# Chapter 1
# A Research Agenda for the Socio-Technical Design of Ubiquitous Computing Systems

**Kurt Geihs and Holger Hoffmann**

**Abstract** While technologies make Ubiquitous Computing a reality today, proper engineering methods for creating successful systems are still lacking or inadequate. The result is that mere "trial-and-error" approaches are used when developing novel UC systems. In this chapter we present an overview over the major development challenges, focussing on both social as well as technical aspects of UC system development. These range from the embedding of systems into a social context, sensing and adapting to different usage context and emergent system properties to the need for multidisciplinary cooperation during system development. Furthermore we analyse existing socio-technical development approaches from literature and their shortcomings in relation to the development challenges before introducing the VENUS research approach. We conclude this chapter by giving an outlook for the application of the VENUS research results and chances for further research.

## 1.1 Introduction

With the advances of technology in recent years, especially the development of mobile devices and pervasive applications, the vision of Ubiquitous Computing (UC) as described by Weiser [71] almost 25 years ago is a reality today. In a nutshell, UC is a computing concept where computing is taking place around us while the computing devices are made effectively invisible. Especially the recent

K. Geihs (✉)
Distributed Systems, Kassel University, Pfannkuchstr. 1, 34121 Kassel, Germany
e-mail: geihs@uni-kassel.de

H. Hoffmann
Information Systems, Kassel University, Pfannkuchstr. 1, 34121 Kassel, Germany
e-mail: hhoffman@uni-kassel.de

K. David et al. (eds.), *Socio-technical Design of Ubiquitous Computing Systems*,
DOI 10.1007/978-3-319-05044-7_1,

increase in smartphones and the multitude of pervasive applications for all aspects of personal as well as professional life result in technology that is “interwoven into the fabric of everyday life” [71]. This new level of proximity between users and computers holds the potential to combine the users’ high level cognitive capabilities and creativity with the computers’ interconnectedness and ability for high speed data processing—a symbiosis likely to become transformative in many different application domains [9].

However, the development of UC systems faces numerous challenges. Engineering methods and tools for creating successful systems are still lacking. Bernstein et al. [9] argue that most applications are built using mere “trial-and-error” approaches and research and practice only have a frail understanding of why most of them still fail. They come to the conclusion that in the future, developers of such systems will have to exceed the traditional role as software architects who implement algorithms and also incorporate organisational and societal aspects into system development. This premise brings to mind the socio-technical systems paradigm, which was first mentioned in 1951 by Trist and Bamforth [67] in a work-related context. In the traditional socio-technical perspective, e.g. found in [12, 13, 50], the technical perspective is often condensed to a minimum. Baxter and Sommerville [7, 64] however describe a more holistic approach in which the balance between social/behavioural aspects of system development are matched with technical aspects of system development.

This book presents the results of the VENUS project. VENUS is an research cluster at the Interdisciplinary Research Center for Information System Design (ITeG) at Kassel University, funded by the State of Hesse as part of its LOEWE initiative to foster excellence in research and development.[1] The long-term goal of VENUS is the definition and evaluation of a comprehensive interdisciplinary development methodology for the design of socially aware UC systems. In particular, VENUS focuses on the interactions between the new technology, the individual user and the society. Therefore, four disciplines are represented in VENUS, i.e. computer science, information systems, human-computer interaction, and law, contributing to the research of development methods and tools for ubiquitous applications and taking into account theories, methods and tools described in the context of socio-technical system design.

The following chapters of this book will show how VENUS has responded to these challenges. In this first chapter we give an overview over the research strategy of VENUS and the common challenges that developers face when creating ubiquitous applications, and we discuss the current state of the art of socio-technical system design.

[1] https://hmwk.hessen.de/loewe.

## 1.2 Challenges

The realisation of the potential present in UC raises a number of research questions. We see seven major challenges for the development of socio-technical ubiquitous systems. They are derived from current literature on ubiquitous/pervasive computing and socio-technical system design. Their focus is on challenges for an integrative approach rather than fine grained research challenges in the individual, domain-specific subtasks, e.g., as outlined by [6]. Consequently, the following discussion represents a balanced perspective on the challenges most prevalent in current literature on the development of ubiquitous systems and socio-technical systems research.

### *1.2.1 Embedding in Social Context*

As the socio-technical systems viewpoint and Weiser's definition of UC [71] suggest, the types of systems that are in the focus of our research share the characteristic of being tightly integrated into their users' social context. While this integration into the users' everyday lives does offer a wide range of opportunities, e.g. combining the flexibility of human problem solving with the speed and accuracy of computer algorithms [9], it also requires the inclusion of social aspects during system development, e.g. concerning user attitude or legal compliance.

The most prominent challenge when trying to develop successful socio-technical ubiquitous systems embedded in the users' social context is posed by the users' attitude towards a system [44, 70]. For traditional systems a wide variety of acceptance models is available and evaluated [19, 20, 69]. Current research applies these models to UC, too [18, 72]. These models focus on the users' perception of the system's usefulness and ease of use to determine their intention to use a system [19, 20]. In current acceptance research, Gefen et al. [25, 26] as well as Lee and See [45] argue for the integration of another aspect into these models: They make the argument that the user's trust in a system is a crucial factor for user acceptance of systems that are integrated into the user's personal social context, performing tasks with only implicit interaction. The resulting development methods represent a multidisciplinary approach combining behavioural aspects and engineering principles [37, 63].

