MARVIN
Modeling Environments with Ubiquitous Computing

Caio Stein D'Agostini and Patricia Vilain
Departamento de Informtica e Estatstica, Universidade Federal de Santa Catarina
Caixa Postal 476 – 88.040-900, Florianpolis, SC, Brazil

Keywords: Ubiquitous, pervasive, modeling, context.

Abstract: When different devices and applications start to work with one another, physical and behavioral aspects from the environment have to be taken into account when designing a system. It is impractical, if not impossible, to list all possible devices, applications and new characteristics and behaviors that might emerge as result from the autonomy of each part from the parts of the system, different data interpretation and self-organization capability. In order to cope with this difficulty, this paper combines the contributions from different works as a way to improve the modeling process of this type of computational environment.

1 INTRODUCTION

The concept of ubiquitous computing was first developed by Mark Weiser in the year of 1988 (Weiser, 1994). The ideia is making tools invisible to user, keeping the user's focus on the task been done and not on the tool. To make ubiquitous computing possible, the computers should be integrated into the real world since the first stages of applications development. Ubiquitous appllications should be capable to use information about the current context to adapt themselves. Those contexts are defined by information that, if available, can alter the result of an action. (Satyanarayanan, 2001) defines context information as attributes that, when available, can be used to make decisions without the need to interrupt the user frequently.

The analysis and design methods and the modeling tools currently available are usually insufficient to construct ubiquitous systems (Ingstrup, 2003). This results on time consuming *ad-hoc* developments (Sheng and Benatallah, 2005). To provide better design conditions, some aspects that need attention during the modeling process are listed: physical and digital representantion of objects in the environment; representation of spatial aspects (size, area, etc); identification of the information needed by each service; capability to model how the system affects the environment and how the environment affects the system; and the identification of emergent behavioue and how to treat it. These items are explained in details in (D'Agostini, 2007).

The remainder of the paper is organized as follows. Section 2 shows the proposed model, composed of several other models. In Section 3, a case study is presented. Finally, Section 4 highlights our conclusions.

2 MARVIN

This section presents a short description of model proposed. This model is named MARVIn (Portuguese for 'Model for Integrated Real/Virtual Environments'). The model is not indended to be used by itself, as it only provides some solutions to cover some specific modeling aspects referring to ubiquitous computing environments.

2.1 ContextUML: The Starting Point

Aiming to solve the problems presented in the previous section, works from different specific research areas were studied. The work that provided the greatest contribution and a was used as the base for this work was ContextUML (Sheng and Benatallah, 2005), which provides a meta-model class diagram capable of representing the dependency of an application towards an specific context.

ContextUML explicitly separates the operations of a service from the service, explicitly identifies different relevant context information by con-

Stein D'Agostini C. and Vilain P. (2008).
MARVIN - Modeling Environments with Ubiquitous Computing.
In *Proceedings of the Tenth International Conference on Enterprise Information Systems - ISAS*, pages 262-265
DOI: 10.5220/0001698502620265

text objects and binds objects (including services) to these contexts, throught ContextAwarenessMechanisms (Sheng and Benatallah, 2005). This makes possible, for example, to easily identify two services that might interfere with the other's functioning.

However, ContextUML does not covers aspects such as physical aspects of the system, and some of its contributions can be further extended with contributions from other research projetc.

2.2 Extending ContextUML: Other Complementary Works

From (May, 2003), we complemented ContextUML (Sheng and Benatallah, 2005) with the notion of Tangible Objects. These are objects that have a representation both on the physical world (its physical object) and on the virtual/digital world (the application). The interaction of the Tangible Objects in the system and with the physical world is done throught the use of Spatial Diagrams (Ingstrup, 2003). These diagrams are maps featuring the location of relevant physical and tangible objects in the environments, areas of coverage (sensors coverage for example), etc.

While (Sheng and Benatallah, 2005) and (May, 2003) deal with the static views of the environment, the aspects related to the dynamic views of the systems are left uncovered. These includes time restrictions, context evolution and changes on services' availability, etc. Some contributions to cover the dynamic aspects of the applications in the environment come from (Ingstrup, 2003) (Castro et al., 2006) (Derntl and Hummel, 2005) (Chung and Nixon, 1995) and (Damasceno et al., 2006). The goal when dealing with the dynamics of the system, is not to model all the different actions and situations that can be contemplated for the system. Instead, the designer should worry about the situations he or she considers that should not happen. We take this approach because we do not want to model all the different applications in the environment, but how they can work togheter and the results of those interactions.

These solutions are presented in seven steps, but this does not imply the need to follow the same ordering. These steps are to be applied using the contributions from ContextUML, but considering some modifications on its metamodel in order to better model the relation between services and operations. Figure 1 shows this new metamodel. The extensions made to (Sheng and Benatallah, 2005) are in different color.

Each step presents one or some contribution from related workd and how they contribute to solve the limitations from their own solutions. To illustrate each step, a case study was chosen.

