at data and there is no natural unique
way to represent it. While the real world is supposed to be unique, its represen-
tation depends on the intended purpose: every representation of reality conveys
a user-specific representation of the real world. Thus, different applications that
share interest in the same real-word phenomena may have different perceptions
and therefore require different representations. Differences may arise in all facets
that make up a representation: what amount of information is kept, how it is
described, how it is organized (in terms of data structures), how it is coded,
what constraints, processes and rules apply, how it is presented, what are the
associated spatial and temporal frameworks, etc.

Most frequently, work on ontologies aims at developing a single-world ontology, i.e. an ontology that represents a given conceptualization of the real world from a given perspective. However, many researchers recognize that semantics is context-dependent: there is no single truth, and interpretation of concepts in particular depends on the context in which the concepts are used. A non-normative ontology should provide contextual definitions and contextual data structures to represent the diversity of perceptions and focuses. One example of interests where context is needed is in semantic interoperability. Ontologies play an essential role in semantic interoperation of information systems. Unfortunately, semantic interoperability is not easy to achieve because related knowledge is most likely to be described in different terms, using different assumptions and different data structures. Multi-representation ontologies can enable this integration. For instance, [9] introduces the concept of role to link the same concept described in several ontologies used to integrate information systems. Moreover, context may be used to define user profiles in ontologies and thus allows to select a subset of the ontology. For instance, in [13], context allows to filter results of multimedia database querying according to user profiles.

Domain ontologies are constructed by capturing a set of concepts and their links according to a given context. A context can be viewed as various criteria such as the abstraction paradigm, the granularity scale, interest of user communities, and the perception of ontology developer. So, the same domain can have more than one ontology, where each one of them is described in a particular context. We call each one of these ontologies a **MonoRepresentation ontology (MoRO)**. Thus, concepts in MoRO are defined with one and only one representation. Our motivation is to see how we can describe ontology according to several contexts at the same time. We shall call such ontology a **MultiRepresentation Ontologies (MuRO)**. A MuRO is an ontology that characterizes an ontological concept by a variable set of properties or attributes in several contexts. So, in a MuRO, a concept is defined once with several representations, such that a single representation is available for one context.

The multi-representation problem is commonly known in the discipline of spatial databases and object-oriented modeling. Spaccapiatra et al. [19] have investigated the problem in spatial databases to associate different geometry, resolution and scale with the same spatial object. In their work, they proposed an extension of existing ER-based model called MADS [17]. A stamping mechanism of data elements (concepts, attributes, instances) and relationships is suggested to enable manipulations of data elements from several representations. In object-oriented modeling, several object-role languages have been proposed to represent an object by a single structure which is augmented dynamically by specifc information (roles) related to its different facets [18, 14].

Existing ontology languages are not capable of defining a single MuRO ontology. For instance, there isn't any possibility to define a unique road concept

with different sets of properties at the same time. So, our motivation is to deal with the problem of multirepresentation in the context of ontology languages. In our research, we focus on the logic-based ontology languages. As a matter of fact, we will consider only languages that are based on description logics (DLs). DLs are a subset of first order logic used to describe knowledge in terms of concepts and roles to automatically derive classification taxonomies and provide reasoning services. Concepts in DL are intentionally described to specify the properties that individuals must satisfy in order for them to belong to the concept. DL-based ontology languages namely OWL [22], OIL, and DAML+OIL [21] are gaining support from many organisations and technical institutions because of their importance in data, information, and knowledge interoperation and exchange.

1.2 Contribution and organisation of the paper

Our objective is to define a contextual ontology language to support multiple representations of ontologies. The underlying key idea of our work is to adapt the stamping mechanism proposed in [19] to serve the needs of multirepresentation in spatial databases to our needs of ontology language requirements. To achieve this goal, we first propose a sub-language of DL as an ontology language and then we extend it by introducing stamping mechanism to the constructs of the DL language to allow multiple representations of concepts. The remainder of the paper is organised as follows. Section 2 presents the sub-language of DL that we propose as ontology language. Section 3 briefly summarises the multi-representation paradigm based on stamping mechanism. Section 4 is devoted to the presentation of some extension of DL to support multirepresentation needs. Section 5 presents some related work and finally, section 6 will conclude the paper.

