



This is an author manuscript post-peer-reviewing (accepted version) of the original publication. The layout of the published version may differ .

Contextual Ontologies: Motivations, Challenges and Solutions

Benslimane, Djamal; Arara, Ahmed; Falquet, Gilles; Maamar, Zakaria; Thiran, Philippe; Gargouri, Faiez

How to cite

BENSLIMANE, Djamal et al. Contextual Ontologies: Motivations, Challenges and Solutions. In: Proceeding of the 4th Biennial International Conference on Advances in Information Systems - ADVIS 2006. Izmir (Turkey). Berlin : Springer, 2006. p. 168–176. (Lecture Notes in Computer Science (LNCS))
doi: 10.1007/11890393_18

This publication URL: <https://archive-ouverte.unige.ch/unige:46276>

Publication DOI: [10.1007/11890393_18](https://doi.org/10.1007/11890393_18)

Contextual Ontologies

- *Motivations, Challenges, and Solutions* -

Djamal Benslimane⁽¹⁾, Ahmed Arara⁽¹⁾, Gilles Falquet⁽²⁾
Zakaria Maamar⁽³⁾, Philippe Thiran⁽⁴⁾, and Faiez Gargouri⁽⁵⁾

¹ LIRIS Laboratory, Lyon 1 University, Villeurbanne, France
{djamal.benslimane,ahmed.arara}@liris.cnrs.fr

² University of Geneva, Geneva, Switzerland
Gilles.Falquet@cui.unige.ch

³ Zayed University, Dubai, U.A.E
Zakaria.Maamar@zu.ac.ae

⁴ University of namur, Namur, Belgium
pthiran@fundp.ac.be

⁵ ISIM, University of Sfax, Sfax, Tunisia
faiez.gargouri@fsegs.rnu.tn

Abstract. Contextual ontologies are ontologies that characterize a concept by a set of properties that vary according to context. Contextual ontologies are now crucial for users who intend to exchange information in a domain. Existing ontology languages are not capable of defining such type of ontologies. The objective of this paper is to formally define a contextual ontology language to support the development of contextual ontologies. In this paper, we use description logics as an ontology language and then we extend it by introducing a new contextual constructor.

Keywords. Context, Description logics, Ontology.

1 Introduction

Known as shared, common, representational vocabularies, ontologies play a key role in many information integration applications. They offer a basis for consistent communication among heterogeneous and autonomous systems [5,7,11]. Ontologies are now so large that users in the same domain with different interests cannot identify concepts relevant to their needs. In fact, existing ontologies [9,12,14] are context-free with respect to their concept representation and definition. Contexts appear in many disciplines as meta-information to characterize the specific situation of an entity, to describe a group of conceptual entities, and to partition a knowledge base into manageable sets or as a logical construct to facilitate reasoning services [3].

Domain ontologies are developed by capturing a set of concepts and their links according to a given context. A context can be seen from different perspectives. For instance it could be about abstraction level, granularity scale, interest of user communities, and perception of ontology developer. Therefore,

the same domain can have several ontologies, where each ontology is described in a particular context. We refer to this as **MonoContext Ontology (MoCO)**. Concepts in a MoCO are defined for one and only one context. The motivation of our research is to see how an ontology can be described according to several contexts at a time. We refer to this as **MultiContext Ontology (MuCO)**. A MuCO characterizes an ontological concept by a variable set of properties in several contexts. As a result, a concept is defined once with several representations while a single representation is available in one context.

Current ontology languages do not permit defining a single MuCO. They do not offer any possibility to hide or filter the ontology content. For example, it is not possible to hide the irrelevant *Road* concept from a user who is only interested in land coverage. Information on green areas and constructed areas are more relevant to him. The objective of this paper is to propose a contextual ontology language to support the development of Multi-Context ontology. We only consider languages that are based on description logics (DLs). DLs are a subset of first order-logic describing knowledge in terms of concepts and roles to automatically derive classification taxonomies, and provide reasoning services. Concepts in DL intentionally specify the properties that individuals must satisfy.

