HAL
open science

CADEAU: Supporting Autonomic and User-Controlled Application Composition in Ubiquitous Environments

Oleg Davidyuk, Ivan Milara Sanchez, Jukka J. R. Riekki

To cite this version:

Oleg Davidyuk, Ivan Milara Sanchez, Jukka J. R. Riekki. CADEAU: Supporting Autonomic and User-Controlled Application Composition in Ubiquitous Environments. Apostolos Malatras. Pervasive Computing and Communications Design and Deployment: Technologies, Trends, and Applications, IGI Global, 2010. inria-00509108v1

HAL Id: inria-00509108

https://inria.hal.science/inria-00509108v1

Submitted on 10 Aug 2010 (v1), last revised 8 Jun 2011 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

CADEAU: Supporting Autonomic and User-Controlled Application Composition in Ubiquitous Environments

Oleg Davidyuk[1,2], Iván Sánchez Milara[2] and Jukka Riekki[2]

[1] ARLES Research Team, INRIA Paris-Rocquencourt,
Domain de Voluceau, Le Chesnay, 78153, France
oleg.davidyuk@inria.fr

[2] Dept. of Electrical and Information Engineering, and Infotech Oulu,
P.O. Box 4500, University of Oulu, 90014,
Oulu, Finland
{firstname.secondname}@ee.oulu.fi

Abstract. Networked devices, such as consumer electronics, digital media appliances and mobile devices are rapidly filling our everyday environments and changing them into ubiquitous spaces. Composing an application from resources and services available in these environments is a complex task which requires solving a number of equally important engineering challenges as well as issues related to user behavior and acceptance.

In this chapter we introduce CADEAU, a prototype that addresses these challenges through a unique combination of autonomic mechanisms for application composition and methods for user interaction. These methods differ from each other in the degree to which the user is involved in the control of the prototype. They are offered so that users can choose the appropriate method according to their needs, the application and other context information. These methods use the mobile device as an interaction tool that connects users and resources in the ubiquitous space. We present the architecture, the interaction design and the implementation of CADEAU and give the results of a user study that involved 30 participants from various backgrounds. This study explores the balance between user control and system autonomy depending on different contexts, the user's needs and expertise. In particular, the study analyses the circumstances under which users prefer to rely on certain interaction methods for application composition. We argue that this study is a key step towards better user acceptance of future systems for the composition of ubiquitous applications.

1 Introduction

Our everyday living, working and leisure environments are rapidly becoming ubiquitous due to the wide availability of affordable networking equipment, advances in consumer electronics, digital media appliances and mobile devices. This, combined with the increasing importance of web technologies for communication (e.g., Web Services, Cloud Computing and Social Networking) is resulting in the emergence of innovative ubiquitous applications. These applications usually involve multiple resources and Web

ACM Reference Format: Davidyuk, O., Sánchez, I. & Riekki, J. (2010). CADEAU: Supporting Autonomic and User-Controlled Application Composition in Ubiquitous Environments. In Apostolos M. (Ed.), Pervasive Computing and Communications Design and Deployment: Technologies, Trends, and Applications. IGI Global. pp. –

Services at the same time. Examples of such resources are mobile devices, displays, portable players and augmented everyday objects. Web Services utilize these resources and provide the interfaces through which users can interact and control the ubiquitous environment. Ubiquitous applications differ from traditional applications that are static and bound to resources as specified at design time. Ubiquitous applications, on the other hand, are composed (or realized) from the available resources and Web Services at run-time according to user needs and other context information.

Depending on the degree of autonomy, application composition can be autonomic or user-controlled. A system supporting autonomic composition fully controls all processes (including the application's behavior) and does not assume any user involvement. In contrast, user-controlled composition systems involve users in control. These systems can be further classified as manual composition systems (users themselves control everything) and semi-autonomic composition systems (both users and the system collaborate to control the composition through, e.g., a visual interface). For instance, a semi-autonomic system can rely on a mixed initiative interface which guides users through a sequence of steps that result in a composed application.

In general, systems for autonomic application composition aim to ensure better usability by keeping user distraction during the composition to minimum (although user attention may be distracted while (s)he is using the composed application). These systems focus on abstracting user activities from their system-level realization and allow users to concentrate on what they need, rather than on how these activities have to be realized by the system (Sousa et al., 2006, 2008b; Masuoka et al., 2003). User activities are users' everyday tasks that can be abstractly described in terms of 1) the situation (context) in which the tasks take place, 2) the system functionalities required to accomplish the activities, and 3) user preferences relating to QoS, privacy and other requirements. In order to support the user in these activities, the automatic system captures the user's goals and needs by means of user context recognition facilities (Ranganathan & Campbell, 2004) or through dedicated user interfaces (Davidyuk et al., 2008a; Sousa et al., 2006; Kalasapur et al., 2007). Some systems allow users to express their intent vaguely, for example in their natural language as suggested by Lindenberg et al. (2006). Then, the system reactively or even pro-actively searches for possible ways to compose the required application using the appropriate resources.

In spite of the advantages in autonomic application composition, users might feel out of control, especially when the system does not behave as anticipated or when the resulting application does not match the users' original goal. Moreover, as pointed out by Hardian et al. (2006) and confirmed through user experience tests by Vastenburg et al. (2007), involving users in application control is essential to ensure that users accept autonomous prototypes, especially those intended for home or office automation domains. In addition, our earlier studies on user control for application composition (Davidyuk et al., 2008a) reveal that users still need to be provided with control interfaces even if the system is autonomic and users do not intend to control the composition of each application.

In this chapter, we present CADEAU, a prototype that supports the composition of applications from ubiquitous resources and Web Services. This prototype uses the user's mobile device as the interaction tool that can control the application composition as

well as the application itself. This prototype is a complete solution that supports both automatic and the user-controlled composition. CADEAU provides three interaction methods, namely the autonomic, the manual and the semi-autonomic method, which differ from each other in how much the user is involved in the control of the application composition. These methods are offered in order to let the users to choose the most suitable means of interaction according to their needs. As the main contribution of the chapter we present the implementation of the prototype, the example application and the results of a user study. This user study involved 30 participants and aimed to explore the balance between user control and system autonomy in application composition in different contexts, depending on users' needs and experience with technologies. In particular, the study addresses the question of the autonomy domain of the system, i.e. the issues that users allow the system to take decisions on. This study also analyzed the circumstances under which the users prefer to rely on certain interaction methods for application composition. We are not aware of any other user evaluation study of a fully implemented system for application composition.

The chapter begins by reviewing the related work on application composition in ubiquitous environments. Then, we introduce the application scenario and overview the conceptual architecture of both the CADEAU prototype and the example application. Then, we present the interaction methods and the user interfaces of the application. The main contribution of the chapter, which is the implementation of the prototype and the user evaluation study, are then described. Finally, we discuss the main findings of the chapter and outline future work.

2 State of the art

Various solutions that tackle ubiquitous application composition have been proposed. These solutions focus on service provisioning issues (Chantzara et al., 2006; Takemoto et al., 2004), context-aware adaptation (Preuveneers & Berbers, 2005; Rouvoy et al., 2009; Bottaro et al., 2007), service validation and trust (Bertolino et al., 2009; Buford et al., 2006), optimization of service communication paths (Kalasapur et al., 2007), automatic generation of application code (Nakazawa et al., 2004), distributed user interface deployment (Rigole et al., 2005, 2007) and design styles for developing adaptive ubiquitous applications through composition (Sousa et al., 2008a; Paluska et al., 2008). In contrast, the work described in this chapter focuses primarily on providing user control in application composition. Hence, we classify related work in these categories according to the extent of user control: autonomic, semi-autonomic and manual application composition.

Autonomic composition. Systems in this category usually aim to minimize user distraction while (s)he is composing an application. These systems assume that users do not wish to be involved in the control, and thus all processes are carried out autonomously. Most research on autonomic composition deals with activity-oriented computing[1] (Masuoka et al., 2003; Sousa et al., 2006, 2008b; Ben Mokhtar et al., 2007; Messer et al., 2006; Davidyuk et al., 2008b). These systems take a user-centric view

[1] Some researchers alternatively called it task-based computing.

and rely on various mechanisms to capture users' needs and intentions that are automatically translated into abstract user activity descriptions. These descriptions can be provided to the system implicitly through user context recognition facilities (Ranganathan & Campbell, 2004), explicitly through dedicated user interfaces (Sousa et al., 2006; Messer et al., 2006; Davidyuk et al., 2008a) or they can be supplied by application developers at design time (Beauche & Poizat, 2008; Ben Mokhtar et al., 2007). After the system receives an activity description, it carries out the activity by composing an application that semantically matches the original description according to some specified criteria and a matching (or planning) algorithm. Issues related to semantic matching for application composition have been studied, e.g., by Ben Mokhtar et al. (2007) and by Preuveneers & Berbers (2005). Planning algorithms for application composition have been proposed among others by Beauche & Poizat (2008), Ranganathan & Campbell (2004), Rouvoy et al. (2009) and Sousa et al. (2006). The prototype presented in this chapter also builds on the activity-oriented infrastructure and uses a planning algorithm (Davidyuk et al., 2008b) to realize autonomic application composition.

Semi-autonomic composition. In general, these solutions assume that the applications are composed as the result of collaboration between users and the system. Semi-autonomic composition may vary from computer-aided instruction to intelligent assistance that involves two-way dialogue with the system (also known as mixed initiative interface). For example, DiamondHelp (Rich et al., 2006) uses a mixed initiative control interface based on the scrolling speech bubble metaphor (i.e. resembles an online chat) which leads the user through a set of steps in order to control or manipulate appliances at home. Another approach that provides a set of interactive tools for composing applications has been developed by Wisner & Kalofonos (2007). Their first tool allows users to see the devices available in the home network and compose applications by simply connecting these devices in this interface. Then, another tool interactively assigns events and actions to all devices chosen by the users and guides them through the process of specifying the application's behavior, after which the application can be started. The semi-autonomic composition used in CADEAU resembles an interface for computer-aided instruction, i.e. users control the composition process by choosing from the options that are dynamically produced by the system.

