A formal framework for understanding context-aware behavior in ubiquitous computing

Isabel Cafezeiro\textsuperscript{1}, José Viterbo\textsuperscript{2}, Alexandre Rademaker\textsuperscript{2}, Edward Hermann Haeusler\textsuperscript{2}, Markus Endler\textsuperscript{2}

\textsuperscript{1} Departamento de Ciência da Computação
Universidade Federal Fluminense
Rua Passo da Patrícia, 156 - Bloco E - 3 andar, Boa Viagem
24210-240 – Niterói – Brasil

\textsuperscript{2} Departamento de Informática
Pontifícia Universidade Católica do Rio de Janeiro
Rua Marquês de São Vicente 225, Gávea
22453-900 – Rio de Janeiro – Brasil

Abstract. A formal framework to contextualize ontologies was proposed providing several ways of composing ontologies, contexts or both. This algebra gave flexibility to model applications in which the meaning of an entity depends on environment constraints or where dynamic changes in the environment have to be considered. In this article we consider this formal algebra to formalize the problem of interpreting context information in ubiquitous systems, in order to verify not only how the formal approach might contribute with ubiquitous computing, but also how this concrete application domain may contribute with the formal framework.

1 Introduction

Before using a formal model, method or language for a specific problem domain, it is worth thinking about the expected benefit and the potential risks of this endeavor. In fact, formalization usually helps to develop a better understanding of the problem domain and its scope, as well as clearly define the major concepts involved. Hence, by presenting an applied problem in terms of a more abstract theory it is sometimes possible to directly adapt results of the theoretical framework to the applied problem, so as to yield interesting, and previously unforeseen, results, such as about the inherent complexity of a problem.

In [1] we proposed a formal framework to contextualize ontologies, providing several ways of composing ontologies, contexts or both. This algebra was flexible enough for modeling applications in which the meaning of an entity depends on environment constraints or where dynamic changes in the environment should be considered. This algebra emphasized the relationships of contexts with entities, considering that contexts are essential to clarify the meaning of entities, and that applications that consider dynamic changes of the environment require
new forms of representing the context. In this article we use this formal algebra to formalize the problem of interpreting context information in ubiquitous computing systems.

Ubiquitous computing is a particularly interesting domain for applying this formal algebra. As a fundamental requirement, ubiquitous applications must capable of responding to dynamic changes in their environments with minimal human interference. Users should be able to take full advantage of the local capabilities within a given environment and be able to seamlessly roam between several environments, even as resources (like available bandwidth) change [7]. Hence, ubiquitous computing systems strongly rely on context data.

Through this experiment we intend to show not only how the formal approach may contribute with a specific application domain, but also how a concrete application domain may contribute with the formal framework, either minoring the gap between the theoretical framework and its possible applications or validating its mechanisms.

This article is organised as follows. In Section 2 we discuss the algebra of contextualized ontologies. In Section 3 we describe ubiquitous environments and present a specific scenario. In Section 4 we apply the algebra to formalize the ontologies discussed in the scenario. Finally, in Section 5 we present our conclusions.

The formal framework that we consider in this article is founded in Category Theory [1], [2]. In [1] and [2] the reader can find the formalization of the algebra and associations between categorical concepts and the ontology terminology. Along this article, concepts are presented in an informal way, avoiding the use of the categorical jargon. For categorical definitions and proves we guide the reader to [1] and [2].

2 The algebra of contextualized ontologies

The algebra of contextualized ontologies are designed for applications where additional information is required in order to describe an entity. This information, that we call context, may be some kind of meta-data or any information related to the entity but not particular to it. This is the case of ubiquitous computing applications [3] [9]. Under this paradigm, information concerning either physical or computational environment is a relevant part of the application. Besides, the overall information available for an application — i.e. the context where it is immersed — constantly suffers dynamic changes.

