

# Pervasive Communication: The Need for Distributed Context Adaptations

Jorge Vallejos, Brecht Desmet, Pascal Costanza, Wolfgang De Meuter

Programming Technology Lab – Vrije Universiteit Brussel  
Pleinlaan 2 - 1050 Brussels - Belgium  
{jvallejo,bdesmet,pascal.costanza,wdmeuter} @vub.ac.be

**Abstract.** This paper focuses on the effects of pervasive computing on today’s software technology. We present an existing software application for communication as a case study and establish a set of requirements this application should accomplish to become a pervasive computing application. We define what a pervasive communication is and identify the need for distributed context adaptation schemes.

## 1 Introduction

In the pervasive computing paradigm, software applications vanish into their users’ surroundings, spreading their functionality across computers integrated into everyday devices [1]. This pervasive condition entails a number of new challenges which we illustrate in the following scenario.

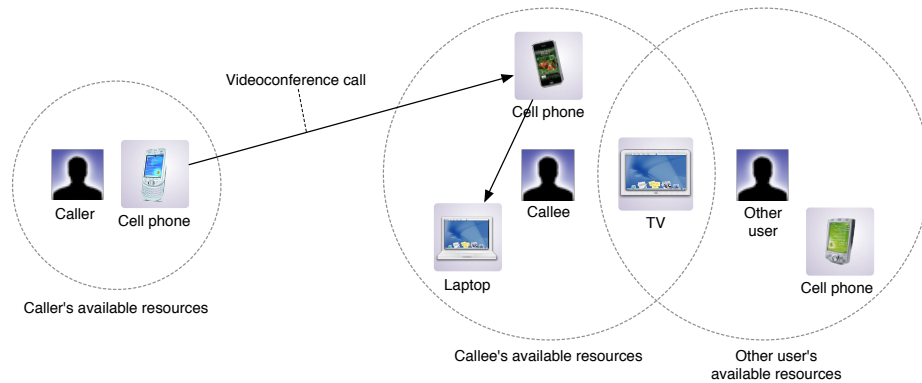
Consider the case of a software application for communication like *Skype* [2], *Google Talk* [3] or *iChat* [4]. The main property of such an application is to offer *multiple* traditional and new communication services to its user (based on text, audio and video), who only requires *one* identity to use all these services. Assume that this application – named *ContextCom* in this paper – runs in a pervasive computing environment composed of a set of devices provided with processing and communication capacity: an interactive TV, a cell phone and a laptop. Although *ContextCom* is available at any of these devices, its user may have some preferences on where to use the different communication services. For instance, the user may prefer the interactive TV for having videoconferences whereas he opts for his cell phone to chat via text messages. Additionally, the user’s preferences may be conditioned to the context in which the communication occurs. For example, if somebody else is also watching the TV at the moment the user receives a videoconference call, then he may prefer to use his laptop instead. Finally, the fact that the services of *ContextCom* are available in different devices should not imply for the user to have independent instances of this application in every device. It should be possible that the user has a single account that he can simultaneously use on all devices.

Based on the scenario above, we argue that a pervasive computing application for communication should be able to *dynamically distribute* the communication across the devices available in the user’s environment, to *adapt* this dynamic

distribution to the context, and to *preserve* the user's identity regardless of the dynamic distribution. In the following sections, we further analyse these requirements and discuss their implications for the development of such an application.

## 2 Understanding Pervasive Communication

To understand how a pervasive computing application for communication (such as *ContextCom*) differs from existing communication applications, we first need to understand what a pervasive communication is. We generically define a pervasive communication as an interaction that occurs through the environment of its participants, i.e. the devices with processing and communication capacity available in this environment (see Figure 1). This kind of communication has a number of particularities that we describe in this section.



**Fig. 1.** A pervasive communication.

**Distributed Presence of Identity** In a pervasive communication, the availability of a user is determined by the set of devices available in his surroundings. This means that the user is capable to communicate with others (e.g. to initiate or to receive a call) as long as he has at least one device available for this purpose. This also implies that in case of having more than one device available, the identity of the user should be *distributed* in all these devices so that he can, for instance, be reached by a call at any of these devices. The constitution of this identity is dynamically reconfigurable as the devices become (un)available in his environment.

In the figure above, we also observe that same device can be used by more than one user (e.g. the TV). The device, in this case, may be part of the identities of several users.

**Dynamic Context Adaptations** The devices in a pervasive communication may exhibit different behaviours. A device, for instance, may not only hold a communication (e.g. a videoconference call) but also delegate it to another device in the user's environment (e.g. the cell phone of the callee delegates the videoconference call to his laptop). This adaptation of behaviour occurs dynamically and depends on the context of the communication in which the device is involved. By context, we mean any piece of information which is computationally accessible [5]. Some examples of context information we consider in this scenario are the type of communication, the availability and hardware characteristics of the devices (e.g. the TV is the most convenient device for a videoconference with respect to video and audio quality), and the presence and preferences of the users (e.g. the laptop may be a better alternative for the videoconference if there is more than one user watching the TV).

**Coordination of Distributed Context Adaptations** A pervasive communication may also require the adaptation of several devices. Such a case requires a coordination scheme between the devices involved in the communication. For instance, while having a videoconference at the laptop, a user may want to be notified of all the incoming calls or messages at this device. This implies that if any of the other devices receives such calls or messages during the time of the videoconference, it should send a notification to the laptop.

