Verena Elisabeth Majuntke

Application Coordination in Pervasive Systems
1. Introduction

This chapter serves the purpose to give an overview of the thesis at hand and the addressed problem statements. At first, it describes the vision of pervasive computing and discusses the trends towards pervasive computing and its importance in the industrial sector. Subsequently, the problem of interferences in pervasive systems is identified and the need for their management is motivated. The motivation is followed by the definition of a research aim and a summary of the contributions of this thesis. The introduction of the research aim closes with an overview of the thesis structure.

1.1. Pervasive Computing

The notion of Ubiquitous Computing, or Pervasive Computing, was first introduced by Mark Weiser in 1991. In his essay The Computer for the 21st Century [Wei91], Weiser described his vision of the human-computer interaction. He predicted that the future human environment would be pervaded by a multitude of information processing devices. Being equipped with respective hard- and software, these devices will be able to form networks and to cooperate in the interest of their users. Through their cooperation they would provide functionality to users assisting them seamlessly in their everyday tasks. The explicit human-computer interaction would transform into an implicit use of the functionality the networks provide. As a result, the user’s environment would become intelligent, sensing the user’s need and aiming at an optimal user support at any time and anywhere.

The vision of Mark Weiser has brought forth a multitude of approaches that contribute to the realization of pervasive computing. Early approaches in this area were projects such as Aura [GSSS02] or Gaia [RHC+02] which addressed a variety of aspects in pervasive
computing and yielded system software for their realization. More than twenty years later, a truly pervasive system as described in Weiser’s vision is yet to be realized. However, the trend towards pervasive computing has become visible in different areas. One of the areas which is discussed in the following is the business sector.

1.2. Business Applications

In 2006, a study entitled “Pervasive Computing: Trends and Impacts” [BSI06] was developed by order of the German Federal Office for Information Security (BSI). The study was conducted in cooperation with VDI/VDE Innovation und Technik GmbH, Fraunhofer Institute for Secure Information Technology and Sun Microsystems GmbH. One goal of the study was to identify trends in pervasive computing and to analyze impacts on the industrial sector. The results of the study reflect the knowledge of international experts that was gathered through a comprehensive online survey and a variety of interviews. The following discussion on pervasive computing in the industrial sector and the observations are extracted from the study.

The study revealed that a variety of areas exist in which “pervasive computing is already recognisable and is very likely to play a decisive role in the future” [BSI06, p. 22]. An area in which the trend towards pervasive computing is evident is the sector of production and logistics [BSI06, Section 4.1, pp. 23-25]. Nowadays, IT-based controlling and monitoring systems are an integral part of production-specific and logistical systems. The aim of such systems is the optimization and automation of production, transport and supply along the entire supply chain. The integration of physical objects is realized by attaching artifacts to the objects providing them with digitally ascertainable data. In earlier systems, the use of bar code was the prevailing standard. However, a disadvantage of bar code was its requirement to physically access the artifact to retrieve the stored data. The use of RFID, in contrast, enables a remote access to the artifacts and thus the data. Thus, the tracking and tracing of objects without the need of additional physical actions has become viable. While a complete automation and optimization has yet to be achieved, a trend towards intelligent and autonomous systems is obvious.
1.3. Motivation

E-commerce has been identified as another area in which pervasive computing has become recognizable [BSI06, Section 4.6, pp. 30-31]. An enabler for e-commerce has been the fact that today’s users can typically be identified through pervasive computing objects such as their smart phones. The exploitation of user profiles and preferences provides a large potential for user-tailored marketing and location-based services [VMG01]. The use of location-based services has contributed to the sharing of costly products such as bicycles and cars. Depending on the user’s location, the availability of such objects can be determined and their use can be precisely recorded and billed. As a result, such systems enable the shared utilization of capital-intensive objects and can provide an attractive business model to users.

Another example is the area of medical care [BSI06, Section 4.7, pp. 31-33]. Medical and health-related systems have been identified as a large application area for pervasive computing. The optimization and automation of core processes in this area promise a large potential for cost reductions. As an example, pervasive systems could be employed to monitor patients at home to avoid long-term stays in hospitals for observation reasons. Likewise, the state of patients with chronic illnesses could continuously be monitored in order to develop an optimal treatment plan and to adapt it if necessary. Besides the cost factor, this also has the potential to improve the patient’s quality of life. Instead of being bound to the hospital bed, the treatment could be realized in an environment familiar to the patient.

