Context-Aware Services in Smart Environments: A Review

Oliver Brdiczka

Telecooperation Group
TU Darmstadt
Hochschulstraße 10
D-64283 Darmstadt
brdiczka AT tk.informatik.tu-darmstadt.de

Abstract This paper addresses the topic of providing context-aware services in smart environments. We review recent work in smart environments, and further define and debate essential aspects concerning context-aware service supply. Future challenges for the research in context-aware smart environments are mentioned and discussed.

1 Introduction

Technological progress in the domains of sensor networks, computer vision and artificial intelligence has enabled the construction of augmented “smart” environments. These environments are equipped with numerous sensors and interaction devices, permitting to sense, interpret and react to human activity in the scene. Real context-awareness constitutes a major challenge for developers of smart environments and goes beyond simply sensing and reacting to human actions. Issues like intelligibility, scrutability and trust need to be considered when realizing a useful context-aware system. This paper aims at reviewing and discussing current research issues in smart environments and context-aware service supply. We first present several examples of smart environments including offices, homes, and classrooms. Based on these examples, essential notions like calm technology or implicit human computer interaction are defined and important issues like calibration of trust are introduced and discussed. An outline and description of remaining challenges concludes this paper.

2 Smart Environments

In the 1980s and early 1990s, Xerox PARC researcher Mark Weiser developed the concept of ubiquitous computing, following the principle that:

“The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it.” [43]
The integration of computing devises into every-day environments has been one of the predominant trends over the last decade. Cell phones, PDAs and laptop computers as well as WLAN networks have become part of almost every household. This trend enables computer-everywhere environments. These environments are augmented with multiple sensors and interaction devices. Coen [16] defines the term of “intelligent” environments as “spaces in which computation is seamlessly used to enhance ordinary activity”. The objective is to make computers not only user-friendly but also invisible to the user. Interaction with them should be in terms of forms that people are naturally comfortable with.

One of the very first intelligent environments, the Intelligent Room [15], has been realized at MIT AI Laboratory. The Intelligent Room is laid out like an ordinary conference room, with a large table surrounded by chairs (Figure 1). Mounted at various places in the conference area are twelve video cameras, which are used by computer vision systems. Two video projectors, several video displays as well as audio devices and wireless microphones further augment the environment. The objective of the Intelligent Room was to experiment with different forms of natural, multimodal human-computer interaction (HCI) during what is traditionally considered non-computational office activity. Numerous computer vision, speech and gesture recognition systems are used to detect what inhabitants are doing and saying.

A similar office environment has also been developed at INRIA Rhône-Alpes. The SmartOffice [28] comprises a whiteboard area and a large office
desk completed with a computer workstation in the center of the room. 50 sensors (cameras and microphones) and three actuators (a video projector and two speakers) are installed within the environment. The MagicBoard [24] is the main “actuator” for the SmartOffice, letting users combine digital and physical information on the whiteboard. Mobile and wide-angle cameras permit the use of computer vision recognition systems. Eight microphones distributed across the ceiling are used for speech recognition. The objective was to monitor the user in order to anticipate user intentions and to augment the environment in order to communicate useful information.

At XRCE, an intelligent workplace environment has been realized [4]. The intelligent workplace environment is laid out like a normal individual workplace, comprising a desktop computer, a PDA device and an office telephone. The environment is augmented with PC and phone usage sensors, PDA location and ambient sound sensors as well as a PDA user feedback form. The PDA form was used by the users to give feedback on their current office activity. The objective of the intelligent workplace at XRCE was to sense individual office activity and to provide sensed information to other users (e.g. in order to derive possible availability).

Mozer [31] developed one of the first intelligent home environments at the University of Colorado. The Adaptive House has been implemented in an actual residence that was renovated in 1992, at which time the infrastructure needed for the project was incorporated into the house. The home laboratory is equipped with an array of over 75 sensors which provide information about the environmental conditions that are monitored – temperature, ambient light levels, sound, motion, door and window openings – and actuators to control the furnace, space heaters, water heater, lighting units, and ceiling fans. The objective of the Adaptive House was to make life more comfortable for inhabitants and to conserve energy at the same time. By using inferred occupancy and usage patterns in the home, the Adaptive House was to adjust automatically room heating, water heating and room illumination. Explicit Sensing and recognition of human activities in the house was not the focus of the Adaptive House Project.

