The Knowledge Engineering Review, Vol. 29:2, 171–185. © Cambridge University Press, 2014
doi:10.1017/S0269888914000046

Specifying ubiquitous systems through the algebra of contextualized ontologies

ISABEL CAFEZEIRO[1], JOSÉ VITERBO[1], ALEXANDRE RADEMAKER[2], EDWARD HERMANN HAEUSLER[2] and MARKUS ENDLER[2]

[1]*Departamento de Ciência da Computação, Universidade Federal Fluminense, Niterói, Brasil;*
e-mail: isabel@dcc.ic.uff.br, jviterbo@ic.uff.br;
[2]*Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, Brasil;*
e-mail: markus.endler@gmail.com, arademaker@inf.puc-rio.br, hermann@inf.puc-rio.br, endler@inf.puc-rio.br

Abstract

In order to be able to specify and design ubiquitous applications, it is necessary to rely on a precise and formal representation of context. Aiming at reducing the gap between the formal specification and the design of a concrete ubiquitous application, we have proposed an approach to specify a context-aware system at three levels of abstraction. This specification uses the Algebra of Contextualized Ontologies, which is based on Category Theory and takes contextualization as a basic notion, proposing a small set of simple and powerful operations to compose and decompose contextualized ontologies. In this paper, we present a revised version of the algebra and algorithms to compute the algebraic operations. We exemplify our approach along with a concrete ubiquitous computing scenario. Moreover, by taking advantage of Category Theory foundations, we show that it is possible to derive the algorithms for each algebraic operation from simple modifications of a single basic algorithm.

1 Introduction

The ubiquitous computing paradigm was first introduced in Weiser (1991), which describes it as 'the calm technology that recedes into the background of our lives'. According to Mark Weiser's vision, people would be able to have seamless access, anytime and anywhere, to services and information providers. Ubiquitous computing is normally associated with a large set of electronic devices—either embedded into physical environments or in form of portable, wireless devices—that compute, sense and interact with each other to collectively enable the seamless services to mobile users.

These ubiquitous services aim at supporting the users in their daily activities in a way that their presence becomes unnoticed, while their absence may cause some inconvenience. Ubiquitous computing thus is inherently distributed, dynamic and involves heterogeneous systems and devices. Moreover, ubiquitous applications must respond dynamically to changes in the environment with no or little user interference. Thus, these systems must be context-aware, that is, they must be capable of gathering, interpreting and reacting to several sorts of context information of the environment in which they are used.

In this paper we show how to specify a ubiquitous system using an algebra that was specifically tailored to formal context modeling. This algebra, named Algebra of Contextualized Ontologies, has as central principles a *homogeneous* and *independent* description of entities and contexts, and a representation of their *semantics through a net of relationships*. By *homogeneous description* of entities and contexts, we mean the use of a uniform representation of the knowledge pertaining to

https://doi.org/10.1017/S0269888914000046 Published online by Cambridge University Press

these elements, that is, we use ontologies to describe both entities and contexts. The homogeneous description not only facilitates a uniform mapping between entities and contexts, but also allows to regard an entity both as a single component of a system, or as a complete system. By *independent description* of entities and contexts, we mean that any component of a system, be it a device or an environment, is handled as an independent element with corresponding attributes and encapsulated state. The relation of an entity to its context, however, is made explicit through well-defined connections. This form of description supports modular and reusable specifications.

Finally, by *semantics through a net of relationships* we mean that it is just through the semantic links (connections) that one determines 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, for example, a meta-context of the first context. In addition, it is also possible for an entity to have several contexts (represented by several links with the same domain), or for a context to contextualize several entities (represented by 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.

1.1 Scope of this work

The Algebra of Contextualized Ontologies was first proposed in Cafezeiro *et al.* (2008a), in which basic concepts to support structural composition of contextualized ontologies were discussed. In Cafezeiro *et al.* (2008b), the applicability of this algebra was first discussed in the scope of ubiquitous computing.

With the aim of reducing the gap between the formal specification and the development of the concrete ubiquitous application, in this paper we adopt an approach to formalize a system at different levels of abstraction reflected by three different, but related, diagrams: (i) the design of the ubiquitous system, (ii) the integration of components using the algebra operations and (iii) detailed formalization in the Algebra of Contextualized Ontologies.

As a result of balancing the formal proposal and the experimental insights, we present a revised version of the algebra incorporating non-structural constraints considering the basic description logic *ALC*, as tackled in Rademaker and Haeusler (2008). Besides, we present new algorithms to compute algebraic operations, taking advantage of Category Theory foundations. This makes possible to derive all algorithms for implementing of the operations of the algebra from simple modifications of a single basic algorithm.

Thus, the main purpose of this paper is not to expose the mathematical background of the Algebra of Contextualized Ontologies, nor to explain the specification process or environment, which has been done in Cafezeiro *et al.* (2009), but to present the new algorithms and the extension of the operations to the axioms of each contextualized ontology, and to exemplify the specification of an ubiquitous system under this revised version. Although in this work we focus solely on ubiquitous computing, the generality of our approach indicates that it can also be applied to any system where context plays a key role.

This paper is organized as follows. In Section 2, we revisit the Algebra of Contextualized Ontologies. In Section 3, we discuss the methodology based on the algebra used in the specification of context-aware systems, present our scenario in more detail and formalize it using the proposed approach. In Section 4, we present algorithms to compute the algebraic operations. In Section 5, we compare our approach with related work. Finally, Section 6 concludes the paper.

