Considerations upon Interoperability on Pervasive Computing Environments

Andres Flores[1] **and Macario Polo Usaola**[2]

[1]Department of Computer Science, University of Comahue,
Buenos Aires 1400, 3400 Neuquen, Argentina. Email: aflores@uncoma.edu.ar

[3]Escuela Superior de Informática, Universidad de Castilla-La Mancha,
Paseo de la Universidad 4, Ciudad Real, España. Email: macario.polo@uclm.es

Abstract. Component-based Development is a challenging paradigm, though Pervasive Computing Environments, as a special case of such systems, carry even more complications. Applications must be kept available for users anytime, anywhere no matter the user location. This implies transparency on interconnection of components at run-time. Since components may not be previously evaluated when a user's context change happen, integration of new components might blur the actual requirements of that system. This implies to consider Interoperability at different levels. This paper reports on those challenges, also discussing the possibility to improve the description of an approach under a formal basis. Thus solution strategies can be better defined and applied.

Keywords: Component-based Software Engineering, Pervasive Computing Environments

1 Introduction

Pervasive Computing Environments (PvC) implies to come to reality the expectation of being able to interact with equipments without the restriction of a particular location where to perform the daily work. Thus users into the environment expect to have access to information anytime, anywhere, no matter the location [4, 2]. The underlying infrastructure of such environment involves many different protocols, and software/hardware platforms, where diverse small devices come to be the regular and helpful manner to proceed.

Users moving around in the environment make in-use applications to be affected, since the user interaction changes from one used device to another. This implies a run-time connection of new devices and to carry out processes to either search for similar applications inserted into that device or migrate components to fulfill such requirement.

The exposed scenario can be better understood under the following concept: "Interoperability". This is the expected quality on PvC environments under which completely separated developed components have the capability to interrelate with other components or systems. Such concept can be analysed at different levels, which involve diverse particular elements to be considered on each of them [3]. An approach working at a syntactic level is presented in [6], where the WAP protocol has been its basis. Our work is currently focused on the semantic level, which we found has been very much neglected so far. *Semantic Interoperability* implies for a component to satisfy the expected requirements according to a particular context. Considering context changes being a common feature of Pervasive Computing Environments, semantics verification becomes an important contribution for the realization of this computing paradigm at a reliable standard.

As an extra consideration, besides dynamic component integration, capabilities of the devices involved in a user task may also include very particular requisites from the underlying infrastructure which could severely complicate the effective usage of components [5].

As can be seen many are the challenges upon PvC Environments, which must be carefully distilled in order to arrive to a proper solution for each of them. This paper reports different considerations about systems into such environments and particularly from the point of view of interoperability.

Thus next sections discuss levels of interoperability, strategies to build PvC Systems, approaches to rearrange running applications with the likely need of applying assessment and selection techniques, and the benefits of using formality as an underlying support.

2 Levels of Interoperability

Interconnecting components implies the capability to exchange expressions or messages between them, which conform a sort of communication protocol. Several possible levels of Interoperability must be considered upon the requirement of understanding those exchangeable elements [3]:

- Encoding: being able to segment the representation in characters;
- Lexical: being able to segment the representation in words (or symbols);
- Syntactic: being able to structure the representation in structured sentences (or formulas or assertions);
- Semantic: being able to construct the meaning of the representation (context sensitive);
- Semiotic: being able to construct the pragmatic meaning of the representation (related to natural language).

Some work at Syntactic and Semantic levels have been proposed by different authors. In [6] is presented an approach based on testing for the WAP protocol which corresponds to a syntactic level. At a semantic level, the work in [8] uses ontologies to represent different contexts which may give meaning to concepts involved in a user daily task. Our work is based on an ontology infrastructure as well, though we intent to solve semantic interoperability by applying different testing strategies.

3 Architecture Building Strategies

Usually Component-based Development is considered a mix of activities of both Top-Down decomposition and Bottom-Up composition. At the purest form, we only refer to the latter, which pretends to assemble existing parts into a meaningful configuration that reflects predefined system requisites. The concern here involves this kind of process coming to conform a system which does not satisfy the original user requirements [9].

When systems are considered upon PvC Environments their building process corresponds to a Bottom-Up component-based development. Here the conflicting situation involves assuring for systems to adjust to continuously changing requirements. Users mobility implies to reconsider how to continue providing the expected functionality and support connections of new and possibly different devices as well. Each device may provides its own components or even migration code processes could be carried out from/to those devices. Hence every aspect related to Interoperability must be well thought-out according to levels on the previous section.

