Specifying Ubiquitous Systems through the Algebra of Contextualized Ontologies

Isabel Cafezeiro[1], José Viterbo[2], Alexandre Rademaker[2],
Edward Hermann Haeusler[2], Markus Endler[2]

[1] Departamento de Ciência da Computação
Universidade Federal Fluminense
Niterói – Brasil
[2] Departamento de Informática
Pontifícia Universidade Católica do Rio de Janeiro
Rio de Janeiro – Brasil

Abstract. In this paper we present the algebra of contextualized ontologies and an approach to specify context-aware systems. The algebra is designed to support context moddeling and aims at the specification of modular and scalable description of arbitrarily complex systems. It takes contextualization as a basic notion and proposes a small set of simple and powerful operations to compose and decompose contextualized entities. The specification approach considers the gap between the formal specification and the real application and split the specification process in three levels variyng from the system design to the complete formalization using the algebra.

1 Introduction

Ubiquituous Computing is a new paradigm in which several sorts of computing devices - both small ones embedded into environments, as well as wireless portables - interact with each other to provide seamless services and information to mobile users. These ubiquitous services aim at supporting the users at their daily activities in a way that their presence becomes unnoticed, while their absence may cause some inconvenience. The paradigm of Ubiquitous Computing thus is inherently distributed, dynamic and involves heterogeneous systems and devices. Moreover, ubiquitous systems must respond dynamically to changes in the environment, with little, or without, human interference. Hence, these systems' behaviour must be context aware, i.e. several sorts of information about the context wherein they are supposed to operate are key for their proper execution.

Concerning context-awareness, several researchers [1, 2] have stressed the importance of precisely defining and modelling the central concepts of the paradigm, but unfortunately so far, most of them are informal and lack a solid, mathematical foundation. A formal treatment of context-awareness may contribute to the field by providing a sound understanding of concepts and methods, giving insights to design and implementation decisions, enabling some kind of formal verification at the early stages of design, suggesting new software development methods and tools for building correct and trustworthy systems, and finally, offering means of applying theoretical results to application-specific domains, yielding interesting and original results for concrete and applied problems. We present in this paper an algebra to formalize context modelling which has as central principles an *homogeneous* and *independent* description of entities and contexts, and a representation of their semantics through *net of relationship.*

By *homogeneous description* of entities and contexts we mean the adoption of a unique mechanism for representing knowledge, i.e. we use ontologies to describe both entities and contexts. The homogeneous description not only facilitates the uniform mapping between entities and contexts, but also gives the possibility to regard an entity either as a system component, or as an entire system. By *independent description* of entities and contexts we mean that, any component of a system, be it a device or an environment, is an independent element with its encapsulated attributes and state. However, the interfaces that connect them, i.e. the relation of an entity to its context is made explicit, what enhances the degree of modularity and reuse. By *semantics through net of relationships* we mean that, because of the homogeneous description of entities and contexts through ontologies, it is only through the semantic link between them that we define which ontology represents an entity (i.e., the domain of the link) and which represents the context (i.e., the codomain of the link), in each situation.

Since links compose associatively, one context can act as an entity of a different context. Thus this second context acts as a meta-context of the first entity. In addition, it is also possible for an entity to have several contexts (represented as several links with the same domain), or for a context to contextualize several entities (represented as several links with the same codomain). These structures can be extended arbitrarily, forming a net of entities and contexts that altogether define the semantics of a complex situation.

The algebra of contextualized ontologies was first proposed in [5] where basic concepts to support structural composition of contextualized ontologies were discussed. In [6] the applicability of the algebra was discussed through a scenario of ubiquitous computing. As a result of balancing the formal proposal and the experimental insights we present in this paper a revised version of the algebra with new algorithms and incorporating non structural constraints considering the basic description logic $\mathcal{ALC}$ [7]. Besides, in order to minimize the gap between the formal specification and the real application we propose in this paper a three layered approach to formalize a system where the top level focuses on the design of the ubiquitous system, the middle level describes the integration of components using the algebra operations and the lower level is the complete formalization in the algebra of contextualized ontologies. We focus on ubiquitous computing, although the generality of the framework that seems to be adequately applied to any situation where contexts and ontology are present.

This paper is organized as follows: in section 2 we present contextualized entities and the algebra of contextualized entities. In section 3 we consider a layered division of the algebra in order to facilitate the process of specification of systems where context information must be taken into account. We also present a scenario on ubiquitous systems and formalize it in the proposed approach. In section 4 we present algorithms to compute the operations, discuss complexities and present logical remarks about the approach. In section 5 we compare our approach with related work and section 6 concludes the paper.

2 The Algebra of Contextualized Ontologies

Contextualized ontologies are composed by three parts: an entity and a context, which are both ontologies, and a link between them. When referring to ubiquitous computing, we can think about the entity as being a computational device. The context can be thought of as the environment where the entity operates, which can either be a physical environment or another computational device. The link is the immersion of the entity into its context. Under this model, information concerning either physical or computational environment is treated as a relevant part of the application, deserving its own representation. This representation is self contained, what gives a high degree of mobility. The following definition is adapted from [18]:

Definition 1 (Ontology Structure) *An ontology structure is a tuple* (C, R, H^C, rel, A). *The components, in the same order that appear in the tuple, are: Concepts, Relations, which are disjoint sets.* $H^C \subseteq C \times C$ *is a hierarchy of concepts - a taxonomic relation.* $rel : R \rightarrow 2^{(C \times C)}$ *is a function that relates concepts non-taxonomically. Axioms specify other properties of concepts and relations. We assume that* H^C *is a partial order. By* $(x_1, x_0) \in H^C$ *we mean that* x_1 *is a subconcept of* x_0.