The rest of this paper is organized as follows. Section 2 gives the syntax and semantics of the proposed contextual language and presents some equivalence rules needed during syntax manipulations. Section 3 discusses the subsumption problem in this language. Section 4 surveys some works that are relevant to the issue of multiple viewpoints of ontologies. Finally, Section 5 concludes the paper.

2 Towards an enhanced version of description logics

For the purpose of our research on multiple context ontologies, we adopt the term contextual ontology to emphasize the importance of context in first, solving the multiple representation problem and second, providing a better visibility and access to ontological information elements (concepts, roles, individuals). The term contextual ontology is used as well to indicate that the ontology we deal with is context dependent. Therefore, a contextual ontology consists of two key words context and ontology. To meet the contextual ontologies' requirements, we propose the notion of contextual concepts. Contextual concepts are basically derived from atomic concepts by using a set of non-contextual and/or contextual constructors. To formally define a contextual concept, we propose adding a new constructor known as projection to the syntax given in Definition 1. This projection constructor is expressed in Definition 3. Definition 4 gives the new contextual interpretation of concepts.

2.1 Contextual constructors

Definition 1. Syntax of contextual concept terms *Let s_1, \dots, s_m be a set of context names. Contextual concept terms C can be formed according to the following syntax:*

$$\begin{aligned} C &\longrightarrow (C)[S] \text{ (contextual restriction)} \\ S &\longrightarrow \text{list of context names} \end{aligned}$$

The definition of non-contextual concepts remains always possible. Such concepts will exist in all contexts with a single representation. The semantics of a non-contextual language is extended with the contextual notion as per Definition 1.

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 contextual interpretation $\mathcal{I} = (\mathcal{I}_0, \mathcal{I}_1, \dots, \mathcal{I}_j, \dots, \mathcal{I}_t)$ is a t -tuple indexed by the contexts $\{1, \dots, t\}$ where each \mathcal{I}_j is a (non-contextual) interpretation $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}^j})$, 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 $A \in \mathcal{C}$ onto a subset $A^{\mathcal{I}^j} \subseteq \Delta^{\mathcal{I}}$ and each role name $R \in \mathcal{R}$ onto a subset $R^{\mathcal{I}^j} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$.*

The extension of $\cdot^{\mathcal{I}^j}$ to arbitrary concepts is inductively defined as follows:

$$\begin{aligned} \perp^{\mathcal{I}^j} &= \emptyset \\ \top^{\mathcal{I}^j} &= \Delta^{\mathcal{I}} \\ (C \sqcap D)^{\mathcal{I}^j} &= C^{\mathcal{I}^j} \sqcap D^{\mathcal{I}^j} \\ (C \sqcup D)^{\mathcal{I}^j} &= C^{\mathcal{I}^j} \sqcup D^{\mathcal{I}^j} \\ (\exists R.C)^{\mathcal{I}^j} &= \{x \in \Delta^{\mathcal{I}} \mid \exists y : (x, y) \in R^{\mathcal{I}^j} \wedge y \in C^{\mathcal{I}^j}\} \\ (\forall R.C)^{\mathcal{I}^j} &= \{x \in \Delta^{\mathcal{I}} \mid \forall y : (x, y) \in R^{\mathcal{I}^j} \rightarrow y \in C^{\mathcal{I}^j}\} \\ (\leq nR)^{\mathcal{I}^j} &= \{x \in \Delta^{\mathcal{I}} \mid \|\{y \mid (x, y) \in R^{\mathcal{I}^j}\}\| \leq n\} \\ (\geq nR)^{\mathcal{I}^j} &= \{x \in \Delta^{\mathcal{I}} \mid \|\{y \mid (x, y) \in R^{\mathcal{I}^j}\}\| \geq n\} \\ ((C)[S])^{\mathcal{I}^j} &= \begin{cases} C^{\mathcal{I}^j} & \text{if } j \in S \\ \emptyset & \text{otherwise} \end{cases} \end{aligned}$$

2.2 Examples

The following suggests some concept definitions in multiple contexts.