The EasyLiving Project [10] at Microsoft Research was concerned with the development of an architecture and suitable technologies for intelligent home environments. The focus of EasyLiving laid on technologies for middleware (to facilitate distributed computing), geometric world knowledge and modeling (to provide location-based context), perception (to collect information about environment state) as well as service abstraction and description. Input devices can include an active badge system, cameras, wall switches, and sensitive floor tiles. Output devices can include home entertainment systems, wall-mounted displays, speakers, and lightening. Stereo computer vision tracking is used to derive the location of people in the environment as well as to maintain their identity while they are moving around. Radio-frequency (RF) wireless-LAN-enabled mobile devices
are located based on the signal strength of known infrastructure access points. A geometric world model is used to derive the spatial relationship between entities in the environment. The location is used to infer a person’s intent or activity based on his or her position. The objective of EasyLiving was to enable typical PC-focused user activities to move from a fixed desktop into the environment as a whole. Several intelligent space applications like movable desktop sessions or location-based media control have been implemented.

![Figure 2: Layout of the augmented home environment MavHome at University of Texas at Arlington (picture from [47])](image)

The MavHome Project [17] developed a smart home environment at the University of Texas at Arlington. The MavHome acts as an autonomous intelligent agent that perceives its environment through the use of sensors, and can act upon the environment through the use of actuators. Perception is managed through light, humidity, temperature, smoke, gas, infrared motion, and switch sensors deployed in the environment. Main actuators are the control of lightening and blinds, water heater, different video and screen displays, sprinkler and VCR. Location-based media control and tracking is also provided. The objective was to manage the home automatically in a way that maximizes productivity and comfort of its inhabitants, minimizes the costs of operating the home, and ensures the maximum security of the home and collected/personal data.

At MIT Media Lab, Bobick et al. [9] constructed an early example of an intelligent entertainment environment. The KidsRoom was a
perceptually-based, interactive, narrative playspace for children. The environment, which resembles a children’s bedroom, uses two large back-projected video screens, four speakers, theatrical lightening, three video cameras, and a microphone array to perceive and to interact with the children. Computer-vision algorithms on the video images of the scene are used to identify the activity of several children. Constant tracking of the positions of up to four children and a strong story context are used to limit the number possible children’s activities. Images, music, narration, light, and sound effects generated by the system guide the children through the story. The strong story context defines the possible children’s activities at the actual state of the play and the appropriate reactions to be taken by the system. The objective of the KidsRoom was to explore the design of interactive spaces and to develop suitable computer vision techniques.

Figure 3: Overview of the Smart Classroom system at Tsinghua University (picture from [41])

The eClass Project [2] (formerly known as Classroom 2000 Project) concerned the development of an intelligent education environment at Georgia Tech. The project constructed a prototype classroom environment and the necessary software infrastructure to seamlessly capture much of the rich interaction that occurs in a typical university lecture. The classroom is augmented with single audio-video stream recording facilities, electronic whiteboards, and personal pen-based interfaces. Further, software and WWW access facilitate automatic capture and content-based access of multimedia information in the educational setting. The objective
Oliver Brdiczka

of the eClass Project was to automate the capture of individual and group activity in the classroom and to provide an easily accessible interface that integrates this information together.

The Smart Classroom Project [46] constructed an intelligent classroom environment at Tsinghua University. The augmented classroom has two wall-size projector screens, one on the front wall and the other on a side wall, and several cameras that are deployed in the environment (Figure 3). Additional cameras are installed on the computers of remote students. The teacher wears a wireless headset microphone to capture his or her speech. A touchsensitive board further enhances the room. Voice-recognition, computer vision techniques and activity recognition are used to permit the simultaneous instruction of local and remote students. The objective of the Smart Classroom Project was to seamlessly integrate tele-education and traditional classroom activities. The system turns a physical classroom into a user interface for tele-education.

This section has presented different examples of intelligent environments in the domains of workplace, housing and education. These augmented environments involve various research disciplines, ranging from computer science, over social science to psychology. In computer science, ubiquitous or pervasive computing [36] integrates computing into these environments, nomadic computing [26] mobilizes computing devices, and ambient intelligence [39] helps making these environments smart(er). In the field of ambient intelligence, we consider sensing and responding to human and environmental context to be a key feature for achieving augmented intelligent environments. The following section defines and discusses the terms context and context-awareness as well as proactive services that context-aware systems supply.

3 Context-Aware Services: towards unobtrusive, proactive system behavior

In 1996, Weiser and Brown [44] introduced the notion of calm technology, described as:

“If computers are everywhere they better stay out of the way, and that means designing them so that the people being shared by the computers remain serene and in control.”

Computing systems should “stay out of the way”, while providing useful and enriching services. In the context of smart environments and smart artifacts, Streitz et al. [42] distinguish two types of service behavior: system-oriented, importunate smartness and people-oriented, empowering smartness. System-oriented, importunate smartness enables the environment to take certain self-directed actions, while people-oriented, empowering smartness focuses on empowering the users to make decisions and take responsible actions.
System-oriented, yet unobtrusive smartness constitutes a major challenge as it addresses two important issues:

1. sensing, and recognizing user behavior, needs and intents,

2. while keeping the user informed and in control.