Usually, an ontology is a set of DL (description logic) sentences representing a theory – the knowledge base. From the semantic point of view, when this set of sentences is consistent, we have a model (here called an ontology structure). In an ontology structure $O = (C, R, H^C, rel, A)$ for a given ontology Ω, C and R are the set of non logical symbols that appear in the sentences of Ω (in C, the unnary symbols and in R, the n-ary symbols). H^C embodies the *is–a* relation of Ω, and rel is an extensional description of the non taxonomical relations. A is the set of sentences of Ω, as long as it is consistent. For any ontology structure O, there exists a set of corresponding valid sentences: the ontology Ω induced by the given ontology structure (of course, it is provided modulled ontology isomorphism). In this paper we emphasise structure, then, the relationships between ontologies will be given at the semantic level. It is clear that any semantic mapping induces a mapping between theories, as we discuss afterwards.

We define a link between ontology structures (semantic mapping) as:

Definition 2 (Link Between Ontology Structures) *A link between ontology structures is a triple $(f, g, h) : O \rightarrow O'$ where $O = (C, R, H^C, rel, A)$ and $O' = (C', R', H^{C'}, rel', A')$ are ontology structures, $f : C \rightarrow C'$, $g : R \rightarrow R'$ are functions such that (i) if $(c_1, c_2) \in H^C$ then $(f(c_1), f(c_2)) \in H^{C'}$, for $C_1, C_2 \in C$; (ii) if $(c_1, c_2) \in rel(r)$ then $(f(c_1), f(c_2)) \in rel'(g(r))$, for $c_1, c_2 \in C$ and $r \in R$; and $h : A \rightarrow A'$ exists if and only if (iii) for all $a \in A$ there exists $s' \in Th(A')$, where s' is the sentence that results from the translation of a to the vocabulary of O' by f and g.*

By condition *(i)*, links preserve hierarchy of concepts. By *(ii)*, links preserve relations. By *(iii)*, axioms of the domain ontology structure, when properly translated to the vocabulary of the codomain ontology structure, hold for the codomain. The process of translation is a canonical process over the structure of the sentence. We will denote by $trans_{fg}(a)$ the translation of an axiom a by f and g. By $Th(A')$ we mean the set of sentences that are provable from axioms A'. From *(iii)*, we see that a link between ontology structures induces a mapping between theories:

Let $O = (C, R, H^C, rel, A)$ be an ontology structure for Ω, and $(f, g, h) : O \rightarrow O'$ be a link between ontology structures, where $O' = (C', R', H^{C'}, rel', A')$. Then, by (iii) of definition 2, if $a \in A$ then $A' \vdash tranfs_{fg}(a)$. O' induces an ontology Ω' whose axioms are A' and the translation of the axiomatization of H^C. Then there is a syntactical morphism between Ω and Ω': for all $a \in \Omega$, $\Omega' \vdash trans_{fg}(a)$.

As we assume the semantic approach we use, along this paper, the term *ontology* to refer to ontology structure. Only in section 4.1, where logical remarks make necessary the reference to both syntactic and semantic, we will use the terms *ontology* and *ontology structur*e to distinguish syntactic from semantic. We remark that, to not increase the complexity of worst case, we opt for not distinguishing individuals, that are, thus, viewed as concepts.

Definition 3 (Contextualized Ontologies) *A contextualized ontology is a triple (e, l, c), also represented by* $e \xrightarrow{l} c$*, where l is a link between the ontologies e (domain ontology) and c, codomain ontology.*

2.1 Operations on Contextualized Ontologies

The algebra of contextualized ontologies is formed by two set of operations devoted to compose and decompose contextualized ontologies in several ways. The dual aspect between actions of composing and decomposing is considered. In the formal sense, this duality takes the form of reversing arrows, and is clearly noted by abstracting the role of an ontology as entity or context. It is sufficient to define one of the sets of operations, and then, by duality, to deduce the other. This homogeneity in the definition is also reflected in the algorithms that compute the algebra operations, what contributes a lot to the simplification of the framework as a whole. A unique algorithm can be taken as reference from which all others are derived.

The operations have the general purpose of manipulating information of ontologies in order to produce a coherent and concise body of information over which a particular device may operate. For this, align and coalign play the role of establishing binary relations between contexts or between entities. Context integration, collapsed union, entity integration and relative intersection are devoted to produce new entities, contexts or contextualized entities according to the related information. The composition operations are Alignment, Context Integration and Collapsed Union. Decomposition operations are Coalignment, Entity Integration and Relative Intersection.

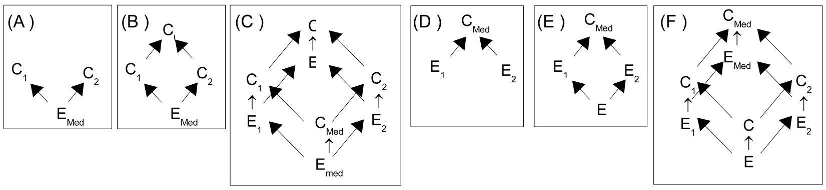

Fig. 1. (A) Alignment (B) Context Integration (C) Collapsed Union (D) Coalignment (E) Entity Integration (F) Relative Intersection

Alignment.(figure 1-A) A situation where an entity has more than one context is an alignment: $C_1 \leftarrow E_{Med} \rightarrow C_2$. By defining a binary relation, the alignment makes possible the partial mapping between contexts. This feature makes possible to deal with situations where a concept of a context does not make sense in the other context. On the other hand, the entity must be totally mapped on both contexts: all concepts of the entity must be understood in both contexts.