Example 1. An employee is defined in context s_1 as anyone who has an employee number and in context s_2 as anyone who works for a company.

$$Employee = (\exists EmployeeNumber.Number)[s_1] \sqcup (\exists WorksFor.Company)[s_2]$$

Example 2. In context s_1 a student is a person who is enrolled in at least one course, while in s_2 a student is a person who has an id-card.

$$Student = Person \sqcap ((\exists EnrolledIn.Course)[s_1] \sqcup (\exists Has.StudentIDCard)[s_2])$$

Example 3. In context s_1 a married man is a man who has exactly one wife, while in s_2 he may have up to 4 wives and in s_3 he may have an unlimited number of wives.

$$MarriedMan = Man \sqcap \exists wife.Woman \sqcap ((\leq 1wife)[s_1] \sqcup (\leq 4wife)[s_2] \sqcup (\top)[s_3])$$

The expression $\top[s_3]$ is interpreted as the whole domain $\Delta^{\mathcal{I}}$ in s_3 , which expresses the absence of number constraint on *wife* in s_3 .

2.3 Algebraic manipulations

It is straightforward to prove the following equivalences

$$\begin{aligned} C[s] \sqcup D[s] &\equiv (C \sqcup D)[s] \\ C[s] \sqcap D[s] &\equiv (C \sqcap D)[s] \\ \exists R.(C[s]) &\equiv (\exists R.C)[s] \\ C[s] &\equiv C \sqcap \top[s] \end{aligned}$$

For negations and universal quantifiers the rules are slightly more complex. In fact we have

$$\begin{aligned} (\neg C)[s] &= \neg C \sqcap \top[s] \equiv \neg(C[s]) \sqcap \top[s] \\ \neg(C[s]) &\equiv \top[\bar{s}] \sqcup (\neg C)[s] \\ (\forall R.C)[s] &\equiv (\forall R.C[s]) \sqcap \top[s] \\ \forall R.(C[s]) &\equiv (\forall R.C)[s] \sqcup \forall R.\perp \end{aligned}$$

where \bar{s} is the complement of the set of contexts s .

These equivalences can thus be used to shift the projection operator inside or outside expressions.

3 Subsumption in multiple contexts

In a contextual ontology it is necessary to redefine the notion of subsumption to take into account the challenging of managing different contexts.

Definition 3 (Contextual subsumption). *The contextual concept description D subsumes the contextual concept description C (written $C \sqsubseteq D$) iff for all contextual interpretations $\mathcal{I} = (\mathcal{I}_1, \dots, \mathcal{I}_t)$ $C^{\mathcal{I}^k} \subseteq D^{\mathcal{I}^k}$, $k = 1, \dots, t$.*

According to this definition, D subsumes C if for each interpretation and for each context, the interpretation of C is a subset of the interpretation of D .

Using the contextual restriction operator in an ontology with contexts $\{1, \dots, t\}$ the condition $C \sqsubseteq D$ is equivalent to $C[1] \sqsubseteq D[1]$ and \dots and $C[t] \sqsubseteq D[t]$.

Decidability of subsumption

It is possible to prove that a contextual subsumption is decidable by adapting and extending the classical tableau algorithm [1] for description logics. Note that this algorithm works only on concept descriptions in negative normal form, i.e. expressions where the negations occur only at the lowest level, just in front of the concept names. Thanks to the transformation rules of Section 2.3, any contextual concept description can be put into a negative normal form.

Let us start by defining the notion of contextual ABox⁶.

Definition 4. Let N_I be a set of individual names. A contextual ABox is a finite set of assertions in the form $C(a) : s$ (contextual concept assertion) or $r(a, b) : s$ (contextual role assertion), where C is a concept description, r a role name, s a context name, and a, b are individual names.

A contextual interpretation \mathcal{I} , which assigns elements $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ to the corresponding individual name a of N_I , is a model of an ABox \mathcal{A} iff $a^{\mathcal{I}} \in C^{\mathcal{I}^k}$ holds for all assertions $C(a)/k$ and $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}^k}$ holds for all assertion $r(a, b)/k$ in \mathcal{A} .