The algebra of contextualized ontologies is founded in two basic features: (i) a uniform representation of entities and context and (ii) the emphasis on the relationship. Concerning (i), we use ontologies for representing both entities and contexts. This enhances the flexibility of the framework avoiding to determine a priori the role of an ontology: an ontology may represent an entity, a context or even both an entity and a context. Concerning (ii), the framework puts the focus on the relationship among the components of a systems and not on the components themselves. In this way, the internal constitution of an entity is protected,
and descriptions are built in a modular and reusable way. The benefits of emphasizing relationships are similar to those well known in systems constructions since the 70's: *Every module (...) is characterized by its knowledge of a design decision which it hides from all others. Its interface or definition was chosen to reveal as little as possible about its inner workings.* [?] The combination of (i) and (ii) makes possible the reuse of descriptions in a wide sense. Also, as it is the relationship that determines, at any time, the role of an ontology as entity or context the meaning of the subject being described is given by a net of relationships, what enables more accurate descriptions.

2.1 Contextualized Ontologies

By ontology we refer to a structure composed by concepts organized in a taxonomy, relations that determine non-taxonomical relationships, and logical axioms that set restrictions among relationships. The axioms are given in some expressive language whose model-theoretic semantics provides meaning.

Contextualized Ontologies are described as a structure preserving link between two ontologies. The source of the link is the entity and the target is the context. By *structure preserving* we mean that the context respects the hierarchical structure and the ontological relations of the entity. In other words, the entity is coherent with respect to its context. Formally, this means that if an ontology $O$ has a relation $f(c_1, c_2)$ where $c_1, c_2$ are concepts of the ontology. Then a link $F : O \rightarrow O'$ from $O$ to a context $O'$ is such that $F(f(c_1, c_2)) = F(f)[F(c_1), F(c_2)]$.

In order to avoid violating internal constitution of entities, few constraints must be stated about links: (i) there is an identity link for any entity or context, that maps the entity/context to itself. Thus the entity may be viewed as a (non-informative) context of itself; (ii) an entity is called *domain* of a link, while a context is called *codomain* of a link; (iii) links can be composed in an associative way if the codomain of the first is the domain of the second. The notation of a triple (entity, link, context), also represented by $e \rightarrow c$, will used any time we want to identify the ontologies that act as entity or context in a contextualized ontology. We will use the symbol “$\circ$” to denote the associative composition of contextualized ontologies.

In the sequel we present modular constructs that can be applied to contextualized entities, in order to coherently combine entities, contexts or both. We divide the operations in three classes: Entity Integration, Context Integration and Combined Integration. We use the term “component” to refer to concepts or relations of ontologies.

**Entity Integration.** (Figure 1, A) Operations in this class have the purpose of integrating entities ($E_1$ and $E_2$) that share the same context: $E_1 \rightarrow C_{Med} \leftarrow E_2$. As entities are coherent with respect to their context, the integration has the context as mediator. The result is 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 is mapped in the same component of the context.

**Context Integration.** (Figure 1, B) These operations consider situations where a single entity $E_{Med}$ has more than one context ($C_1$ and $C_2$): $C_1 \leftarrow E_{Med} \rightarrow C_2$. The context integration produces a new context $C_1 \rightarrow C \leftarrow C_2$ that combines information of the original context preserving the coherence with the entity. This operation can be used in situations where a single entity can be viewed in many ways, according to the considered context. The integration performs the amalgamated union of contexts, collapsing components that are images of the same component in the original entity.

**Combined Integration.** This class of operations embodies two subclasses: relative intersection and collapsed union. They consider the need to integrate the contextualize ontology as a whole, without making distinction between entity or context.

**Relative Intersection.** (Figure 1, C) Is the intersection of two contextualized ontologies mediated by a third contextualized ontology. It produces a new contextualized ontology having just the components of the originals that are mapped in the mediator.

**Collapsed Union.** (Figure 1, D) Is the amalgamated union of two contextualized ontologies mediated by a third contextualized ontology. It produces a new contextualized ontology having all components of the original but collapsing those components of the original that are image of the same component of the mediator.
3 Ubiquitous Environments

In the vision of ubiquitous computing, computer systems will seamlessly be incorporated into our everyday lives, providing services and information anytime and anywhere [9]. Compared to traditional distributed systems, ubiquitous computing systems feature increased dynamism and heterogeneity [6]. The underlying ubiquitous computing infrastructures are more complex and bring into the foreground issues such as user mobility, device disconnections, join and leave of devices, heterogeneous networks, as well as the need to integrate the physical environment with the computing infrastructure [2].