Context Integration. (figure 1-B) In an alignment $C_1 \leftarrow E_{Med} \rightarrow C_2$, the context integration produces a new context C, to which C_1 and C_2 are linked: $C_1 \rightarrow C \leftarrow C_2$. This new context combines information of C_1 and C_2 preserving the coherence with the entity E_{Med}. The integration performs the amalgamated union of contexts, collapsing components that are related by the alignment $C_1 \leftarrow E_{Med} \rightarrow C_2$. The result contains all information of the original contexts, but identifies parts related by the mediator entity.

Collapsed Union. (figure 1-C) Is the amalgamated union of two contextualized ontologies mediated by a third contextualized ontology. It is the combined composition of entities and contexts, where the ontology links ensures the preservation of structure, relations, and axioms of each ontology and coherence of each entity with respect to contexts. It produces a new contextualized ontology with all components of the original ones, but collapsing components that have the same source in the mediator.

Coalignment. (figure 1-D) It is a mechanism of establishing a correspondence between vocabularies of two ontologies by the use of an intermediate target ontology: $E_1 \rightarrow C_{Med} \leftarrow E_2$. It configures a binary relation between the two ontologies, where related components are those that are mapped in the same component of the intermediate target. By defining a binary relation, the coalignment makes possible the partial mapping between entities, what means that not all concepts of one entity make sense for the other entity. Both entities, however, must be totally mapped in the context.

Entity Integration. (figure 1-E) Given a coalignment $E_1 \rightarrow C_{Med} \leftarrow E_2$, the entity integration produces a new entity E contextualized by the original ones (and by transitivity, by the original context C_{Med}). The entity integration performs the semantic intersection of the entities under the mediation of the context, that is, the new entity will embody all, and nothing more than, information of the original entities that are related by the coalignment $E_1 \rightarrow C_{Med} \leftarrow E_2$.

Relative Intersection. (figure 1-F) Is the intersection of two contextualized ontologies mediated by a third contextualized ontology. It is the combined intersection of entities and contexts, where the ontology links ensure the preservation of structure, relations, and axioms of each ontology and coherence of each entity with respect to its context. It produces a new contextualized ontology having just the components of the originals that are mapped in the mediator.

3 Specifying Context-Aware Applications

We use three kinds of diagrams to make possibe dealing with different levels of abstractions at the same time.

(i) **Diagram of Devices:** specify all computational devices (pieces of software/hardware) that compose the ubiquitous system and the kind of information that is exchanged among them. Each device of a ubiquitous system is a node of the diagram, which is linked to other nodes by a directed arrow labelled with the information that is to be transmited from a device to another.

(ii) **Diagram of Entities.** An *entity* is any component of a system that produces a reaction by devices. For example, a Professor, that must be identified when arriving at work. Each device of the previous diagram is related to a diagram of entities. Through the algebra of contextualized ontologies the entities of this diagram can be operated generating new entities wherein the devices can extract the necessary information to work on.

(iii) **Diagram of Ontologies.** The diagram of ontologies provides a detailed view of the diagram of entities, where each entity is described by an ontology and the links are consistently defined in the algebra of contextualized entities.

These three diagrams are related in the following way: each device of the diagram of devices correspond to a diagram of entities. In the diagram of entities, details of entities and their connection are hidden. These details appear in the diagram of ontologies, where entities specifications and their connections are sufficiently detailed in order to make possible consistency tests.

3.1 A simple scenario

An extended version of this scenario is presented and formalized in [6]. In this paper we present a subset of it, to illustrate the features that are introduced in this paper.

We consider two universities in Brazil, PUC-Rio and UFF, which are collaborating in some research projects, e.g. the UbiForm Project. He is a professor at PUC-Rio, and is also participating in the UbiForm Project. He carries with him his smart phone, which hosts some context-aware applications. When he arrives at PUC-Rio, an Ambient Management Service (AMS) detects his smart phone (SMP_{Silva}) and identifies that it belongs to him. The Ubiform Project Agenda (UPA), a service of AMS, informs the members of UbiForm Project about Silva's arrival. A Personal Agenda application running on SMP_{Silva} contacts UPA with a request to be notified about the beginning of events. Another application executing at SMP_{Silva}, the Configuration Management Service (CMS), requests to be notified whenever Silva is in a room in which an activity (e.g. a technical presentation, a brainstorming session) has started, so that it may switch the smart phone to silent mode. When Silva arrives at UFF SMP_{Silva} gets a connection to the local wireless network, and based on its GPS coordinates finds out that its owner (Silva) is at UFF. It then determines that this university is a partner institution of PUC-Rio, obtains the IP address of the AMS at UFF and registers with it, indicating the user's identity and his preferences. The AMS registers SMP_{Silva} and identifies that it belongs to Silva. The system verifies that Silva is member of the collaboration project and configures a workspace for him. Although Silva is identified as a visitor at UFF, he will also be perceived as being a professor (from PUC-Rio). According to UFF's Ubicom resource access policy any professor can access the meeting room's printers at UFF. Then, AMS will also recognize this access permission.

3.2 Formalizing the Scenario

Diagrams of Devices: Figure 2 composes the diagram of devices of the scenario described in subsection 3.1. It shows AMS, CMS, UPA and PA that interchange information as a result of the perception of the presence Professor Silva. Two of these devices are applications running on Silva's smart phone: the CMS, that configures the phone mode to silent, alarm or vibration according to the room or activity where Professor Silva is engaged, and PA, a personal agenda that must be synchronized with the project agenda. The other two devices (AMS and UPA) are applications running at the environment and play the role of monitoring information about the environment and notifying the SMP applications. Figure 2-A pictures the exchanging of information when Professor Silva arrives at PUC. In figure 2-B, we show the diagram of devices that formalize the arrival of Professor Silva at UFF. In this case, it is performed a communication between AMS, at PUC, and AMS at UFF, via GPS/WiFi. Through this communications, Professor Silva is recognized at UFF and resources can be allocated according to his preferences and permissions.

