# Contextualization of a DL Knowledge Base

Krzysztof Goczyła, Aleksander Waloszek, Wojciech Waloszek

Gdańsk University of Technology, Department of Software Engineering,
ul. Gabriela Narutowicza 11/12, 80-952 Gdańsk, Poland
{kris,alwal,wowal}@eti.pg.gda.pl

**Abstract.** In the paper we propose a method of structuring a knowledge base into hierarchically related contexts and present how this arrangement influences the structure of TBox and ABox. We introduce a possibility to attach to a single TBox many ABoxes describing different parts of a domain and we show how to interpret such an ontology. Practical application of the method offers very interesting possibilities, like shortening the time of inference or storing mutually contradictory pieces of information in a single knowledge base. We analyse how such a structure changes purpose and mechanisms of reasoning, and we discuss their soundness and completeness. We also describe some related work about contexts.

## 1 Introduction

Contextualizing knowledge bases is an approach to make reasoning process more effective and to avoid inconsistencies in large ontologies. There are many types of relationships between contexts. In our approach we try to distinguish a group of relationships similar to inheritance. This kind of relation can be applied to both TBox and ABox. In case of TBox we separate axioms into groups that in the top of hierarchy define more general notions while passing down the hierarchy more specific and specialized concepts. In case of ABox every subset of assertions is attached to a particular context (i.e. a particular group of axioms). Moreover, this subset may also be divided into smaller groups called *context instances*. This subdivision limits the flow of conclusions and enables us to store inconsistent statements in one knowledge base.

The limited space in this paper does not allow us to present full spectrum of possible applications of a knowledge base organized according to our proposal. Our aim is to enable such tasks like modelling space-time situations, possibilities, believes or intentions, processes or even logical metalevels, i.e. such contexts where ABox contains description of ABox and TBox of another context. We realize that described method does not fulfil all of these requirements and it needs further development. But even now with its simple rules it has big potential of practical applications.

Section 2 presents formal definition of contextualized knowledge base. Section 3 contains description of reasoning problems in contextual knowledge bases. In Section 4 we try to review shortly another works on contexts. Section 5 summarizes the paper.

## 2 Formal definition of contextualized ontology

Our main goal was to introduce a kind of arrangement into large ontologies. We strive to allow for:

– introducing a hierarchical arrangement into large ontologies in order to describe various fragments of knowledge at different level of detail,
– holding contradictory assertions if they describe the same problem from different points of view,
– making it possible to integrate information from different points of view at a desired level of generality.

We propose a way of reaching these goals by introducing a notion of context into the knowledge base. First of all, we introduce contextualized TBox (contextualized terminology) that can be composed of several parts. These parts (being standard DL TBoxes), called *contexts*, remain with each other in a relation of generalization/specialization (inheritance, see Fig. 1 for an example).

**Definition 1.** *A* contextualized TBox $\mathbf{T} = (\{T_i\}_{i \in I}, \trianglelefteq)$ *consists of a set of TBoxes whose elements are called* contexts*, and a* generalization relation $\trianglelefteq \subseteq I \times I$ *which is a partial order established over the set of indexes* $I$*. The poset* $(I, \trianglelefteq)$ *is a tree containing the least element* $m$*. We also introduce the following notions:*

$T_m$ *called the* root context *of the contextualized TBox* $\mathbf{T}$*,*
$T_i$ generalizes $T_j$ *iff* $i \trianglelefteq j$*,*
$T_i$ specializes $T_j$ *iff* $j \trianglelefteq i$*.*

The idea behind such hierarchical arrangement of contexts was to allow for constrained interactions between parts of terminology. The general rule here is that more specialized terminologies may "see" more general ones, but more general terminologies are unaware of the existence of more specialized ones.

Introduced contexts encompass only terminology. To deal with assertional part of the knowledge base we allow for creation of many ABoxes for one terminology. We call these ABoxes *context instances*.