Another challenge when developing ubiquitous systems is as old as the paradigm of UC itself [8]: The protection of users' privacy while using ubiquitous applications embedded in their social context [44, 68]. This challenge arises from both the fact that ubiquitous systems are usually context-aware and adapt to their environment, i.e. applications collect data [29], and thus users provide data about themselves when using such applications [60]. As this challenge is based upon fundamental requirements originating from social norms and values—some of which are codified in legislation [21, 33, 34, 60], proper technical solutions for this challenge cannot be reached by software experts alone. It requires contributions from other disciplines.

The challenges originating from the integration of the computer system into the social and organisational context of the user imply that an approach for developing successful socio-technical systems has to take into account the user's perception and trust concerning the system on the one hand. On the other hand it also has to ensure that the usage of user-related data to provide novel functionalities in ubiquitous systems adheres to laws, regulations and social norms.

### *1.2.2 Context Awareness*

One of the key characteristics of ubiquitous socio-technical systems is their context awareness and their dynamic adaptation to context changes in order to deliver an ideal system behaviour towards the user. While prior research already exists concerning techniques for context awareness and application adaptation, many questions remain revolving around detecting, modeling and predicting contexts, as well as basic infrastructures for context aware applications [27, 62]. Furthermore, the user's preferences and profile information are also part of the system context and hence have an influence on the application behaviour. For these integral parts of ubiquitous systems research concerning feasible user interfaces is lacking and consequently one of the challenges for a systematic development.

While research results concerning the integration of environmental context parameters (e.g. noise, light, temperature) and their contextual influence on interface design has already been published, the user's or user group's contextual influence is not well researched and presents another challenge for socio-technical system development. Some parallels can be drawn to research on heterogeneous user models [52] and age-diverse designs [14, 61], the diversity of users in a socio-technical system concerning perceptive, cognitive and motor skills is challenging [55]—especially for those users without much technical expertise. One aspect concerning user preferences for personalised services that is both important and not yet well addressed is the creation of such preference profiles and the matching of profiles to users. Today's practice of creating and configuring them manually is not very practical. This creates a need for algorithms that help generate user profiles from monitoring the system usage in order to understand and predict the users' intentions.

As the usage context is by definition closely related to the user and his personal surroundings, context detection raises the question of "context privacy" and how to protect the users data. While some prior research exists on the legal consequences, e.g. [56, 57, 66], most of it is limited to data protection for RFID-Systems [43] and the legal assessment of such systems [40]. This, however, is one of the driving forces for trust development in systems, structures and organisations where context awareness is part of the socio-technical applications and processes. One challenge lies in understanding the underlying cause-and-effect relationships in trust development as well as methods for developing trust-supporting elements for UC systems.

### *1.2.3 Application Adaptation*

With the advent of smartphones and tablet computers more and more mobile computing devices are available that incorporate a wide range of different sensors and have the processing and memory capacities to run sophisticated applications. Using context information gathered from the built-in sensors, applications running on these mobile devices are able to react to context changes for example in respect to brightness, temperature or the presence of certain objects—identified by RFID tags. To control this behaviour, a set of basic adaptation techniques is known already [47]. The most extensive form of software adaptation is compositional adaptation. It allows changes to the application's architecture at runtime. The result is an application that is able to continuously adapt its behaviour dynamically to its context. An overview over the state of the art and current challenges of self-adaptive systems is presented by [27].

For adaptive applications, the adaptive design of the user interface is a major challenge where interface design does not simply rely on perception skills of the user, but rather supports the cognition by incorporating mental models into system design and development [58]. Taking into account the user's knowledge, experiences and habits leads to a number of questions that have to be answered during the development of adaptive socio-technical systems: How and when should the users be informed about an adaptation? How often may a system adapt to a new context before it becomes disruptive for the users' processes? Can adaptation help to learn [14] or will it have a negative impact on the users' performance?

As adaptive systems change, a big challenge for their development is securing the chain of accountability and liability by tracing the behaviour of self-adaptive systems with regard to legal concerns. So far only general questions concerning accountability and liability have been discussed from a legal perspective. What will be needed, however, are legal approaches for guiding the design and development of socio-technical systems with adaptive components alongside novel legal interpretations of such systems and their role in society.

### *1.2.4 Knowledge Discovery*

One basic requirement in order to be able to provide purposeful socio-technical applications is met by building up knowledge of the surrounding environment with data about users, their actions and the context in which they use the application. In order to generate such situational knowledge, methods originating from data mining, collaborative systems, recommender systems and network analysis are employed. First approaches to put this into practice are found for the analysis of RFID-data to provide location-aware systems [16] or how to use ontologies based on Web-2.0-data [31, 48]. The results obtained from this collective intelligence can be regarded as one possible solution to the bottleneck of knowledge discovery [17]. Other methods are applied in similar ways to e.g. discover communities, i.e. sets of people sharing the same interest and/or objectives, in groups of people [15].