Devices may be involved in several interactions at the same time. Since presumably these interactions require also different adaptations, there is a high probability that devices end up with adaptations that conflict with each other. In Figure 1, for example, the TV is a device shared between two users and as such it can be involved in two interactions: the user who is having a videoconference in his laptop and for which the TV should act as a notifier of the incoming calls, and another user that may have a conversation via text messages directly in the TV.

**Distributed Context Reasoning** The context of a pervasive communication is not a monolithic and homogeneous set of information, it can vary with time and from one device to another (e.g. hardware characteristics). Additionally, some context information may not, and probably should not, be known by all the devices (e.g. user preferences). The context-dependent adaptation required for a communication, therefore, cannot be decided in only on device but it requires a distributed reasoning process. In our scenario, for instance, the cell phone of the callee might not have the means to detect that this user is not watching the TV alone. Thus, this cell phone can only decide to delegate the videoconference call to the TV (as the TV has better audio and video attributes) and let the TV decide to delegate the call to the laptop.

A distributed context reasoning scheme also preserves the autonomy of the devices. For example, if the caller could decide the device that the callee should use for the videoconference, the callee would lose the possibility to discern whether and how to receive the calls.

## 2.1 Summary: The Need for Distributed Context Adaptations

In summary, in a pervasive communication (i) the identity of the users should be distributed in the devices found in their environment, (ii) the devices should dynamically adapt their behaviour to the context of the pervasive communication, (iii) the adaptations of several devices should be coordinated, and (iv) the context-dependent reasoning process should be distributed. We refer to all these conditions as the *need for distributed context adaptations*.

## 3 Ongoing Work and Discussion

We are currently working on an object-oriented programming model to address the requirements identified in the previous section. This model is a combination of previous works presented in [6] and [7]. Our intention in this model is to provide dedicated language constructs to define local and distributed context adaptations of applications, i.e. the adaptation of the behaviour of one device and the coordination of several adaptations of different devices. In this model, devices decide their adaptation as well as their participation in a distributed coordination scheme. Finally, this model also enables the distribution of user's identities between several devices.

There are still some open issues of pervasive communication and distributed context adaptations which consequences need to be further investigated. Some of these issues are listed below:

**Context beyond proximity** Different from the notion of context that is commonly described in context-aware systems and which is associated to proximity [8, 9], the context that influences a pervasive communication may be found at completely different physical locations. For instance, in the scenario of the videoconference (Figure 1), the callee may use different configurations based on the caller's identity or location [6].

**Different notions of adaptation scope** In a pervasive communication, we observe two cases of context adaptations that uses two different notions of scope. The first adaptation is related to the user's identity which scope is the user's physical surrounding. This identity changes according to context events that occur in the environment, e.g. new devices or communication types that become available. The second adaptation is the one required for a communication which scope is related to the execution time and the part of the environment this communication affects. For instance, the adaptation for a videoconference is required only during the time this videoconference occurs and may only affect some devices.

**User-assisted context adaptations** There might be situations in which the adaptation for a pervasive communication cannot be automatically decided and requires the interaction of the user. The user may also want to create new adaptations or modify the existing ones. For such cases, the application for communication should provide means to involve the user in the context reasoning process.

**Changes of context during the communication** During a communication, the devices that hold this communication may change their context conditions, e.g. if they become (un)available. Such changes may require different adaptations that the one decided at the beginning of the communication.

## References

1. Weiser, M.: The computer for the twenty-first century. *Scientific American* (1991) 94–100
2. Zennström, N., Friis, J.: Skype. <http://www.skype.com> (2007)
3. Google Inc.: Google Talk. <http://www.google.com/talk/> (2007)
4. Apple Inc.: iChat. <http://www.apple.com/macosx/features/ichat/> (2007)
5. Hirschfeld, R., Costanza, P., Nierstrasz, O.: Context-Oriented Programming. Submitted to *Journal of Object Technology*. <http://www.jot.fm> (2007)
6. Vallejos, J., Ebraert, P., Desmet, B., Cutsem, T.V., Mostinckx, S., Costanza, P.: The Context-Dependent Role Model. In Indulska, J., Raymond, K., eds.: 7th IFIP International Conference on Distributed Applications and Interoperable Systems (DAIS 2007), Paphos, Cyprus. LNCS 4531, Springer (2007)
7. Costanza, P., Hirschfeld, R.: Language Constructs for Context-Oriented Programming - An overview of ContextL. In: *Dynamic Languages Symposium*. (2005)
8. Barron, P., Cahill, V.: Using stigmergy to co-ordinate pervasive computing environments. In: *WMCSA '04: Proceedings of the Sixth IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'04)*, Washington, DC, USA, IEEE Computer Society (2004) 62–71
9. Sørensen, C.F., Wu, M., Sivaharan, T., Blair, G.S., Okanda, P., Friday, A., Duran-Limon, H.: A context-aware middleware for applications in mobile ad hoc environments. In: *MPAC '04: Proceedings of the 2nd workshop on Middleware for pervasive and ad-hoc computing*, New York, NY, USA, ACM Press (2004) 107–110