**Definition 2.** *A* contextualized ABox $\mathbf{A} = (\{A_j\}_{j \in J}, inst, \ll)$ *of contextualized TBox* $\mathbf{T} = (\{T_i\}_{i \in I}, \trianglelefteq)$ *is a triple consisting of:*

1. *A set of ABoxes* $\{A_j\}_{j \in J}$*, each of which is called an* instance of context*,*
2. *The function* $inst : J \to I$ *relating each ABox from* $\{A_j\}_{j \in J}$ *with TBox from* $\{T_i\}_{i \in I}$*,*
3. *The* aggregation relation $\ll \subseteq J \times J$*, which is a partial order established over the set of indexes* $J$*. We require that:*
   a. *The poset* $(J, \ll)$ *is a tree containing the least element* $n$*,*
   b. $inst(n) = m$*, where* $m$ *is the least element of the relation* $\trianglelefteq$*,*
   c. *For each* $j \ll k$ *such that* $j \neq k$ *holds* $inst(j) \trianglelefteq inst(k)$ *and* $inst(j) \neq inst(k)$*.*

*We also say that:*

$A_n$ *is called the* root context instance *of the contextualized ABox* $\mathbf{A}$*,*
$A_j$ *is an* instance of the context $T_i$ *iff* $inst(j) = i$*,*
$A_j$ aggregates $A_k$ *iff* $j \ll k$*,*
$A_j$ is aggregated by $A_k$ *iff* $k \ll j$*,*
$A_j$ *is an* aggregating context instance *iff* $\exists k : j \ll k$*.*

The idea of assigning several ABoxes to a single TBox (like in Fig. 1 where the context instances $A_7$, $A_8$, and $A_9$ are assigned to the context $T_5$) gives us distinctive opportunities: different ABoxes may contain different (consistent locally but possibly inconsistent with other ABoxes) sets of assertions.

It is worth noting that in a contextualized ABox a context instance aggregating all other context instances appears (in Fig. 1 it is $A_1$). The consequence of this fact is that all context instances have to be consistent with each other at the highest (defined in a contextual TBox) level of generality. This fact justifies calling the pair of contextualized TBox and ABox the contextualized knowledge base.

**Definition 3.** *A* contextualized knowledge base $\mathbf{K} = (\mathbf{T}, \mathbf{A})$ *consists of a contextualized TBox* $\mathbf{T}$ *and a contextualized ABox* $\mathbf{A}$ *of* $\mathbf{T}$*.*

Contextualized knowledge base is given the interpretation in a specific way: each context instance is given its own interpretation. Such an approach gives some level of locality within context instances.

**Definition 4.** *A* contextualized interpretation $\mathcal{I}$ *of contextualized knowledge base* $\mathbf{K} = (\mathbf{T}, \mathbf{A})$ *where* $\mathbf{T} = (\{T_i\}_{i \in I}, \trianglelefteq)$ *and* $\mathbf{A} = (\{A_j\}_{j \in J}, inst, \ll)$*, is a set of interpretations* $\{\mathcal{I}_j\}$ *where* $j \in J$*.*

The next definition specifies what conditions the local interpretations have to satisfy in order to make the global interpretation a model of a knowledge base.

**Definition 5.** *A contextualized interpretation* $\mathcal{I} = \{\mathcal{I}_j\}$ *of a contextual knowledge base* $\mathbf{K} = (\mathbf{T}, \mathbf{A})$ *where* $\mathbf{T} = (\{T_i\}_{i \in I}, \trianglelefteq)$ *and* $\mathbf{A} = (\{A_j\}_{j \in J}, inst, \ll)$*, is a model of the knowledge base* $\mathbf{K}$ *iff:*

1. *For every individual name* $a$*, there do not exist two interpretations* $\mathcal{I}_j, \mathcal{I}_k \in \mathcal{I}$ *such that* $a^{\mathcal{I}_j} \neq a^{\mathcal{I}_k}$*,*
2. *For every context instance* $A_j$*:*
   a. $\mathcal{I}_j \models A_j$*,*
   b. $\mathcal{I}_j \models \bigcup_{i \in \{i : i \trianglelefteq inst(j)\}} T_i$*,*
   c. *for every* $k$ *such that* $j \ll k$*:*
      i. $\Delta^{\mathcal{I}_k} \subseteq \Delta^{\mathcal{I}_j}$*,*
      ii. *for every concept* $C$ *from* $\bigcup_{i \in \{i : i \trianglelefteq inst(j)\}} T_i$*:* $C^{\mathcal{I}_j} \cap \Delta^{\mathcal{I}_k} = C^{\mathcal{I}_k}$*,*
      iii. *for every role* $R$ *from* $\bigcup_{i \in \{i : i \trianglelefteq inst(j)\}} T_i$*:* $R^{\mathcal{I}_j} \cap (\Delta^{\mathcal{I}_k} \times \Delta^{\mathcal{I}_k}) = R^{\mathcal{I}_k}$*.*