Manual composition. These approaches allow the users themselves to decide how applications are composed. In this case, the role of the system is to provide some means of user interaction (e.g. a graphical user interface) through which users can specify the structure and the functionality of their applications. Solutions that focus on application composition for home networks have been suggested by Bottaro et al. (2007), Chin et al. (2006), Gross & Marquardt (2009), Mavrommati & Darzentas (2007), Newman & Ackerman (2008), Newman et al. (2008) and Rantapuska & Lahteenmaki (2008). Manual application composition in the museum domain has been suggested in Ghiani et al. (2009). Somewhat related is the solution proposed by Kawsar et al. (2008). Although their work focuses mainly on the end-user deployment of ubiquitous devices in home environments, they also tackle some application composition issues. In particular, their system allows users to install various devices (i.e. by physically plugging and wiring them) and then to develop simple applications by manipulating smart cards associated

with the installed devices. Another approach, presented by Sánchez et al. (2009), uses an RFID-based physical interface that allows users to choose visual resources (that have to be used with an application) by simply touching RFID tags attached to resources.

Applications for delivering multimedia content in ubiquitous spaces based on RFID technology have been proposed, for instance, in the prototypes of Broll et al. (2008) and Sintoris et al. (2007). These solutions, however, focus on the interaction of users with RFID tags and do not support application composition.

Several researchers have studied the issue of balancing user control and autonomy of the system. For example, Vastenburg et al. (2007) conducted a user study in order to analyze user willingness to delegate control to a proactive home atmosphere control system. They developed a user interface which provided three modes of interactivity: manual, semi-automatic and automatic. However, the automatic behavior of their system was 'wizard-of-oz', in that it was remotely activated by a human observer during the experiment. Another attempt to address the issue of balancing user control and autonomy solution has been made by Hardian et al. (2008). Their solution focuses on context-aware adaptive applications and attempts to increase user acceptance by explicitly exposing the system's logic and context information used in the application adaptations.

CADEAU differs from the related work presented above, because our prototype supports autonomic, semi-autonomic and manual composition at the same time. Moreover, we are not aware of any other user evaluation experiment of a fully implemented composition system that has studied the balance between user control and system autonomy.

3 Overview of CADEAU

Applications in CADEAU are composed of ubiquitous resources and Web Services. CADEAU supports the resources that provide multimedia, computational or other capabilities. These resources are used and controlled by Web Services. The applications in CADEAU can be composed automatically or manually depending on the extent to which users wish to be involved in control. In the first case the applications are composed according to abstract descriptions provided by the application developers. These descriptions define what ubiquitous resources are needed and what particular characteristics (or capabilities) these resources must have in order to compose these applications. Applications are realized automatically during the composition process, whose primary goal is to produce application configurations, i.e. the mappings of application descriptions to concrete resources. Once an application configuration has been produced, CADEAU reserves the resources and Web Services for the user and executes the application.

Depending on the amount of resources available in the environment together with their characteristics and capabilities, the same application description may correspond to multiple application configurations. Assuming that the resource characteristics vary, some of the application configurations will be more attractive to users than others. For example, if the user needs to watch a high-quality video file, (s)he will prefer the application configuration option which utilizes an external display with higher reso-

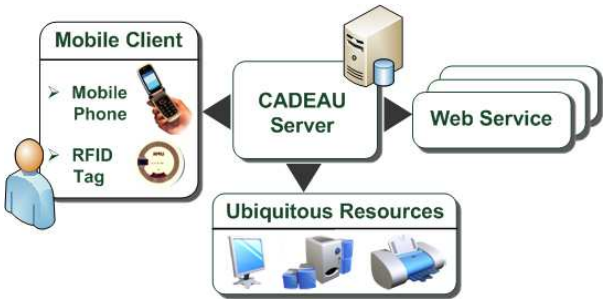


Fig. 1: CADEAU architecture.

lution and faster network connection. In order to address this issue, CADEAU uses the optimization criteria that allow 1) to compare application configurations and 2) to encode the user's preferences regarding various resource characteristics.

User-controlled composition in CADEAU is based on semi-autonomic and manual interaction techniques. The semi-autonomic method also relies on the automatic composition process, but provides a user interface for selecting alternative application configurations produced by CADEAU. Users can browse these configurations, compare them and choose the one that suits them best. The manual method allows users to fully control the application composition through a physical user interface. This interface consists of a NFC-enabled mobile device (NFC Forum, 2010a) and RFID tags which are attached to ubiquitous resources in the environment. A user composes an application by simply touching corresponding tags with his or her mobile device. This action triggers CADEAU to reserve these resources for this user and to start the application.

The general overview of the CADEAU prototype is shown in Figure 1. The main components are the CADEAU server, mobile clients, ubiquitous resources and Web Services. The CADEAU server is build upon the REACHeS system (Riekki et al., 2010) and provides the communication facilities for other components, performs the composition of applications and allocation of resources. In particular, the CADEAU server includes the Composition Engine that is responsible for finding the application configurations matching the user's needs and the situation in the environment. The role of Web Services is to enable the control and interaction with ubiquitous resources. They also implement application logic and provide access to data used in applications. The mobile clients are used as interaction devices that connect users, resources and the server. The mobile clients are also a part of the CADEAU user interface which consists of (i) the user interface on the mobile devices and external displays and (ii) the physical interface through which the users interact with the ubiquitous environment. While the first play the primary role in CADEAU, the physical interface provides the input for the user interaction. That is, the physical interface in CADEAU bridges the real and digital worlds, so that the user is able to interact with augmented objects and access appropriate ubiquitous resources. The physical interface of CADEAU is comprised of RFID tags and mobile devices with integrated RFID readers.

In order to explain the features and illustrate CADEAU, we present an application

(a) (b)

Fig. 2: The prototype of a 'smart' newspaper with embedded tags (a) and a user interacting with the smart newspaper (b).

scenario that we implemented with the prototype and used for the user evaluation study[2]. It should be noted, however, that CADEAU supports various kinds of Web Services and resources and the application scenario presented below is only one example.

> *John is reading a newspaper in a cafeteria. This newspaper has some hidden multimedia content (audio narration, video and images) that can be accessed by touching the newspaper with a mobile phone (see Figure 2). John touches the newspaper titles and, shortly after, CADEAU prompts John to browse the hidden content on a nearby wall display. John browses the list of articles which are linked with multimedia files. He selects the most interesting files by pressing buttons on the mobile phone's keypad. When files are chosen, CADEAU stores files' links in John's mobile phone. Later that day, John decides to watch the videos and listen to the audio narration in a conference room at work. John decides to use a semi-autonomic method to choose appropriate resources. CADEAU proposes several combinations of a display, an audio system and Web servers that host the multimedia files. John chooses the combination named 'nearest resources' and starts the application that plays the multimedia files using these resources. John can control the playback (stop, pause, next/previous) by pressing the phone's buttons.*

The multimedia information that John accessed in the CADEAU application is organized as shown in Figure 3. All content is categorized by subjects that are mapped to RFID tags in the newspaper. Each subject is related to a cluster of articles which are

[2] This application scenario is also available as the video from `http://www.youtube.com/watch?v=sRjCisrdr18`

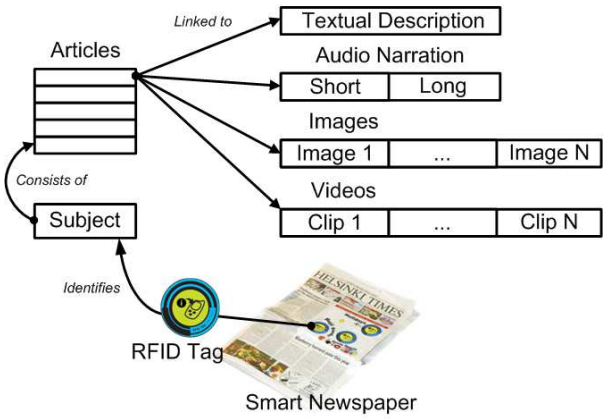


Fig. 3: Conceptual structure of the multimedia content in the example application.

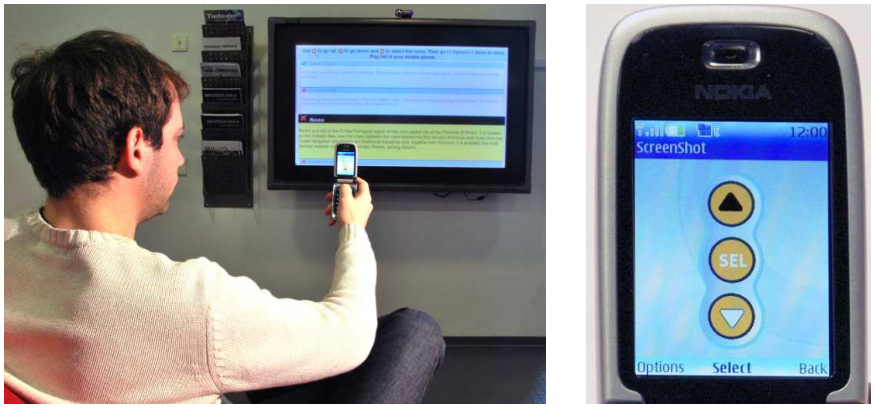
represented with short textual descriptions that resemble an RSS feed. Users acquire a subject by touching the appropriate tag. Then users browse related articles on an external display. Textual descriptions act as links to the multimedia files (audio, video and image slideshows) that are related to the articles. Thus, if a particular article is chosen while browsing some topic, the user's mobile device acquires links to all multimedia files associated with this article. Among these, the audio narration provides the most important information, while videos and images are supplementary material whose role is to enrich and augment the user experience with the application. The advantage of the audio narration feature stems from the fact that it provides information which is normally printed in the newspaper. Each audio narration consists of two parts, a short version and a long version. The short version narrates the overview of the article, while the long one is a thorough description that goes into greater detail. By default, CADEAU assumes that users listen only to the short version of audio. However, if interested, users can request the long version. As video and image files do not have the same importance as audio narration, the playback of audio narrations is controlled separately from the playback of the other types of files.