4 Adaptability Approaches

When a user change its physical location applications into the environment must be rearranged in order to enable for users to continue with their current task. From considerations on the previous section two main approaches emerge to give a likely solution:

Self-Adaptable Applications. Applications including functionality to change themselves on occurrence of a user context change. Thus the responsibility of such kind of specific behaviour fall on *providers* of those applications. We may say that the environment is laying its confidence on component providers, who may actually not be aware of the involving reliability concern. Additionally, users may come to find that only a few range of applications are available into the environment. They are those that may adjust themselves in order to make users to feel that the environment follow them around. This approach is used in [8].

Dynamic Re-Construction by the Environment. The environment is aware of changes on user context of operation and reacts re-assembling applications by selecting and assessing COTS components. Hence reliability and availability are enhanced with respect to the previous approach. The environment is developed with the spirit of supporting ubiquitousness and as such it is the ideal to coordinate processes of re-building applications on run-time by using every component that it may locate. In addition, the range of components to be utilized is not limited, since they conform the regular COTS components available in the marketplace. This is the approach we are intended to apply on our work.

5 Assessment and Selection

On a change of a user's context of operation applications may be in the need of incorporating new components. Quite much approaches on assessment and selection have been proposed when on a development stage [1]. Though PvC Environments require to carry out those processes at run-time, and alike approaches here are still a matter of research.

Our work is based on an ontology infrastructure, from where components can be evaluated. Currently we are developing an adequate method to carry out this process. Metadata inserted on components is the key to both access internal aspects without violating their encapsulation, and provide contextual information in order to perform a proper analysis. Thus the strategy is applying component-based testing techniques.

6 Rigor and Formality

The usage of formality on an infrastructure for interoperability certainly may enhance its underlying properties. Both components and their exchangeable elements can be better analysed through the use of formal specifications, since they may help disambiguating expressions. A formal model of contexts is described in [8], which is based on first order logic. Our intention is to apply Propositional Linear Temporal Logic [7], from where we may improve the description of dynamic interactions of components. In particular it will be used for specifying the testing process. From such formal framework testing strategies may be better defined and applied.

7 Conclusions and Future Work

Achieve interoperability on PvC Environments at its different levels is still a matter of research. Many strategies could be applied which carry diverse considerations which where reported on this paper.

As was exposed, our work is currently focus on Semantic Interoperability where both the understanding of exchangeable concepts and the interconection of components, is achieved by using an ontology infrastructure from where we believe is possible to certify semantic matching.

Our work will continue by developing an abstraction of an architecture for implementing testing approaches who will give support to accomplish our intent.

References

[1] Alejandra Cechich and Mario Piattini Velthuis, *Quantifying COTS Component Functional Adaptation*, ICSR'04, 8^{th} Int'l Conference on Software Reuse (Madrid, Spain), LNCS, Springer-Verlag, July 2004.

[2] D. Garlan et.al, *Software Architecture-based Adaptation for Pervasive Systems*, ARCS'02, 2002, pp. 67–82.

[3] J. Euzenat, *Towards a principled approach to Semantic Interoperability*, IJCAI'01, Workshop on Ontologies and Information Sharing (Seattle, WA, US), August 2001.

[4] A. Flores, *CSCW Systems on PvC Environments*, 5th Workshop of Researchers on Computer Science (Tandil, Argentina), May 2003.

[5] A. Flores and A. Cechich, *Quality Considerations on Ubiquitous Systems*, JCC'02, II Chilean Workshop of Software Engineering (Copiapo, Chile), November 2002.

[6] Ousmane Koné, *An Interoperability Testing Approach to Wireless Application Protocols*, JUCS, Journal of Universal Computer Science **9** (2003), no. 10, 1220–1243.

[7] Z. Manna and Pnueli A., *The temporal logic of reactive and concurrent systems (specification)*, Springer-Verlag, 1992.

[8] A. Ranganathan, R. McGrath, R. Campbell, and Mickunas D., *Ontologies in a Pervasive Computing Environment*, 18th IJCAI, Workshop on Ontologies and Distributed Systems (Acapulco, Mexico), August 2003.

[9] Rizzo Vincenzi et.al., *Component-based software quality: Methods and techniques*, LNCS, vol. 2693, ch. Component-based Software: An Overview of Testing, pp. 99–127, Springer-Verlag, 2003.