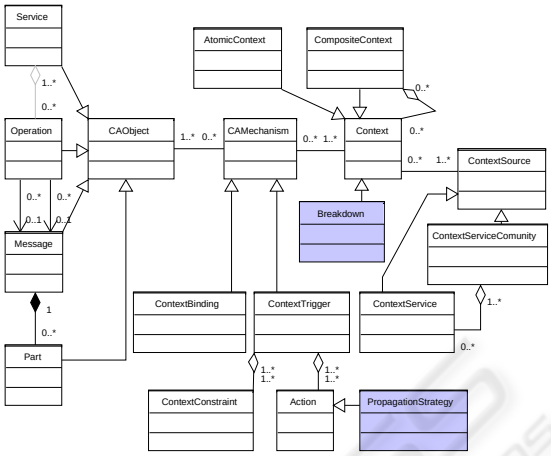

Figure 1: MARVIn's metamodel.

3 CASE STUDY: INSTANT MESSENGER APPLICATION

This application is supposed to locate the addressee's location for a message so it can be forwarded to the instant messenger client application closer to the user at the time. By keeping track of the user's location, there is no need for the user to log-in and logout of a client application everytime it moves around the environment.

The example application is constructed coordinating the following type of services available in the environment: a Client application that acts as an interface to the user; a Message Router capable os storing a message and forwarding to the addressee once its location is known; a Locator service capable of providing a target's location; and Authenticator service so user's can gain access to the other services; an Entrance service which keeps track of users entering and leaving the environment.

All this services can are independent (their implementation) from the others, but they need to share information among them to work properly. On this section the MARVIn's steps are applied to model how services and devices inside a research laboratory are coordinated to provide an instant messenger application. Note that the resources originally provided by ContextUML (Sheng and Benatallah, 2005) are not illustrated here.

1. **Creation of a graph of goals for identification of conflicting requirements and emergent behaviours.** The instant messenger application modeled here is supposed to be precise, meaning that messages sent to a user are supposed to be received by the closest computer executing a client

service at the time. But it is also supposed to be cheap and easy to implement, meaning no expensive or sophisticated sensors and devices are to be used. Figure 2 shows how these goals relate.

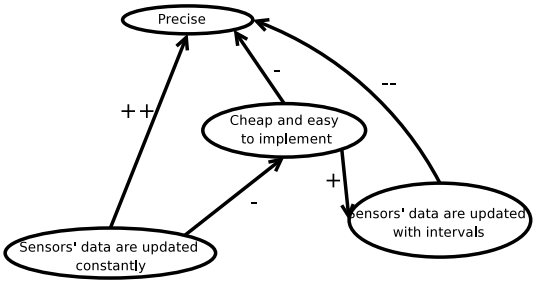

Figure 2: Decomposed objectives to predict the system behaviour in the presence of other services (D'Agostini, 2007).

2. **Identification of all the objects (physical or virtual) relevant to the application(s).** Besides the already cited services (client, router, locator, authenticator and entrance), the instant messenger application needs to know about the devices available to execute the client application. Those are the desktop and personal computers. It also need to know about the users (as real people).
3. **Mapping the localization of all relevant objects.** The router service needs information about the computers' placement in the room, as well about the users' location. These objects are tangible objects, since they are relevant both in the physical as in the digital world. Figure 3 shows the mapping of the environment and identifies the tangible objects.

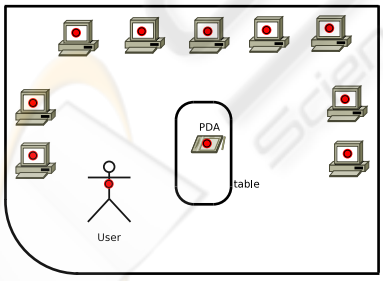

Figure 3: Map of the environment with the instant messenger application (D'Agostini, 2007).

4. **Identification of the different contextual information** The messenger needs to know about location, both of the users and the computers. It also needs to know about the users' current status (logged-in or out), which informs if they are present at the environment and authorized to use the available services. Another information is related to the messages that are exchanged between users.
5. **Grouping of the services and devices based on their functionalities** For this case study there is only one type of each service (all computers have the same client application, there is only one authenticator service, etc), but to illustrate this step it will be considered as if several types of location and authentication related services where available, resulting in these scopes:
   - Authentication: Authentication service executing on the computers that uses logs-in and a biometric device that identifies the users by their fingerprints.
   - Location: Image tracking service, which tracks the user through a camera and a bluetooth service that detects the users' cell-phones location.
6. **Identification of possible system's breakdowns and strategies to deal with it.** The breakdown and propagation strategy identified for this example are related to the provision of information about location. Next, Figureetapa6 presentes a brief description of a use case describing the message routing, which depends on the location of the addressee user.

```
>>Message Routing
1.Message is  sent to a user in the room
2.Router requests the addresse's location
3.The image tracking service receives the request
  3A. The service locates the user (go to 4)
  3B. The service can not locate because of poor
lighting conditions - BREAKDOWN - AlternativeLo-
cationProvider Strategy
4. ...
>>AlternativeLocationProvider Strategy
1.The location request is sent again, but for the
user's cell-phone(bluetooth)
2.Awaits for answer
```

Figure 4: Decomposed objectives to predict the system behaviour in the presence of other services (D'Agostini, 2007).

