# Applications Suitability on PvC Environments



**Andres Flores**
*University of Comahue, Argentina*

**Macario Polo Usaola**
*Universidad de Castilla-La Mancha, Spain*

## INTRODUCTION

Pervasive computing (PvC) environments should support the *continuity* of users' daily tasks across dynamic changes of operative contexts. Pervasive or ubiquitous computing implies computation becoming part of the environment. Many different protocols and operating systems, as well as a variety of heterogeneous computing devices, are interrelated to allow accessing information anywhere, anytime in a secure manner (Weiser, 1991; Singh, Puradkar, & Lee, 2005; Ranganathan & Campbell, 2003).

According to the initial considerations by Weiser (1991), a PvC environment should provide the feeling of an enhanced natural human environment, which makes the computers themselves vanish into the background. Such a disappearance should be fundamentally a consequence not of technology but of human psychology, since whenever people learn something sufficiently well, they cease to be aware of it.

This means that the user's relationship to computation changes to an implicit human-computer interaction. Instead of thinking in terms of doing explicit tasks "*on the computer*"—creating documents, sending e-mail, and so on—on PvC environments individuals may behave as they normally do: moving around, using objects, seeing and talking to each other. The environment is in charge of facilitating these actions, and individuals may come to expect certain services which allow the feeling of "*continuity*" on their daily tasks (Wang & Garlan, 2000).

Users should be allowed to change their computational tasks between different operative contexts, and this could imply the use of many mobile devices that help moving around into the environment. As a result, the underlying resources to run the required applications may change from wide memory space, disk capacity, and computational power, to lower magnitudes. Such situations could make a required service or application inappropriate in the new context, with a likely necessity of supplying a proper adjustment. However, users should not perceive the surrounding environment as something that constraints their working/living activities. There should be a continuous provision of proper services or applications. Hence the environment must be provided with a mechanism for *dynamic applications suitability* (Flores & Polo, 2006).

## PERVASIVE COMPUTING ENVIRONMENTS

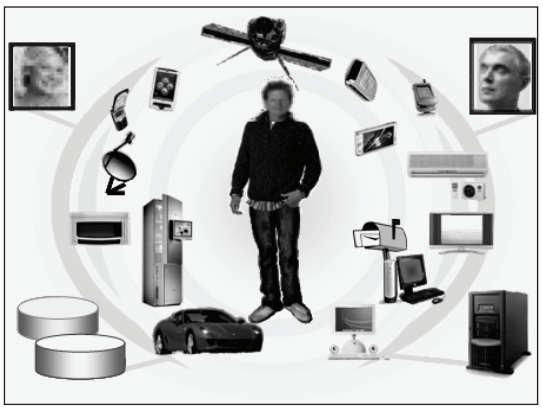
In the field of PvC there is still a misuse of some related concepts, since often PvC is used interchangeably with ubiquitous computing and mobile computing. However, nowadays consistent definitions are identified in the literature as follows (Singh et al., 2005).

Mobile computing is about elevating computing services and making them available on mobile devices using the wireless infrastructure. It focuses on reducing the size of the devices so that they can be carried anywhere or by providing access to computing capacity through high-speed networks. However, there are some limitations. The computing model does not change considerably as we move, since the devices cannot seamlessly and flexibly obtain information about the context in which the computing takes place and adjust it accordingly. The only way to accommodate the needs and possibilities of changing environments is to have users manually control and configure the applications while they move—a task most users do not want to perform.

PvC deals with acquiring context knowledge from the environment and dynamically building computing models dependent on context. That is, providing dynamic, proactive, and context-aware services to the user. It is invisible to human users and yet provides useful computing services (Singh et al., 2005). Three main aspects must be properly understood (Banavar & Bernstein, 2002). First is the way people view mobile computing devices and use them within their environments to perform tasks. A device is a portal into an application/data space, not a repository of custom software managed by the user. Second is the way applications are created and deployed to enable such tasks to be performed. An application is a means by which a user performs a task, not a piece of software that is written to exploit a device's capabilities. And third is the environment and how it is enhanced by the emergence and ubiquity of new information and functionality. The computing environment is the user's information-enhanced physical surroundings, not a virtual space that exists to store and run software.

Ubiquitous computing uses the advances in mobile computing and PvC to present a *global computing environment* where seamless and invisible access to computing resources

*Figure 1. Vision of an enhanced physical environment by ubiquitous computing*



is provided to the user. It aims to provide PvC environments to a human user as s/he moves from one location to another. Thus, it is created by sharing knowledge and information between PvC environments (Singh et al., 2005). Figure 1 shows the vision a user may have of a physical environment that is enhanced by ubiquitous computing.

Some approaches for PvC are concerned with interconnecting protocols from different hardware artifacts and devices, or solving problems of intermittent network connections and fluctuation on bandwidth. Therefore, their applications are quite general or low level, yet mainly related to communication tools which still requires a big effort for a user to accomplish a working task. Other approaches are focused on solving problems of prohibited access to information or even to a closed or restricted environment. If we consider that the environment is populated with an enormous amount of users, each intending accesses to different hardware and software resources, the security concerns increase proportionally (Kallio, Niemelä, & Latvakoski, 2004).

On the other side, there are approaches particularly concerned with providing higher level services more related to users tasks, in order to help them reduce the working effort (Roman, Ziebart, & Campbell, 2003; Becker & Schiele, 2003; Chakraborty, Joshi, Yesha, & Finin, 2006; Gaia Project, 2006; Aura Project, 2006). Most of them have been conceptualized with some sort of self-adjusted applications or by applications relying on basic services provided by the underlying platform (e.g., CORBA).

No matter how users need a transparent delivery of functionality, so they could have a sense of continued presence of the environment. Therefore, any unavailability of a required service implies that a user understand that the underlying environment cannot provide all that is needed, thus destroying the aspiration of transparency.

## SUITABILITY FOR PERVASIVE APPLICATIONS

Functionality on a PvC environment is usually shaped as a set of aggregated components that are distributed among different computing devices. On changes of availability of a given device, the involved component behavior still needs to be accessible in the appropriate form according to the updated technical situation. This generally makes users be involved on a dependency with the underlying environment and increases the complexity of its internal mechanisms (Iribarne, Troya, & Vallecillo, 2003; Warboys et al., 2005).

Applications composed of dynamically replaceable components imply the need of an appropriate integration process according to component-based software development (CBSD) (Cechich, Piattini, & Vallecillo, 2003; Flores, Augusto, Polo, & Varea, 2004). For this, an application model may provide the specification of a required functionality in the form of the aggregation of component models, as can be seen in Figure 2. A component model provides a definition to

*Figure 2. Connection of models and components to integrate an application*