The classical tableau algorithm aims at constructing a model of a concept description C_0 . It does achieve this by starting with an initial (singleton) set of ABoxes $\hat{S} = \{\mathcal{A}_0\}$ where $\mathcal{A}_0 = \{C_0(x_0)\}$ and then exhaustively applying transformation rules. These rules either add new assertions to an ABox or create new ABoxes in \hat{S} . C_0 is satisfiable if and only if the set in \hat{S} of ABoxes obtained by this process contains at least one consistent ABox (without clash). To test if $C \sqsubseteq D$ amounts to prove that $C \sqcap \neg D$ is unsatisfiable.

In a contextual case, a concept description C_0 is satisfiable if there is at least one context k such that the tableau algorithm applied to $\hat{S} = \{\{C_0(x_0)/k\}\}$ yields one consistent ABox. The rules to apply are the same as those of [1] plus the following rule that deals with contextual restrictions.

Contextual restriction-rule

Condition: The ABox \mathcal{A} contains $C[s](x)/k$ but neither $C(x)/k$ nor $\perp(x)$.

Action: if $k \in s$ then $\mathcal{A}' := \mathcal{A} \cup \{C(x)/k\}$ else $\mathcal{A}' := \mathcal{A} \cup \{\perp(x)\}$.

(In the terms of [1] $\perp(x)$ should be expressed as $Q(x) \sqcap \neg Q(x)$, where Q is any arbitrary concept name.)

This rule creates a clash ($P(x)$ and $\neg P(x)$ in the same ABox) when the ABox contains a restriction $C[s](x)/k$ with $k \notin s$, which is clearly unsatisfiable, otherwise the restriction operator is dropped.

4 Related work and discussions

This section presents some of the works that are inline with developing multiple and/or contextual ontologies.

⁶ A knowledge base in a description logic system is made up of two components: (1) the *TBox* is a general schema concerning the classes of individuals to be represented, their general properties and mutual relationships; (2) the *ABox* contains a partial description of a particular situation, possibly using the concepts defined in the *TBox*. The *ABox* contains descriptions of (some) individuals of the situation, their properties and their interrelationships.

4.1 Distributed description logics

Distributed description logics (DDLs) are proposed to better present heterogeneous information in distributed systems by modeling the relations between objects and concepts of heterogeneous information systems [10]. Formal semantics of DDLs is proposed in [2]. Borgida and Serafini argue that there is no single global view of a real world but correspondences between different local conceptualizations should be provided through directed import feature and mapping. They suppose that there are binary relations r_{ij} and r_{ji} that describe the correspondences (at the instance level) between two ontologies O_i and O_j . A bridge rule concept is proposed to constrain these correspondences. A bridge rule from ontology O_i to ontology O_j is expressed in the following two forms:

$$\begin{aligned} o_i : C &\rightarrow \sqsubseteq o_j : D \\ o_i : C &\rightarrow \supseteq o_j : D \end{aligned}$$

These bridge rules allow concepts of an ontology to subsume a concept or to be subsumed by a concept of another ontology. These rules mean that the interpretation of C in O_i , once mapped onto O_j through r_{ij} , must be a subset (resp. a superset) of the interpretation of D in O_j .

Let us consider the example presented in [2] where a concept *Book_on_the_shelves* in ontology O_1 is defined to represent all the books that are not currently on loan in a given library. Assume the existence of the role *locate_at* in ontology O_2 which associates a book with a location on the shelves. To combine both ontologies in a distributed way, the following bridge rule can be defined: $O_1 : \textit{Book_on_shelf} \sqsubseteq O_2 : \exists \textit{located_at_}\{\textit{lyon_library}\}$. This bridge rule formalizes the fact that people know something is located in *lyon.library* only if it is a book that is not on loan there. DDLs have a solid logical ground and look very attractive to deal with multiple ontologies but coordination through mapping/bridge rules is necessary for any pair of ontologies that need to collaborate.