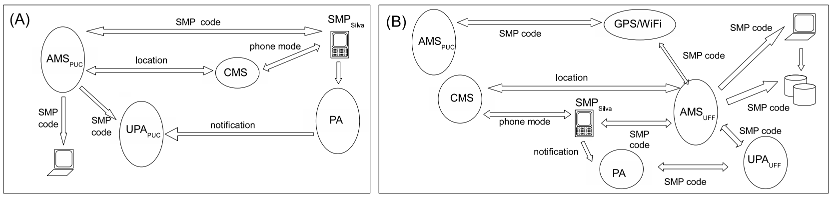

Fig. 2. Diagrams of devices: (A) Professor Silva at PUC.(B) Professor Silva at UFF.

Diagrams of Entities: Each device of the diagram of devices must be detailed in terms of a diagram of entities. For example, for AMS it is necessary to combine personal and professional information of Professor Silva in order to use personal preferences to configure professional environment. The algebra of contextualized entity is used in this level of specification to guide the construction of a coherent and consistent body of the information over which each device works. Details of the the algebraic operation will appear just in the diagrams of ontologies. In order to illustrate the use of diagrams of entities we describe in the following the diagram for AMS.

The Ambient Management Service (AMS). Figure 3 concerns the situation where AMS informs other members of Silva's team about his arrival. $Prof_i$ at PUC results from the context integration of each i and the ontology describing information of PUC, under the mediation of professional information of each i (Prof. i). Figure 3-A shows this context integration for $i = Silva$. The entity integration of each $Prof_i$ and $Prof$ $Silva$ under the context of PUC will make the connection among the i professors of PUC and Professor Silva. The resulting entity will be composed by each professor (Figure 3-B).

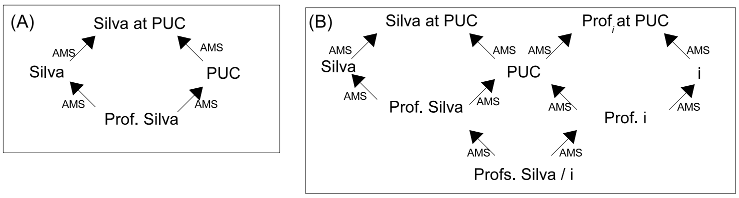

Fig. 3. (A) Context integration of Silva and PUC, under the mediation of professional information of Silva (Prof. Silva). (B)Entity integration of several professors.

Later, Professor Silva is visiting UFF, where he is registered as a visitor researcher. Within the context $SilvaAtUFF$ that results from integration $Silva \overset{AMS}{\longleftarrow} Prof.Silva \overset{AMS}{\longrightarrow} UFF$ (lateral square at left in figure 4-A), AMS can properly set the professor's workspace. But some of Silva's permissions for the use of resources come from the fact that he is a Professor at PUC. Thus, information about Silva's status at PUC must also be taken into account. The context integration $UFF \overset{AMS}{\longleftarrow} Prof.Silva \overset{AMS}{\longrightarrow} PUC$ generates a context where AMS can find information about Silva as a PUC professor and as a UFF visitor researcher in the joint project UFF/PUC (base square of Figure 4-A). The context integration

$SilvaAtUFF \stackrel{AMS}{\longleftarrow} Silva \stackrel{AMS}{\longrightarrow} SilvaAtPUC$ generates a context where AMS can find not only information about Silva as a PUC professor or as a UFF visitor researcher, but also personal information about Silva (top square of Figure 4-A). Note that Figure 4-A also pictures a combined integration: the collapsed union of the contextualized entities $UFF \rightarrow SilvaAtUFF$, $PUC \rightarrow SilvaAtPUC$ mediated by $Prof.Silva \rightarrow Silva$.

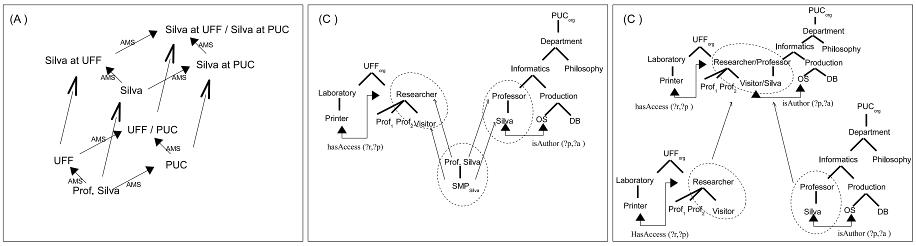

Fig. 4. (A) Diagram of entities of AMS: Each face of the cube shows a context integration. The complete cube is the collapsed union of the contextualized entities $UFF \rightarrow SilvaAtUFF$, $PUC \rightarrow SilvaAtPUC$ mediated by $Prof.Silva \rightarrow Silva$. (B) Diagram of ontologies: Alignment of UFF and PUC under the mediation of SMP_{Silva}. The mediator captures the fact that Silva is a professor and properly map this information in the ontology of UFF. (C) Diagram of ontologies: Context integration over the alignment of (B).