1.2 Motivating scenario

In order to clarify the concepts involved in the algebra of contextualized ontologies and to explain the proposed specification methodology, let us consider the following ubiquitous computing scenario. Silva is a professor at a university, for example, PUC-Rio. As a researcher, he is also

https://doi.org/10.1017/S0269888914000046 Published online by Cambridge University Press

involved in a project called UbiForm, where he collaborates with professors of another university, say UFF. The collaborating partners hold regular meetings in each university, according to a common agenda previously arranged, and share a set of documents about the project. Consider also that, in addition their notebooks, Silva and the other UbiForm researchers use other mobile devices, such as smart phones, that have embedded positioning technology (e.g. GPS), and which execute context-aware applications. Moreover, in both universities, the meetings take place in computer-mediated environments, where users' activities are supported through specific ubiquitous services, that is, their mobile devices are able to discover and interact with cloud services or services provided by the ambient infrastructure of the visited places. When attending a project meeting at UFF, Silva may rely on these services to get instant access to devices (e.g. a printer) and to project documents, or to be notified about events such as the beginning of a meeting or the delay of a participant.

In such scenario, from a certain level of perspective, Silva may be regarded as an entity, linked to a context that may comprise simply his smart phone preferences and his current location. On the other hand, a location, such as a classroom, may itself be regarded as an entity that is linked to a context that may aggregate its description plus context data obtained from sensors located inside the room (e.g. presence sensors). All these pieces of information may be used, for example, to enable the context-aware application at Silva's smartphone to automatically set the device's ring mode to silent whenever he is within a classroom of PUC-Rio, since his current activity there is usually lecturing.

However, when attending a project meeting at UFF, at another level of perspective, Silva may be regarded as a different entity: 'Silva at UFF'. In this case, the context to be considered may be his role (e.g. spokesman in the UbiForm project) and his current agenda entry (e.g. a UbiForm project meeting). In this case, although Silva is not in any classroom of his university, there is still this meta context—the UbiForm meeting at UFF and Silva's role in this project—that may cause the mobile application to automatically switch the devices's ring mode to silent also in this situation. Moreover, sometimes, it may happen also that an entity is linked to more than one context. For example, Silva may be simultaneously the spokesman at the UbiForm meeting at UFF, and at the same time a father-to-be that wants to be reached at any possible location, if his wife calls him.

Context-aware applications that interact with Silva in such different situations have to be able to identify the different contexts that must be used to trigger the system's behaviors. To set the ring mode to silent, an application running on Silva's smartphone has to consider different contexts if he is at PUC-Rio or participating in a meeting at UFF. In the latter case, this application must concomitantly take into account contexts that describe Silva's personal preferences and his role and activity at UFF, or his private situation (e.g. father-to-be). On the other hand, in the same situation, a service running in the ambient infrastructure at UFF, would also have to take into account the contexts related to Silva whenever he is visiting that institution. In both cases, there is the need of a methodology to support the modeling and specification of such entities and contexts at different levels of abstraction.

2 The algebra of contextualized ontologies

Contextualized ontologies are composed of 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, for example, as being a mobile device. The context can be thought of as the environment where the entity operates, which, in this case, could be a physical environment where this mobile device is located. The link is the immersion of the entity into its context. Under this model, information concerning either the 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 modularity. The following definition is adapted from Maedche and Staab (2001):

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*

https://doi.org/10.1017/S0269888914000046 Published online by Cambridge University Press

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, which we call 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). However, for some ontologies Ω there are no corresponding ontology structures. For example, for $\Omega = \{\neg p \sqcup p\}$. In this paper we focus on ontology structures, that is, the relationships between ontologies will be given at the semantic level. It is clear that any semantic mapping induces a mapping between theories.

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) of Definition 2, 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 $A' \vdash trans_{fg}(a)$. 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 $a \in \Omega, \Omega' \vdash trans_{fg}(a)$.

Since we adopt a semantic-oriented approach, in the remainder of this paper we use the term *ontology* as a placeholder for ontology structure.

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.*

Example 1. *Figure 1 shows a simplification of the representation of Prof. Silva and its contextualization in PUC-Rio. The ontology at left describes Silva as a concept Prof. Silva and a subconcept* SMP_{Silva}*, the smart phone of Prof. Silva. In the ontology at right there is a simplified representation of PUC-Rio. In this situation, the ontology representing the professor plays the role of entity and the ontology representing the institution PUC-Rio plays the role of context. The contextualized entity shows how Prof. Silva is considered as a member of PUC-Rio. In this figure, one can notice that the hierarchy of concepts is preserved.*

https://doi.org/10.1017/S0269888914000046 Published online by Cambridge University Press

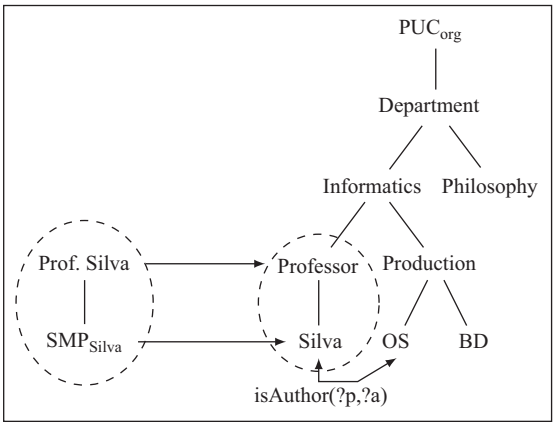


Figure 1 An entity: Silva, a contex: PUC-Rio, and a contextualized entity: Silva at PUC-Rio