2 Multirepresentation paradigm based on Stamping mechanism

A stamping mechanism to characterise database elements was proposed in [20] to support multiple representations of the same data. Stamps, used to characterise modeling contexts defined in a database, have a twofold semantic: they allow to distinguish the multiple representations of the same phenomenon and also to filter access to data during querying. Hence, a concept can be used in one or more context and each representation of a concept is stamped or labeled differently.

This proposal can be illustrated through use of an example. Let us consider two representations of the real world, identified by the stamps s_1 and s_2, corresponding to *Road traffic* and *Driving school* contexts.

The description of the stamped concept $Vehicle$ is as it follows:

Type $Vehicle$ (s_1,s_2)
s_1: Speed (1,1) : number,
s_1: VehicleType (1,1) : string,
s_1, s_2: LicencePlate (1,1) : string,
s_1, s_2: AssurancePolicy (1,1) : string,
s_1: RegistrationDate (1,1) : date,
s_2: RegistrationDate (1,1) : string,
s_1: Driver (1,1) : string,
s_2 : CarModel (1,1) : string

Stamps are written as arguments after type name of concept (i.e. type $Vehicle$ (s_1, s_2). A list of stamps is written before each atribute name. Stamping attributes allows for the definition of personalized concepts whose definition is varying according to stamps. Thus, in the context identified by s_1, the Vehicle concept is described by the attributes *Speed*, *VehicleType*, *LicencePlate*, *AssurancePolicy*, *RegistrationDate* (of domain Date) and *Driver*. In the context identified by s_2, it is described by the attributes *LicencePlate*, *AssurancePolicy* (as in context s_1), *RegistrationDate* but of a different attribute domain (string) and *CarModel*.

Stamps apply at the schema level (meta-data) thus allowing attributes to have:

- a single definition visible in a particular context, like for instance the attribute Speed,
- several definitions, i.e. different cardinalities or different domains of values according to the context, like the attribute RegistrationDate.

Stamps also apply at the data level and thus allow to partition the population of concepts into subpopulations having different stamps. It is useful when some instances of a multirepresentation concept are not shared. Stamping an instance allows to specify by which user an instance may be accessed: If instances corresponding to motor vehicles are stamped only with stamp s_2 and all other vehicles with stamps s_1 and s_2, users of the Driving school database have access only to motor vehicles. Like concepts, links that connect concepts can be stamped with the same semantics and according to the same rules.

The stamping technique is particularly of interest to deal with multiple representations of data as it allows:

- to define a representation context with a stamp that is a tuple of values corresponding to any number of meta-data items (viewpoint, resolution, time).
- to associate stamps with concepts, links, attributes, instances. Thus, the constructors of our model are orthogonal.

3 Adding multirepresentation mechanism to description logics

This section presents an extension of the description logics fragment given in annexe. This extension is able to describe the context varying aspect of concepts. Such concepts will be called contextual concepts, i.e. concepts with more than one representation, each representation is available in a given context.

3.1 Contextual constructors

In DL, we have the notions of concepts (that we denote classical concepts for more clarity) as a set of individuals (unary predicates) and roles (binary predicates) as attributes or relationships. Complex concepts are derived from atomic concepts by applying DL constructors (also will be denoted as classical constructors). For our requirements of multi-representation ontologies, we propose the notion of contextual concepts that may describe concepts associated with different contexts.

Contextual concepts are derived basically from atomic concepts by using a set of classical and/or contextual constructors. Contextual constructors are defined by specializing the classical ones to allow the construction of a concept that is partially or completly available in some contexts.

For the needs of multirepresentation, we add the following rule to the syntax defined in classical description logics language.

Definition 1. (Syntax of contextual concept terms) *Let $s_1, \cdots, s_m$ be a set of context names. Contextual concept terms C and D can be formed by means of the following syntax:*

$C, D \longrightarrow$
$\qquad \exists_{s_1, \cdots, s_m} R.C \mid$ (contextual existential quantification)
$\qquad \forall_{s_1, \cdots, s_m} R.C \mid$ (contextual value restriction)
$\qquad (\leq_{s_1, \cdots, s_m} nR) \mid$ (contextual at most number restriction)
$\qquad (\geq_{s_1, \cdots, s_m} nR) \mid$ (contextual at least number restriction)
$\qquad C \sqcap_{s_1, \cdots, m} D$ (contextual conjunction)

The definition of non-contextual concepts remains always possible. Such concepts will exist in all contexts with single representation.