Another challenge for knowledge discovery is, that socially acceptable and economically viable information provision requires and is dependent on reliable findings on the user's information requirements and adequate information channels in the socio-technical system. For trusted information processing, media selection and the initiation and implementation of exchange relations in socio-technical systems can indeed build upon previous work concerning uncertainties in online exchange relationships [54]. However, these relationships often show substantial differences from pervasive and ubiquitous socio-technical systems [59]. Also, legal aspects of this new, socio-technical integration based on technologies in UC and Web-2.0, are currently only represented to a very limited extend with only general previous considerations on legal requirements concerning the technical integration [32, 65].

Due to the extended possibilities for interaction of socio-technical systems in ubiquitous settings as well as the increase in information available to the user through the system, the precise user tasks to be supported by the system need to be taken more into concern during system development. One key challenge in ubiquitous socio-technical applications is the notion of implicit user interaction, which has not received much attention in research so far. However, human-computer interaction in special usage contexts that don't allow the user to focus on his interaction with the system can be found in other areas of research. The most prominent examples are user-oriented systems in the car, e.g., driver assistance systems and driver information systems. These systems also have to take the usage environment—the driver with only limited resources for interacting with the system—into account and hence adapt to the driver's attention and stress state in order to offer a user interface that is appropriate for both the task to accomplish as well as the usage environment [24, 36, 42, 53].

### *1.2.5 Multidisciplinary Development*

One of the major differences between developing socio-technical systems and traditional software engineering is that multiple disciplines are involved at different stages of the software lifecycle [64]. While including stakeholders from a wide range of disciplines is essential for the development of complex socio-technical systems, disciplinary boundaries often hinder the success of joint development teams [7]. However, publications originating from either side of the behavioural/explanatory to engineering/creatory spectrum of research, mostly see this joint effort as a necessary requirement for the development of socio-technical systems [7, 13, 30, 50, 51, 64]. They argue that only an interdisciplinary approach enables developers to follow a problem-oriented approach and address all relevant aspects of the system, e.g. as represented in Sommerville's [64] socio-technical systems stack (cf. Fig. 1.1).

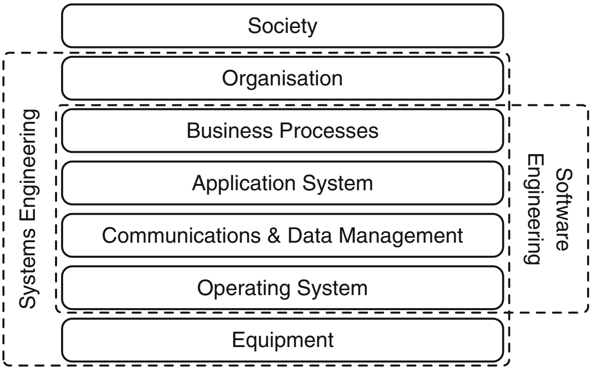


**Fig. 1.1** Socio-technical Systems Development Stack by Sommerville [64]

The "social" challenges of socio-technical system development are strongly related to the top four layers of the stack, ranging from "society" to the "application system". They include, e.g., legal requirements and requirements for fostering user trust and acceptance to address social aspects that have to be realised during system design and implementation. The "technical" challenges on the other hand are mostly related to the bottom four layers of the stack, ranging from the "application system" to the "equipment". They include, e.g., sensor equipment for context sensing, the composition of context information from their values and the appropriate adaption to this context.

Sommerville [64] highlights two major challenges resulting from this: First, experts from different disciplines use different vocabularies, resulting in homonyms where the same word has two different meanings or synonyms where each disciplines uses their own word to refer to the same matter. One example for this is "transparency". In computer science the term indicates that something is unnoticeable, e.g. in the sense that a distributed system appears as a single system to a user [64], while in legal documents and juridical language the same word stands for complete and comprehensive disclosure of all details and facts. Second, representatives make assumptions about the other disciplines' capabilities based on their individual mental models coined by their discipline. The results are often conflicting requirements, e.g. wide ranging data collection and interpretation versus the protection of the users' privacy.

An approach for developing successful socio-technical systems hence has to mediate between different groups of stakeholders and help foster a common understanding [35]. In order to allow proper communication, an approach needs to support the stakeholders in overcoming misunderstandings due to different vocabularies early in the process. Additionally, it has to provide the means to clarify the different disciplines' capabilities and competences in order to determine shared requirements, create a common design and perform joint evaluations of the developed system.

### *1.2.6 Utilizing Human-Computer Networks*

One of the major opportunities socio-technical ubiquitous systems open up is described in the vision of Bernstein et al. [9]—combining the creativity and cognitive capabilities of users with the speed and dependability of algorithms as well as the interconnectedness of computers. At the heart of this vision lies the capability of being able to combine and process data from a heterogeneous set of sources. Possible sources for data are found in a wide variety of sensors [4, 5], but also in user-generated data, e.g. images, texts or audio.