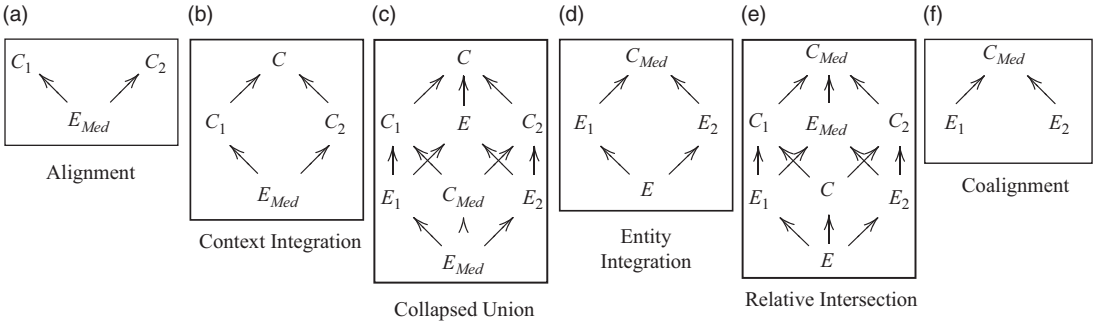


Figure 2 The algebra of contextualized ontologies operations

2.1 Operations on contextualized ontologies

The Algebra of Contextualized Ontologies is formed by two sets of operations devoted to compose and decompose contextualized ontologies in several ways. There is a duality between the actions of composing and decomposing which, in the formal sense, takes the form of reversing arrows, and is clearly noted by abstracting the role of an ontology to 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 to the simplification of the framework as a whole.

The operations have the general purpose of manipulating information of ontologies in order to produce a coherent and concise body of information. For that sake, to play the role of establishing binary relations between contexts or between entities, we define *alignment* and *coalignment* operations. We define also *context integration*, *collapsed union*, *entity integration* and *relative intersection* operations, which 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*.

Alignment. It is a situation where an entity has more than one context: $C_1 \leftarrow E_{Med} \leftarrow C_2$, as shown in Figure 2(a). 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. In an alignment $C_1 \leftarrow E_{Med} \leftarrow C_2$, the context integration produces a new context C, to which C_1 and C_2 are linked: $C_1 \rightarrow C \leftarrow C_2$, as shown in Figure 2(b). This new context

https://doi.org/10.1017/S0269888914000046 Published online by Cambridge University Press

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 $C_1 \leftarrow E_{Med} \leftarrow C_2$. The result contains all information of the original contexts, but identifies parts related by the mediator entity.

Collapsed union. It is the amalgamated union of two contextualized ontologies mediated by a third contextualized ontology, as shown in Figure 2(c). It is the combined composition 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 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. It is a mechanism of establishing a correspondence between vocabularies of two ontologies by the use of an intermediate target ontology: $E_1 \leftarrow C_{Med} \leftarrow E_2$, as shown in Figure 2(f). It constitutes 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, that is, not all concepts of one entity make sense for the other entity. Both entities, however, must be totally mapped in the context.

Entity integration. Given a coalignment $E_1 \leftarrow C_{Med} \leftarrow E_2$, the entity integration produces a new entity E contextualized by the original ones (and by transitivity, by the original context C_{Med}), as shown in Figure 2(d). The entity integration performs the semantic intersection of the entities under the mediation of the context, that is, the new entity will embody all information of the original entities that are related by the coalignment $E_1 \leftarrow C_{Med} \leftarrow E_2$.

Relative intersection. It is the intersection of two contextualized ontologies mediated by a third contextualized ontology, as shown in Figure 2(e). 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.

The reader who is familiar with Category Theory can find in Cafezeiro *et al*. (2008a), a detailed explanation about the categorical foundation of each operation. In this paper, we briefly remark that the contextualization of an entity is a morphism in the category of ontology structures. *Alignment* is a *pushout* in the same category, and the dual *coalignment* is a *pullback*. *Context integration* is a *pushout* in the category of ontology structures and the dual *entity integration* is a *pullback* in the same category. *Collapsed union* is a *pushout* in the category of links between ontology structures, and the dual *relative integration* is the *pullback* in the same catogory.

3 Specifying context-aware applications

Considering that it would be very difficult to specify a system by working directly with ontologies and ontology links defined by the algebra, we adopt in this approach three kinds of diagrams: *diagram of reactive elements*, *diagram of entities* and *diagram of ontologies*. These diagrams give structure to the process of specification by supporting different levels of expression. The *diagram of reactive elements* gives a general view of the system by representing the exchange of information between its components. The *diagram of entities*, sketches the meaning of a single entity, making possible to describing an entity without considering all the formal details of a specification through ontologies and ontology links. Finally, the *diagram of ontologies* introduces all formal details of a system. These three diagrams do not have the purpose of establishing a step-by-step process of specification, but to offer a simple way of expressing each level of abstraction.

Diagram of reactive elements. It specifies all computational elements (pieces of software/hardware) that compose the ubiquitous system and the kind of information that is exchanged among them. Each element 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 an element to another.

https://doi.org/10.1017/S0269888914000046 Published online by Cambridge University Press

Diagram of entities. An *entity* is any element of the system that produces a reaction caused by the interaction with another element of the system. For example, a Professor, that must be identified when arriving at work. Each element 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 element being described can extract the necessary information to work on.

Diagram of ontologies. It 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 Ontologies.