The semantics of the non-contextual language is extended with the contextual notions. To define the semantics of the contextual constructors, we assume having a set of context names $S = \{s_1, s_2, ..., s_t\}$.

Definition 2. (Semantics of contextual concept terms) *The semantics of the contextual part of the language is given by a contextual interpretation defined in a context j over $\mathcal{S}$. A context j is either a simple context name belonging to*

$\mathcal{S}$ or a composed context defined as a conjunction of simple contexts[3]. A contextual interpretation $\mathcal{CI} = (\mathcal{I}_0, \mathcal{I}_1, \cdots, \mathcal{I}_j, \cdots, \mathcal{I}_t)$ is obtained by associating to each context j a non-contextual interpretation $\mathcal{I}_j = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}j})$, which consists of an interpretation domain $\Delta^{\mathcal{I}}$, and an interpretation function $\cdot^{\mathcal{I}j}$. The interpretation function $\cdot^{\mathcal{I}j}$ maps each atomic concept $A \in \mathcal{C}$ to a subset $A^{\mathcal{I}j} \subseteq \Delta^{\mathcal{I}}$ and each role name $R \in \mathcal{R}$ to a subset $R^{\mathcal{I}j} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. Let $Co(j)$ be a function which returns a set of all context names appearing in a simple/composed argument context j[4]. The extension of $\cdot^{\mathcal{I}j}$ to arbitrary concepts is inductively defined as follows:

$(\exists_{s_1, \cdots, s_m} R.C)^{\mathcal{I}j} = \{x \in \Delta^{\mathcal{I}} \mid \exists y : (x, y) \in R^{\mathcal{I}j} \wedge y \in C^{\mathcal{I}j} \wedge Co(j) \cap \{s_1, \cdots, s_m\} \neq \emptyset\}$
$(\forall_{s_1, \cdots, s_m} R.C)^{\mathcal{I}j} = \{x \in \Delta^{\mathcal{I}} \mid \forall y : (x, y) \in R^{\mathcal{I}j} \rightarrow y \in C^{\mathcal{I}j} \wedge Co(j) \cap \{s_1, \cdots, s_m\} \neq \emptyset\}$
$(\leq_{s_1, \cdots, s_m} nR)^{\mathcal{I}j} = \{x \in \Delta^{\mathcal{I}} \mid Co(j) \cap \{s_1, \cdots, s_m\} \neq \emptyset \wedge \|\{y \mid (x, y) \in R^{\mathcal{I}j}\}\| \leq n\}$
$(\geq_{s_1, \cdots, s_m} nR)^{\mathcal{I}j} = \{x \in \Delta^{\mathcal{I}} \mid Co(j) \cap \{s_1, \cdots, s_m\} \neq \emptyset \wedge \|\{y \mid (x, y) \in R^{\mathcal{I}j}\}\| \geq n\}$
$(C \sqcap_{s_1, \cdots, s_m} D)^{\mathcal{I}j} = \{x \in \Delta^{\mathcal{I}} \mid x \in C^{\mathcal{I}j}\}$ if $Co(j) \cap \{s_1, \cdots, s_m\} = \emptyset$,
$= \{x \in \Delta^{\mathcal{I}} \mid x \in C^{\mathcal{I}j} \cap D^{\mathcal{I}j}\}$ if $Co(j) \cap \{s_1, \cdots, s_m\} \neq \emptyset$

In the following, we briefly describe contextual constructors accompanied by simple examples.

The contextual value restriction constructor $(\forall_{s_1, \cdots, s_m} R.C)$: It will define a new concept all of whose instances are related via the role R only to the individuals of class C and in the contexts s_1 to s_m. For example, in two contexts s_1 and s_2, the concept Employee is defined as individuals with either an attribute EmployeeNumber in context s_1 or an attribute LastName in context s_2. The concept is expressed as it follows:

$$Employee = \forall_{s_1} EmployeeNumber.Number \sqcup \forall_{s_2} LastName.String$$

Outside of the two contexts s_1 and s_2, the concept $Employee$ corresponds to an empty set.