This combination of data originating from users and computers creates two novel research aspects: how can users be included in the system to work with machine generated data, and how can computers process data generated by humans. The question of how the users can be included in handling data that cannot be processed well or not at all by algorithms is covered by numerous recent approaches—presented by, e.g., von Ahn et al. [1–3]. Data generated by users is already used in machine learning approaches, e.g., for context determination or user identification [28]. However, only little attention has been paid to the challenge of how to include user generated data using "soft" factors related to human perception or sensation—like innovativeness, beauty or readability. This is a crucial prerequisite when large amounts of such data need to be processed quickly and reliably, as for example in systems based on collective intelligence.

In order to be able to support the socio-technical development of ubiquitous applications that combine the users' and computers' unique capabilities, a solution approach has to support the handling of machine-generated data by humans where algorithms fail and processing human-generated data by computers where speed, reliability and determinism are important. In the latter case, the consideration of factors considering human perception or sensation poses an additional challenge.

### *1.2.7 Emergent System Properties*

Ubiquitous socio-technical systems are by definition comprised of multiple components that interact with one another. This is the micro-level viewpoint. From the macro-level viewpoint, these local interactions often lead to the emergence of properties that apply to the entire complex system and cannot be attributed to specific components [64]. Emergent properties are not limited to computer systems, but are also known in other disciplines, e.g. biology, physics and chemistry [46]. Sommerville [64] distinguishes between functional and non-functional emergent system properties. Functional emergent system properties represent the intended purpose of the complex system and only emerge after all system components have been integrated [23]. Non-functional emergent properties on the other hand relate to the behaviour of a system in the context it is operated in. For ubiquitous systems, Drossos et al. [23] argue that these emergent properties result from the actual use of the system.

Proper consideration and handling of emergent properties is crucial for the success of socio-technical applications. However, three major challenges arise. First, as context itself may be seen as an emergent property [22], it is impossible to precisely specify the context in which those applications are used or to precisely know how (and which) users interact with the system—both aspects render e.g. the specification of requirements difficult [22, 64]. Second, emergent properties cannot be assessed a priori, but can only be experienced and measured, if at all, once the system is operational [64]. Third, evaluations of socio-technical systems need to be conducted in a real-life context in order to be able to incorporate an assessment of emergent system characteristics into the system evaluation [4, 5, 18].

Thus, a system that features emergent properties can only be developed successfully when addressing those emergent properties during the stages of system development. A development approach needs to support the elicitation of contextual system requirements that only arise when the system is put to use in its social or organisational context. Accordingly, the approach needs to support in-situ evaluations, because simulations and evaluations in laboratory settings can only cover a subset of a system's emergent properties—and thus only imprecise conclusions can be drawn.

## 1.3 Existing Development Approaches

Historically, socio-technical system development originates from the need to analyse and optimise work practices. Mining was the target of the first description of a socio-technical approach [67]. The goal was to optimise processes and tools then in use to enable mine workers to fulfil their tasks more efficiently and effectively. Today the notion of socio-technical system development is used in multiple disciplines—with very different focal points of research, in different societies and cultures, and last but not least has been subject to many changes since its introduction. As a consequence, a plethora of development paradigms and development approaches can be found in the literature [7]. In 2006 Mumford [51] published an extensive review of the research done, highlighting details such as cultural aspects, work organisation etc. in the individual methods.

Her work was picked up by Baxter and Sommerville [7] with a focus on IT support in commercial enterprises. They accentuate the need for a careful embedding of a new IT system into the existing organisational processes and usage environments. The core of their proposal is a pragmatic framework for the engineering of socio-technical systems. They group existing development approaches for socio-technical systems and evaluate how these approaches support the analysis, design and evaluation phase of systems development (cf. Fig. 1.2). In addition to the support for distinct development phases design principles are proposed that provide abstract general guidance when developing socio-technical systems. While they present a concise overview, we classify two of the assessments in their list differently. First, since Beyer and Holtzblatt [10, 11] advise developers

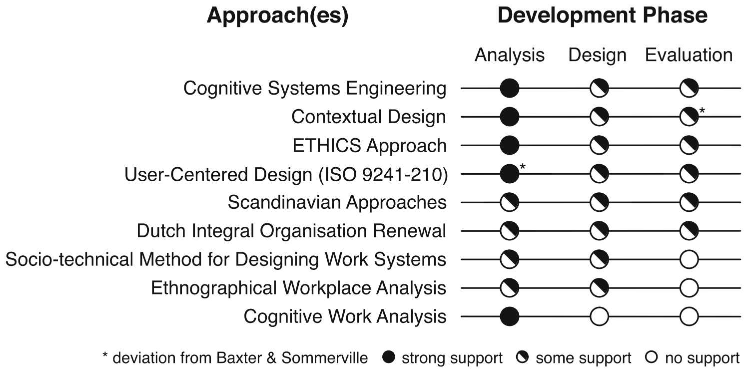


**Fig. 1.2** Scope of socio-technical system development approaches, adapted from [7]

to use prototyping techniques to interact with the future system, the Contextual Design approach does support the evaluation of socio-technical systems, at least to the extent of (visual) design decisions. Second, the User-Centered Design approach [41] is designed to support the analysis of the users' requirements and needs—with a special focus on ergonomic aspects—and hence has strong support for this development phase. These two deviations from Baxter's and Sommerville's assessment are highlighted in the figure below.