A fundamental characteristic of a software infrastructure for ubiquitous applications is context-awareness, which is the ability of a system to sense the current environment and autonomously perform appropriate adaptations in regard to its optimal operation, general behavior and user interaction. When a user enters a new context, it is desirable that the applications on his devices be able to adapt to the new situation, and the environment be able to adapt its services to the presence of the new user.

Ontology has been widely adopted for representing context information in ubiquitous systems. It has not only the advantage of enabling the reuse and sharing of common knowledge among several applications [5], but also of allowing the use of logic reasoning mechanisms to deduce high-level contextual information [8]. In the following subsection we describe a simple scenario — based on the Campus Project [4] —, which illustrates the use of context in a ubiquitous environment, and highlights some concepts such as location-specific context, reasoning, heterogeneous contexts and semantic mediation.

3.1 Scenario

Silva is a Brazilian professor and researcher who works at PUC-Rio, in the Computer Science Department. At present, Silva is participating in the Campus project, which aims to develop a middleware that provides services to collect and interpret context information, in support to other ubiquitous services and applications. Silva carries with him his smart phone and his notebook, and both devices host some context-aware applications that respond different to situations, according to his preferences and environment conditions.

When he arrives at PUC-Rio, an Ambient Management Service (AMS) registers his smart phone ($SM_{Silva}$) and determines that it belongs to him. The system verifies that Silva works there as a professor and sets his workspace (1). This system also informs other members of Silva’s team about Silva’s arrival (2). A Personal Agenda application running on $SM_{Silva}$ contacts the context infrastructure to be notified about the beginning of each event involving the whole project team, based on the project schedule and the location (3). Another application on $SM_{Silva}$, a Configuration Management Service, requests to be notified whenever Silva is in a room in which an activity has started, so that it may set the smart phone to silent, and as soon as the activity ends, switch it back to the ring mode. But if Silva’s wife sends him a message during the
meeting, he wishes that the phone vibrates so that he can discreetly check the subject (4).

From the example proposed above, we may see that the ubiquitous applications described rely on a wide variety of context information to trigger their actions. While the Ambient Management Service and the Personal Agenda are concerned with context information that describe Silva’s role in the institution, the Configuration Management Service take into consideration also Silva’s personal preferences. Thus, we notice that the context that fully describes Silva comprises not only the context that describes PUC-Rio, but also the context that describes the Campus Project and the context that describe Silva’s personal preferences and features. When Silva is at home or somewhere else other than at PUC-Rio — e.g. at an Airport (5) —, the Configuration Management Service will be imerse in an different overall context. In such cases, formalization may helps to describe and understand how different contexts form a specific combined view.

Now lets suppose that Silva is visiting LIP6 with several other researchers and he carries with him his smart phone and his notebook, both executing the Campus middleware services. Their purpose is to have joint workshops related to a collaboration project. When Silva arrives at LIP6, Wi-Fi and GPS enabled $SM_P_{Silva}$ connects to the network, and using the current GPS data, queries a location service to find out that its user (Silva) is at LIP6 (6). It then determines that this university is a partner institution of PUC-Rio; obtains the IP address of the Ambient Management Service at LIP6 (AMS’) and registers with it, indicating the user’s identity and preferences. The Ambient Management Service registers $SM_P_{Silva}$ and determines that it belongs to Silva, a visiting professor from PUC-Rio. The system verifies that Silva is involved with the collaboration project and sets a workspace for him (7).

Notice that when the Personal Agenda and the Configuration Management Service interact with the Ambient’s local context provider at LIP-6, while Silva is identified as a visitor at that institution, he can still be perceived as a professor at PUC-Rio. Hence, supposing that only professors can have access to prints at LIP6, when setting Silva’s workspace, AMS’ will recognize this permission and configure his applications to use a printer. Once again, we identify that one of the main requirements of ubiquitous systems is the ability to adapt services/behaviors to the current context view. Again formalization may be useful to describe a correspondence between different contexts in the form of a resulting aligned view.

4 Formalizing the application

4.1 High level diagrams

In this section, we refer to the numbers that appear in section 3.1 to draw high level diagrams of the scene.