The contextual existential quantification constructor $(\exists_{s_1, \cdots, s_m} R.C)$ will construct a new concept all of whose instances are related via the role R to at least one individual of type C and only in contexts s_1 to s_m. For example, the following expression describes that student is an individual that has at least one graduation diploma in context s_1 and in the same time participates in at least one course in context s_2:

$$Student = \exists_{s_1} Diploma.Graduate \sqcap \exists_{s_2} Register.Course$$

It should be noted that the interpretation of the expression Student in the context s_1 or s_2 separately will give us an empty concept. On the contrary, the interpretation of Student in the context $(s_1 \wedge s_2)$ will correspond to some individuals satisfying the two conditions of the expression.

---

[3] $j = s_1$ and $j = s_1 \wedge s_2$ are two examples of simple context and composed context respectively.
[4] $Co(s_1 \wedge s_2) = \{s_1, s_2\}$ and $Co(s_1) = \{s_1\}$ are two examples of the function Co.

**The contextual number (at most, at least) restriction construc-
tors** ($\leq_{s_1,\cdots,s_m} nR, \geq_{s_1,\cdots,s_m} nR$): They specify the number of role-fillers. The
$\leq_{s_1,\cdots,s_m} nR$ is used to indicate the maximum cardinality whereas, the expres-
sion $\geq_{s_1,\cdots,s_m} nR$ indicates the minimum cardinality in the given contexts s_1
to s_m. The following example illustrates two cardinalities in different contexts:
context s_1 where a man is allowed for at most one wife at one period of time,
and context s_2 where he is allowed to have at most four wives at the same time.
In the two contexts s_1 and s_2, the type of $NumberOfWife$ is $Number$.

$$\text{Man} = (\leq_{s_1} 1Number0fWife \sqcup \leq_{s_2} 4Number0fWife) \sqcap$$
$$\forall_{s_1,s_2} NumberOfWife.Number$$

The contextual conjunction ($C \sqcap_{s_1,\cdots,s_m} D$): It will define either a con-
cept resulting of the conjunction of the two concepts C and D in the defined
contexts s_1 to s_m, or a concept equivalent to concept C outside of all the given
contexts (s_1 to s_m). For example, the following expression decsribes a Manager
as being a person of sex female and who either has at least one responsability in
the context s_1 or manages at leat one project in the context s_2. Outside of the
two contexts s_1 and s_2, Manager is only defined as being a person of sex female
and nothing else.

$$Manager = (Person \sqcap \forall Sex.Female) \sqcap_{s_1,s_2}$$
$$(\exists_{s_1} Responsability.String \sqcup \exists_{s_2} Manage.Project)$$

A terminological axiom $A = D$ is satisfied by a contextual interpretation $\mathcal{CI}$
if $A^{\mathcal{I}j} = D^{\mathcal{I}j}$ for every context j. A contextual interpretation $\mathcal{CI}$ is a model for
a $TBox$ $\mathcal{T}$ if and only if $\mathcal{CI}$ satisfies all the assertions in $\mathcal{T}$. Finally, a concept D
subsumes a concept C iff $C \sqsubseteq D$ is satisfied in every contextual interpretation $\mathcal{CI}$.
The contextual constructors proposed in this paper can be viewed as an adaptive
constructors that specialize the non-contextual ones. Thus, the subsumption for
the language will be decidable since it constitutes a subset of the description
logic $\mathcal{ALCNR}$ [7].

3.2   Example of multirepresentation ontology

Table 1 presents a very simple multirepresentation ontology described in our
language. Two contexts are considered: the traffic control and driving school
designated by s_1 and s_2 respectively.

4   Related work and discussion

In the following, a brief discussion of how this paper is related to other ap-
proaches that deal with the issue of modeling and reasoning about conceptual
schemas in various contexts.

Table 1. Multirepresentation Ontology example for Road Traffic and Driving School Contexts