1.3. Motivation

The results of the BSI study reveal that pervasive computing is increasingly present in the human’s daily life. The promise of pervasive computing is the optimization and automation of processes core to the respective area. In the area of logistics, core processes can be all processes involved in the management of stock such as tracking of pallets and ordering on demand. In the context of smart homes this could be the realization of any tasks to ease its user life. Conceivable examples are the adjustment of the temperature to its user’s needs or the redirection of a phone call to the room the user is currently in.
The vision of Mark Weiser has brought forth a variety of research work that aims at the realization of pervasive computing. Technically, pervasive computing is realized by a pervasive system. A pervasive system consists of users, devices, and the physical space they reside in. In order to provide functionality to users, pervasive applications are executed. A pervasive application is a distributed application which makes use of resources and capabilities currently available in the pervasive system. To provide their functionality anytime and anywhere, pervasive applications are context-aware and adaptive. According to Dey [Dey01], “context is any information that can be used to characterize the situation of an entity”. The entity may be a user, a specific location, or any kind of object that may have an impact on the application’s behavior. This context-awareness allows the application to incorporate the context information into configuration decisions. The adaptivity enables the application to adjust to changing contexts, pursuing an optimal configuration at all times. As a result, a pervasive application is able to continuously provide functionality in different functional configurations.

While pervasive applications have been specifically designed to adapt themselves to changing environments, the application’s ability to influence the environment and thus to change the context itself is often neglected. Such a context influence can either be produced implicitly as a side effect of employed resources or explicitly through the use of available actuators. As a consequence, the relationship between an application and the context is bidirectional. The context influences a pervasive application and vice versa. The fact that applications interact with the context and not only react to context changes makes them context interactive.

When an application is run in isolation, its ability to influence the context can be neglected. The execution of multiple applications, however, leads to new challenges if the applications are executed in the same physical space. The challenges arise from the fact that applications share the physical environment as common context and interact with it. One on one hand, they react to changes in the context by adapting themselves. On the other hand, they change the context according to their needs. As a consequence, pervasive applications are directly related with each other via the context they share.
Consider the following example: User Anne is in the living room reading on her e-book reader. In order to provide a good contrast, her application has turned on the lights. After a while Bob enters the living room. His intention is to watch a movie on the projector installed in the environment. To do so, his application closes the blinds and turns off the light to provide the optimal atmosphere for a movie. Bob’s application clearly has an impact on Anne’s application. The changing of the light level impairs the functionality provision of Anne’s application. In this scenario, the e-book application has two options to deal with the context change. It can 1) adapt the context according to its need again, i.e. turn on the light and open the blinds or 2) adapt itself, i.e. by redirecting its output to another device for example. The first option may result in both applications taking turns adapting the context. The second option may lead to a situation where the e-book application cannot provide its functionality anymore.

The described problem is referred to as an interference throughout this thesis. An interference is an application-produced context that impairs the functionality provision of another application. The problem can be reduced to the fact that applications which are executed in the same physical space share and interact with a common context. As a consequence, they can have a direct impact on each other through the commonly shared context.

1.4. Research Aim

The fact that interferences can occur becomes more problematic given that future user environments are likely to comprise not only multiple applications within a single system, but also multiple pervasive systems. With the continuous development of pervasive computing, it is very unlikely that the world will be managed by only a single system. Instead, a variety of different pervasive systems will exist in parallel. As a consequence, pervasive applications which are executed in the same physical space are likely to interfere with each other even if they are executed in different pervasive systems. Ideally, the occurrence of interferences should be avoided to enable the unobstructed provision of functionality by multiple applications in the same physical space. In practice, however,
the context-interactivity is one of the major characteristics of pervasive applications. As a consequence, interferences cannot be avoided and thus must be detected and resolved at runtime to allow an undisturbed pervasive system experience. The goal of the thesis at hand is to develop an approach to manage interferences in terms of their detection and resolution. The proposed solution must be able to manage interferences across multiple systems. It should consider the needs of its users and handle interferences in their interest.

Up to the present, the management of interferences as defined in this thesis has not been addressed in its entirety. Some research work exists which addresses the management of subsets of the interference problem, e.g., [KMW03], [MD06], [SHW05], or [RC03]. Other work focuses on the realization of frameworks to detect and resolve problems between multiple applications such as [MD06] and [BRK06]. However, their work remains on a theoretical level and has not been developed to handle the addressed problems at runtime. Further approaches exclusively focus on the task of interference detection, e.g., [PLH05], [SW09], and [AKM06], or specific interference resolution strategies, e.g., [JCL11], [HME+06], and [SW05]. In summary, none of the existing approaches is able to handle the problem of interferences as addressed in this thesis.

1.5. Contribution

The contribution of this thesis is a coordination framework that manages interferences between applications in pervasive systems. The management is split into two tasks, interference detection and interference resolution. For interference detection, applications are required to provide information to the framework about their context interaction in their current functional configuration. Based on this knowledge and a context model, interferences can be detected. For interference resolution, applications are required to specify and provide interactions for alternative functional configurations. If the framework detects an interference, it determines interference-free context interactions for each application to resolve the interference. The applications are then requested to instantiate the respective functional configuration that complies with the context interaction. Thus, a coordinated application adaptation is performed. Specifically, the contributions of this thesis are:
1.5. Contribution 7

(I) Interference Model and Detection: The thesis analyzes the problem of interferences in detail and introduces a formal model for interferences based on monadic predicate logic [Löw31]. Based on the interference model, the problem of interference detection is discussed and two algorithms, a basic statement evaluation and an optimized version, are presented.