4 CADEAU Interaction Design

In this section we present the interaction methods supported by CADEAU and explain them using the application presented in the previous scenario as an example. The example application can be functionally divided into *collecting* and *delivering* phases, as shown in Figure 4. The goal of the first phase is to allow users to choose multimedia content from the 'smart newspaper', while the second phase focuses on delivering this content using multiple ubiquitous resources.

During the *collecting* phase, users interact with the tags embedded in the newspaper by touching them with their mobile devices. Each of these tags is augmented with a graphical icon and a textual description, as shown in Figure 2a. The action of touching

Fig. 4: The interaction workflow of the example application.



(a) (b)

Fig. 5: The user interface for browsing on the external display (a) and the remote controller user interface (b).

prompts the user to browse the chosen subject on a public wall display (located nearby) using his or her mobile device as a remote display controller. The user interfaces of the wall display and the mobile device are shown in Figures 5a and 5b.

The user interface of the wall display comprises a list of articles that are associated with the subjects chosen by the user (see Figure 3). Users can choose multiple articles. When the user selects an article of interest, the application acquires the article's reference number and adds it to the user's playlist which is stored in the memory of the mobile device. The *collecting* phase ends when the user closes the application.

The second phase of the application scenario, the *delivering* phase, involves using an application in a large ubiquitous environment with multiple resources. In order to

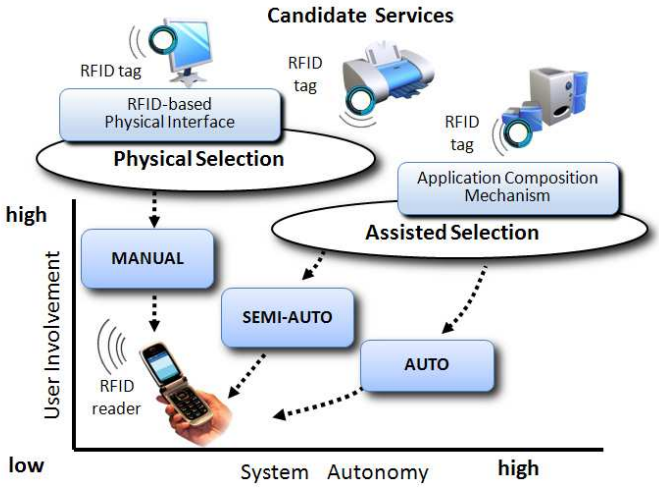


Fig. 6: CADEAU interaction methods.

(a) (b) (c)

Fig. 7: The user interface for the manual method: user interacting with a display service (a), the mobile phone user interface (b) (shows that two services are selected) and the control panel for the remote services (c).

help users to choose the right combination of resources the CADEAU offers three alternative interaction methods, namely, manual, semi-autonomic and automatic. These methods are shown in Figure 6 where they are arranged according to the levels of user involvement and system autonomy that the methods provide. The users can always switch from one interaction method to another as required by the situation in the ubiquitous environment, the application being composed or the user's personal needs. Once the application has been composed by means of any of these methods, it can be used with the chosen resources. Next, we present the motivation and explain these interaction methods in detail.

(a) (b) (c)

Fig. 8: The UI of the semi-autonomic method (b), starting the CADEAU application using the autonomic method (a) and the UI of the remote control (c).

The manual method is an interaction technique which addresses the need of the users to fully control the composition of the CADEAU application (see Figure 7). The manual method relies on the physical interface to allow the users themselves to choose the resource instances. Figure 7a demonstrates a user choosing a display resource by touching the attached RFID tag with a mobile device. Whenever a resource tag is touched, it uniquely identifies the resource instance associated with that tag. The CADEAU application requires multiple resource instances to be chosen, hence the interface on the mobile device suggests to the user what other resources are needed in order to compose the whole application. The user interface on the mobile device plays an essential role in the manual method. It provides feedback to the user's actions (i.e. it visualizes the information the user collects by touching tags) and also suggests to the user what other resources (or services) (s)he needs to choose before the application can be started. Figure 7b presents the user interface on the mobile device after the user has chosen two resource instances (a display and a speaker resource). The CADEAU application starts as soon as the user chooses the last necessary service instance.

The resource instances that cannot be equipped with tags are represented in the ubiquitous space with remote control panels. Such resources are typically non-visual services that are either abstract (i.e. exist only in a digital world) or are located in places that are hard to reach (e.g., on the ceiling). Figure 7c shows an example of such a control panel for a video projector resource that is mounted on the ceiling of the ubiquitous space.

The semi-autonomic method allows the application composition to be controlled by the Composition Engine as well as by the user. The key role in this interaction method is played by the list of application configurations that appears on the mobile device when a user touches the start tag (see Figure 8a) in the ubiquitous space. Each entry in this list is comprised of a set of service instances required by the application. This list is dynamically produced and organized by the system according to the user

Fig. 9: Semi-autonomic method helps users to identify resources in the ubiquitous space.

defined criteria. Thus, the list always starts with the most attractive application configuration for the user. Figure 8b shows the user interface of the list with three alternative application configurations. In this case, each application configuration is a combination of two resource instances represented by small circular icons. These icons visualize the type of the resource instance (i.e. a display or a speaker) while supplementary textual descriptions (e.g. 'closest headset') indicate the instances from the ubiquitous space that are engaged in this particular application configuration.

Often the users are unfamiliar with the ubiquitous space and hence may experience difficulties when associating particular resource instances to their textual descriptions. In addition, the users may need to preview a certain application configuration before starting it. In these cases, users can optionally browse the list of application configurations and identify the resource instances using the keypad of their mobile devices. This action commands the resources in the ubiquitous space to respond to the user: the services providing display capabilities respond by showing a 'splash screen' (see Figure 9) while the audio services play a welcoming audio tone. However, users can omit this step and proceed directly to starting the preferable application configuration by highlighting it and pressing the phone's middle button, as shown in Figure 8b. Sometimes, none of the application configurations offered by the Composition Engine may suit the user, and, in this case, (s)he can switch to the manual method which provides a greater degree of user control.

The automatic method is an interaction technique which is based on the so-called-principle of least effort (Zipf, 1949) and aims to start the application while keeping user distraction to minimum. This method assumes that the user does not want to control the application composition and (s)he prefers to delegate the task of choosing an application configuration to the system. The user starts the application by touching the start tag in the ubiquitous space (see Figure 8a) after which the system responds by automatically choosing and starting an application configuration.

When an application configuration is started by means of one of the methods men-

tioned above, the application changes the user interface of the mobile device into a remote control, as presented in Figure 8c. Simultaneously, the wall display shows the user interface of the playlist composed by the user during the *collecting* phase. From this point, the user can control the application by giving commands from his or her mobile device. For example, the user can start, stop and pause playing the items from the playlist, as well as jump to the next or the previous item. The user can also mute, increase and decrease the volume of the speakers. During the playback, the user can optionally listen to the long versions of audio files by pressing the 'more' button from his or her mobile device. This command loads the file of longer duration to the playlist. The CADEAU application can be stopped at any time by closing the application from the user interface. This action stops the CADEAU application, erases the playlist and releases the resource instances for the next user.

5 Implementation

The implementation of the CADEAU prototype is built upon the REACHeS platform (Riekki et al., 2010) and reuses its communication and the Web Service remote control functionalities. CADEAU extends the basic composition mechanism used in REACHeS with the Composition Engine and, in addition, provides interaction methods for controlling the composition process. Figure 10 shows the architecture of the prototype which consists of the mobile clients, the CADEAU server, ubiquitous resources and the Web Services.

5.1 CADEAU architecture.

The client-side functionality is implemented using J2ME-MIDlets running on Nokia 6131 NFC mobile devices. The prototype supports Nokia 6131 and 6212 mobile devices which are, to date, the only market available mobile phones equipped with NFC readers. The MIDlets implement the user interface on the mobile device and also implement the interaction with RFID tags. The latter is realized by two components, the NFC Listener and the MIDlet Launcher (part of the mobile device's OS). The NFC Listener is responsible for detecting RFID tags while the MIDlet Launcher maintains the registry of RFID tags and the MIDlets associated with these tags. The NFC Listener component is built using the Java Contactless Communication API JSR-257. When the NFC Listener detects that a tag has been touched, it either (i) triggers the MIDlet Launcher to start the User Interface (UI) MIDlet or (ii) dispatches the information read from the tag directly to the UI MIDlet, if the MIDlet is already started.