4.2 Contextualized Ontology ($C - OWL$)

In [4], Bouquet et al. consider that an ontology is built to be shared while a context is built to be kept local. To take advantage of both notions (ontology and context), they propose combining them in a unique framework. Thus, they propose the contextual ontology notion as an ontology with a local interpretation. This means that its contents is not shared with other multiple ontologies.

To cope with the semantic-heterogeneity problem, Bouquet et al. argue that imposing a single schema will always cause a loss of information. Their theoretical framework considers the following: (i) different conceptualizations provide a set of local ontologies that can be autonomously represented and managed, (ii) inter-relationships between contextualized ontologies can be discovered and represented, and (iii) the relationships between contextualized ontologies can be used to give semantic-based services and preserving their local "semantic identity". The OWL language is extended with respect to its syntax and semantics to meet the contextualized ontology's requirements. The new C-OWL language

is augmented with rules (or bridge rules) that relate (syntactically and semantically) concepts, roles, and individuals of different ontologies.

4.3 E-connections

E-connections is proposed in [8] as a formalism (i) to provide an expressive way for combining different heterogeneous logical formalisms such as description, modal, and epistemic logics, and (ii) to ensure the decidability and computational robustness of the combined formalism. The key idea in E-connections is to consider that the domains of the ontologies to combine are completely disjoint. The ontologies are then interconnected by defining new links between individuals belonging to distinct ontologies. For example, assume that O_1 and O_2 are two disjoint ontologies dealing with people and books respectively. The combination of both ontologies can be done by defining new links between individuals of O_1 and individuals of O_2 , and creating new concepts from the existing concepts of both ontologies. Hence, the link *buy* can be defined in O_1 to represent the fact that a person can buy books. In a similar way, the concept *FrequentBuyer* can be added to O_2 to define persons who buy at least one book is described as follows: $FrequentBuyer = Person \sqcap \exists buy.Book$. A framework is proposed in [6] to combine multiple, disjoint OWL ontologies. It is important to note that *E-connections* does not allow concepts to be subsumed by concepts of another ontology, which limits the expressivity of the language.

4.4 Modal logics

Modal logics [13] are a formalism for expressing dynamic aspects of knowledge such as beliefs, judgments, intuitions, obligations, time, actions, etc. In modal logics, the semantics of expressions or formula is defined in terms of things' trustworthiness in different worlds or contexts. This contrasts with the classical description logic, where things are just true or false, and not true in one context and false in another. The syntax of a modal description logics consists of the classical description logic constructs and the modal operators ($\Box_i C$, $\Diamond_i C$) known as necessity and possibility operators respectively. Modal concept C is defined as follows: $C \longrightarrow \Box_i C \mid \Diamond_i C$. For example, a faithful wife who loves her husband is expressed in classical DLs as: $faithfulWife = wife \sqcap \exists loves.husband$. If we would like to emphasize the fact that a faithful wife necessary loves her husband, which means that she always loves her husband, we need to express the same concept as: $faithfulWife = wife \sqcap \Box_i \exists loves.husband$.

4.5 Discussions

We highlight how our proposed approach is different from the aforementioned approaches. Both *DDLs*'s and *E-connections*'s objective is to preserve the independence of each local ontology. To work with multiple ontologies, they propose either a set of axioms (bridge rules) or *e-connections* concept in order

to establish interconnections between ontologies. Our approach differs from both approaches. Indeed, while *DDL* and *E – connections* are concerned with how to work and reason about multiple ontologies, our approach deals with how a single ontology is defined in a way that different perspectives are included. This means that with *DDL* and *E – connections*, different ontologies are available and represented in a classical way and then a new mechanism is added to link them. In our approach, we aim at creating a single ontology in which the definition of a concept includes the notion of multiple contexts in order to separate concepts from one context to another.

Our approach’s objective is rather closed to the ones targeted in the ontology views and modal description logics approaches. Our approach can be different from the ontology views approach by the fact that a view is extracted by querying the ontology content and this assumes that users have to master both a query language and the ontology content. With our approach, users only need to specify the current context to extract a sub-ontology from another one. A modal description logics approach allows modal interpretations of concepts. Our contextual interpretation is somehow a special case of modalities as well as temporal and spatial description logics. Hence, this work is different from ours in the sense that it does not give us the ability to explicitly designate context names.