Baxter and Sommerville conclude that the analysis phase of socio-technical systems is well supported by many approaches, while the design phase only receives limited support. Most importantly, the evaluation of socio-technical systems is neglected by five of the development approaches they analysed, the remaining six only deliver limited support for system evaluations—including our differing assessment of evaluation support by the Contextual Design approach. These shortcomings are in line with the challenges identified in the previous section, especially the challenge posed by the emergent system properties and the diverse data sources.

For our discussion we thus exclude the approaches that only offer substandard support for the phases in socio-technical systems development. Moreover, the ETHICS Approach as described by Mumford [49] is a collection of practices and can thus be applied to software-engineering methodologies only to a very limited extent. Thus, we concentrate on the following three approaches for further consideration as a methodological foundation of our work: Cognitive Systems Engineering [38, 39], Contextual Design [10, 11] and Human-Centered Design [41].

**Cognitive Systems Engineering** is a multidisciplinary approach, uniting a technical perspective with a cognitive perspective of the system. The users are in the center of interest, e.g. during cognitive task analysis. Their input is used for the two distinct steps of deriving a technical task definition—from a technical viewpoint—and incorporating man-machine guidelines—following a cognitive research approach. The final result of the process is a suggestion for a socio-technical system (here: man-machine-system). However, Cognitive Systems Engineering lacks thorough system development principles or evaluation strategies.

**Contextual Design** focuses mainly on the analysis of user tasks as well as the elicitation and validation of user requirements. The design elements, especially user interfaces, are evaluated by asking for the users' feedback on early prototypes. Although not mentioned explicitly—and not reflected by the visual representation of the approach—Contextual Design can be used as an iterative development approach.

**User Centered Design**, as defined in the ISO 9241-210 norm [41], is the most recent approach. As the name implies, the focus of the approach is on understanding the users, their requirements and the context in which the system is to be used. One of the strengths of the approach is the iterative nature, where the designs are evaluated against the initial requirements and earlier phases can be re-iterated, depending on the outcome of the evaluation. The approach does not specifically include viewpoints from other disciplines or go beyond the design stage in systems development.

## 1.4 The VENUS Approach and Research Agenda

The lack of a systematic development methodology that not only considers the technical requirements and functionality of a product but also takes into account its social embedding is a great challenge for the development of new technologies such as UC. VENUS provides a comprehensive interdisciplinary and integrated methodology for the development of UC applications. We hope that this methodology will boost the development of new UC applications that meet the technical and non-technical user expectations.

The research within the scope of the VENUS project provides improvements for the challenges identified in the previous section while incorporating and extending the current state of the art of socio-technical system design. From the three approaches described above, three main characteristics are evident. First, all three approaches put a strong emphasis on the user focus in the development process. We follow this approach and also put the user in the center of attention for our research. Second, in all approaches the social and/or organisational environment is regarded as the context for the engineering task of system development. Both aspects are reflected in our research, too. Third, both the design and the evaluation phase of socio-technical systems have received only limited support by the approaches reviewed by Baxter and Sommerville [7], some approaches even ignore design and/or evaluation completely. As we consider a proper design and evaluation as crucial for the success of a system, we include support for both development phases in our research.

As an inherently interdisciplinary research project VENUS involves experts from four different disciplines: computer science, information systems, human-computer interaction, and law. VENUS addresses foundations, design methodology, and evaluation of context-aware, self-adaptive UC applications that comply with technical as well as non-technical requirements. The work program of VENUS is structured into three activity groups, i.e. Foundations, Methodology, and Laboratory.

In *Foundations* we build on and extend the state of the art in each of the involved disciplines in view of the specific requirements and characteristics of UC. In *Methodology* we develop a common, interdisciplinary design method that covers all phases of the software lifecycle. The unique distinctive characteristic of the VENUS method is the systematic integration of non-technical concerns, i.e. concerns about the social embedding of the technology, right from the start of the development process. In the activity group *Laboratory* we conduct practical experiments and evaluations with the new design methodology, i.e. we build and thoroughly evaluate demonstrators of innovative context-aware, self-adaptive UC applications.

Each activity group consists of several individual projects. In addition, certain overarching concepts, e.g. interdisciplinary design patterns for UC, are discussed and advanced in interdisciplinary research teams. The organisation of this book reflects the structure of the VENUS work program:

Part II presents results achieved in the activity group Foundations, providing the disciplinary ground work for VENUS.
Part III focusses on the research related to methodology, funnelling the development activities into a common approach.
Part IV covers the design of the demonstrators that were built in the course of the project in order to develop and evaluate the VENUS Development Method; this includes discussions of the lessons learned.
Part V is dedicated to the evaluation phase that plays an important role in supporting the user acceptance of the new technology.