Diagrams of Ontologies: The diagrams of ontologies are a refinement of the diagrams of entities, were each entity appears described as an onlology and the connections are links between ontologies (that is, they preserve ontology properties). We selected a illustrative diagram of entities to show how this framework provides the required information to adapt services or behaviours according to the context changes. We consider a situation in which information coming from one context enables decisions about an entity in a different context. For instance, Professor Silva is allowed to use the printer at UFF as a consequence of the fact that, at PUC, he is a professor.

Considering the base square of Fig. 4-A, the mediator $Prof.\ Silva$ of the context integration $UFF \overset{AMS}{\longleftarrow} Prof.\ Silva \overset{AMS}{\longrightarrow} PUC$ must capture the fact that Silva is a professor and properly map this information into the ontology of UFF. Diagram of ontologies in Figure 4-B depicts the ontology for UFF and PUC and shows this alignment. Note that, as the concept *Professor* at *PUC* is related to *Researcher* at *UFF*, the relation *hasAccess(?p,?d)* will hold for Professor Silva and Printer in the resulting context (in Fig. 4-C). Also, note that, in this resulting context, information about Professor Silva's production is available to be used by AMS.

4 Algorithms and Complexities

The contextualization of an entity is the definition of a link between ontologies, where the source is the entity and the target is the context. According to definition 2, a link preserves hierarchy, relations and logical properties, enabling, thus, a consistent and coherent mapping of meanings of the entity into the context. Contextualization is not an automatic process, as it reflects the intended semantics for the entity. It can, however, be automatically validated, that is, it is possible to define an algorithm to verify if the structure of concepts and relations is preserved and logical constraints are respected. We consider two ontology structures $O = (C, R, H^C, rel, A)$ and $O' = (C', R', H'^{C'}, rel', A')$, and a link $(f : C \rightarrow C', g : R \rightarrow R', h)$.

Algorithm 1 <u>*link:(f,g,h)⟶ logical*</u>

For all $(c_1, c_2) \in H^C$ *if* $(f(c_1), f(c_2)) \notin H'^{C'}$ *returns false;*
For all $r \in (c_1, c_2) \in rel$ *if* $(f(c_1), f(c_2)) \notin g(r)$ *returns false;*
For all $a \in A$ *if* $trans_{fg}(a) \notin Th(A')$ *returns false*

The complexity of the algorithm for validating a link $O \overset{(f,g,h)}{\longrightarrow} O'$ is dependent of the complexity of the theorem prover for the considered logic. Considering just the structure of the ontology, the complexity is linear on the size of O.

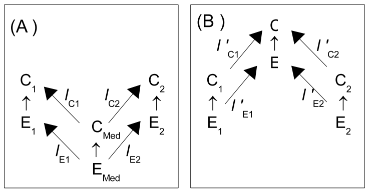

Fig. 5. (A) Alignment of contextualized entities. (B) Coalignment of contextualized entities

The contextualization of an entity is the fundamental operation over which all the operations of the algebra are constructed. For example, **the alignment of an entity** is its contextualization in two ontologies. By duality, **the coalignment of two entities** is the contextualization of both in the same contex. In a similar way, **the alignment of a contextualized entity** is the consistent combination of alignments in contexts and entity. By duality, **the coalignment of two contextualized entities** is the consistent combination of coalignments in contexts and entity. By consistent we mean that the ontological structure of the entity (co)alignment is preserved in the context (co)alignment. Referring to figure 5-A, this means that E_{Med} is mapped in the same way in C_1 following $E_{Med} \rightarrow C_{Med} \rightarrow C_1$ or following $E_{Med} \rightarrow E_1 \rightarrow C_1$, and the same for C_2. In the sequel we present the algorithm for verifying that a square such as $C_1 \leftarrow C_{Med} \leftarrow E_{Med} \rightarrow E_1 \rightarrow C_1$ is consistent. We consider four links between ontologies forming a square as: $O' \overset{l'_1}{\leftarrow} O_1 \overset{l_1}{\leftarrow} O \overset{l_2}{\rightarrow} O_2 \overset{l'_2}{\rightarrow} O'$, where, $l_1 = (f_1, g_1, h_1), l'_1 = (f'_1, g'_1, h'_1), l_2 = (f_2, g_2, h_2), l'_2 = (f'_2, g'_2, h'_2)$.

Algorithm 2 *square:*$(l_1, l'_1, l_2, l'_2) \longrightarrow$ *logical*

For all $c \in H^C$ *if* $(f'_1(f_1(c))) \neq (f'_2(f_2(c))$ *returns false;*
For all $r \in rel$ *if* $(g'_1(g_1(r))) \neq (g'_2(g_2(r)))$ *returns false;*

The algorithm verifies whether concepts and relations are consistently mapped. Axioms are not verified because the transitivity is ensured by the underliyng logic (if $trans_{f'_1 g'_1}(trans_{f_1 g_1}(a))) \in Th(A')$ then $trans_{f'_2 g'_2}(trans_{f_2 g_2}(a))) \in Th(A')$).

As in the previous case, the complexity of the algorithm is linear on the size of O.

Note that *(i)* the same algorithm can be used to verify the consistency of alignments and coalignments; *(ii)* Given three contextualized entities and a coalignment of their entities, the consistent coalignment on their contexts is unique and can be trivially computed; thus, for composing contextualized entities it is enough to coalign the entities. In the same way, given three contextualized entities and an alignment of their entities, the consistent alignment on their contexts is unique and can be trivially computed; thus, for decomposing contextualized entities it is enough to align the entities.

The algorithm to compute the **collapsed union** produces, from an alignment of contextualized entities (figure 5-A), a coalignment of contextualized entities (figure 5-B). In the following we describe the algorithm to compute the collapsed union. Figure 6 illustrates each step.