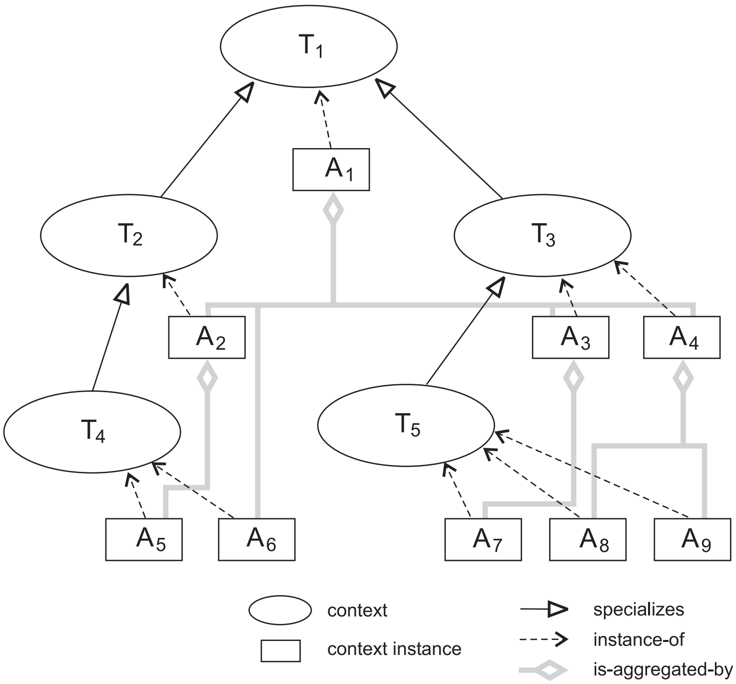


**Fig. 1.** An example of contextualized knowledge base. Relationships between context instances and between context instances and contexts are depicted in the form of graph, e.g. the instance $A_4$ aggregates instances $A_8$ and $A_9$. For the sake of clarity only the transitive reductions of generalization and aggregation relations have been depicted

The rules in the above definition may be divided into several categories. Rules 2.a and 2.b are called **local conformance** rules. They ensure that each interpretation satisfies the ABox and the TBox of the context instance it is assigned to and all TBoxes being its ancestors. An immediate corollary from this is the fact that terminologies that have any context instances assigned to cannot contradict any of their ancestors.

Other rules introduce the desired level of interaction between interpretations. Rule 1 is called **uniformity of names**. This rule was introduced to facilitate gathering pieces of information about one individual from various context instances (without necessity of defining mappings) during reasoning.

Rules 2.c (**aggregation conformance** rules) establish relations between aggregating context instance and context instances being aggregated. Rule 2.c.i introduces **aggregation conformance of domains** and states that the domain of the interpretation of the aggregating context must cover domains of interpretations of all context instances being aggregated.

Rules 2.c.ii and 2.c.iii establish **aggregation conformance of denotation**. They state that within the limited domain of the context instance $A_k$ being aggregated by $A_j$, at the level of generality of the terminology $T_i$ $(inst(j) = i)$,