---

Concepts defintion

Road= (($\leq_{s_1,s_2} 1Name$) $\sqcap$ ($\geq_{s_1,s_2} 1Name$) $\sqcap(\forall_{s_1,s_2} Name.String$)) $\sqcap$
((($\leq_{s_1} 1StartPoint$) $\sqcap$ ($\geq_{s_1} 1StartPoint$) $\sqcap(\forall_{s_1} StartPoint.Point$) $\sqcap$
($\leq_{s_1} 1EndPoint$) $\sqcap(\geq_{s_1} 1EndPoint$) $\sqcap$ ($\forall_{s_1} EndPoint.Point$) $\sqcap$
($\leq_{s_1} 1With$) $\sqcap$ ($\geq_{s_1} 1With$) $\sqcap$ ($\forall_{s_1} With.Number$) $\sqcap$
($\leq_{s_1} 1Lanes$) $\sqcap$ ($\geq_{s_1} nLanes$) $\sqcap$ ($\forall_{s_1} Lanes.Number$) $\sqcap$
($\leq_{s_1} 1SpeedLimitWith$) $\sqcap$ ($\geq_{s_1} 2SpeedLimit$) $\sqcap(\forall_{s_1} SpeedLimit.Number$)) $\sqcup$
(($\leq_{s_2} 1RoadType$) $\sqcap$ ($\geq_{s_2} 1RoadType$) $\sqcap(\forall_{s_2} RoadType.String$)))

Vehicle= (($\leq_{s_1,s_2} 1LicencePlate$) $\sqcap$ ($\geq_{s_1,s_2} 1LicensePlate$) $\sqcap$
($\forall_{s_1,s_2} LicensePlate.String$) $\sqcap$
($\leq_{s_1,s_2} 1InsurancePolicy$) $\sqcap$
($\geq_{s_1,s_2} 1InsurancePolicy$) $\sqcap(\forall_{s_1,s_2} InsurancePolicy.String$)) $\sqcap$
(($\leq_{s_1} 1Speed$) $\sqcap$ ($\geq_{s_1} 1Speed$) $\sqcap(\forall_{s_1} Speed.Number$) $\sqcap$
($\leq_{s_1} 1VehicleType$) $\sqcap(\geq_{s_1} 1VehicleType$) $\sqcap$ ($\forall_{s_1} VehicleType.String$)) $\sqcup$
(($\leq_{s_2} 1CarModel$) $\sqcap$ ($\geq_{s_2} 1CarModel$) $\sqcap(\forall_{s_2} CarModel.String$))

RoadSign=($\leq_{s_1} 1Type$) $\sqcap$ ($\geq_{s_1} 1Type$) $\sqcap(\forall_{s_1} Type.string$) $\sqcap$ ($\leq_{s_1} 1Location$) $\sqcap$
($\geq_{s_1} 1Location$) $\sqcap$ ($\forall_{s_1} Location.String$) $\sqcap(\leq_{s_1} 1Purpose$) $\sqcap$
($\geq_{s_1} nPurpose$) $\sqcap(\forall_{s_1} Purpose.String$)

Student= Person $\sqcap_{s_2}$ (($\leq_{s_2} 1Name$) $\sqcap$ ($\geq_{s_2} 1Name$) $\sqcap(\forall_{s_2} Name.String$) $\sqcap$
($\leq_{s_2} 1Birthdate$) $\sqcap(\geq_{s_2} 1Birthdate$) $\sqcap$ ($\forall_{s_2} Birthdate.Date$))

Instructor= Person $\sqcap_{s_2}$ (($\leq_{s_2} 1Name$) $\sqcap$ ($\geq_{s_2} 1Name$) $\sqcap(\forall_{s_2} Name.String$))

. . .

Modal description logics. The classical description logics are not intended to deal with the dynamic aspects of knowledge representation such as spatial, temporal, beliefs, obligations, etc [23]. Modal descriptions logics, however, are adequate to represent such notions. Modal description logics can be defined as description logics (DLs) augmented by some modal operators like (a) necessity to express that a concept holds in some given worlds (i.e. all worlds in the future), (b) possibility to express the existence of a world in which a concept holds. Different modal description logics have been proposed to deal with this dynamic aspect of concepts. As a matter of fact, we found out that temporal logics are the most suitable and closely related to our approach. In temporal description logics, some temporal operators (for example Until and Since) are introduced to define a concept in the past, present and future tense [4]. In a similar manner, we can consider the tense in general (past, present and future) as a characterisation of a context. According to such characterizing criteria, the temporal description

logics give us the evolution feature of the whole concept from one context to another. Hence, this work is different from ours in the sense that it does not give us the ability to explicitly designate context names of contexts as in the temporal case where a precise time is given.