Consider that Silva, PUC, LIP6, Campus Project and Airport are ontologies that, describe, respectively, personal information about Professor Silva, PUC-
Rio and LIP6 hierarchical structure, and information about the Campus Project and a given Airport. These ontologies are not contextualized. Their contexts will appear as we proceed in the construction of the formal model.

We start by (1), when the Ambient Management Service (AMS) registers Professor Silva’s smart phone. This process concerns the integration of the ontologies Silva and PUC with respect to the smart phone of Professor Silva. We construct a very simple ontology: $SMP_{Silva}$ to be contextualized in Silva and PUC. This means that concepts and relations of $SMP_{Silva}$ will be linked into correspondents of Silva and PUC, respecting the structure of the ontologies. $SMP_{Silva}$ will act as mediator of Silva and PUC in a context integration Silva $\overset{AMS}{\rightarrow}$ SMP$_{Silva}$ $\overset{AMS}{\rightarrow}$ PUC. The integration will result a new ontology that we will name SilvaAtPUC. It will embody all components of Silva, all components of PUC, and will have the images of concepts of the mediator $SMP_{Silva}$ collapsed. Operating in this integrated context (SilvaAtPUC), AMS will have enough information to identify the presence of Professor Silva at PUC (Figure 2). A similar diagram can be considered for each member of the project that is present at the moment.

\[\text{Fig. 2. Considering the ontology of Professor Silva and the ontology of PUC, AMS generates the ontology SilvaAtPUC where it can reason about the presence of Professor Silva at PUC.}\]

Then, in (2), AMS informs other members of Silva’s team about his arrival. Considering that, for any member $Prof_i$, a context integration $Prof_i \overset{AMS}{\rightarrow} SMP_{Prof_i} \overset{AMS}{\rightarrow} PUC$ has been generated, the entity integration of each $SMP_{Prof_i}$ and $SMP_{Silva}$ under the context of PUC will make the connection among the smart phones of the $i$ professors of PUC and the smart phone of Professor Silva (Figure 3). The resulting entity will be composed by the identifications of the smart phone of each professor.

In (3), the Personal Agenda (PA) of Silva’s smart phone contacts the Campus Project Agenda to be notified about events. The entity integration $SMP_{Silva} \overset{PA}{\rightarrow} CampusProject \overset{PA}{\rightarrow} SMP_{Prof_i}$ results in the synchronization of professors with respect to the Campus Project agenda. In the resulting ontology the Personal
Agenda can process information about events in which professors $i$ and Silva take part (Figure 4).

In (4) the Configuration Management Service (CMS) (running on $SMP_{Silva}$) requests to the Campus Project Agenda to be notified when any activity is about to start. AMS is aware of the location of Professor Silva at PUC, and hence of his presence in a room where a Campus activity is being held. It also considers Silva’s personal information in order to properly configure his phone alarm. A context integration $CMS_{CampusProject} \xrightarrow{CMS} SMP_{Silva} \xrightarrow{CMS} Silva$ results in a context $SilvaAtCampus$ which combine personal information about Silva and the present Campus activity in which he is involved (Figure 5). Similar situation occurs when Silva is at somewhere else, as at the airport, for instance (5). The context integration $CMS_{Airport} \xrightarrow{CMS} SMP_{Silva} \xrightarrow{CMS} Silva$ results in the context $SilvaAtAirport$ wherein the CMS can configure his phone alarm according to his preferences.

Then (6) Silva is visiting LIP6, where he is registered as a visitor researcher. Within the context $SilvaAtLIP6$ that results from integration $AMS_{Silva} \xrightarrow{AMS} SMP_{Silva} \xrightarrow{AMS} LIP6$, AMS can properly set the professor’s workspace. But some of Silva’s permissions on the use of resources come from the fact that he is a Professor at
CMS considers personal information about Silva and his physical position at the Campus.

PUC (7), thus, information about his status at PUC must also be taken into account. The context integration $LIP6 \xrightarrow{AMS} SMPSilva \xrightarrow{AMS} PUC$ generates a context where AMS can find information about Silva as a PUC professor and as a LIP6 visitor researcher in the joint project LIP6/PUC (base square of Figure 6).