Algorithm 3 *Given an alignment, construct a coalignment by the following steps:*
1. For all component x of the alignment mediator
2. add a new component y to the coalignment mediator
3. create a link from the image of x by the left link to y
4. create a link from the image of x by the right link to y
5. For all component x that are not in the image of the left or right links from the alignment mediator
6. add a new component y to the coalignment mediator
7. create a link from x to y
8. Perform the union of axioms, with the proper translations.

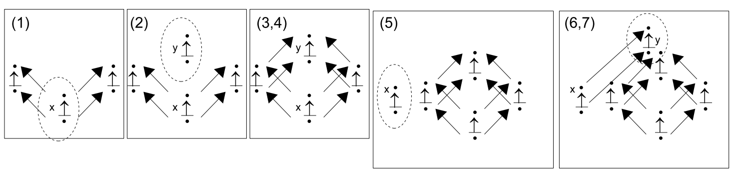

Fig. 6. Steps of the algorithm to compute the collapsed union.

The resulting coalignment embodies the least informative integration of the given alignment. The complexity of the algorithm is linear on the size of the given alignment, as each component of each ontology is visited a unique time.

The **context integration** of $C_1 \leftarrow E \rightarrow C_2$ can be derived from the collapsed union by contextualizing each entity or context in itself, using the identity as link. Thus, the collapsed union $C_1 \rightarrow C_1$ and $C_2 \rightarrow C_2$ under the mediation of $E \rightarrow E$ results the context integration of $C_1 \leftarrow E \rightarrow C_2$.

The algorithm for **Relative Intersection** is the dualization (2.1) of the algorithm for collapsed unoion:

Algorithm 4 *Given a coalignment, construct an alignment by the following steps:*
1. For all component x of the coalignment mediator that is in the image of left and right links
2. add a new component y to the alignment mediator
3. create a link from y to the domain of x by the left link
4. create a link from y to the domain of x by the right link
5. Perform the intersection of the disjunction of axioms, with the proper translations.

While the collapsed union algorithm results a coalignment that embodies the least informative integration of the given alignment, this algorithm results an alignment that embodies the more informative 'intersection' (common part) of the coalignment. While the collapsed union algorithm requires additional lines (5 to 7) *to augment* the generated coalignment with information that is particular to the left or right side, the relative intersection algorithm requires the condition of line 1 *to restrict* the generated alignment to information of the coalignment (avoiding, thus, information that is particular to the left or right side). The complexity of the algorithm for relative intersection is linear on the size of the given coalignment in the structural part.

The **entity integration** of $E_1 \rightarrow C \leftarrow E_2$ can be derived from the relative intersection by contextualizing each entity or context in itself. Thus, the relative intersection $E_1 \rightarrow E_1$ and $E_2 \rightarrow E_2$ under the mediation of $C \rightarrow C$ results the entity integration of $E_1 \rightarrow C \leftarrow E_2$.

4.1 Logical remarks on Collapsed Union and Relative Intersection

Description Logics (DL) [19] are quite well-established as underlying logics for knowledge reasoning. $\mathcal{ALC}$ is a basic description language whose syntax of concept descriptions is: $\phi_c ::= \bot|A|\neg\phi_c|\phi_c \sqcap \phi_c|\phi_c \sqcup \phi_c|\exists R.\phi_c|\forall R.\phi_c$ where A stands for atomic concepts and R for atomic roles (non-taxonomic relations). In a broad sense, a knowledge base specified in any description logic having $\mathcal{ALC}$ as core is called an ontology. A DL theory presentation is a set of axioms in the DL logical language. Considering $\mathcal{ALC}$, we discuss, in this subsection, the construction of theory presentation for Collapsed Union and Relative Intersection. We take A and B as theory presentations of the contextualized entities to be operated by Collapsed Union or Relative Intersection. Without loosing generality, we assume that both sets are given over the same vocabulary.

We know that for any sets of formulas, A and B the following relations hold: *(i)* $Th(A) \cup Th(B) \subseteq Th(A \cup B)$ and *(ii)* $Th(Th(A) \cup Th(B)) \equiv Th(A \cup B)$. The argument for the former is the following: - If $\phi \in Th(A) \cup Th(B)$ then $\phi \in Th(A)$ or $\phi \in Th(B)$, equivalently, $A \vdash_{DL} \phi$ or $B \vdash_{DL} \phi$, and, in both cases $A \cup B \vdash_{DL} \phi$, and hence, $\phi \in Th(A \cup B)$. The later is based on the fact that if $\phi \in Th(A \cup B)$ then $A \cup B \vdash_{DL} \phi$ and so, by (Craig) interpolation there is σ, s.t. either , $A \vdash_{DL} \sigma$ and $\{\sigma\} \cup B \vdash_{DL} \phi$, or , $B \vdash_{DL} \sigma$ and $\{\sigma\} \cup A \vdash_{DL} \phi$. In both cases we have $\phi \in Th(Th(A) \cup Th(B))$. It is worth noting that $Th(A) \cup Th(B) \not\equiv Th(A \cup B)$, counterexample provided by $A = \{p_1\}$ and $B = \{\neg p_1 \sqcup p_2\}$, where p_1 and p_2 are distinct atomic DL concepts. It is also important to note that the interpolation is valid for $\mathcal{ALC}$ (proved as a consequence of cut elimination, [7]), but we are not aware whether the Craig interpolation holds for other description logics. We reach the conclusion that a theroy presentation for the collapsed union operation is $Th(Th(A) \cup Th(B))$. From *(i)* and *(ii)*, $Th(A) \cup Th(B)$ can be taken as an axiomatization, and hence so $A \cup B$ can be. It is worth remarking that this operation may produce set of properties is logically inconsistent.