These three diagrams are related in the following way: each element of the *diagram of reactive elements* 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 Formalizing the scenario

Following, we give more details on the scenario presented in Subsection 1.2, highlighting the main elements involved.

Silva carries with him his smart phone (SMP_{Silva}), which hosts some context-aware applications, such as the Configuration Management Service (CMS) and the Personal Agenda (PA). While CMS is responsible for dynamically setting the configurations of SMP_{Silva} (e.g. switching it to silent mode), PA is responsible for managing Silva's personal agenda, notifying him about important events. The ambient infrastructure in each university also hosts context-aware applications, such as the Ambient Management Service (AMS) and the UbiForm Project Agenda (UPA). AMS is responsible for identifying the devices (and users) connected to the local wireless network and setting up users' workspaces, that is, providing access to the files they will need. UPA is responsible for managing the project agenda and tracking the project's participants.

When Silva arrives at PUC-Rio, the AMS of PUC-Rio (AMS_{PUC}) detects SMP_{Silva} and identifies that it belongs to him. The UbiForm Project Agenda (UPA_{PUC}) informs the members of UbiForm Project about Silva's arrival. The PA application running on SMP_{Silva} contacts UPA_{PUC} with a request to be notified about the beginning of events. The 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 SMP_{Silva} 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 *Silva* is at UFF. The the AMS of UFF (AMS_{UFF}) registers SMP_{Silva} and identifies that the device belongs to Silva, a member of UbiForm project who works at a partner institution. AMS_{UFF} contacts AMS_{PUC}, authenticating the user and recovering his preferences. It then configures a workspace for him. Although Silva is identified as a visitor at UFF, he will also be perceived as being a professor in his institution. According to UFF's UbiForm resource access policy any professor (local or visitor) can access the meeting room's printers at UFF. Then, AMS_{UFF} will also recognize this access permission.

3.1.1 Diagrams of reactive elements

Figure 3 composes the diagram of reactive elements of the scenario described above. It shows AMS_{PUC}, UPA_{PUC}, CMS, PA, AMS_{UFF} and UPA_{UFF} that exchange information as a result of the perception of the presence of Silva. Two of these elements are applications executing on Silva's smart phone: the CMS, which configures the phone ring mode according to the location or activity of Prof. Silva, and PA, a personal agenda that must be synchronized with the project agenda. The other elements, AMS_{PUC}, UPA_{PUC}, AMS_{UFF} and UPA_{UFF} are applications executing at each university's ambient infrastructure and play the role of monitoring information about the environment and providing specific functionalities for the smart phone's applications (e.g. notifying the arrival of a project participant). Figure 3(a) pictures the exchange of information when

https://doi.org/10.1017/S0269888914000046 Published online by Cambridge University Press

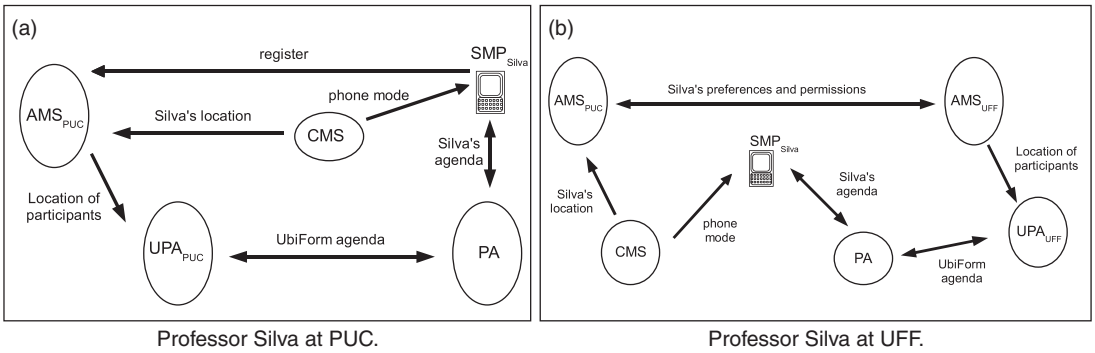


Figure 3 Diagrams of reactive elements

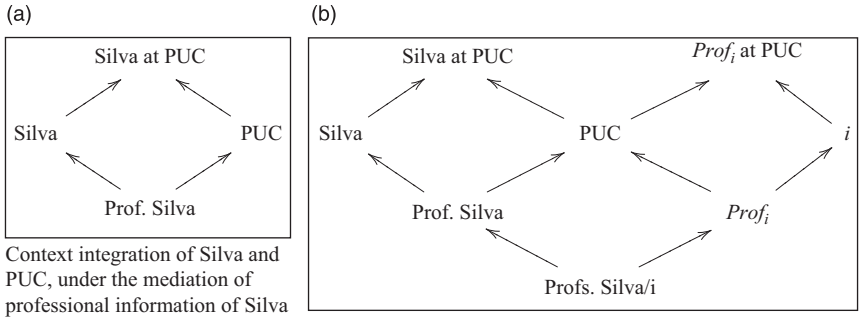


Figure 4 Diagrams of entities

Silva arrives at PUC. In Figure 3(b), we show the diagram of reactive elements that describes the arrival of Silva at UFF. In this case, AMS_{UFF} interacts with AMS_{PUC}, so that Silva is authenticated at UFF and the resources at UFF can be allocated for him according to his preferences and permissions.