Overall the contributions in this book are meant to provide a comprehensive overview of the manifold facets of the socio-technical design of UC systems. We expect that the different views on the solution space and the proposed concrete solutions provide guidelines and stepping stones for the socially aware development of new UC applications.

## 1.5 A Wider Perspective

Although project VENUS specifically targets UC application scenarios, we claim that the developed methodological framework—aiming at a systematic social embedding of such technology in order to assure user acceptance—can and should be applied to other application domains as well where social awareness plays an important role. Clearly, this claim needs further confirmation through future research projects. The interdisciplinary Research Center for Information System Design (ITeG) at Kassel University, which is the organisational home of project VENUS, will be conducting such research.

During recent years the general awareness for issues related to the social embedding and acceptance of IT technology has increased substantially. VENUS is one of the first projects that have addressed these challenges in a systematic, multidisciplinary, integrated way—with a focus on UC applications and their implications. Other application domains, such as ambient assisted living, service robotics, large-scale collective intelligence systems, disaster recovery support systems etc. have similar as well as additional concerns, requirements, and constraints. Future research projects hosted in the ITeG research center will continue to explore these fields and provide support for socially aware computing solutions. We invite the readers to take this book as a compendium for understanding the requirements as well as a source of inspiration, for the development of acceptable and accepted innovative IT solutions.

## References

1. von Ahn, L., Blum, M., Langford, J.: Telling humans and computers apart automatically. Comm. ACM **47**(2), 57–60 (2004)
2. von Ahn, L., Dabbish, L.: Labeling images with a computer game. In: Conference on Human Factors in Computing Systems. Vienna (2004)
3. von Ahn, L., Maurer, B., McMillen, C., Abraham, D., Blum, M.: reCAPTCHA: Human-based character recognition via web security measures. Science **321**(12), 1465–1468 (2008)
4. Bannach, D., Amft, O., Lukowicz, P.: Rapid prototyping of activity recognition applications. IEEE Pervasive Comput. **7**(2), 22–31 (2008)
5. Bannach, D., Kunze, K., Weppner, J., Lukowicz, P.: Integrated tool chain for recording and handling large, multimodal context recognition data sets. In: UbiComp '10, pp. 357–358. Copenhagen (2010)
6. Bardram, J., Friday, A.: Ubiquitous computing systems. In: Krumm, J. (ed.) Ubiquitous Computing Fundamentals, pp. 37–94. CRC Press, Boca Raton (2010)
7. Baxter, G., Sommerville, I.: Socio-technical systems: from design methods to systems engineering. Interact. Comput. **23**(1), 4–17 (2011)
8. Bellotti, V., Sellen, A.: Design for privacy in ubiquitous computing environments. In: European Conference on Computer-Supported Cooperative Work, pp. 77–92. Kluwer Academic, Milan (1993)
9. Bernstein, A., Klein, M., Malone, T.W.: Programming the global brain. Comm. ACM **55**(5), 41–43 (2012)
10. Beyer, H., Holtzblatt, K.: Contextual Design: Defining Customer-Centered Systems. Academic Press, London (1997)
11. Beyer, H., Holtzblatt, K.: Contextual design. Interactions **6**(1), 32–42 (1999)
12. Bostrom, R.P., Heinen, J.S.: MIS problems and failures: a socio-technical perspective. Part I: the causes. MIS Quarterly **1**(3), 17–32 (1977)
13. Bostrom, R.P., Heinen, J.S.: MIS problems and failures: a socio-technical perspective. Part II: The application of socio-technical theory. MIS Quarterly **1**(4), 11–28 (1977)
14. Bruder, C., Blessing, L., Wandke, H.: Gestaltung und Untersuchung einer adaptiven Benutzungsschnittstelle zur Lernunterstützung älterer Benutzer elektronischer Geräte. Prospektive Gestaltung von Mensch-Technik-Interaktion **22**(25), 71–76 (2007)
15. Cattuto, C., Baldassarri, A., Servedio, V., Loreto, V.: Emergent community structure in social tagging systems. In: European Confeence on Complex Systems. Dresden (2007)
16. Cattuto, C., Van den Broeck, W.: Exposing Contact Patterns (2008). URL http://www.sociopatterns.org/2008/06/exposing-contact-patterns/
17. Christiaens, S.: Metadata mechanisms: from ontology to folksonomy . . . and back. In: On the Move to Meaningful Internet Systems (OTM2006). Montpellier (2006)
18. Connelly, K.: On developing a technology acceptance model for pervasive computing. In: Ubiquitous Systems Evaluation at the Ninth International Conference on Ubiquitous Computing. Innsbruck (2007)