RDF context. Resource Description Framework (RDF) is an emerging standard for the representation and exchange of metadata in the semantic web. RDF and RDF schema (RDFS) have been exploited in a variety of applications only to name few: resource discovery, cataloguing , digital libraries,etc. Recently, RDF(S) has become a good candidate for ontological knowledge representation. RDFS, as an ontology language, however, suffers from the power of expressivity and automated reasoning. A. Delteil and C. Faron-Zucker have proposed in [8] an extension of RDFS to go beyond the existing triple RDF statement- that is: the resource (subject), the property (predicate) and the value (object). They suggested a Defined Resource Description Framework (DRDFS) that enables the definition of class, property, and axiom definition to express contextual knowledge on the Web. The approach is based on the conceptual graph (CG) features. In this work, contextual annotations are proposed using extended RDF primitives namely *context*, *isContextOf*, and *referent*. The problem with this approach is that: it considers a context as a whole cluster of RDF statements (annotations). Moreover, rules based on the translation of CG graphs are needed to construct RDF contexts. In our approach, we consider a concept that may co-exist in several contexts but with variable sets of properties.

Topic Maps. Topic Maps (TMs) [1] are an emerging new tool used for organisation, management, and navigation of large and interconnected corpora of information resources. TMs collect key concepts of information sources and tie them together. The key concepts in topic maps are topics, associations, and occurrences. A topic is a resource within the computer that stands in for (or 'reifies') some real-world subject. An association is a link used to tie related topics. Occurrences are a set of information resources that may be linked to a topic. Topics can have base name, and variants of base names. This naming scheme allows the applicability of different names to be used in different specific contexts or scopes such as for multilingual representations. Topics have types expressed by class-instance relation (Rome is instance of a city). Topic types are also topics according to TMs standard (ISO/IEC 13250-2000). The notion of scope (the limit of validity) in TMs is one of the key distinguishing features of the topic maps paradigm; scope makes it possible for topic maps to incorporate diverse worldviews. In a given context, a topic has a specific scope. Different TMs can form different layers above the same information pool and provides us with different views of it. In fact, TMs capture contextual validity through the aspects of scope. Another filtering mechanism used in TMs is the facet which is used to assign metadata to the information resources. Thus, facets are based on the properties of information resources whereas scope is based on the properties of the topics.

Object-role. There has been a large amount of work in databases addressing specific facets of multi-representation. The first technique that one could think of is the well-known view mechanism. Views allow one to derive a new (virtual) representation from the representations already defined in the database. In relational DBMS the view mechanism is quite powerful (e.g., new, virtual relations may be built from any number of pre-existing relations). In object-oriented DBMS,however, views have less flexibility and are mostly restricted to simple selection views. Traditionally the generalisation link, which by definition links two representations (one more generic, one more specific) of the same real world entity, is supported in semantic and object-oriented approaches to describe several classifications of the same phenomenon. *Is-a links*, however, form static hierarchies that come with an associated inheritance mechanism, subtyping constraints and instantiation rules, that restrict modeling expressiveness. This restriction has prompted lots of works searching for a more flexible classification scheme. Those approaches assume that an analysis of the object life cycle determines the set of relevant possible representations for an object. These alternative representations, called roles, aspects or deputy objects, are described as types with specific rules, and more flexible inheritance mechanisms. These propositions allow, for instance, objects to move from one type to another or to acquire new instances as needed ([2, 12, 24, 5, 18]) or to be dynamically regrouped to form new classes, while keeping their membership in their original classes (e.g., [16]). Another solution, ([20]), inspired by the work in temporal databases proposes to describe stamped multi-representation types. Stamps allows to characterize the different representations and also to filter access to data.

5   Conclusion

In this paper, we have presented a solution to the problem of multirepresentation ontologies. In this respect, we have illustrated how an ontological concept may be represented in several facets/contexts according to its intended uses. A contextual ontology language is presented. It is based on description logics and stamping technique. Specific constructors are adapted from the existing ones to deal with the multirepresentation aspect.

As future work, we aim at finalizing and implementing the proposed constructs. Further, we intend to validate and test the proposed language in the domain of urbanism where we expect a wide range of contexts like transportation, land use, urban planing, etc. A link with existing standard ontology languages based on description logic (OIL, DAML+OIL) will be established in the future phase of our study. Finally, we intend to adapt the class-based logic query language [6] to deal with the need of browsing and searching concepts in large multirepresentation ontologies.