3.1.2 Diagrams of entities

Each element of the diagram of reactive elements should then be detailed in terms of a diagram of entities. For example, for AMS_{UFF} it is necessary to combine personal and professional information of Silva in order to use personal preferences to configure a workspace for him, providing quick access to the files he will be manipulating. The Algebra of Contextualized Ontologies is used at this specification level to guide the construction of a coherent and consistent body of the information over which each element reacts. Details of the algebraic operation will appear only in the diagrams of ontologies. In order to illustrate the use of diagrams of entities in the following, we describe the diagram for the AMS.

Figure 4 represents the situation where AMS informs other participants of UbiForm project 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 4(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 and is shown in Figure 4(b).

Later, Professor Silva is visiting UFF, where he is registered as a visitor researcher. Within the context *SilvaAtUFF* that results from integration $Silva \overset{AMS}{\leftarrow} Prof.Silva \overset{AMS}{\rightarrow} UFF$ (lateral square at left in Figure 5), AMS_{UFF} 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.

https://doi.org/10.1017/S0269888914000046 Published online by Cambridge University Press

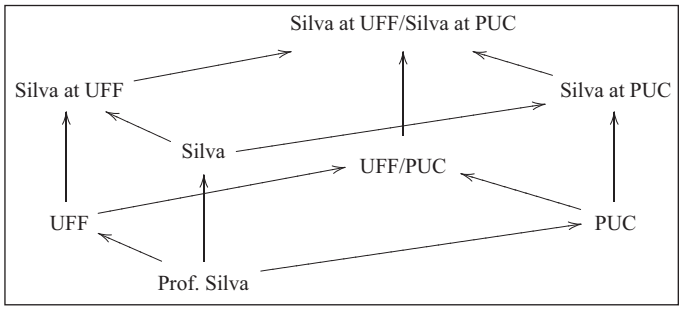


Figure 5 Diagram of entities of AMS: each face of the cube shows a context integration. The complete cube is the collapsed union of the contextualized entities $UFF \rightarrow SilvaAtUFF$, $PUC \rightarrow SilvaAtPUC$ mediated by *Prof. Silva* $\rightarrow$ *Silva*. AMS=Ambient Management Service

Thus, information about Silva's status at PUC must also be taken into account. The context integration $UFF \overset{AMS}{\leftarrow} Prof.Silva \overset{AMS}{\rightarrow} PUC$ generates a context where AMS_{UFF} 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 5). The context integration $SilvaAtUFF \overset{AMS}{\leftarrow} Silva \overset{AMS}{\rightarrow} SilvaAtPUC$ generates a context where AMS_{UFF} 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 5). Note that Figure 5 also pictures a combined integration: the collapsed union of the contextualized entities $UFF \rightarrow SilvaAtUFF$, $PUC \rightarrow SilvaAtPUC$ mediated by *Prof. Silva* $\rightarrow$ *Silva*.

3.1.3 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 of the entity diagrams are mapped to links between ontologies. We selected an 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, as any professor (local or visitor) can access the meeting room's printers at UFF, Silva is allowed to use that printer as a consequence of the fact that he is identified as a professor at PUC.

Considering the base square of Figure 5, the mediator *Prof. Silva* of the context integration $UFF \overset{AMS}{\leftarrow} Prof.Silva \overset{AMS}{\rightarrow} PUC$ must capture the fact that Silva is a professor and properly map this information into the ontology of UFF. Diagram of ontologies in Figure 6(a) 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 Prof. Silva and Printer in the resulting context (in Figure 6(b)). Also, note that, in this resulting context, information about Prof. Silva's production is available to be used by AMS_{UFF}.

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 (algorithm 1, left). We consider two ontology structures $O = (C, R, H^C, rel, A)$ and $O' = (C', R', H'^{C'}, rel', A')$, and a link $(f : C \rightarrow C', g : R \rightarrow R', h)$.

The complexity of the algorithm for validating a contextualization $O \overset{(f,g,h)}{\rightarrow} O'$ is dependent of the complexity of the theorem prover for the considered logic. As we are considering ALC, the

https://doi.org/10.1017/S0269888914000046 Published online by Cambridge University Press

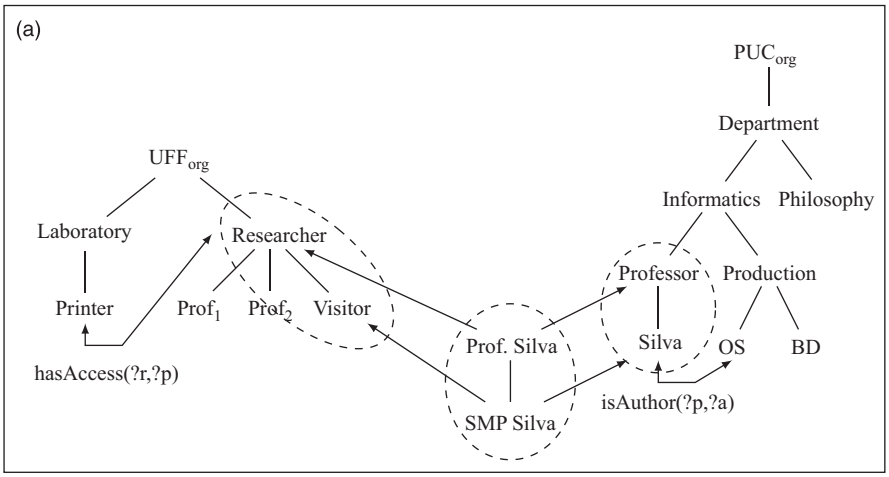


Diagram of ontologies: Alignment of UFF and PUC under the mediation of SMPSilva. The mediator captures the fact that Silva is a professor and properly maps this information into the ontology of UFF.