The physical interaction is realized using ISO/IEC 14443 RFID tags (Mifire 1k type) that are attached to physical objects (i.e. ubiquitous resources). These RFID tags store data which is used for two purposes: (i) to describe an application that has to be invoked and provide the parameters that are needed for its invocation, and (ii) to specify the parameters that are needed to control the execution of an application (e.g., the events generated by the CADEAU user interface). Data is stored inside memories of tags as NDEF messages (NFC Forum, 2010b) that may consist of multiple NDEF records. Each record contains NDEF flags and a variable payload. In CADEAU, a payload is an ASCII

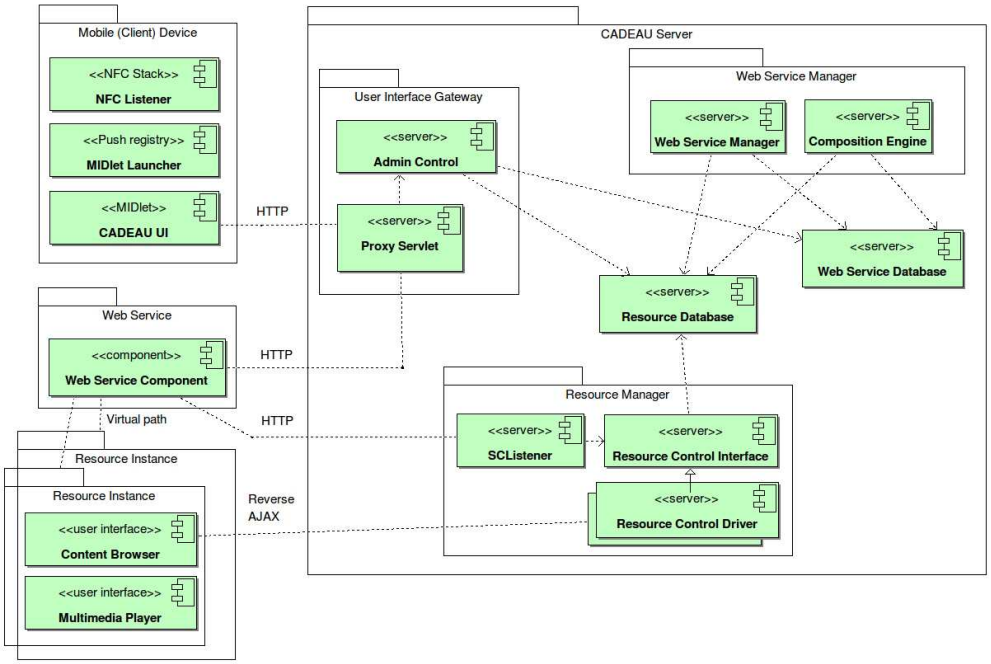


Fig. 10: The architecture of the CADEAU prototype.

string which encodes a pair of a parameter name and the corresponding parameter value. In particular, these parameters are used in the communication protocol described in the next section. NDEF messages can be read from the memories of tags using an NFC enabled mobile device.

CADEAU Server. As shown in Figure 10, the CADEAU server comprises three subsystems, the User Interface Gateway (UIG), the Web Service Manager (WSM) and the Resource Manager (RM). In addition, the server-side includes the databases that store the information related to the Resource Instances and the Web Services. The server-side functionality is implemented using Java Servlet API 2.2. running on Apache Tomcat 6, although the Composition Engine (part of the Web Service Manager subsystem) is implemented in C++ to achieve better performance.

The goal of the UIG is to provide the communication between the user interface on mobile devices and the other subsystems. The UIG consists of the Proxy Servlet and the Admin Control components. The Proxy Servlet processes the messages sent by the UI MIDlet or a resource instance and dispatches them to the appropriate components in the system. Certain messages are dispatched to the Admin Control component, whose role is to keep the information in the Resource and the Web Service databases up-to-date. The first database stores the information about the available Resource Instances and those which are allocated for each Web Service. The second database contains the information that associates the Web Service instances with the sessions opened by

different CADEAU MIDlets. It should be noted that one application may use multiple MIDlets.

The Resource Manager subsystem connects the ubiquitous resources to the server and consists of two components, the Resource Control Driver and the SCListener. The former realizes the Resource Control Interface and implements the resource-specific control and communication protocol. Each ubiquitous resource is assigned with its own Resource Control Driver instance, one part of which is executed within the CADEAU server while the other part of the driver is executed within the Resource Instance.

Specifically, the Resource Control Driver implements a Reverse Ajax protocol based on HTTP Streaming[3] for Display and Speaker resources. The SCListener is responsible for dispatching commands (i.e. asynchronous messages) from Web Service Components to the appropriate Resource Control Drivers.

A Resource Instance (RI) is a standalone PC embedded into the ubiquitous space whose functionality is used by Web Services. Certain RIs in the prototype (e.g. ones that offer multimedia facilities) are provided with user interfaces that are realized using web browsers and JavaScript. Because the CADEAU prototype does not require deploying any additional software onto the RI's PC, any PC equipped with an Internet connection and a web browser can be turned into a new CADEAU Resource Instance by opening the browser and typing in the HTTP registration request. This triggers the RI's web browser to load the necessary scripts that belong to the Resource Control Driver. After that, the RI can be communicated with and controlled through the Web Service interface.

CADEAU application. The example application that we presented in the overview is implemented as a set of MIDlets on the mobile device, the Content Browser, the Media Web Services, and MIDlets to assist the composition phase. The Content Browser Web Service enables users to browse dynamically generated HTML pages on a remote Display RI. The first phase of the application (i.e. the *collecting* phase) starts when the user chooses a topic by touching an RFID tag in the newspaper. This action initializes the Content Browser and also loads the UI MIDlet so that the user can control the Content Browser from his or her mobile device. Upon receiving a command from the user (i.e. from the UI MIDlet), the Content Browser generates an HTML page and sends it to the dedicated Display RI which loads the page into the web browser (i.e. displays the page to the user). The user navigates on this HTML page and checks and unchecks articles by sending commands from the mobile device. These commands are forwarded to the Content Browser which either generates a new HTML page or commands the RI to update the page that is currently being displayed. User selections are stored in an XML file which is used as a playlist during the second phase of the application (i.e. the *delivering* phase). The second phase of the application involves the playback of the multimedia content chosen by the user. The multimedia files are stored on multiple Media Web Services which provide access to the files on request. The playback is realized by the Display and Speaker RIs that implement the open source JWMediaPlayer and the JWImageRotator Flash players[4]. These RIs support

[3] `http://ajaxpatterns.org/HTTP_Streaming`

[4] JWMediaPlayer and JWImageRotator Flash players are available from `http://www.joerenwijering.com`

Parameter	Mandatory	Example Value	Description
Service	Yes	MultimediaPlayer	Id of the target Web Service
Event	Yes	Play	Describes the event to be sent to the Web Service
ResourceId	No	000000001	List of RIs to be allocated to a Web Service
IsAsync	No	True	Has to be set 'true', if the event does not require setting up a session
Playlist	No	playlist.xml	The URL of the playlist to be shown at a Display RI

Table 1: The parameters used in the communication between the MIDlets, the UIG and the Web Services.

rendering of multimedia files in various formats, support streaming over the network and accept dynamic playlists.

Although the example application presented utilizes only the multimedia facilities, the CADEAU prototype supports other types of ubiquitous resources whose functionality can be accessed using a Web Service.

5.2 The CADEAU communication protocol

The components of the CADEAU server, the Web Services and the RIs communicate with each other using the HTTP protocol. The messages, sent between the Web Services and the UI MIDlet, are encapsulated into the HTTP GET requests, while the messages, sent between the Web Services and the RIs, are transmitted using the POST requests. The HTTP requests include several parameters in the GET URL, while the POST requests include a message with several commands in the POST body. Each message can accommodate multiple parameters. These parameters are either mandatory or optional (e.g. service-specific) parameters. Example parameters for GET requests are listed in Table 1. The mandatory parameters always specify the recipient of the message (i.e. the target Web Service or the subsystem) and the event to be sent. The events are the administrative commands or the service-specific actions that are used to change the state of the RIs (e.g. to update the UI of the resource).

The administration commands are always dispatched to the Admin Control Component that processes and performs the requested commands (e.g. adds a new RI description to the Resource database). Unlike the administration commands, the service-specific and error messages are dispatched directly to the target Web Services and then routed to the RIs through dedicated Resource Control Drivers.

Figure 11 illustrates how the CADEAU subsystems communicate during the browsing phase of the application scenario. As can be seen, the phase starts when the user touches an RFID tag in the newspaper and then presses a button on his or her mobile device to scroll down the displayed HTML page.

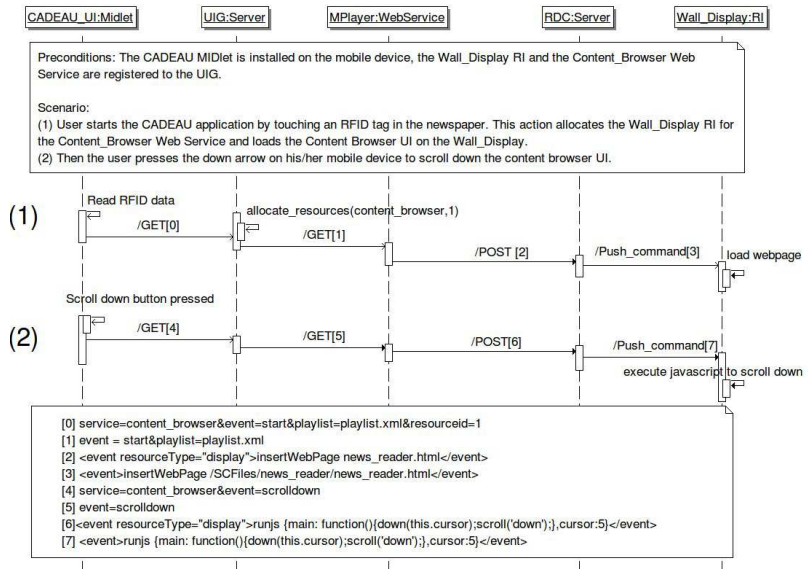


Fig. 11: Communication between the subsystems of CADEAU.

5.3 The application composition process

The CADEAU prototype supports the user-controlled composition (based on the manual and the semi-autonomic methods) as well as the automatic composition. The composition process of the latter is presented in Figure 12. In this case, the key role is played by the Composition Engine which produces the application configuration according to the predefined optimization criteria. This Composition Engine is based on the application allocation algorithm that we reported earlier in (Davidyuk et al., 2008b). This Composition Engine is implemented as a C++ library that takes two XML files as input, (i) the list of available RIs and (ii) the application model. The first one is created by extracting from the resource database the descriptions of the RIs that are physically located in the same ubiquitous space as the user. This file is dynamically created at the beginning of the composition process. The second file, the application model, is a static XML file which is provided by the application developers. It encodes the structure of the CADEAU application, i.e., it specifies what types of RIs are needed and how they have to be connected. In addition, the application model describes the properties of RIs (e.g., the minimum bandwidth, screen resolution and such) that are required by the application. Example applications and platform model files are presented in Figures 13a and 13b, respectively.