Appendix: Syntactical and semantics aspects of (non-contextual) description logics

Definition 3. (Syntax of concept terms) *Let $\mathcal{C}$ be a set of concept names, $\mathcal{R}$ be a set of role names and $n \in \mathcal{N}$. Non-contextual concept terms C and D can be formed by means of the following syntax:*

$$
\begin{aligned}
C, D \longrightarrow \quad & A \mid & & \text{(atomic concept)} \\
& \top \mid \bot \mid & & \text{(top, bottom)} \\
& C \sqcap D \mid & & \text{(conjunction)} \\
& C \sqcup D \mid & & \text{(disjunction)} \\
& \neg C \mid & & \text{(complement)} \\
& \exists R.C \mid & & \text{(existential quantification)} \\
& \forall R.C \mid & & \text{(value restriction)} \\
& (\leq nR) \mid & & \text{(at most number restriction)} \\
& (\geq nR) & & \text{(at least number restriction)}
\end{aligned}
$$

Definition 4. (Semantics) *The semantics of the language is given by an interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ which consists of an interpretation domain $\Delta^{\mathcal{I}}$, and an interpretation function $\cdot^{\mathcal{I}}$. The interpretation function $\cdot^{\mathcal{I}}$ maps each atomic concept $A \in \mathcal{C}$ to a subset $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ and each role name $R \in \mathcal{R}$ to a subset $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The extension of $\cdot^{\mathcal{I}}$ to arbitrary concepts is inductively defined as follows:*

$$
\begin{aligned}
\top^{\mathcal{I}} &= \Delta^{\mathcal{I}} \\
A^{\mathcal{I}} &\subseteq \top^{\mathcal{I}} \\
(C \sqcap D)^{\mathcal{I}} &= C^{\mathcal{I}} \cap D^{\mathcal{I}} \\
(C \sqcup D)^{\mathcal{I}} &= C^{\mathcal{I}} \cup D^{\mathcal{I}} \\
(\neg C)^{\mathcal{I}} &= \top^{\mathcal{I}} \setminus C^{\mathcal{I}} \\
(\exists R.C)^{\mathcal{I}} &= \{x \in \top^{\mathcal{I}} \mid \exists y : (x,y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\} \\
(\forall R.C)^{\mathcal{I}} &= \{x \in \top^{\mathcal{I}} \mid \forall y : (x,y) \in R^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}}\} \\
(\leq nR)^{\mathcal{I}} &= \{x \in \top^{\mathcal{I}} \mid \quad \|\{y \mid (x,y) \in R^{\mathcal{I}}\}\| \leq n\} \\
(\geq nR)^{\mathcal{I}} &= \{x \in \top^{\mathcal{I}} \mid \quad \|\{y \mid (x,y) \in R^{\mathcal{I}}\}\| \geq n\}
\end{aligned}
$$

Let A be a concept name and let D be a concept term. Then $A = D$ is a terminological axiom (also called defined-as axiom). A terminology ($TBox$) is a finite set $\mathcal{T}$ of terminological axioms with the additional restriction that no concept name appears more than once in the left hand side of the definition. An interpretation $\mathcal{I}$ is a model for a $TBox$ $\mathcal{T}$ if and only if $\mathcal{I}$ satisfies all the assertions in $\mathcal{T}$.

Definition 5. (Subsumption) *Let C, D be two concepts. D subsumes C (for short $C \sqsubseteq D$) iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for all interpretations $\mathcal{I}$. C is equivalent to D (for short $C \equiv D$) iff $C \sqsubseteq D$ and $D \sqsubseteq C$, i.e., $C^{\mathcal{I}} = D^{\mathcal{I}}$ for all interpretations $\mathcal{I}$.*