5 Conclusion

In this paper, we argued that both ontology and context complement each other to achieve the goal of resolving partially the semantic heterogeneity in the scope of multiple context ontology where the same concept may need to be shared by more than one application. The notion of contextual ontologies approach was presented and formalized based on using the description logic language. Throughout this paper, we advocated the multi-representation rather than mono-representation of real world entities. Our rationale for the need of multi-context definition of concepts is: new requirements and information needs impose that many systems to coordinate, access shared entities of one another, and query autonomous, heterogeneous information sources. 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.

References

1. F. Baader and U. Sattler. An overview of tableau algorithms for description logics. *Studia Logica*, 69:5–40, 2001.
2. Alexander Borgida and Luciano Serafini. Distributed description logics: Directed domain correspondences in federated information sources. In Robert Meersman and Zahir Tari, editors, *CoopIS/DOA/ODBASE*, volume 2519 of *Lecture Notes in Computer Science*, pages 36–53. Springer, 2002.

3. A. Day and G. Abowd. The context toolkit: Aiding the development of context-aware applications. *Software Engineering for wearable and Pervasive computing*, 2000.
4. P. Pouquet et al. C-owl: Contextualizing ontologies. In *Proceedings of the 2nd International Semantic Web Conference (ISWC2003), 20-23 October 2003, Sundial Resort, Sanibel Island, Florida, USA.*, Oct 2003.
5. A. Gangemi, D.M. Pisanelli, , and G. Steve. An overview of the onion project: Applying ontologies to the integration of medical terminology. *Data and Knowledge Engineering*, 31, 1999.
6. Bernardo Cuenca Grau, Bijan Parsia, and Evren Sirin. Working with multiple ontologies on the semantic web. In Sheila A. McIlraith, Dimitris Plexousakis, and Frank van Harmelen, editors, *International Semantic Web Conference*, volume 3298 of *Lecture Notes in Computer Science*, pages 620–634. Springer, 2004.
7. J. Martins H.S. Pinto, A. Gomez-Prez. Some issues on ontology integration. In *Proceedings of IJCAI 1999 Workshop on Ontologies and Problem Solving Methods: Lessons Learned and Future trends*, pages 7.1–7.2, 1999.
8. O. Kutz, C. Lutz, F. Wolter, and M. Zakharyashev. E-connections of abstract description systems. *Artificial Intelligence*), 156(1):1–73, 2004.
9. Douglas B. Lenat. Cyc: A large-scale investment in knowledge infrastructure. *Commun. ACM*), 38(11):32–38, 1995.
10. Yinglong Ma and Jun Wei. A default extension to distributed description logics. In *IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'04)*, pages 38–44, 2004.
11. Thomas Strang, Claudia Linnhoff-Popien, and Korbinian Frank. Cool: A context ontology language to enable contextual interoperability. In Jean-Bernard Stefani, Isabelle M. Demeure, and Daniel Hagimont, editors, *DAIS*, volume 2893 of *Lecture Notes in Computer Science*, pages 236–247. Springer, 2003.
12. P.C.H. Wariyapola, N.M. Patrikalakis, S.L. Abrams, P. Elisseeff, A.R. Robinson, H. Schmidt, and K. Streitlien. Ontology and metadata creation for the poseidon distributed coastal zone management system. In *IEEE Forum on Research and Technology Advances in Digital Libraries*, pages 180–189, May 1999.
13. F. Wolter and M. Zakharyashev. Satisfiability problem in description logics with modal operators. In *In Proceedings of the 6th Int. Conf. on Knowledge Representation and Reasoning, KR'98, Trento, Italy, June 1998.*, pages 512–523, 2001.
14. Toshio Yokoi. The edr electronic dictionary. *Commun. ACM* 38(11), 38(44):42–44, 1995.