The system services are to be supplied without interrupting the user’s current task and activity. In addition, they should be predictable for the user (principle of least surprise [5]). These services will not replace human-computer interaction itself because depending on the complexity of the current task of the user, deriving user behavior, intent, or needs may be too difficult. The main purpose is to reduce the communication workload of the user when working on his tasks. Obviously necessary actions may be automated and so the user can concentrate on essential work and human-computer interaction tasks.

The automatic supply of system services is addressed by the term proactive system behavior. Salovaara and Oulasvirta [35] outline that:

“... the concept proactive refers to two critical features of a system: 1) that the system is working on behalf of (or pro) the user, and 2) is taking initiative autonomously, without user’s explicit command.”

Proactive systems are thus acting on their own initiative on behalf of the user. Schmidt [39] extends this notion to implicit human computer interaction (iHCI). iHCI is the interaction of a human with the environment and with artifacts which is aimed to accomplish a goal. Within this process the system acquires implicit input from the user and may present implicit output to the user. Implicit input are actions and behavior of humans, which are done to achieve a goal and are not primarily regarded as interaction with a computer, but captured, recognized and interpreted by a computer system as input. Implicit output is not directly related to an explicit input and is seamlessly integrated with the environment and the task of the user.

Actions and behavior of humans, captured, recognized and interpreted by a computer system are the input and basis for iHCI. The computer systems must hence be aware of what the humans are doing in the environment and determine the context in which human actions take place. This issue is normally addressed by the term context-aware computing.

The word context is composed of “con” (with) and “text” and refers thus to the meaning that must be inferred from adjacent text. Winograd [45] refers to context as a shared reference frame of ideas and objects that are suggested by a text. Context is a consensual space, called “common ground” [14], that establishes a framework for communication based on shared experience. Such a shared framework provides a collection of roles and relations with which to organize meaning for a phrase.

Schilit and Theimer [37] first defined the term context-awareness by:
“location information [that] enables software to adapt according to its location of use, the collection of nearby people and objects, as well as the changes to those objects over time”

This definition is particularly useful for mobile computing applications. An example is the context-aware tourist guide system proposed by Cheverst et al. [11]. The system combines mobile computing technologies with a wireless infrastructure to present visitors to the city of Lancaster (UK) with information tailored to both their personal and environmental contexts.

However, more complex context-aware computing applications need to be built on notions of context that encompass more than only location information [38, 40]. Pascoe [33] defines context as a subset of physical and conceptual states of interest to a particular entity. This definition has sufficient generality to apply to a system that recognizes human actions and behavior. Dey [19] reviews definitions of context and provides a definition of context as any information that characterizes a situation related to the interaction between humans, application and the surrounding environment. Situation refers here to the current state of the environment. Context specifies the elements that must be observed to model a situation. An entity refers to a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves [3, 20].

Recent definitions of context-awareness go even further by defining context as part of a never ending evolution process of interaction in an augmented environment. Coutaz et al. [18] observe hence that:

“Context is not simply the state of a predefined environment with a fixed set of interaction resources. It’s part of a process of interacting with an ever-changing environment composed of reconfigurable, migratory, distributed, and multiscale resources.”

Social and interactional aspects of context must also not be neglected. Individual behavior may not be the correct unit of analysis. Evidence shows that about 40 % of the variance in human behavior may be attributable to non-linguistic social context [34]. Dourish [21] further highlights the interactional nature of context. Contextuality is a relational property that holds between objects and activities. Context is particular to each occasion of activity or action; the scope of contextual features is redefined dynamically. Context is not just a fixed part of the environment, but context arises from human activity.

Some scientists claim, however, that context-awareness in real-world applications is simply impossible. An exhaustive enumeration of the set of existing contextual states of the environment seems difficult [23]. Lueg [30] even states that “context-aware artifacts are far from being able to
recognize situation”. Further, we cannot always know which information determines a specific contextual state. As consequence, determining which appropriate action should be taken by the system autonomously seems impossible [23]. Erickson [22] summarizes that “computers are good at gathering information, humans are good at recognizing context and determining what is appropriate”. Human should hence be kept in the control loop, and context-aware computing should rather do visualization of contextual information than recognition and reasoning on human intentions.