In the case of the relative intersection, from the conterexample $A = \{p_1 \sqcap p_2\}$, $B = \{p_1, p_2\}$ we see that $Th(A \cap B) \subseteq Th(A) \cap Th(B)$. Thus $A \cap B$ can not be taken as an axiomatization for the relative intersection. The theory presentation, in this case, is done by the disjunction of formulas, based on the following fact: *(iii)* $Th(A) \cap Th(B) \equiv Th(\{\alpha \sqcup \beta / \alpha \in A \text{ and } \beta \in B\})$.

The argument for *(iii)* is: If $\phi \in Th(A) \cap Th(B)$ then $\phi \in Th(A)$ and $\phi \in Th(B)$, equivalently, $A \vdash_{DL} \phi$ and $B \vdash_{DL} \phi$, thus, $\{\alpha \sqcup \psi / \alpha \in A\} \vdash_{DL} \phi$ for any ψ and $\{\beta \sqcup \psi / \beta \in B\} \vdash_{DL} \phi$ for any ψ. Then, $\{\alpha \sqcup \beta / \alpha \in A \text{ and } \beta \in B\} \vdash_{DL} \phi$, and thus, $\phi \in Th(\{\alpha \sqcup \beta / \alpha \in A \text{ and } \beta \in B\})$. Conversely, if $\phi \in Th(\{\alpha \sqcup \beta / \alpha \in A \text{ and } \beta \in B\})$ then $\{\alpha \sqcup \beta / \alpha \in A \text{ and } \beta \in B\} \vdash_{DL} \phi$. As a consequence of $\sqcup$-elimination, $\{\alpha / \alpha \in A\} \vdash_{DL} \phi$ and $\{\beta / \beta \in B\} \vdash_{DL} \phi$, and $\phi \in Th(A)$ and $\phi \in Th(B)$, hence $\phi \in Th(A) \cap Th(B)$.

In the relative intersection, the theory generated from consistent sets of axioms is always consistent.

5 Related Work

In [8–10] a purely logical approach to deal with modular ontologies is taken. Following the approach to distributed first-order logic proposed in [9], the authors define distributed versions of Description Logics (DDLs) able to logically specify the interconnection between concepts/roles between component ontologies. For example, let us consider that one needs to specify that a concept D, described by an ontology i[1], is associated to a concept E, described by an ontology j. Their approach says, in this case, that the specifier should set up the (distributed) subsumption $i : D \sqsubseteq j : E$ interconnecting these two concepts by means of

[1] To say that D is described by an ontology means that D is defined by means of a Description Logic term-concept in this ontology.

an external ontology. They also propose modifications on the existing reasoning algorithms for DLs, in order to cope with this distributed version. The result is that the reasoner uses axioms from different theories (ontologies) guided by the axioms present in the external ontology and the queries submitted [11]. In [12], DDL was used to reason on Change Management in (Modular) Ontology.

The DDL approach is an interesting and elegant solution for the general problem of modularizing ontologies. It is somehow related to our approach on integration of ontologies. Let us compare them. The focus of our approach is on context-awareness ontological specification. In this sense, we are able to locally identify contextualized entities (concepts/roles descriptions), dealing separately with their respective context as well as context integration. The distributed Description Logic approach, by itself, does not provide an answer to context-awareness. We would adopt the same basic idea proposed in our approach if we used DDLs instead of ordinary DLs. Besides that, our use of an algebra for expressing the structure of contextualized ontologies maintains the information on the structure itself, allowing posterior decomposition and integration at the theories (ontologies) level. We think this is important in any approach to structuring ontology. Regarding the work discussed in this paragraph we can say that our approach would add context-awareness and structure representation to DDLs, while DDLs would bring us more elegant and efficient reasoning, regarding memory/storing. Using ordinary DLs (as in OWL-DL) we must either build the whole structured contextualized ontology, as described in section 2 or implement a backward chain reasoner, aware of this structure. In this later case the reasoner performas the integration at execution time, what implies in a loss of efficience. One other point is that DL's subsumption is a particular case of the links between ontologies that are used in our approach. Using DDLs, the external ontology should be defined in order to reflect structured contextualized ontologies. This later issue is subject of further research. An interesting point also to be addressed is the fact that we can use our approach on contextualized ontologies to deal with change management of ontologies (see [12]). We can take a changing of ontologies as changing of contexts.

There are several other works that deal with formalization of context modelling using different theoretical approaches. In [6] we presented a detailed comparison between our approach and Context UNITY [13, 14], *Context-Aware Action System* [15] and Bigraphical Reactive System (BRS) Models [1]. The first approach aims at the specification of applications that use a mechanism for defining contexts which is transparently maintained as the environment evolves. The second approach presents an extension of the classical action system formalism with the addition of the notion of context; and the third approach has as main goals to model ubiquitous systems and to be a meta-theory encompassing existing calculi for concurrency and mobility (such as CCS, π-calculus). Beside these, we cite CommUnity [4] who as well as our approach uses Category Theory as theoretical basis to express integration and contexts. It also emphasises the separation between systems behaviour and context modelling. As it evolved from a previous work on distribution and mobility in software architectures, it adopts concepts of communication by channels and location variables. We believe our approach is more general than it, since CommUnity context mappings happen always via channels and location variables while ours is based on any conceptual piece of the ontology.