If the semi-autonomic method is used, the Composition Engine produces three alternative application configurations of the same application (see Figure 14). These configurations are displayed on the mobile device for the user, who can browse these configurations and choose the most suitable one. Then, this configuration is sent to the UIG which commands the WSM to reserve the RIs listed in the configuration and, after that, invokes the CADEAU application. If the automatic method is used, the

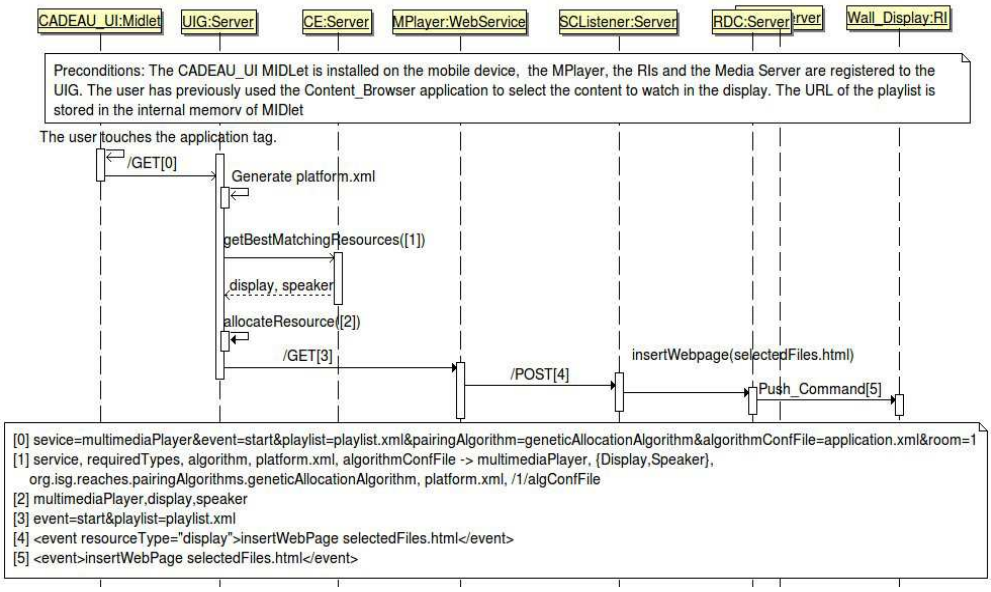


Fig. 12: The composition process of the autonomic method.

Composition Engine produces only one application configuration which is sent directly to the UIG, as shown in Figure 12.

The composition process of the manual interaction method differs from the other two methods, as it does not use the Composition Engine. Instead, the RIs are chosen and provided to the system when the user touches RFID tags with his or her mobile device. This action commands the UI MIDlet to send the Id numbers of the chosen RIs to the UIG. Then, the UIG requests the WSM to allocate the chosen RIs and, after that, invokes the CADEAU application (see Figure 15).

6 User Study and Evaluation

We followed a design process that involved multiple iterations, including the development of the initial prototype followed by a preliminary usability study with some experts in IT. Lack of space does not allow us to report the process in detail, nor the intermediate results. However, the resulting CADEAU prototype and the application are described in the sections 'CADEAU Interaction Design' and 'Implementation', correspondingly. Therefore, in this chapter we describe only the setup, the procedure and our findings from the final user study; this study was conducted with the fully implemented CADEAU prototype.

The primary goal of this user study was to make an assessment of the trade-off between user control and autonomy of the system for application composition, dictated by the user's needs, situation and expertise. This was carried out by comparing the

```
<?xml version="1.0" encoding="UTF-8"?>
<data>
   <Displays>
       <Display id="1">
           <resourceQ>30</resourceQ>
           <bandwidth>20</bandwidth>
           <isDisplay>true</isDisplay>
           <isSpeaker>false</isSpeaker>
           <quality>0</quality>
           <cpu>0</cpu>
       </Display>
   </Displays>
   <Speakers>
       <speaker id="2">
           <resourceQ>30</resourceQ>
           <bandwidth>20</bandwidth>
           <isDisplay>false</isDisplay>
           <isSpeaker>true</isSpeaker>
           <quality>0</quality>
           <cpu>0</cpu>
       </speaker>
   </Speakers>
   <Servers>
       <server id="3">
           <resourceQ>0</resourceQ>
           <bandwidth>20</bandwidth>
           <isDisplay>false</isDisplay>
           <isSpeaker>false</isSpeaker>
           <quality>10</quality>
           <cpu>20</cpu>
       </server>
   </Servers>
</data>
```

(a)

```
<?xml version="1.0" encoding="UTF-8"?>
<data>
<Displays>
    <Display id="259">
        <resourceQ>30.0</resourceQ>
        <bandwidth>30.0</bandwidth>
        <isDisplay>true</isDisplay>
        <isSpeaker>false</isSpeaker>
        <quality>0.0</quality>
        <cpu>0.0</cpu>
    </Display>
                ...
</Displays>
<Speakers>
     <Speaker id="514">
          <resourceQ>50.0</resourceQ>
          <bandwidth>50.0</bandwidth>
          <isDisplay>false</isDisplay>
          <isSpeaker>true</isSpeaker>
          <quality>0.0</quality>
          <cpu>0.0</cpu>
     </Speaker>
                ...
</Speakers>
<Servers>
     <Server file="http://www.ee.oulu.fi/
          ~jiduran/video1.flv" id="100">
         <resourceQ>0.0</resourceQ>
         <bandwidth>30.0</bandwidth>
         <isDisplay>false</isDisplay>
         <isSpeaker>false</isSpeaker>
         <quality>40.0</quality>
         <cpu>20.0</cpu>
     </Server>
                ...
</Servers>
</data>
```

(b)

Fig. 13: Listing of the application model (a) and the platform model (b) files used in the prototype.

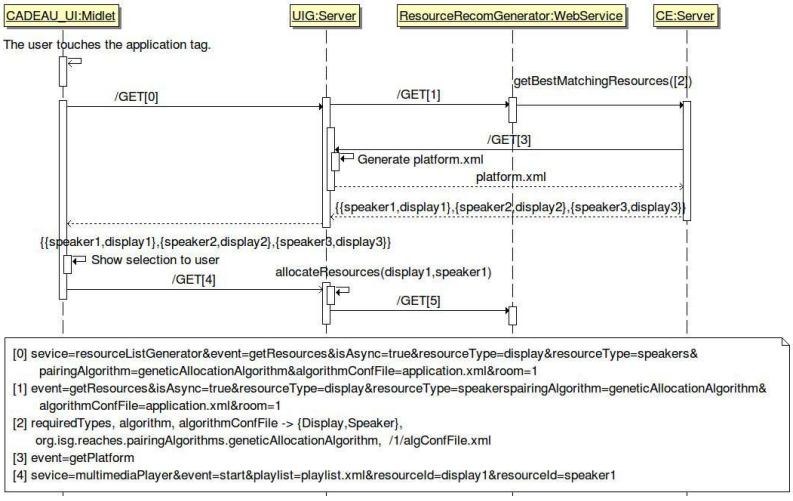


Fig. 14: The composition process of the semi-autonomic method.

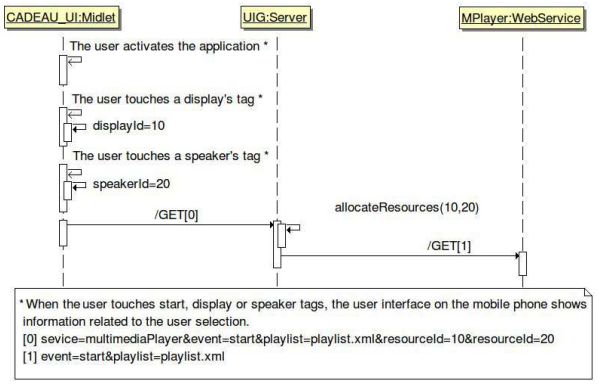


Fig. 15: The composition process of the manual interaction method.

interaction methods and analyzing the factors (e.g. the amount of feedback) contributing to the user's comfort and feeling of control in different contexts. We concentrated our efforts on identifying the issues that were difficult for users to comprehend. The last goal was to gain insights for future research by understanding user experiences, especially the breakdowns perceived by users when they were carrying out tasks in CADEAU.

6.1 Methodology

Thirty participants from the local community of the City of Oulu (Finland) took part in the study. These participants were recruited according to their background and previous experience with technologies, and were assigned to one of three focus groups that each consisted of 10 individuals.

The first group, group A, consisted of IT professionals and students studying towards a degree in computer science or engineering. This group represented experts who deal with mobile technologies on a daily basis as well as having some previous experience with ubiquitous applications. These users were chosen in order to give an expert opinion and provide feedback from a technical point of view. The second group, group B, consisted of less computer-savvy individuals who represented average technology users. As they reported later in the survey, 50% of them never or very rarely use mobile phones beyond calling or texting. The participants in the last group (group C) were carefully screened to ensure that their computer skills and experience were minimal and none of them had any technical background. This group represented a variety of professions, including economists, biology students, a manager, a planning secretary and a linguist. These individuals were chosen to represent conservative users who are less likely to try new technologies and applications. The distribution of gender and age across the participants of these three groups is shown in Table 2. All the users were trying the CADEAU prototype for the first time and two persons had watched the video of the application scenario.

Group	Gender		Age		
	M	F	≤25 y.o.	26-30 y.o.	30+
(A) IT experts	70%	30%	30%	40%	30%
(B) Average users	70%	30%		80%	20%
(C) Non-tech. users		100%	60%	30%	10%

Table 2: Demography of the user study.

The study was carried out at the Computer Science and Engineering Laboratory at the University of Oulu. Two adjacent meeting rooms were converted into ubiquitous spaces prior to the experiment. Each of these spaces was fitted with 6-8 multimedia devices of different kinds. The experiment began in the nearby lobby, so that the users could see the spaces only during their testing session.