19. Davis, F.D.: Perceived usefulness, perceived ease of use, and user acceptance of information technology. MIS Quarterly **13**(3), 318–340 (1989)
20. Davis, F.D.: User acceptance of information technology: system characteristics, user perceptions and behavioral impacts. Int. J. Man Mach. Stud. **38**(3), 475–487 (1993)
21. DeCew, J.W.: In Pursuit of Privacy: Law, Ethics and the Rise of Technology. Cornell University Press, Ithaca (1997)
22. Dourish, P.: What we talk about when we talk about context. Pers. Ubiquitous Comput. **8**(1), 19–30 (2004)
23. Drossos, N.I., Goumopoulos, C.A., Kameas, A.D.: A conceptual model and the supporting middleware for composing ubiquitous computing applications. J. Ubiquitous Comput. Intell. **1**(2), 1–13 (2006)
24. Fereidunian, A., Lehtonen, M., Lesani, H., Lucas, C., Nordman, M.: Adaptive Autonomy: Smart Cooperative Cybernetic Systems for More Humane Automation Solutions. In: 2007 IEEE International Conference on Systems, Man and Cybernetics. Montréal (2007)
25. Gefen, D., Benbasat, I., Pavlou, P.A.: A research agenda for trust in online environments. J. Manag. Inform.Syst. **24**, 275–286 (2008)
26. Gefen, D., Karahanna, E., Straub, D.W.: Trust and TAM in online shopping: an integrated model. MIS Quarterly **27**(1), 51–90 (2003)
27. Geihs, K.: Selbst-adaptive Software. Informatik Sprektrum **31**(2) (2008)
28. Girardin, F., Blat, J., Calabrese, F., Dal Fiore, F., Ratti, C.: Digital footprinting: uncovering tourists with user-generated content. IEEE Pervasive Comput. **7**(4), 36–43 (2008)
29. Heckmann, D.: Integrating privacy aspects into ubiquitous computing: a basic user interface for personalization. In: Artificial Intelligence in Mobile Systems Workshop at UbiComp. Seattle (2003)
30. Helbing, D., Balietti, S., Bishop, S., Lukowicz, P.: Understanding, creating, and managing complex techno-socio-economic systems: Challenges and perspectives. Eur. Phys. J. **195**(1), 165–186 (2011)
31. Heyman, P., Garcia-Molina, H.: Collaborative creation of communal hierarchical taxonomies in social tagging systems. Tech. rep., Stanford (2006)
32. Hoeren, T., Sieber, U.: Handbuch Mulitmediarecht. Beck, München (2008)
33. Hoffmann, A., Jandt, S., Hoffmann, H., Leimeister, J.M.: Integration rechtlicher Anforderungen an soziotechnische Systeme in frühe Phasen der Systementwicklung. In: 6. Konferenz Mobile und ubiquitäre Informationssysteme MMS 2011. Kaiserslautern (2011)
34. Hoffmann, A., Schulz, T., Hoffmann, H., Jandt, S., Roßnagel, A., Leimeister, J.M.: Towards the use of software requirement patterns for legal requirements. In: 2nd International Requirements Engineering Efficiency Workshop (REEW 2012). Essen (2012)
35. Hoffmann A., Bittner, E.A.C.L.J.M.: The emergence of mutual and shared understanding in the system development process. In: 19th International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ). Essen (2013)
36. Hoffmann, H., Leimeister, J.M.: Evaluating application prototypes in the automobile. IEEE Pervasive Comput. **10**(3), 43–50 (2011)
37. Hoffmann, H., Söllner, M.: Incorporating behavioral trust theory into system development for ubiquitous applications. Pers. Ubiquitous Comput. **18**(1), 117–128 (2014)
38. Hollnagel, E., Woods, D.D.: Cognitive systems egnineering: new wine in new bottles. Int. J. Man Mach. Stud. **18**, 583–600 (1983)
39. Hollnagel, E., Woods, D.D.: Joint Cognitive Systems: Foundations of Cognitive Systems Engineering. Taylor & Francis Group, Boca Raton (2005)
40. Holznagel, B., Bonnekoh, M.: Rechtliche Dimensionen der Radiofrequenz-Identifikation. Tech. rep., Köln (2006)
41. International Organization for Standardization: ISO 9241-210:2010 Ergonomics of human-system interaction: Human-centred design for interactive systems (2010)
42. Karrer, K., Rötting, M.: Müdigkeitserkennung im Fahrzeug—Analyse der Auswirkung verschiedener Rückmeldevarianten auf das Verhalten des Fahrers. Prospektive Gestaltung von Mensch-Technik-Interaktion **22**(25), 191–196 (2007)