The context integration $SilvaAtLip6 \xrightarrow{AMS} Silva \xrightarrow{AMS} SilvaAtPUC$ generates a context where AMS can find not only information about Silva as a PUC professor or as a LIP6 visitor researcher, but also personal information about Silva (top square of Figure 6). Note that figure 6 also pictures a combined integration: the collapsed union of the contextualized entities $LIP6 \rightarrow SilvaAtLIP6$, $PUC \rightarrow SilvaAtPUC$ mediated by $SMPSilva \rightarrow Silva$.

### 4.2 A zoom into ontologies and morphisms

As the limitation of space forbids the detailed description of the whole scenario, we select two diagrams of the previous section to illustrate how this framework provides the required information to adapt services or behaviors due to changes of context. First, we consider a situation in which information coming from one context makes possible to make decisions about an entity on a different context. For instance, (3), Professor Silva is allowed to use the printer as a consequence of the fact that, at PUC, he is a professor. In addition to this, suppose that AMS would like to make available for Professor Silva the publications of LIP6 that are related to his production. For this, AMS must be aware of Professor Silva’s production.

Considering the base square of figure 6, the mediator $SMPSilva$ of the context integration $LIP6 \xrightarrow{AMS} SMPSilva \xrightarrow{AMS} PUC$ must capture the fact that Silva is a professor and properly map this information in the ontology of LIP6. Figure 7 pictures the ontology for LIP6 and PUC and shows this alignment.

Note that, as the concept Professor at PUC is related to Research at LIP6, the relation $HasAcces(Researcher, Printer)$ will hold for Professor Silva and Printer in the resulting context (in Figure 8). Also, note that, in this resulting context information about Professor Silva’s production is available to be used by AMS.

Secondly, we show how the integration can filter information in order to affect just a selected set of entities. We consider the situation (3), where the
Fig. 6. Each face of the cube shows a context integration: in $LIP6 \xrightarrow{AMS} SMP_{Silva} \xrightarrow{AMS} Silva$, AMS configures Silva at LIP6. In $Silva \xleftarrow{AMS} SMP_{Silva} \xrightarrow{AMS} PUC$, AMS configures Silva at PUC. At the base $LIP6 \xrightarrow{AMS} SMP_{Silva} \xrightarrow{AMS} PUC$, AMS integrates PUC and LIP6 Silva according to the SMP of Silva. At the top $Silva_{LIP6} \xleftarrow{AMS} Silva \xrightarrow{AMS} Silva_{PUC}$, AMS integrates PUC and LIP6 Silva according to personal information of Silva. The cube can also be considered as the collapsed union of the contextualized entities $LIP6 \rightarrow Silva_{AtLIP6}$, $PUC \rightarrow Silva_{AtPUC}$ mediated by $SMP_{Silva} \rightarrow Silva$. 
Fig. 7. Alignment of LIP6 and PUC under the mediation of $SM_{P_{Silva}}$. The mediator captures the fact that Silva is a professor and properly map this information in the ontology of LIP6.
Fig. 8. The context integration of the alignment of figure 7: the relation HasAccess(Researcher, Printer) holds for Professor Silva and Printer and information about Professor Silva’s production is available.
Personal Agenda of Silva’s smart phone contacts the Campus Project Agenda to be notified about events. Diagram of figure 4 pictures this situation, showing the integration of SMP of Professor i and SMP of Professor Silva under the context of the Campus Project. Figure 9 shows the alignment of SMP of Professor i and SMP of Professor Silva with respect to the context of the Campus Project. Figure 10 shows the resulting entity, in which appears only the events that both take part.

Fig. 9. The alignment of SMP of Professor i and SMP of Professor Silva with respect to the agenda of the Campus Project. Event 1 and 2 are in the agenda of Professor Silva. Event 2 and 3 are in the agenda of Professor i.

5 Conclusions

A formalization is usually also embedded in a wider theoretical framework, e.g. Category Theory, in our case, which has its theorems and results, some of which are very powerful. However, one should also be aware of the limitations and potential risks of applying formal methods to a concrete problem. When we use a formal model for any subject we always abstract from some issues or entities which apparently seem less relevant. In real systems these issues might well have a significant impact on the real system’s behavior, and should ideally be accounted for. Hence, whenever we develop a formal model of a system, there is
always a trade-off between the model’s degree of realism, its complexity and its underlying set of applicable basic results.

References