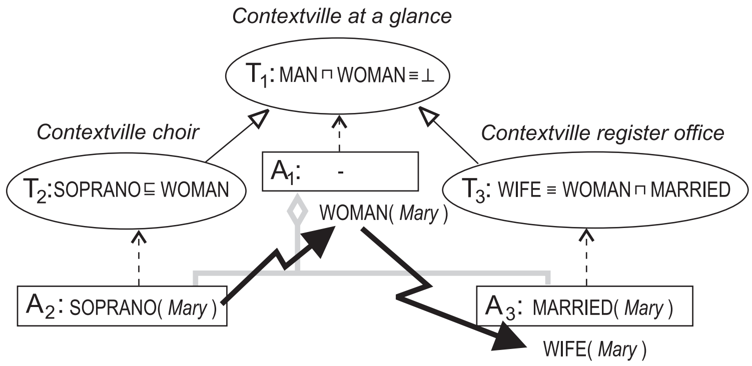


**Fig. 2.** An example of the aggregation conformance of denotation. Here we have three contexts: $T_1$ that describes general notions of `WOMAN` and `MAN`, $T_2$ that specializes $T_1$ towards description of voices in a choir, and $T_3$ that also specializes $T_1$ but towards description of social relations. Context instance $A_1$ aggregates context instances $A_2$ and $A_3$. Although $A_1$ is empty, according to the rule 2.c.ii, interpretation $\mathcal{I}_1$ in order to be a model of the knowledge base has to assign `Mary` to the concept `WOMAN` (i.e. $\mathtt{Mary}^{\mathcal{I}_1} \in \mathtt{WOMAN}^{\mathcal{I}_1}$). As a consequence of this, the same rule enforces that in the interpretation $\mathcal{I}_3$ `Mary` is assigned to the concept `WIFE` (i.e. $\mathtt{Mary}^{\mathcal{I}_3} \in \mathtt{WIFE}^{\mathcal{I}_3}$), as the information about `Mary` being a woman "flows" down the aggregation relationships

all the concepts and roles must be interpreted in $A_j$ in the same way (have the same extensions) as in $A_k$. These rules ensure flow of conclusions between aggregating context instance and the context instances being aggregated. The flow is bidirectional, as shown in the example from. Fig 2. An interpretation of $A_j$ must take into account all information from $A_k$, but due to the fact that is attached to a more general TBox this information must be reinterpreted in more general terms. This is also the way to avoid inconsistencies—to aggregate instances containing contradictory statements on the level of generality where the inconsistencies do not exist.

## 3 Reasoning in contextual knowledge bases

Reasoning in contextual knowledge bases is relevant to a single context instance.

**Definition 6.** *Entailment in contextual knowledge base* $\mathbf{K} = (\mathbf{T}, \mathbf{A})$ *where* $\mathbf{T} = (\{T_i\}_{i\in I}, \trianglelefteq)$ *and* $\mathbf{A} = (\{A_j\}_{j\in J}, inst, \ll)$*:*

1. $C \sqsubseteq D$ *is entailed by* $\mathbf{K}$ *in the context* $T_i$ *(denoted* $\mathbf{K} \models^i C \sqsubseteq D$*) iff for every contextual interpretation* $\mathcal{I} = \{\mathcal{I}_j\}_{j\in J}$ *that is a model of* $\mathbf{K}$ *for every* $j$ *such that* $inst(j) = i$ *it is true that* $\mathcal{I}_j \models C \sqsubseteq D$*,*
2. $C(a)$ *(and analogically* $R(a,b)$*) is entailed by* $\mathbf{K}$ *in the context instance* $A_j$ *(denoted* $\mathbf{K} \models^j C(a)$ *and* $\mathbf{K} \models^j R(a,b)$*, respectively) iff for every contextual interpretation* $\mathcal{I} = \{\mathcal{I}_j\}_{j\in J}$ *that is a model of* $\mathbf{K}$ *it is true that* $\mathcal{I}_j \models C(a)$ *(*$\mathcal{I}_j \models R(a,b)$*, respectively).*

To show the possibility of employment of known reasoning algorithms for contextual knowledge bases we will use a method similar to the one exploited by A. Borgida and L. Serafini in [1]. For the sake of brevity we assume that all contexts and context instances use the same Description Logics $\mathcal{DL}$. In [1] axioms and assertions included in different Information Systems ($\mathcal{IS}$s) are translated to appropriate statements in a single global knowledge base. We will use similar kind of translation to transfer the contents of a contextual knowledge base $\mathbf{K}$ to the global non-contextual knowledge base $K$.