Each participant came to our laboratory individually for a session that took approximately an hour. At the beginning the users were given a short introduction in which the functionality of the system was demonstrated using the newspaper and one display located in the lobby. The users, who were unfamiliar with RFID technology, were given additional explanations and time to practice reading RFID tags using the mobile device. Then, each user was asked to perform first the *collecting* task and then the *delivering* task from the CADEAU example application (see the section 'Overview of CADEAU' for details). All participants had to perform each task twice, using different interaction methods in the two ubiquitous spaces. That is, each participant used one of the following combinations: manual and autonomic, manual and semi-autonomic or automatic and semi-autonomic. The experiment was organized in such a way that each of the three methods was used an equal number of times in each focus group.

The participants were encouraged to ask questions, give comments, point out difficult issues and think out loud during the experiment. Since most of the users had had little or no experience with similar systems in the past, all the users were explicitly told that they could not break the system or do anything wrong. After the tasks were completed, the users were asked to fill in an anonymous questionnaire and then discuss their experience with the observers. We used the questionnaire to compare the interaction methods while the interview focused on collecting feedback on the concept and the system in general.

6.2 Results

Although we did not set a strict time limit for completing the assignments and asked the users to finish when they felt they understood how the application and the system work, the users belonging to group A completed the tasks in a significantly shorter time than users from the other two groups (B and C). This is because, in most cases, the experts omitted the preamble and thus could proceed directly to the experiment with the CADEAU application.

User willingness to delegate control. In this section we present an analysis of user preferences towards certain interaction methods in different contexts. This anal-

ysis is based on the anonymous questionnaires and the user feedback collected during interviews.

1. **Manual method.** User opinions regarding the manual method were similar across all the focus groups. Users preferred to rely on the manual method when they had already thought about some specific application configuration and hence wanted the system to realize this exact configuration. As an example, one participant described a situation where he was giving a talk and needed a configuration with two display resources cloning each other. This example also points to another important factor - the reliability of the interaction method. Our users felt that the manual method provides the most control. This was best expressed by a non-technical user 'I really feel that I control it [*the resource*] when I touch it'. Another factor affecting the choice of the manual method was familiarity with the ubiquitous space. The manual method was certainly preferred when users were familiar with the environment and knew the location of each resource. Almost all the users mentioned their homes and work places as such environments. As for the public environments, the users chose to rely on the manual method when they wanted privacy (e.g. when browsing a family photo album in a cafeteria) or when they wished to avoid embarrassing situations involving other individuals. This last finding is in line with the results of the experiments reported by Vastenburg et al. (2007). They concluded that users in general prefer to rely on the manual selection if they are involved in a social activity. As ubiquitous applications can be composed of non-visual resources as well (i.e. content providers, servers, so called 'hot spots' and many others), participants were asked whether they prefer to manually choose these resources as well. Surprisingly, users from all three groups answered that they trust the system to choose these resources automatically, and find a configuration that leads to the best overall application quality.
2. **Semi-autonomic method.** Users liked this method as they could control everything on the mobile device without needing to walk anywhere. This feature was also found useful when users wanted to 'hide' their intentions (i.e. while preparing to use a resource) in certain cases. As we were told by a non-technical user (she was assigned to compare this and the manual method), she would always prefer the semi-autonomic method as she felt uncomfortable when touching resources in front of bystanders. We hope such attitudes will change when the RFID technology becomes a part of our daily lives. Some of the expert users found this method useful, too. However, they stated that they wanted to know the selection criteria before they could fully trust the method. The fact that the criteria was hardwired in the application seemed to be the major shortcoming of the method. Besides, as an expert user later admitted, he would trust the method more if he were able to use it for longer periods of time. Thus, a better approach would be to run the experiment over the course of several days and compare the initial user evaluation scores with the scores obtained at the end of the experiment. In particular, Vastenburg et al. (2007) observed in their experiment that user confidence and ease of use increase over time. Several users (groups A and B) admitted that the semi-autonomic method is preferable in situations where one is in a hurry. They pointed out that the UI of the method displays the configurations on the mobile

phone, so users can quickly take a look before starting the configuration if they are hesitating about the choice proposed by the system. One user suggested that this method could save his last choice (i.e. the application configuration used in some similar context) and suggest this configuration among the other options. We believe this feature will increase the usefulness of the method in future.

3. **Automatic method.** Although the expert users were cautious about using this method on a daily basis, they found it useful in several situations. For example, someone is entering a ubiquitous space with an open application on his or her device and (s)he is hesitating (or confused) to choose a configuration on his or her own. In that case the system could automatically choose an application configuration after a short delay. However, the majority of the expert users admitted during the interview, that they need to feel that the method is reliable in order to rely on it. According to these users, reliability means for them that the outcome of the method is predictable. As one expert commented, 'I need to know what happens next and if this system is still surprising me, this surprise has to be a positive one'. The users from group B suggested a public space with many possible combinations of resources as another example situation where this method could be used. But, like in the case with the semi-autonomic method, they requested to know the decision (i.e. the choice) of the system and what information was used by the system to take it, in order for this choice to be corrected if necessary. This confirms theoretical findings reported by (Hardian et al., 2008), where the authors suggested exposing the context information and logic to users in order to ensure that the actions (e.g. adaptation to context) taken on behalf of the users were both intelligent and accountable. The non-technical users were more enthusiastic towards the automatic method than their expert colleagues. Some non-technical users suggested that this method could be used in most situations. As one of them commented, 'it is just nice when things are done automatically'. Although she added that she prefers other methods if she needs to hide her application or its content. The autonomic method was also appreciated for its speed and easiness. These factors were dominant for non-technical users in cases where a person is in a hurry.

Subjective comparison. These results were collected using questionnaires where users had to answer questions like 'how easy was the method to use' (1='very difficult', 5='very easy') or 'did it require apparent effort in order to understand the method' (1='I did not understand it immediately and it took me a long before I understood it', 5='I understood it immediately and did not have to ask any questions'). The results of the comparison between the three methods are shown in Figure 16.

1. **Easiness.** As can be seen from the graphs, the expert users (group A) graded the automatic method as the easiest to use (4.8 pts), while the manual and the semi-autonomic methods scored the second (4.4 pts) and the third (4 pts) place, respectfully. The non-technical (group C) users gave the automatic method the highest grade (4.6 pts) while the manual method was given the lowest (3.7 pts). The group B users gave approximately the same scores to all three methods. Although the scores received from the experts and the average user groups were somewhat expected, the non-technical users surprisingly gave the lowest grade to the manual

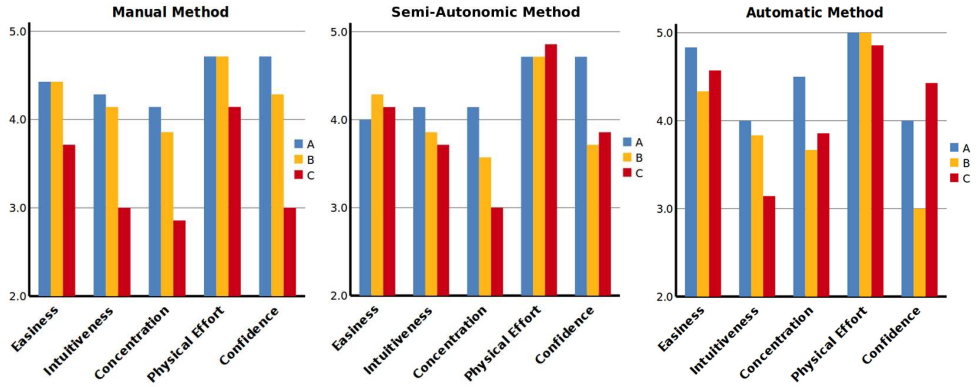


Fig. 16: Comparing the manual (left), the semi-autonomic (middle) and the autonomic (right) methods regarding easiness, intuitiveness, concentration and other characteristics across three focus groups (A=experts, B=average users, C=non-technical users).

method. A possible explanation could be that none of them had any previous experience with RFID technology, thus the users did not feel comfortable using it.

2. **Intuitiveness.** The answers given by the expert and average users followed each other hand in hand, although the average users gave lower overall grades: they chose the manual method as the most intuitive (4.1 pts) and they gave the lowest grade (3.8 pts) to the automatic method. As one user from this group (B) commented, the automatic method was not very intuitive because its choice criteria was not clear at all. We believe this is caused by the fact that the users did not have access to the optimization criteria on the UI. The non-technical users named the manual method as the least intuitive (3 pts). As in the case with the 'easy to use' characteristic, we believe this to be due to a lack of experience and, hence, difficulties with understanding the RFID technology. One user from group C could not complete the assignment using the manual method, but had to interrupt the experiment and ask the observers for explicit instructions on what she has to do. As was later revealed, she always preferred to use some 'default configuration' when she was working on her computer, thus she was confused when she herself had to make a choice in the first place.
3. **Concentration.** The expert users found that the manual and the semi-autonomic methods are equally demanding (4.1 pts) and require higher concentration efforts than the automatic method (4.5 pts). The results of the average user group showed a similar tendency, although these users gave lower grades to all three methods. The answers given by the non-technical users were in line with the results of the other groups.
4. **Physical effort.** We expected our users to choose the manual method as the most demanding and the automatic one as the least demanding in terms of physical effort. Although we guessed right in the case of the non-technical group, the other two groups (A and B) gave equal scores to both the manual and the semi-autonomic

methods. As we observed during the experiment, these two groups behaved actively and were walking around to identify resources also when using the semi-autonomic method. On the other hand, the group C users preferred to stay on the spot and were focusing on the mobile phone's UI during the experiment. We believe that this observation is also linked to the confidence factor, which we discuss next.