One strategy to respond to these critics is to provide feedback to the user about the reasoning of the system. Cheverst et al. [13] propose the term comprehensibility to suggest that the user “can look through the outer covering (e.g. glass box) to examine the inner workings of the device”. The motivation is that users fear the lack of knowledge of what some computing system is doing, or that something is being done 'behind their backs' [10]. Bellotti and Edwards [8] go further by defining the term intelligibility by:

“Context-aware systems that seek to act upon what they infer about context must be able to represent to their users what they know, how they know it, and what they are doing about it.”

The term scrutability further refers to the ability of a user to interrogate her user model in order to understand the system’s behavior. Kay et al. [25] describe this process as:

“...when the user wants to know why systems are performing as they are or what the user model believes about them, they should be able to scrutinize the model and the associated personalization processes.”

The issue of comprehensibility and scrutability is closely related to the issue of control over the system. Kay et al. [25] see scrutability as a foundation for user control over personalization. The user needs to be able to understand what the system is doing, and also be able to overrule system decisions and processes if necessary. Cheverst et al. [12] mention the obvious motivation: people often want to perform a non-standard action in a given context. Figure 4 relates scrutability and system/user control in a two-dimensional design space. Different augmented environments approaches, presented in section 2, are plotted into this design space.

Barkhuus and Dey [7] report on findings from a study in the context of a mobile scenario, where they investigated the relationship between user control and service automation. In their particular setting, mobile phone users were willing to give up control in exchange for services such as tracking the location of friends or recommendation of nearby restaurants at lunch time. Although their study relied on the participants to imagine
Figure 4: Two-dimensional design space spanning control and scrutability dimensions (adapted from [13])

Figure 5: The relationship among calibration, resolution, and automation capability in defining appropriate trust in automation. Overtrust may lead to misuse and distrust may lead to disuse (taken from [27])
their usage patterns if such a service was available, one of their main findings was that users were willing to give up control if the benefits (i.e. the convenience or added value) of doing so was high.

If users are willing to give up control for a number of system services, this implies that they trust the automation process of the system. Muir [32] shows that the usage of an automated and context-aware system will be optimal if the user’s trust corresponds to the objective trustworthiness of the system. Trustworthiness refers here to the system reliability. This process is called calibration of trust [27]. Figure 5 illustrates the problem. Human trust and trustworthiness of the system should ideally cover the same range of services. When trust exceeds system capabilities, this leads to misuse of the system (overtrust). When human trust is lower than system capabilities, this leads to disuse of the system (distrust). Recent user studies indicate further that perceived system usability has a significant effect on user trust in a system [29].

One might also want to consider the cost and consequences of automated action execution. In particular, if we trust in a system, and the system is said to be reliable, on which basis system decisions for actions executions need to be done? This depends, of course, on the action to be automated and the criticity of the environment. For a hospital environment, Bardram et al. [6] summarize that “the triggering of a context-awareness action depends upon the accuracy of the sensed context information, the degree to which you know what action to take in a certain situation, and the consequence of performing this action.”

## 4 Conclusions

The automating companion agent is still not an everyday life experience. Sensing, recognizing human behavior, and automating services in smart environments are currently research issues. This paper intended to summarize and debate major aspects concerning context-aware service supply in smart environments. Several examples of smart environments have been introduced and the fundamental notions of calm technology, unobtrusiveness, implicit human computer interaction (iHCI), intelligibility, scrutability and trust have been defined and discussed. Current research in proactive service supply faces a number of challenges, some of which we would like to outline here:

### Error Rates

If we aim at providing unobtrusive system behavior, the error rates of recognition and learning algorithms are still too high. Apart from permanently improving these algorithms, we can try to limit the action space of the system. Some actions are less critical and disruptive than others and can be automated with higher error rates in recognition algorithms.

### Generation of Explanations

The decisions and reasoning of a context-
aware system must be transparent for the user in order to admit a sufficient level of trust in such a system. One way to do this is to generate explanations for important system decisions. Such explanations are especially important when errors occur. We need to find an unobtrusive way to provide the user with these explanations within the environment. The generated explanations may also be linked with system control, permitting the user to overrule system decisions.

**Controllability and Human-Computer Interaction** The user must be kept in control. Even though our framework and the learning algorithms are based on an intuitive context model, the user might want to take direct control of the whole system. In this case, an important issue is how to visualize contextual information in an intuitive manner and how to enable the user to control system services easily and without high learning effort. Interface design and type of (explicit) human-computer interaction must carefully be chosen according to the expert level of the user and the physical configuration of the environment.

**Interruptability and Action Cost** Concerning unobtrusive service supply, we need to be aware of the disruptive power of each service. Disruptive power refers to the capacity of interrupting current user task or even current human-human interaction. A service or system action can be disruptive even if the service is pertinent. Therefore, the interruptability of the users must be derived. Based on this information, “cost” and benefit of each service must be estimated and balanced before service execution.
Bibliography