6 Conclusion

In this paper we propose a formal framework to support the specification of context modelling of context-aware systems (e.g., mobile or ubiquitous systems). We follow the direction of much of the works ([3, 4]) in context-aware computing, where the application logic is independent of the informational infrastructure, what results in a high degree of reuse and facilitates easy program development. Our modelling framework is based on the Algebra of Contextualized Ontologies [5, 6], which hides its theoretical basis (Category Theory) under a suggestive terminology, takes *contextualization* as a basic notion and proposes a small set of simple and powerful operations to compose and decompose *contextualized entities*. Due to the homogeneous and independent representation of entities and contexts, and the explicit representation of their relationships, it renders a modular and scalable description of arbitrarily complex context-aware systems. It is possible to define different levels of abstractions through diagrams, what enables a stepwise refinement of the description of the system and its components, and the construction of modular specifications, such that the complexity of a context-system gets decomposed into manageable pieces of specification. Using our framework it is furthermore possible to construct both a complete and a minimal description of context information over which each component of a ubiquitous system can reason.

References

1. L. Birkedal, S. Debois, E. Elsborg, T. Hildebrandt, and H. Niss, "Bigraphical Models of Context-aware Systems," The IT University of Copenhagen, Tech. Rep. TR-2005-74, November 2005.
2. K. Henricksen and J. Indulska, "Developing context-aware pervasive computing applications: Models and approach," *Pervasive and Mobile Computing*, vol. 2, no. 1, pp. 37–64, February 2006.
3. C. Julien and G.-C. Roman, "Egocentric context-aware programming in ad hoc mobile environments," in *SIGSOFT '02/FSE-10: Proceedings of the 10th ACM SIGSOFT symposium on Foundations of software engineering.* New York, NY, USA: ACM, 2002, pp. 21–30.
4. A. Lopes and L. Fiadeiro, "Context-awareness in software architectures," in *EWSA 2005: Proocedings of the 2nd European Workshop in Software Architecture.* Springer-Verlag, 2005.
5. I. Cafezeiro, E. H. Haeusler, and A. Rademaker, "Ontology and context," *6th Annual IEEE International Conference on Pervasive Computing and Communications (PERCOM 2008)*, vol. 0, pp. 417–422, 2008.
6. I. Cafezeiro, J. Viterbo, A. Rademaker, E.H.Haeusler, and M. Endler, "A formal framework for understanding context-aware behavior in ubiquitous computing,"

Communications in Computer and Information Science - 3rd International Symposium on Leveraging Applications of Formal Methods, Verification and Validation (ISOLA 2008), vol. 17, 2008.
7. A. Rademaker and E. H. Haeusler, "Toward short and structural -reasoning explanations: A sequent calculus approach." in *SBIA*, ser. Lecture Notes in Computer Science, G. Zaverucha and A. C. P. L. da Costa, Eds., vol. 5249. Springer, 2008, pp. 167–176.
8. A. Borgida and L. Serafini, "Distributed description logics: Assimilating information from peer sources," *Journal of Data Semantics*, vol. 1, p. 2003, 2003.
9. C. Ghidini and L. Serafini, "Distributed first order logics," 1998.
10. A. Borgida and L. Serafini, "Distributed description logics: Directed domain correspondences in federated information sources," in *On The Move to Meaningful Internet Systems 2002: CoopIS, Doa, and ODBase, volume 2519 of LNCS*. Springer Verlag, 2002, pp. 36–53.
11. L. Serafini and A. Tamilin, "Local tableaux for reasoning in distributed description logics," in *Description Logics*, ser. CEUR Workshop Proceedings, V. Haarslev and R. Möller, Eds., vol. 104. CEUR-WS.org, 2004.
12. H. Stuckenschmidt and M. Klein, "Reasoning and change management in modular ontologies," *Data Knowl. Eng.*, vol. 63, no. 2, pp. 200–223, 2007.
13. C. Julien, J. Payton, and G.-C. Roman, "Reasoning About Context-Awareness in the Presence of Mobility," in *Proceedings of the 2nd International Workshop on Foundations of Coordination Languages and Software Architectures (FOCLASA03), co-located with CONCUR 2003*, Marseille, France, September 2003.
14. G.-C. Roman, C. Julien, and J. Payton, "A Formal Treatment of Context Awareness," in *Proceedings of the 7th International Conference on Fundamental Approaches to Software Engineering (FASE'04)*. Barcelona, Spain: Lecture Notes in Computer Science 2984, March 2004, pp. 12–36.
15. L. Yan and K. Sere, "A Formalism for Context-Aware Mobile Computing," in *Proceedings of the IEEE 3rd International Workshop on Algorithms, Models and Tools for Parallel Computing on Heterogeneous Networks (HeteroPar)*, Cork, Ireland, July 2004.
16. A. Shehzad, H. Ngo, K. Pham, and S. Lee, "Formal modeling in context aware systems," in *Proceedings of the First International Workshop on Modeling and Retrieval of Context*, September 2004.
17. X. Wang, D. Zhang, T. Gu, and H. Pung, "Ontology based context modeling and reasoning using OWL," in *Proceedings of 2nd IEEE Conf. Pervasive Computing and Communications (PerCom 2004), Workshop on Context Modeling and Reasoning*. Orlando, Florida: IEEE Computer Society Press, March 2004, pp. 18–22.
18. E. Maedche and S. Staab, "Ontology learning for the semantic web," *IEEE Intelligent Systems*, vol. 16, pp. 72–79, 2001.
19. F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, Eds., *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.
20. J. Viterbo, M. Endler, and J.-P. Briot, "Ubiquitous service regulation based on dynamic rules," in *Proceedings of the 13th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS 2008), Belfast*. IEEE Computer Society Press, 2008, pp. 175–182.