43. Langheinrich, M.: RFID und die Zukunft der Privatsphäre. In: Roßnagel, A., Sommerlatte, T., Winand, U. (eds.) Digitale Visionen—Zur Gestaltung allgegenwärtiger Informationstechnologien, pp. 43–68. Springer, Berlin (2008)
44. Langheinrich, M.: Privacy in ubiquitous computing. In: Krumm, J. (ed.) Ubiquitous Computing Fundamentals, pp. 95–159. CRC Press, Boca Raton (2010)
45. Lee, J.D., See, K.A.: Trust in automation: designing for appropriate reliance. Hum. Factors **46**(1), 50–80 (2004)
46. Lehn, J.M.: Towards complex matter: Supramolecular chemistry and self-organization. Proc. Natl. Acad. Sci. USA (PNAS) **99**(8), 4763–4768 (2002)
47. McKinley, P.K., Sadjadi, S.M., Kasten, E.P., Cheng, B.H.C.: Composing adaptive software. IEEE Comput. **37**(7), 56–64 (2004)
48. Mika, P.: Ontologies are us: a unified model of social networks and semantics. In: 4th International Semantic Web Conference. Galway (2005)
49. Mumford, E.: Effective Systems Design and Requirements Analysis: The ETHICS Approach. Palgrave Macmillan, Basingstoke (1995)
50. Mumford, E.: A socio-technical approach to systems design. Requirements Eng. **5**(2), 125–133 (2000)
51. Mumford, E.: The story of socio-technical design: reflections on its successes, failures and potential. Inform. Syst. J. **16**(4), 317–342 (2006)
52. Naumann, A., Hermann, F., Peissner, M., Henke, K.: Interaktion mit Informations- und Kommunikationstechnologie—Eine Klassifikation von Benutzertypen. Mensch Comput. **2008**, 37–45 (2008)
53. Neerincx, M.A., Hoedemaeker, M., de Gier, E.: Adaptive in-car user interfaces based on personalized work load estimation. In: IEA2006 Congress—Meeting diversity in Ergonomics. Elsevier, Maastricht (2006)
54. Pavlou, P.A., Huigang, L., Yajiong, X.: Understanding and mitigating uncertainty in online exchange relationships: a principal-agent perspective. MIS Quarterly **31**(1), 105–136 (2007)
55. Pikaar, R.N., Koningsveld, E.A.P., Settels, P.J.M. (eds.): Proceedings IEA2006 Congress—Meeting diversity in Ergonomics. Elsevier, Oxford (2006)
56. Roßnagel, A.: Datenschutz in einem informatisierten Alltag. Friedrich-Ebert-Stiftung, Bonn (2007)
57. Roßnagel, A., Jandt, S., Müller, J., Gutscher, A., Heesen, J.: Datenschutzfragen mobiler kontextbezogener Systeme. Deutscher Universitätsverlag, Wiesbaden (2006)
58. Rötting, M.: Mensch-Maschine-Systemtechnik. In: Urbas, L., Steffens, C. (eds.) Zustandserkennung und Systemgestaltung - 6. Berliner Werkstatt Mensch-Maschine-Systeme, pp. 27–32. VDI Verlag, Düsseldorf (2005)
59. Roussos, G., Moussouri, T.: Consumer perceptions of privacy, security and trust in ubiquitous commerce. Pers. Ubiquitous Comput. **8**(6), 416–429 (2004)
60. Schlegel, R., Kapadia, A., Lee, A.J.: Eyeing your exposure: quantifying and controlling information sharing for improved privacy. In: Seventh Syposium on Usable Privacy and Security. ACM, Pittsburgh (2011)
61. Schneider, N., Stöcker, S., Grandt, M., Schlick, C.: Altersdifferenzierte adaption der mensch-rechner-schnittstelle. Zeitschrift für Arbeitswissenschaft **60**(3), 171–180 (2006)
62. Sigg, S., Haseloff, S., David, K.: Minimising the context prediction error. In: IEEE VTC Conference. Dublin (2007)
63. Söllner, M., Hoffmann, A., Hoffmann, H., Leimeister, J.M.: How to use behavioral research insights on trust for HCI system design. In: ACM SIGCHI Conference on Human Factors in Computing Systems (CHI 2012). Austin (2012)
64. Sommerville, I.: Software Engineering, 9th edn. Pearson, Boston (2011)
65. Spindler, G.S.F.: Recht der elektronischen Medien. C.H. Beck, München (2008)
66. TAUCIS: Technikfolgenabschätzung ubiquitäres computing und informationelle Selbstbestimmung. Tech. rep., Berlin (2006)
67. Trist, E.L., Bamforth, K.W.: Some social and psychological consequences of the longwall method of coal-getting: an examination of the psychological situation and defences of a work

group in relation to the social structure and technological content of the work system. Hum. Relat. **4**(3), 3–38 (1951)
68. Trivedi, P., Sagar, K.K., Vernon: Emerging trends of ubiquitous computing. Int. J. Adv. Comput. Sci. Appl. **1**(3), 72–74 (2010)
69. Venkatesh, V., Morris, M.G., Davis, G.B., Davis, F.D.: User acceptance of information technology: Toward a unified view. MIS Quarterly **27**(3), 425–478 (2003)
70. Want, R.: An introduction to ubiquitous computing. In: Krumm, J. (ed.) Ubiquitous Computing Fundamentals, pp. 1–35. CRC Press, Boca Raton (2010)
71. Weiser, M.: The computer for the 21st century. Sci. Am. (9), 94–104 (1991)
72. Yoon, C., Kim, S.: Convenience and TAM in a ubiquitous computing environment: The case of wireless LAN. Electron. Commerce Res. Appl. **6**(1), 102–112 (2007)

http://www.springer.com/978-3-319-05043-0

Socio-technical Design of Ubiquitous Computing Systems
David, K.; Geihs, K.; Leimeister, J.M.; Roßnagel, A.; Schmidt, L.; Stumme, G.; Wacker, A. (Eds.)
2014, X, 353 p. 75 illus., 26 illus. in color., Hardcover
ISBN: 978-3-319-05043-0