We perform the translation on context instance-by-context instance basis. For each context instance $A_j$ we have to establish a kind of separate space allowing for interpretation of concepts and roles different than in its sibling context instances. If $A_j$ is aggregated by other context instance $A_g$ (assume that $g$ is a direct predecessor of $j$, denoted $g = \pi(j)$, i.e. $g \ll j, g \neq j$ and there is no $l$ such that $g \ll l \ll j$), the space must embrace concepts and roles from $T_i$ where $i \trianglelefteq inst(j)$ and $i \ntrianglelefteq inst(g)$. We denote such a set of indexes $\tau(j)$, and for the least element $n$ of $\ll$ we will assume that $\tau(n) = \{inst(n)\}$. Using this technique two assertions $\mathtt{DOCTOR}(\mathtt{John})$ and $\neg\mathtt{DOCTOR}(\mathtt{John})$ from two context instances $A_j$ and $A_k$ (if e.g. John is a doctor in Poland but not in Great Britain from the legal point of view) will be translated to $j : \mathtt{DOCTOR}(\mathtt{John})$ and $k : \top \sqcap \neg k : \mathtt{DOCTOR}(\mathtt{John})$, which will not generate inconsistency.

The mapping $\#(j, E)$ translating an expression $E$ describing a concept/role within the context instance $A_j$ (or the context $T_{inst(j)}$) in $\mathbf{K}$ to an appropriate expression in $K$ is defined as follows:

- $\#(j, \top) = j : \top$
- $\#(j, \bot) = \bot$
- $\#(j, A) = j : A$, for atomic concept $A$ introduced in $T_i$ such that $i \in \tau(j)$
- $\#(j, A) = j : \top \sqcap \#(\pi(j), A)$, for any other atomic concept $A$.
- $\#(j, R) = j : R$, for atomic role $R$ introduced in $T_i$ such that $i \in \tau(j)$,
- $\#(j, R) = \#(\pi(j), R)$, for any other atomic role $R$,
- $\#(j, \rho(E_1, E_2, \ldots, E_n)) = j : \top \sqcap \rho(\#(j, E_1), \#(j, E_2), \ldots, \#(j, E_n))$, for concept constructor $\rho$ taking $n$ arguments (*structural recursion*)

Now we can define rules of transferring axioms and assertions from $\mathbf{K}$ to $K$:
For each $j \in J$ do the following:

1. For each $C \sqsubseteq D$ included in $T_i$ such that $i \in \tau(j)$, insert to $K$:
   $\#(j, C) \sqsubseteq \#(j, D)$
2. For each atomic concept $A$ introduced in $T_i$ such that $i \in \tau(j)$, insert to $K$:
   $\#(j, A) \sqsubseteq j : \top$
3. For each atomic role $R$ introduced in $T_i$ such that $i \in \tau(j)$ restrict their domain and range in $K$:
   $\top \sqsubseteq \forall \#(j, R).j : \top$
   $\neg j : \top \sqsubseteq \forall \#(j, R).\bot$
4. If $j$ is the least element of $\ll$, insert to $K$:
   $\top \sqsubseteq j : \top$
   Otherwise, insert to $K$:
   $j : \top \sqsubseteq \pi(j) : \top$

5. Copy all assertions of $A_j$ to $K$ in the following form:
   $\#(j, C)(a)$ for each $C(a)$ included in $A_j$
   $\#(j, R)(a, b)$ for each $R(a, b)$ included in $A_j$

Following a very similar line of argumentation as the one in [1] we can show that for every $\mathcal{DL}$ in which every concept and role constructor is *local*[1] (e.g. $\mathcal{SHIQ}$) the following holds: $\mathbf{K} \models^i C \sqsubseteq D$ iff for every $j$ such that $inst(j) = i$ it is true that $K \models \#(j, C) \sqsubseteq \#(j, D)$. This result can be extended to ABox: $\mathbf{K} \models^j C(a)$ iff $K \models \#(j, C)(a)$ and $\mathbf{K} \models^j R(a, b)$ iff $K \models \#(j, R)(a, b)$.

The above discussion was intended to show that reasoning from contextual knowledge base is possible with use of existing tools. However the inference algorithm derived directly from the sketched method of translation may turn out to be inefficient. This is the reason why in practice, in the inference engine KaSeA [8] being implemented by our group, we use other technique of reasoning based on translating assertions from an aggregated context instance to the terms appropriate for the aggregating context instance. This task is similar to finding the most specific concept but within the constrained set of terms.