5. **Confidence.** We expected the expert users to demonstrate a higher level of confidence with the automatic method because they deal with similar technologies on a daily basis. The non-technical users were supposed to show greater confidence when using the manual method, because we believed that the outcome of this method was easier for them to predict. Finally, we expected the average users to show results similar to the expert users. The results showed quite the opposite picture. The non-technical users expressed the highest level of confidence when using the automatic method (4.4. pts) and gave lower scores to the mixed initiative (3.9 pts) and the manual (3 pts) methods. The expert users were equally confident with both the manual and the semi-autonomic methods (4.7 pts) and gave lower scores to the automatic method (4 pts). The opinion of the average user group was in line with the experts. Surprising was the fact that although the non-technical group found the automatic method mediocre in terms of intuitiveness (3.1 pts) they nevertheless demonstrated the highest confidence (4.4. pts) with this method. This means that the non-technical users were overconfident when using the automatic method. The experts and the average users demonstrated interesting opinions as well. They were in favor of the manual and the semi-autonomic methods. The explanation of this phenomenon is that these two user groups have, in, general, lower trust in the autonomy of systems. This hypothesis was also confirmed during the exit interviews.

Design of RFID icons and physical browsing. The user evaluation of the prototype helped us to pinpoint two important usability related issues. The first is the graphical design of the icons that appear on the front side of RFID tags and the second issue is so called physical browsing.

Icon design is an essential issue that influences the intuitiveness and ease of use of RFID-based interfaces (Sánchez et al., 2009). The role of icon design is to communicate the meaning of tags to users in a precise and non-ambiguous manner. In other words, it allows users to correctly recognize and interpret the action that is triggered when a certain tag is touched. Therefore, we included the evaluation of the icon design as a part of this user study. We were particularly interested in evaluating the icon design of the start tag (see Figure 8a) which users had to touch on entering the ubiquitous space. The designer of the tag aimed to communicate to the users that they need to touch this tag in order to deliver the information that they have in their mobile devices. Hence, our participants were asked in the questionnaire to describe the action that, according to their option, is best associated with this tag. Similarly, users had to describe three other designs. The icon used was correctly described by 70% of the expert users, 40% of the average users, and 80% of the non-technical users. We found this result satisfactory for this prototype, although the icon design could be refined in the next design iteration.

Another issue that we studied was physical browsing, which is the mechanism that helps users to identify Resource Instances in the ubiquitous spaces. This is especially challenging when users need to preview an application configuration (offered by the system) on their mobile device while they are not familiar with the ubiquitous space. Such a mechanism should allow users to associate each application configuration with corresponding resource instances in the environment. The CADEAU prototype implements this mechanism as part of the semi-autonomic method and allows users to preview (or validate) chosen application configurations by clicking the middle button on the mobile phone. This commands the display RIs to show 'splash screen' and the audio RIs to play an audio tone. We asked our users to suggest alternative mechanisms to identify resources in ubiquitous spaces. Among the most interesting suggestions were the map-like user interface with a compass, concise textual descriptions on the mobile phone (including, e.g. color and size of the resources), a radar-like user interface and an LED-based panel where all resources are marked. In spite of these suggestions, users generally liked the current validation mechanism implemented in CADEAU.

7 Discussion and Future Work

Although ubiquitous technology aims to be autonomous and invisible, there is still a need for user control and intervention. This is best explained by one participant during the exit interview: 'if it [*the application*] does not read my mind, how does it know what I want?'. Based on the results of this study developers of ubiquitous technology could take into account the preferences of users who have varying degrees of expertise. For example, the expert users need to understand the details of application operation and therefore they require most of the adaptation and configuration processes to be explicit. The average computer users have similar requirements to the experts and they also expressed less trust in the autonomy of the system. For example, they need to be able to override the system's choices and adjust the selection criteria. However, these users may in certain situations rely on autonomy of the system. Users with little or no experience in technologies seem to be overconfident when using the system and thus prefer to rely on default or autonomic options. These users, however, still need to be able to control the application or the system, if necessary.

Among the other factors that influence willingness to delegate control to the prototype were named privacy, familiarity with the environment, the presence of other persons, time pressure and predictability of the outcome of the system's choices. These factors were almost equally important across the three user groups involved in the experiment. For example, users explicitly prefer to rely on the manual method when they wish to hide the multimedia content from other persons. Depending on how familiar users were with the environment, they tended to rely on the manual method if they were very familiar (e.g. at home, or in the office) and chose the automatic or the semi-autonomic methods in the environments less familiar to them. In the presence of other persons, users in general tried to avoid choices that might lead to unpleasant and embarrassing situations. For example, many users liked the semi-autonomic method as they could hide their intentions when preparing to use certain resources with the application. However, user preferences in this case depended on how confident the user

was with the prototype. For example, the expert and average users named the manual method as the most preferable to use when other persons are present. On the other hand, the non-technical users were happy to rely on the semi-autonomic method in this situation. Generally, the expert and the average users tended to use the semi-autonomic and the automatic methods if they were able to predict the behavior of the prototype. Otherwise their preferred method was the manual one. Although the non-technical users admitted that the automatic and the semi-autonomic methods were lacking in intuitiveness, they did not impose high requirements on the predictability of the prototype, as the other user groups did.

Another important finding was the fact that the average users expressed opinions similar to those given by the expert group. The average users however, gave lower overall scores in the long run than those given by the expert users. Thus, as a conclusion, we find it acceptable to rely on the expert opinions when evaluating features related to manual or semi-manual system configuration. On the other hand, we find it unacceptable to rely only on expert or even average users when assessing the automatic (or nearly autonomic) features of a system.

Limitations. One of the limitations of our methodology was the fact that we carried out the experiment in the lab. Although CADEAU is meant to be used in various environments including home, office and public spaces, the lab truly represented only the office space. The findings that were related to other environments and were collected during the interviews were based entirely on the personal experiences and the user's subjective understanding of how CADEAU could be used. A better approach could be to perform field studies. However, such an experiment will require significantly greater time and effort. Another limitation was due to the fact that our users were not given a possibility to try the prototype over several days. Although sufficient for our needs, the approach used in the experiment does not study general trends in time. For example, Vastenburg et al. (2007) demonstrate in their experiments that such factors as user confidence and ease of use have a tendency to increase over time. That suggests that the scores obtained in our study could be in fact higher.

Future work. Several promising directions of future research are identified in this study. One of them is the development of control methods that can be adapted to users with various levels of experience in technologies. That is, rather than having a set of 'fixed' control methods that are offered to all users equally, we are interested in developing and evaluating the methods that can be tailored to user expertise and willingness to delegate the control to the system. For example, users themselves could specify the tasks they want to delegate to the system and the tasks they prefer to control manually.

Another issue for future research involves developing a new control method that unites the advantages of the manual and the automatic methods. This new method, the semi-manual method, does not require users to choose all the resources manually. It could work as follows: a user could select some resource instances (s)he wished to use with the application. Then, the missing resources would be assigned and the rest of the configuration would be realized automatically. The major advantage of this new method is that the user could choose the most important resources manually while leaving less important decisions to be made by the system automatically.

An interesting research direction is the end-user composition of applications. This subject studies tools, methods and technologies that allow end-users to develop composite applications in a do-it-yourself fashion. The initial steps towards this research are reported in (Davidyuk et al., 2010).

8 Acknowledgements

CADEAU is the result of a collaborative effort that has been built thanks to the contribution of many people who supported the authors during the development of the prototype and the writing of this chapter. We wish to thank all those who helped us to successfully complete this project, and in particular:

- Marta Cortés and Jon Imanol Duran for taking part in the development of CADEAU;
- Mikko Pyykkonen for the graphical icon design;
- Hanna-Kaisa Aikio and Hillevi Iso-Heiniemi for making the audio narration;
- Jukka Kontinen, Hannu Rautio and Marika Leskelä for their kind support in organizing the user evaluation experiment;
- Simo Hosio, Tharanga Wijethilake and Susanna Pirttikangas for testing the alpha version of CADEAU and for being patient when the prototype did not work;
- All participants in the user evaluation experiment who kindly agreed to take part in lengthy interviews;
- Valérie Issarny and Nikolaos Georgantas from the ARLES team (INRIA Paris-Rocquencourt) for their valuable comments regarding the experimental results;
- Richard James (from INRIA Paris-Rocquencourt) and Minna Katila for English language advice.

This work has been funded by Academy of Finland (as the Pervasive Service Computing project), and by GETA (Finnish Graduate School in Electronics, Telecommunications and Automation).

Bibliography

Beauche, S., & Poizat, P. (2008). Automated Service Composition with Adaptive Planning. In *Proceedings of the 6th International Conference on Service-Oriented Computing (ICSOC'08)*, LNCS 5364, (pp. 530–537). Springer.

Ben Mokhtar, S., Georgantas, N., & Issarny, V. (2007). COCOA: COnversation-based Service Composition in PervAsive Computing Environments with QoS Support. *Journal of Systems and Software*, *80*(12), 1941–1955.

Bertolino, A., Angelis, G., Frantzen, L., & Polini, A. (2009). The PLASTIC Framework and Tools for Testing Service-Oriented Applications. In *Proceedings of the International Summer School on Software Engineering (ISSSE 2006-2008)*, LNCS 5413, (pp. 106–139). Springer.

Bottaro, A., Gerodolle, A., & Lalanda, P. (2007). Pervasive Service Composition in the Home Network. In *Proceedings of the 21st International Conference on Advanced Information Networking and Applications (AINA '07)*, (pp. 596–603).

Broll, G., Haarlander, M., Paolucci, M., Wagner, M., Rukzio, E., & Schmidt, A. (2008). Collect&Drop: A Technique for Multi-Tag Interaction with Real World Objects and Information. In *Proceedings of the European Conference on Ambient Intelligence (AmI'08)*, LNCS 5355, (pp. 175–191). Springer.

Buford, J., Kumar, R., & Perkins, G. (2006). Composition Trust Bindings in Pervasive Computing Service Composition. In *Proceedings of the 4th Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW '06)*, (pp. 261–266). Washington, DC, USA: IEEE Computer Society.