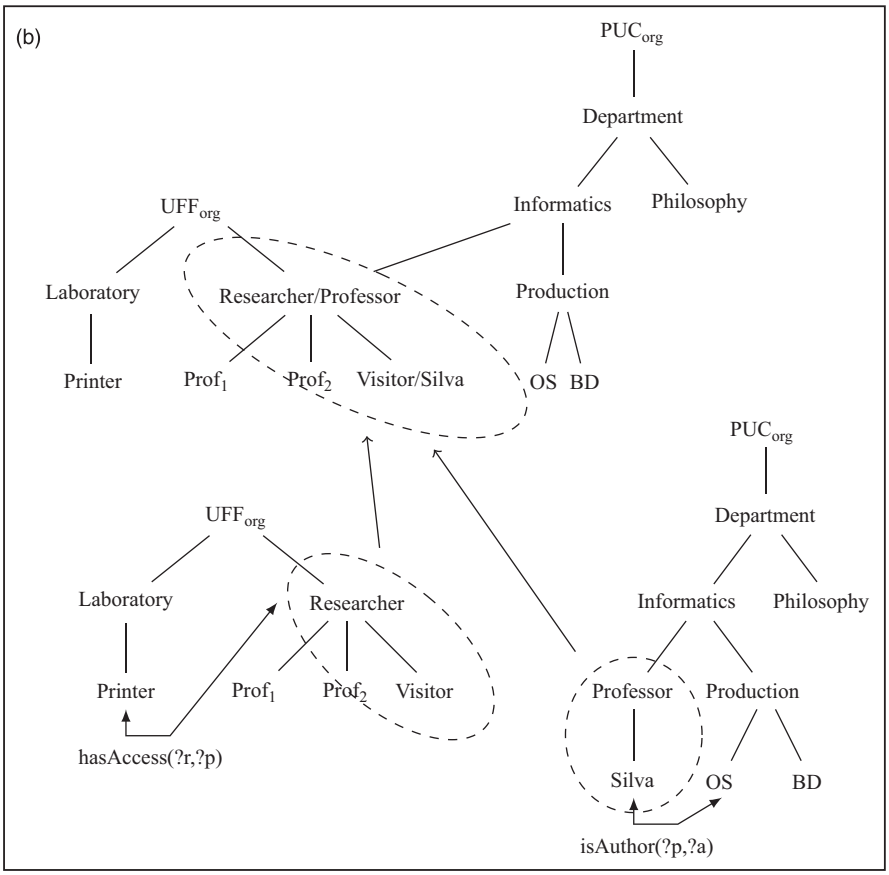


Diagram of ontologies: Context integration over the alignment of (B)

Figure 6 Composition concerning the integration of institutions

complexity is PSPACE Complete, as shown in Baader *et al.* (2003). Considering just the structure of the ontology, the complexity is linear on the size of O .

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 context. In a similar way, *the alignment of a contextualized entity* is the consistent combination of

https://doi.org/10.1017/S0269888914000046 Published online by Cambridge University Press

Algorithm 1 Validation of a contextualization (left) and square (right)

Input: link: (f, g, h)
Output: logical
1: **for all** $(c_1, c_2) \in H^C$ **do**
2: **if** $(f(c_1), f(c_2)) \notin H'^{C'}$ **then**
3: **return false**
4: **end if**
5: **end for**
6: **for all** $r \in R$ and $(c_1, c_2) \in rel(r)$ **do**
7: **if** $(f(c_1), f(c_2)) \notin rel'(g(r))$ **then**
8: **return false**
9: **end if**
10: **end for**
11: **for all** $a \in A$ **do**
12: **if** $trans_{fg}(a) \notin Th(A')$ **then**
13: **return false**
14: **end if**
15: **end for**

Input: square (l_1, l'_1, l_2, l'_2)
Output: logical
1: **for all** $c \in H^C$ **do**
2: **if** $(f'_1(f_1(c))) \neq (f'_2(f_2(c))$ **then**
3: **return false**
4: **end if**
5: **end for**
6: **for all** $r \in rel$ **do**
7: **if** $(g'_1(g_1(r))) \neq (g'_2(g_2(r)))$ **then**
8: **return false**
9: **end if**
10: **end for**

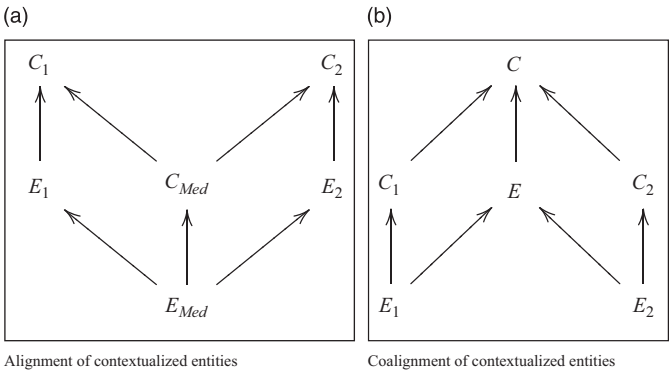


Figure 7 Alignment and coalignment

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 7(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. Algorithm 1, right, verifies if a square $C_1 \leftarrow C_{Med} \leftarrow E_{Med} \rightarrow E_1 \rightarrow C_1$ 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, $l_1 = (f_1, g_1, h_1), l'_1 = (f'_1, g'_1, h'_1), l_2 = (f_2, g_2, h_2), l'_2 = (f'_2, g'_2, h'_2)$.