References

1. ISO 13250. Information technology - sgml applications - topic maps(. Technical report, International Organization for Standardization, ISO/IEC 13250, 1995.
2. Albano A, Ghelli G, and Orsini R. Fibonacci. A programming language for object databases. *Very Large Data Bases Journal*, 4(3):403–444, 1995.
3. J. Angel, D. Fensel, and R.Studer. The model of expertise in karl. In *Proceedings of the 2nd World Congress on Expert Systems, Lisbon,/Estoril,Portugal*, January 1994.
4. Alessandro Artale and Enrico Franconi. A temporal description logic for reasoning about actions and plans. *Journal of Artificial Intelligence Research (JAIR)*, 9:463–506, July 1998. December 1998.
5. Pernici B. Objects with roles. In *In Proceedings of ACM Conference on Office Information Systems, Cambridge, Massachusetts*, pages 205–215, 90.
6. Djamal Benslimane, Mohand-Said Hacid, Evimaria Terzi, and Farouk Toumani. A class-based logic language for ontologies. In *In the proceedings of the Fifth International Conference on Flexible Query Answering Systems. October 27 - 29, 2002, Copenhagen, Denmark. Proceedings. Lecture Notes in Computer Science 1864 Springer 2000, ISBN 3-540-67839-5*, pages 56–70, October 2002.
7. M. Buchheit, F.M. Donini, and A. Scharaef. Decidable reasoning in terminological knowledge representation systems. *Journal of AI Research*, 1:109–138, 1993.
8. A. Delteil and catherine Faron-Zucker. Extending rdf(s) with contextual and definitional knowledge. In *In Proceedings of the IJCAI Workshop on E-Business and the intelligent Web, Seattle, 2001.*, 2001.
9. Fonseca F., Egenhofer M., Davis C., and Câmara G. Semantic granularity in ontology-driven geographic information systems. *Annals of Mathematics and Artificial Intelligence*, 1-2(36):121–151, July 2002.
10. A. Farquhar, R. Fikes, and J.P. Rice. The ontolingua server: A tool for collaborative ontology construction. In *Journal of Human-Computer Studies, 46:*, pages 707–728, 1997.
11. D. Fensel, I. Horrocks, F. Van Harmelen, S. Decker, M. Erdmann, and M. Klein. Oil in a nutshell. In *In Proceedings of the ECAI-2000 Workshop on Applocations of Ontologies and Problem-Solving Methods, Berlin (Allemagne)*, 2000.
12. Gottlob G, Schrefl M, and Röck B. Extending object-oriented systems with roles. *In ACM Transactions on Information Systems*, 14(3):. 268–296, 1996.
13. M.R. Lee. Context-dependent information filtering. In *Proceedings of the International Symposium on Research, Development and Practice in Digital Libraries : ISDL'97 Conference*, November 1997.
14. O. LI and F.H. Lochovsky. Adome: An dvanced object modeling environment. *IEEE Transactions on Knowledge and Data Engineering*, 10(2):255–276, March/April 1998.
15. Enrico Motta. An overview of the ocml modelling language. In *In Proceedings of the 8th Workshop on Knowledge Engineering Methods and Languages (KEML'98), Karlsruhe, Germany*, pages 125–155, January 1998.
16. M.P. Papazoglou, B.J. Kramer, and A. Bouguettaya. On the representation of objects with polymorphic shape and behavior. In *In Proceedings. 13th International Conference on Entity-Relationship Approach, Manchester, UK*, pages 223–240, 94.
17. Christine Parent. Modeling spatial data in the mads conceptual model. In *Proceedings of the International Symposium on Spatial Data Handling,SDH98,Vancouver, Canada.*, July 1998.

18. J. Richardson and P. Schwartz. Aspects: extending objectsto support multiple, idependent roles. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, pages 298–307, 1991.
19. Stefano Spaccapietra, Christine Parent, and Christelle Vangenot. Gis databases: From multiscale to multirepresentation. In *4th Int. Symp., SARA 2000, Texas, USA. Proc. LNCS 1864 Springer.*, pages 57–70, July 2000.
20. C. Vangenot, C. Parent, and S. Spaccapietra. Supporting decision-making with alternative data representations. volume 5, 2001. Available : http://www.geodec.org/.
21. W3C. Daml+oil reference description. Technical report, W3C, http://www.w3.org/TR/2001/NOTE-daml+oil-reference-20011218, December 2001. W3C Note 18 December 2001.
22. W3C. Web ontology language (owl). Technical report, W3C, http://www.w3.org/TR/2002/WD-owl-ref-20021112/, November 2002. W3C Working Draft 12 November 2002.
23. F. Wolter and M. Zakharyaschev. Satisfiability problem in description logics with modal operators. In *In Proceedings of the 6th Int. Conf. on Knowledge Representation and Reasonning, KR'98, Trento, Italy, June 1998.*, pages 512–523, 2001.
24. Kambayashi Y and Peng Z. Object deputy model and its applications. In . *Proceedings of the Fourth International Conference on Database Systems for Advanced Applications, DASFAA'95, Singapore*, pages 1–15, 95.