Besides reasoning problems discussed above the separation of ABoxes and TBoxes gives us a possibility of defining a class of novel inference problems, e.g.: "Find all context instances in which a given assertion is entailed by the contextualized knowledge base" or: "Given a set of context instances $\{A_j\}$, find the lowest level of generality (i.e. the most specific terminology $T_i$) at which they are not inconsistent (i.e. that there might exist a context instance $A_k$ aggregating all context instances $\{A_j\}$ with $inst(k) = i$ not making the contextualized knowledge base inconsistent)". Such problems might be interesting for Semantic Web communities focusing on integration of knowledge. More comprehensive set of similar problems and algorithms for solving them is under preparation.

## 4 Related work

Bouquet at al. in [2] divides the theories of context into two categories. The first category, called *divide-and-conquer (d-a-c)*, contains these theories which state that contextualization is a mean of partitioning a global theory of the world. The second category, called *compose-and-conquer (c-a-c)*, contains those ones which want to perceive a context as a local independent theory which can (but not has to) be integrated with another one with particular integration rules.

Local Model Semantics/Multi-context Systems (LMS/MCS) published in [5][7] form the theoretical basis for the c-a-c approach. There are several works [1][3][9] concerning ontology decomposition in the field of Description Logic. They are based on the foundation of *bridge rules*, a notion originally introduced in [6]. Bridge rules are descriptions of mappings between two portions of information. Although [1] does not use the notion of context (they are called *information*

[1] in practice it means all $\mathcal{DL}$s that do not have role constants and role constructors other than conjunction, disjunction, inverse, composition, role hierarchies and transitive roles; for the formal definition of locality see [1]

*sources* - $\mathcal{IS}$) it gives a method to describe data integration between ontologies. In [9] contexts are called *ontology modules* and bridge rules are replaced by *ontology-based queries*.

The theoretical basis for the d-a-c approach is the Propositional Logic of Context (PLC) introduced by McCarthy and formalized by Buvač and Mason [4]. A model $\mathfrak{M}$ for PLC defines a function, called the *vocabulary*, that associates formulae that are meaningful in a given context to this context. Contexts are arranged hierarchically. *Lifting axioms* play similar role as bridge rules in LMS/MCS.

Our work could be counted among those related with the d-a-c approach but is based on and develops division of propositions between TBox and ABox introduced by DL. By formulating some rules of relating contexts and their instances we intend to eliminate the necessity of defining a significant group of mappings.

## 5 Summary

In our paper we have shown our idea of contextualization of an ontology. We also have proposed an idea of context instances. Then we have described reasoning problems in such knowledge bases. Finally we have tried to place our approach among another work on ontology contexts.

## References


1. A.Borgida, L.Serafini. *Distributed description logics: Assimilating information from peer sources.* In Journal of Data Semantics, pp. 153–184, 2003.
2. P. Bouquet, Ch. Ghidini, F. Giunchiglia, E. Blanzieri. *Theories and uses of context in knowledge representation and reasoning*, Journal of Pragmatics, 35(3), pp. 455–484, 2003.
3. P. Bouquet, F. Giunchiglia, F. van Harmelen, L. Serafini, and H. Stuckenschmidt, C-OWL: Contextualizing Ontologies, Second International Semantic Web Conference (ISWC-2003), vol. 2870 of LNCS, pp. 164–179, Springer Verlag, 2003.
4. S. Buvač, I. A. Mason. *Metamathematics of Contexts.* Fundamenta Informaticae, 23(3), 1995.
5. C. Ghidini, F. Giunchiglia. *Local Model Semantics, or Contextual Reasoning = Locality + Compatibility.* Artificial Intelligence, 127(2):221–259, 2001.
6. F. Giunchiglia. *Contextual Reasoning.* Epistemologia, special issue on I Linguaggi e le Macchine, XVI, pp. 345–364, 1993.
7. F. Giunchiglia, L. Serafini. *Multilanguage hierarchical logics or: how we can do without modal logics.* Artificial Intelligence, 65(1):29–70, 1994.
8. K. Goczyła, T. Grabowska, W. Waloszek., M. Zawadzki: *The Knowledge Cartography A New Approach to Reasoning over Description Logics Ontologies.* SOFSEM 2006: Theory and Practice of Computer Science, vol. 3831 of LNCS, pp. 293–302, Springer, 2006.
9. H. Stuckenschmidt, M. Klein. *Integrity and Change in Modular Ontology.* In Proceedings of the International Joint Conference on Artificial Intelligence - IJCAI'03, pp. 900–905, Acapulco, 2003.