Figure 5 shows the breakdown and the propagation strategy relating to the service request.

7. **Identify the time restrictions needed to ensure the quality of the treated information and consequently of the provided services.** Finally, Figure 6 shows the dynamics of how one service coordinates with the others and how this process affects or is affected by the contextual information.

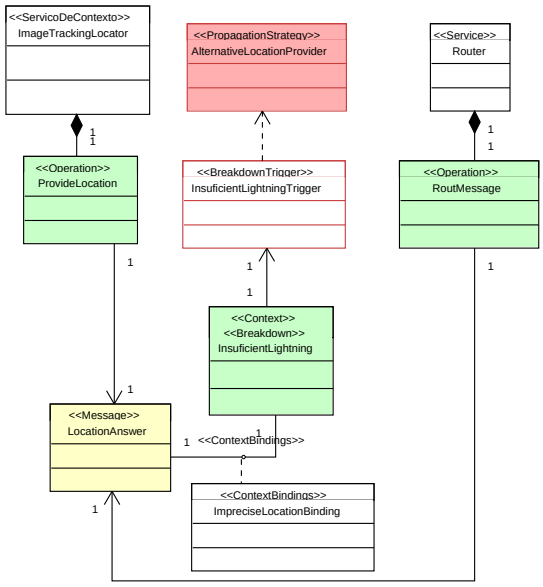

Figure 5: Service request, breakdown and propagation strategy (D'Agostini, 2007).

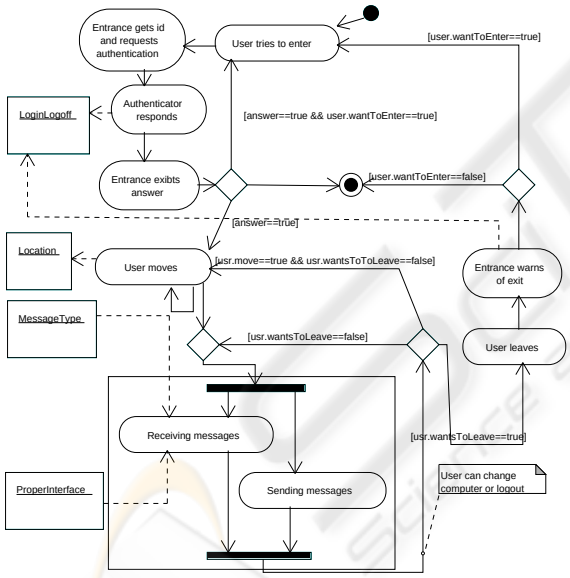

Figure 6: The relation between the actions and the contextual information (D'Agostini, 2007).

4 CONCLUSIONS

From the works studied in order to write this proposal it is easy to see that, while it is clear that traditional development techniques do not apply to complex systems built based on distributed applications and portable devices, there is a lack of development alternatives.

Even if not presented here, comparisons between different related works where done in (D'Agostini, 2007). Our proposal contributes by incorporating contributions from the several of those related works, covering both static and dinamic aspects of the applications, without the need for tools other than those already available for a software designer or programmer. The work also focus on modeling the failures and how to treat them, instead of trying to avoid them, which is not always pratical.

REFERENCES

Castro, J. F., Alencar, F., and Silva, C. (2006). Desenvolvimento de software orientado a agentes. XX Simpsio Brasileiro de Engenharia de Software. Tutorial 3 SBES.

Chung, L. and Nixon, B. A. (1995). Dealing with non-functional requirements: Three experimental studies of a process-oriented approach. In *ICSE17*, pages 25–37. ACMP.

D'Agostini, C. S. (2007). A proposal for modeling environments with ubiquitous computing. Technical report. Bachelors monografy, Universidade Federal de Santa Catarina.

Damasceno, K., Cacho, N., Garcia, A., and Lucena, C. (2006). Tratamento de excees sensvel ao contexto. In *Anais do XX Simpsio Brasileiro de Engenharia de Software (SBES)*, pages 49 – 64.

Derntl, M. and Hummel, K. (2005). Modeling context-aware e-learning scenarios.

Ingstrup, M. (2003). Modeling for pervasive computing: Organizational aspects of system analysis and designs. Master's thesis, University of Southern Denmark.

May, D. C.-M. (2003). Tango: Designing for the digitally pervasive world. Master's thesis, University of Southern Denmark.

Satyanarayanan, M. (2001). Pervasive computing: Vision and challenges. *IEEE Personal Communications*.

Sheng, Q. Z. and Benatallah, B. (2005). Contextuml: A uml-based modelling language for model-driven development of context-aware web services. *Proceedings of the International Conference of Mobile Business*.

Weiser, M. (1994). The world is not a desktop. *Interactions*, pages 7–8.