Dependability in Pervasive Computing: Challenges and Chances

Frank Ortmeier, Otto-von-Guericke-Universität Magdeburg, Germany

ABSTRACT

In 1988, Marc Weiser was one of the first computer scientists who envisioned that computers would become invisible; that computing power and communication technology would become part of many objects of society’s daily life. Many modern systems would not be able without pervasive technology. Today, most such systems might be invisible or wearable, but society is either still aware of them or they only communicate to a limited extent with each other. In the near future many objects of day to day life will be equipped with some kind of computing and communication capability and people won’t be aware of it anymore. The great benefit is that they will offer citizens support and guidance in everyday life. For example, most people do not know that nice features like jam prediction and avoidance rely on feedback of the navigation system to some centralized server clusters. These analyze the data and thus predict possible traffic jams. Although, dependability issues most often form rigid limits. Because the systems are so smoothly integrated into normal life, they are expected to be robust against intended manipulations to guarantee functional requirements and/or to be traceable and understandable for the human user. In addition, the adaptive nature of many Pervasive Computing systems makes them very difficult to analyze and predict.

Keywords: Computing Technologies, Dependability, Enabler Technology, Personal Digital Assistants, Pervasive Computing

INTRODUCTION

Pervasive Computing can be seen as an enabler technology for many new and exciting applications. Indicative examples of envisaged applications include the following: A smart shopping list, which automatically adds items with low supplies to the list. A digital personal assistant, which provides useful information on the basis of the user’s current context and needs (e.g., time schedules). A medical diagnostics system, which supervises vital functions of patients at all times and therefore allows much better diagnostics. A similar system could also be used to monitor heart patients and trigger emergency calls. Even closer in the future are systems, which allow continuous monitoring and tracking of goods. Just think of smart production, logistic or transportation systems. If for example luggage is equipped with RFID tags, long and time-consuming check-in procedures could be omitted, lost luggage would be reduced and customs would be simplified. Already in a prototype state are pervasive driving assistance...
systems. For example (Doshi, Morris, & Trivedi, 2011) report on a pervasive car safety system, which might interfere lane changes. To do this, the system not only monitors traffic and driver but also deduces the driver’s intentions.

All these systems offer a lot of new interesting and useful possibilities, but would not be possible without Pervasive Computing technologies. However, as with every new system (generation) the question of dependability arises as soon as the idea of the system has sprung. Dependability can be typically divided into a number of aspects namely: (functional) correctness, safety, security, reliability, availability, transparency and traceability. Interesting is, that for most existing systems only a very limited number of aspects of dependability are relevant. For example a train control system must be able to tolerate component failures and fulfill its function at all times. Thus it must be safe, reliable and available. Aspects like security, transparency and traceability play little to no role. On the other hand, e-shopping systems need to be secure, transparent and traceable. Safety and reliability are not of much concern (as these systems have virtually no inherent potential for damage).

This is different for Pervasive Computing systems. Most Pervasive Computing systems come with requirements from all aspects of dependability. The reason for this lies in the nature of Pervasive Computing itself. Typically Pervasive Computing systems are very tightly connected with specific users (let it be directly like a medical diagnostic system or indirectly like a smart logistics system, which handles personal luggage at airports). Pervasive Computing systems often gather and store information on behavior, context, habits and plans of their users. This information forms the basis for many benefits the system can offer for the users. For example, a heart attack system can only trigger emergency calls, if it knows the person it is monitoring and its position. Check-in procedures can only be circumvented, if pieces of luggage carry information about their owner respectively their target destination. Additionally, they often rely on wireless communication.

Together this automatically raises requirements like security, transparency and traceability. Secondly, Pervasive Computing systems are normally embedded in the environment not only for gathering information but also for making decisions or at least for decision support. Any decision can be either harmful or not. As a rule of thumb, the closer the connection to the physical world is, the higher the potential for a safety relevant impact will be. A lost emergency signal from a patient’s heart patient might cost his live; wrong loading of transportation containers could make airplanes very instable and cause severe problems during flight. In addition, intended attacks and manipulations of the systems can cause severe consequences (let it be monetary or in the form of injuries or deaths). Just think of smuggling dangerous luggage into airplanes by abuse of the check in system. Therefore, in general Pervasive Computing systems also need to show their safety, reliability and functional correctness.

Besides the need for meeting requirements from many different aspects of dependability, Pervasive Computing also implies restrictions on methods that can be applied. One reason is that in many application scenarios energy and computation power is a very limited resource. So for example, cryptography is not always feasible. Another reason is that very often such systems rely on some sort of learning mechanism, which allows them to dynamically adapt to new situation and function in changing environments. Although this is highly wanted, it brings a lot of restrictions on how the systems can be validated or verified. Unfortunately, there is no generic out-of-the-box solution for guaranteeing dependability for Pervasive Computing available yet. As a matter of fact such a solution does not even exist for traditional IT systems. However, there exist a lot of approaches for different facets of dependability.

This chapter will give an overview on available methods for analyzing dependability and explain their applicability to Pervasive Computing. The Section “Aspects of Dependability” presents a classification of different dependability requirements. Following this clas-
Up In Smoke: Rebuilding After an IT Disaster

www.igi-global.com/chapter/smoke-rebuilding-after-disaster/6388?camid=4v1a