The algorithm verifies whether concepts and relations are consistently mapped. Axioms are not verified because the transitivity is ensured by the underlying 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

https://doi.org/10.1017/S0269888914000046 Published online by Cambridge University Press

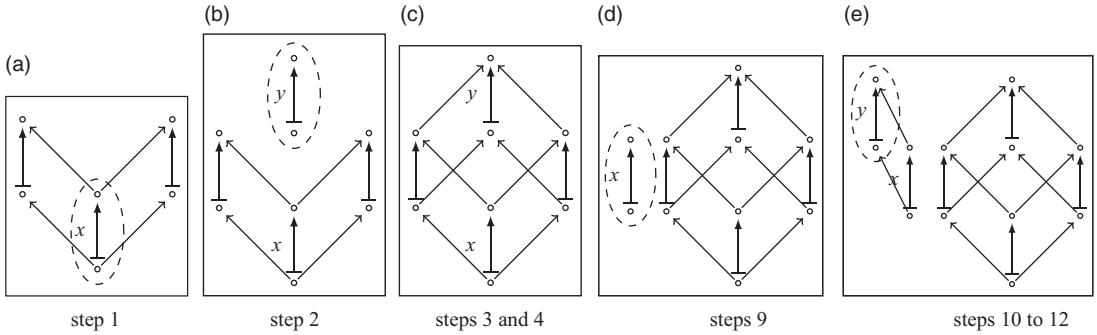


Figure 8 Steps of the algorithm to compute the collapsed union

Algorithm 2 Collapsed union

Input: alignment $CE_1 \overset{l_1}{\leftarrow} CE_{Med} \overset{l_2}{\rightarrow} CE_2$

Output: coalignment $CE_1 \overset{l'_1}{\rightarrow} CE \overset{l'_2}{\leftarrow} CE_2$

1: **for all** component x in CE_{Med} **do**
2: add a new component y to CE
3: create a link from $l_1(x)$ to y
4: create a link from $l_2(x)$ to y
5: **end for**
6: **if** there is a link from $l_i(x)$, $i = 1,2$ to some y' in CE **then**
7: collapse y with y'
8: **end if**
9: **for all** component x not in the image of l_i, $i = 1,2$ **do**
10: add a new component y to CE
11: create a link from x to y
12: **end for**
13: **for all** axioms a_i in the set of axioms of CE_i **do**
14: add $trans_{f_i,g_i}(a_i)$ to the set of axioms of CE
15: **end for**

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 7(a)), a coalignment of contextualized entities (Figure 7(b)). Figure 8 illustrates each step, with the exception of the block in lines 6 to 8, that deals with the particular case when a component of one of the contextualized ontologies is image of more than one component in the alignment mediator. This case must be reflected in the coalignment by collapsing the component introduced with a previously introduced component that is the image of this junction. In the block of lines 13 to 15, we use f_i, g_i to refer to the concept and relation components of each l_i.

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 $C_2 \leftarrow C_2$ under the mediation of $E \rightarrow E$ results the context integration of $C_1 \leftarrow E \rightarrow C_2$. The complexity is linear on the size of the ontologies.

The algorithm for *Relative Intersection* is the dualization of the algorithm for collapsed union.

https://doi.org/10.1017/S0269888914000046 Published online by Cambridge University Press

Algorithm 3: Relative Intersection

Input: coalignment $CE_1 \xrightarrow{l'_1} CE \xleftarrow{l'_2} CE_2$

Output: alignment $CE_1 \xleftarrow{l_1} CE_{Med} \xrightarrow{l_2} CE_2$

1: **for all** component x of the CE that is in the image of l'_i, $i = 1,2$ **do**
2: add a new component y to CE_{Med}
3: create a link from y *to the domain of* x by $l'_i, i = 1,2$
4: **end for**
5: **for all** a_i in the set of axioms of CE_i, $i = 1,2$ **do**
6: add $trans_{f'_1,g'_1}(a_1) \sqcup trans_{f'_2,g'_2}(a_2)$ to the set of axioms of CE_{Med}
7: **end for**

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 (9 to 12) *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). In the logical part, the disjunction of the axioms ensure the provability in both CE_1 and CE_2. The complexity of the algorithm for relative intersection is linear on the size of the given coalignment in the structural part and polynomial on the number of axioms in the logical 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 $E_1 \rightarrow C \leftarrow E_2$.

5 Related work

In this section we discuss the most relevant works that are related to ours from different perspectives, namely categorical-based approaches to ontologies integration, logic-based integration, particularly Distributed Description Logic, and other approaches that formalize context modeling.

Zimmermann *et al.* (2006) presented a categorically based approach for the alignment of ontologies that use *pushouts* (a specific co-limit notion) to provide a rigorous semantics to merging, intersection, union and composition of ontologies. Zimmermann *et al.* define the notion of V-alignment, a span in categorical terminology, raising an abstract notion of merge of two ontologies and an algebra that allows composition of V-alignments, intersection and union also. The approach is then extended with W-alignment, so as to cope also with non-symmetrical relations (like mothership). Both notions, defined in Zimmermann *et al.* (2006), have interesting advantages and drawbacks, as pointed out by the authors themselves. Basically, V-alignments are less expressive than W-alignments, while the latter has a quite complex algebra and composition definition. W-alignments are closer to our approach. The categorical notion underlying V- and W-alignments is known as a *span diagram*, and is built externally to the defined category. Intuitively, this means that A in the V-alignment $O_1 \rightarrow A \rightarrow O_2$, is not of the same type as O_i, $i = 1,2$. This is a major difference to our approach, where the entities and their context are ontologies within the same category. Moreover, our algebra is also simpler, because our definitions of alignment are limits and co-limits in a comma-category over the category of ontologies. This is the main reason why our approach takes advantage of the dualization of notions, which is so important in category theory. Moreover, each morphism in our category represents a set of parallel morphisms in the category of W-alignments. Thus, using contexts we removed the main drawbacks on composition of W-alignments.