(II) Interference Resolution Plan Computation: The first step of interference resolution is the computation of a respective plan. Based on the model of interferences, the problem of interference resolution plan computation is modeled as a constraint satisfaction problem (CSP) and the suitability of algorithm classes for CSPs is discussed for pervasive systems. Furthermore, a heuristic that uses information about an application’s involvement in an interference is introduced realizing an informed backtracking algorithm to compute an interference resolution plan.

(III) Design: The framework is designed to be tailored to pervasive systems which can be heterogeneous, dynamic and open with respect to devices, users and pervasive applications. For this purpose, the framework design is subject to three decisions:

(a) Cross-System Coordination Layer: The framework is designed as a cross-system coordination layer. It coordinates the interaction of pervasive applications with the shared context across different system software. For the realization, the requirements towards application systems are kept at a minimum and abstract from details specific to a particular pervasive system. Besides these requirements, integrated application systems are treated as black boxes.

(b) Extension of Existing Systems: The minimal requirements described in the previous design decision are realized through extensions of existing application systems. For this purpose, the concept of a context configuration is introduced. The context configuration extends a functional configuration with the specification of its context interaction. Furthermore, an adaptation interface is introduced that allows the framework to request the instantiation of a configuration computed by the application itself.
(c) **Strategy-based Application Coordination**: A variety of aspects can be considered when applications are coordinated in order to maintain an interference-free system state. For this purpose, the framework realizes a generic interface for the use of arbitrary resolution strategies. This allows to customize the framework for the needs of different pervasive systems.

(IV) **Development and Evaluation**: A prototypical realization is developed that implements the theoretical concepts of the framework. Furthermore, extensive evaluations are conducted in order to show the utilizability of the coordination framework in practical pervasive systems.

(a) **Component Placement and Communication Sequences**: System characteristics that have an impact on the practical realization are identified and discussed for general pervasive systems. Based on the findings, recommendations for the placement of components that compose the framework and the points in time when data should be exchanged are given.

(b) **Prototype and Measurements**: The prototype COMITY is developed that implements the concepts of the application coordination framework. Furthermore, measurements are conducted to assess the quality of the prototype and the concepts it implements. For this purpose, the memory requirement and the overhead it causes are analyzed. Furthermore, the algorithms for interference detection and resolution are evaluated.

1.6. **Structure**

The remainder of the thesis is structured as follows: Chapter 2 provides the preliminaries for the contributions of this thesis. It introduces the concept of pervasive systems and discusses the notion of pervasive applications to realize functionality in such systems. Furthermore, it identifies major characteristics of pervasive applications and classifies existing approaches along these criteria. The result is an overview of existing approaches and their commonalities yielding a definition for pervasive applications used in this thesis.
Finally, the problem of interferences is discussed as situations which are likely to occur when multiple applications are executed in the same physical space.

Chapter 3 defines the research question of this thesis. For this purpose, the chapter first introduces a system model describing the target system for which a solution is to be developed. Subsequently, it presents the concept of application coordination as the idea to manage interferences in the target systems and defines the goals of this thesis. The chapter closes with the identification and analysis of requirements towards the approach to be taken. The requirements tailor the approach to the pervasive systems and thus refine the research goal.

Chapter 4 discusses related work. At first, related problems and definitions similar to the notion of interferences are analyzed. Then, related work with respect to application coordination is addressed. The chapter introduces comprehensive approaches which address the entire process of application coordination. Finally, existing work with respect to the isolated tasks of interference detection and interference resolution is evaluated.

Chapter 5 presents the framework for application coordination as the approach to handle interferences in the targeted pervasive systems. To start with, the chapter discusses the major design decisions for the framework. Subsequently, it gives an overview of the framework, its compositional parts and describes the mode of operation. The overview is followed by an elaboration on how existing systems need to be extended in order to allow their coordination through the framework. Finally, the tasks of interference detection and interference resolution are addressed. After a thorough analysis of the underlying theory, respective solutions are discussed and developed.

Chapter 6 analyzes the realization of the theoretical application coordination framework for practical pervasive systems. The practical realization covers the component deployment and the points in time when communication is performed. To develop a viable approach, the chapter first identifies system characteristics which have an impact on realization decisions. It then analyzes general pervasive systems with respect to these characteristics. Based on the findings, decisions on component placement and their in-
teraction to realize application coordination are presented. Finally, the dynamism of pervasive systems is addressed and an approach for its handling is discussed.

Chapter 7 presents the prototype COMITY. The prototype implements the concepts described in Chapter 5 and 6. It gives an overview of the classes and discusses the details of the implementation. Finally, the realization of the coordinator for an existing system – the middleware BASE – is presented.

Chapter 8 evaluates the prototype presented in Chapter 7. The chapter analyzes the memory requirements of the coordinator and discusses its overhead in relation to BASE. Furthermore, the chapter conducts performance measurements with respect to the critical path of application coordination and the algorithms implemented for interference detection and resolution.

Chapter 9 closes the thesis with a summary of the results and an outlook on future work.