Chantzara, M., Anagnostou, M., & Sykas, E. (2006). Designing a Quality-Aware Discovery Mechanism for Acquiring Context Information. In *Proceedings of the 20th International Conference on Advanced Information Networking and Applications (AINA'06)*, (pp. 211–216). Washington, DC, USA: IEEE Computer Society.

Chin, J., Callaghan, V., & Clarke, G. (2006). An End-User Tool for Customising Personal Spaces in Ubiquitous Computing Environments. In *Proceedings of the 3rd International Conference on Ubiquitous Intelligence and Computing (UIC'06)*, LNCS 4159, (pp. 1080–1089). Springer.

Davidyuk, O., Georgantas, N., Issarny, V., & Riekki, J. (2010). *MEDUSA: A Middleware for End-User Composition of Ubiquitous Applications*. In Mastrogiovanni, F. and Chong, N.-Y. (eds) Handbook of Research on Ambient Intelligence and Smart Environments: Trends and Perspectives. IGI Global. *To appear.*

Davidyuk, O., Sánchez, I., Duran, J. I., & Riekki, J. (2008a). Autonomic Composition of Ubiquitous Multimedia Applications in REACHES. In *Proceedings of the 7th International ACM Conference on Mobile and Ubiquitous Multimedia (MUM'08)*, (pp. 105–108). ACM.

Davidyuk, O., Selek, I., Duran, J. I., & Riekki, J. (2008b). Algorithms for Composing Pervasive Applications. *International Journal of Software Engineering and Its Applications*, *2*(2), 71–94.

Ghiani, G., Patern, F., & Spano, L. D. (2009). Cicero Designer: An Environment for End-User Development of Multi-Device Museum Guides. In *Proceedings of the 2nd*

International Symposium on End-User Development (IS-EUD'09), LNCS 5435, (pp. 265–274). Springer.

Gross, T., & Marquardt, N. (2009). Creating, Editing and Sharing Complex Ubiquitous Computing Environment Configurations with CollaborationBus. *International Journal of Computer Science and Engineering*, *xx*(xxxx), 1–1.

Hardian, B., Indulska, J., & Henricksen, K. (2006). Balancing Autonomy and User Control in Context-Aware Systems - a Survey. In *Proceedings of the 3rd Workshop on Context Modeling and Reasoning (part of the 4th IEEE International Conference on Pervasive Computing and Communication)*. IEEE Computer Society.

Hardian, B., Indulska, J., & Henricksen, K. (2008). Exposing Contextual Information for Balancing Software Autonomy and User Control in Context-Aware Systems. In *Proceedings of the Workshop on Context-Aware Pervasive Communities: Infrastructures, Services and Applications*.

Kalasapur, S., Kumar, M., & Shirazi, B. (2007). Dynamic Service Composition in Pervasive Computing. *IEEE Transactions on Parallel Distributed Systems*, *18*(7), 907–918.

Kawsar, F., Nakajima, T., & Fujinami, K. (2008). Deploy Spontaneously: Supporting End-Users in Building and Enhancing a Smart Home. In *Proceedings of the 10th International Conference on Ubiquitous Computing (UbiComp'08)*, (pp. 282–291). NY, USA: ACM.

Lindenberg, J., Pasman, W., Kranenborg, K., Stegeman, J., & Neerincx, M. A. (2006). Improving Service Matching and Selection in Ubiquitous Computing Environments: a User Study. *Personal Ubiquitous Computing*, *11*(1), 59–68.

Masuoka, R., Parsia, B., & Labrou, Y. (2003). Task Computing - the Semantic Web meets Pervasive Computing. In *Proceedings of the 2nd International Semantic Web Conference (ISWC'03)*, LNCS 2870, (pp. 866–881). Springer.

Mavrommati, I., & Darzentas, J. (2007). End User Tools for Ambient Intelligence Environments: an overview. In *Human-Computer Interaction, Part II (HCII 2007)*, LNCS 4551, (pp. 864–872). Springer.

Messer, A., Kunjithapatham, A., Sheshagiri, M., Song, H., Kumar, P., Nguyen, P., & Yi, K. H. (2006). InterPlay: a Middleware for Seamless Device Integration and Task Orchestration in a Networked Home. In *Proceedings of the 4th Annual IEEE Conference on Pervasive Computing and Communications*, (pp. 296–307). IEEE Computer Society.

Nakazawa, J., Yura, J., & Tokuda, H. (2004). Galaxy: a Service Shaping Approach for Addressing the Hidden Service Problem. In *Proceedings of the 2nd IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems*, (pp. 35–39).

Newman, M., & Ackerman, M. (2008). Pervasive Help @ Home: Connecting People Who Connect Devices. In *Proceedings of the International Workshop on Pervasive Computing at Home (PC@Home)*, (pp. 28–36).

Newman, M., Elliott, A., & Smith, T. (2008). Providing an Integrated User Experience of Networked Media, Devices, and Services through End-User Composition. In *Proceedings of the 6th International Conference on Pervasive Computing (Pervasive'08)*, LNCS 5013, (pp. 213–227). Springer.

NFC Forum (2010a). Near Field Communication (NFC) Standard for Short-Range Wireless Communication Technology, `http://www.nfc-forum.org`.

NFC Forum (2010b). NFC Data Exchange Format (NDEF), `http://www.nfc-forum.org/specs/`.

Paluska, J. M., Pham, H., Saif, U., Chau, G., Terman, C., & Ward, S. (2008). Structured Decomposition of Adaptive Applications. In *Proceedings of the 6th Annual IEEE International Conference on Pervasive Computing and Communications (PerCom'08)*, (pp. 1–10). IEEE Computer Society.

Preuveneers, D., & Berbers, Y. (2005). Automated Context-Driven Composition of Pervasive Services to Alleviate Non-Functional Concerns. *International Journal of Computing and Information Sciences*, *3*(2), 19–28.

Ranganathan, A., & Campbell, R. H. (2004). Autonomic Pervasive Computing Based on Planning. In *Proceedings of the International Conference on Autonomic Computing*, (pp. 80–87). Los Alamitos, CA, USA: IEEE Computer Society.

Rantapuska, O., & Lahteenmaki, M. (2008). Task-based User Experience for Home Networks and Smart Spaces. In *Proceedings of the International Workshop on Pervasive Mobile Interaction Devices*, (pp. 188–191).

Rich, C., Sidner, C., Lesh, N., Garland, A., Booth, S., & Chimani, M. (2006). DiamondHelp: a New Interaction Design for Networked Home Appliances. *Personal and Ubiquitous Computing*, *10*(2-3), 187–190.

Riekki, J., Sánchez, I., & Pyykkonen, M. (2010). Remote Control for Pervasive Services. *International Journal of Autonomous and Adaptive Communications Systems*, *3*(1), 39–58.

Rigole, P., Clerckx, T., Berbers, Y., & Coninx, K. (2007). Task-Driven Automated Component Deployment for Ambient Intelligence Environments. *Pervasive and Mobile Computing*, *3*(3), 276–299.

Rigole, P., Vandervelpen, C., Luyten, K., Berbers, Y., Vandewoude, Y., & Coninx, K. (2005). A Component-Based Infrastructure for Pervasive User Interaction. In *Proceedings of Software Techniques for Embedded and Pervasive Systems*, (pp. 1–16). Springer.

Rouvoy, R., Barone, P., Ding, Y., Eliassen, F., Hallsteinsen, S. O., Lorenzo, J., Mamelli, A., & Scholz, U. (2009). MUSIC: Middleware Support for Self-Adaptation in Ubiquitous and Service-Oriented Environments. In *Software Engineering for Self-Adaptive Systems*, (pp. 164–182).

Sánchez, I., Riekki, J., & Pyykkonen, M. (2009). Touch&Compose: Physical User Interface for Application Composition in Smart Environments. In *Proceedings of the International Workshop on Near Field Communication*, (pp. 61–66). IEEE Computer Society.

Sintoris, C., Raptis, D., Stoica, A., & Avouris, N. (2007). Delivering Multimedia Content in Enabled Cultural Spaces. In *Proceedings of the 3rd international Conference on Mobile Multimedia Communications (MobiMedia'07)*, (pp. 1–6). Brussels, Belgium: ICST.

Sousa, J. P., Poladian, V., Garlan, D., Schmerl, B., & Shaw, M. (2006). Task-based Adaptation for Ubiquitous Computing. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, *36*, 328–340.

Sousa, J. P., Schmerl, B., Poladian, V., & Brodsky, A. (2008a). uDesign: End-User Design Applied to Monitoring and Control Applications for Smart Spaces. In *Proceedings of the Working IEEE/IFIP Conference on Software Architecture*, (pp. 71–80). IEEE Computer Society.

Sousa, J. P., Schmerl, B., Steenkiste, P., & Garlan, D. (2008b). *Activity-Oriented Computing*, chap. XI, (pp. 280–315). Advances in Ubiquitous Computing: Future Paradigms and Directions. IGI Publishing.

Takemoto, M., Oh-ishi, T., Iwata, T., Yamato, Y., Tanaka, Y., Shinno, K., Tokumoto, S., & Shimamoto, N. (2004). A Service-Composition and Service-Emergence Framework for Ubiquitous-Computing Environments. In *Proceedings of the 2004 Workshop on Applications and the Internet (as part of SAINT'04)*, (pp. 313–318).

Vastenburg, M., Keyson, D., & de Ridder, H. (2007). Measuring User Experiences of Prototypical Autonomous Products in a Simulated Home Environment. *Human Computer Interaction (HCI)*, *2*, 998–1007.

Wisner, P., & Kalofonos, D. (2007). A Framework for End-User Programming of Smart Homes Using Mobile Devices. In *Proceedings of the 4th IEEE Consumer Communications and Networking Conference (CCNC'07)*, (pp. 716–721). IEEE Computer Society.

Zipf, G. K. (1949). *Human Behavior and the Principle of Least Effort*. ix. Cambridge, MA: Addison-Wesley Press.