In Borgida and Serafini (2003), a purely logical approach to deal with modular ontologies is taken. Following the approach to distributed first-order logic proposed in previous works, the

https://doi.org/10.1017/S0269888914000046 Published online by Cambridge University Press

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 an external ontology.

The DDL approach is an interesting and elegant solution for the general problem of modularizing ontologies. It is somehow related to our approach, but we focus 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. 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 performs the integration at execution time, what implies in a loss of efficiency. 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.

There are several other works that deal with formalization of context modeling using different theoretical approaches. In Cafezeiro *et al.* (2008b), we presented a detailed comparison between our approach and Context UNITY (Julien *et al.*, 2003; Roman *et al.*, 2004), *Context-Aware Action System* (Yan & Sere, 2004) and Bigraphical Reactive System (BRS) Models (Birkedal *et al.*, 2005). Beside these, we cite CommUnity (Lopes & Fiadeiro, 2005) who as well as our approach uses Category Theory as theoretical basis to express integration and contexts. It also emphazises the separation between systems behavior and context modeling. 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 approach to support the specification of context modeling of context-aware systems (e.g. mobile or ubiquitous systems). We follow the direction of much of the works (Julien & Roman, 2002; Lopes & Fiadeiro, 2005) 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 approach is based on the Algebra of Contextualized Ontologies presented in Cafezeiro *et al.* (2008a, 2008b), 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 cope, at the same time, with different levels of abstractions through related diagrams, what make possible the

[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.

https://doi.org/10.1017/S0269888914000046 Published online by Cambridge University Press

construction of modular specifications, such that the complexity of a context-system gets decomposed into manageable pieces of specification. 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.

Acknowledgment

This work was partially supported by research grants and scholarships from the Brazilian Council for Technological and Scientific Development CNPq, numbers 303427/2010-8 and 310253/2011-0.

References

Baader, F., Calvanese, D., McGuinness, D. L., Nardi, D. & Patel-Schneider, P. F. (eds). 2003. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press.

Birkedal, L., Debois, S., Elsborg, E., Hildebrandt, T. & Niss, H. 2005. Bigraphical Models of Context-aware Systems. Technical report, TR-2005-74, The IT University of Copenhagen.

Borgida, A. & Serafini, L. 2003. Distributed description logics: assimilating information from peer sources. *Journal of Data Semantics* **1**, 153–184.

Cafezeiro, I., Haeusler, E. H. & Rademaker, A. 2008a. Ontology and context. In *Proceedings of COMOREA 2008*, Hong Kong, 417–422.

Cafezeiro, I., Viterbo, J., Rademaker, A., Haeusler, E. H. & Endler, M. 2008b. A formal framework for understanding context-aware behavior in ubiquitous computing. In *Proceedings of ISOLA 2008*, Kassandra, Chalkidiki, Greece.

Cafezeiro, I., Viterbo, J., Rademaker, A., Hauesler, E. H. & Endler, M. 2009. Designing ubiquitous applications: proposal of a specification environment. In *MUCS'09*, 17–26, ACM, New York, NY, USA.

Julien, C., Payton, J. & Roman, G. C. 2003. Reasoning about context-awareness in the presence of mobility. In *Proceedings of FOCLASA03*, Marseille, France.

Julien, C. & Roman, G. C. 2002. Egocentric context-aware programming in ad hoc mobile environments, In *SIGSOFT '02/FSE-10*, 21–30, ACM, New York, NY, USA.

Lopes, A. & Fiadeiro, L. (eds) 2005. Context-awareness in software architectures. In *Proceedings of EWSA 2005*, Pisa, Italy. Springer-Verlag.

Maedche, E. & Staab, S. 2001. Ontology learning for the semantic web. *IEEE Intelligent Systems* **16**, 72–79.

Rademaker, A. & Haeusler, E. H. 2008. Toward short and structural reasoning explanations: a sequent calculus approach. In *Advances in Artificial Intelligence - SBIA 2008*, Zaverucha, G. & da Costa, A. C. P. L. (eds), Lecture Notes in Artificial Intelligence **5249**, 167–176. Springer.

Roman, G. C., Julien, C. & Payton, J. 2004. A formal treatment of context awareness. In *Proceedings of FASE'04*, Lecture Notes in Artificial Intelligence **2984**, 12–36. Barcelona, Spain.

Weiser, M. 1991. The computer for the twenty-first century. *Scientific American* **265**, 94–104.

Yan, L. & Sere, K. 2004. A Formalism for context-aware mobile computing. In *Proceedings of 3rd IEEE HeteroPar*, Cork, Ireland.

Zimmermann, A., Krötzsch, M., Euzenat, J. & Hitzler, P. 2006. Formalizing ontology alignment and its operations with category theory. In *Proceedings of FOIS 2006* in Frontiers in Artificial Intelligence and Applications, Baltimore, MD, USA, 277–288. IOS Press.

https://doi.org/10.1017/S0269888914000046 Published online by Cambridge University Press