The Journal of Systems and Software 112 (2016) 26–47

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

Considering context in the design of intelligent systems: Current practices and suggestions for improvement

Christine Bauer[a,*], Anind K. Dey[b]

[a] Vienna University of Economics and Business, Department of Information Systems and Operations, Welthandelsplatz 1, D2, 1020 Vienna, Austria
[b] Carnegie Mellon University, Human Computer Interaction Institute, 5000 Forbes Ave, Pittsburgh, PA, USA

ARTICLE INFO

Article history:
Received 22 December 2014
Revised 27 October 2015
Accepted 28 October 2015
Available online 10 November 2015

Keywords:
System design
Design process
Context selection

ABSTRACT

Ubiquitous sensing allows systems to exploit almost any kind of context, and enables the design of intelligent systems that are aware of their context and adapt their behavior accordingly. As such systems have a number of properties, which distinguish them from traditional systems, their design requires a new approach to requirements engineering and to product development. While existing contributions concentrate on individual aspects in the design process, there is a lack of a holistic perspective on the design of intelligent systems. Considering the entire design process, would allow for the creation of better functioning designs, as has been demonstrated in various fields of system design. Furthermore, little is known about how people design intelligent systems. A deep understanding about design practices is, though, a prerequisite for coming up with systematic improvements. The contribution of this paper is twofold: First, based on interviews, we analyze the design processes undertaken by designers, and present five underlying process archetypes. We focus on how designers identify, select, and consider context across the entire design process. Second, we propose an improved design process for intelligent systems that aims at supporting system designers in their design task in order to serve an organization's, and/or users' needs.

1. Introduction

The design of a system is at the heart of software engineering research. Different approaches to system design have been developed, each with different foci: object-oriented design (Booch, 1982), user-centered design (Norman and Draper, 1986), context-centered design (Beyer and Holtzblatt, 1998), *etc*. Over the last decade, with huge advances in technology, we have observed a trend toward more and more sophisticated systems, termed "intelligent", "context-aware", "adaptive", "situated", *etc*., hereafter referred to as intelligent systems. Research efforts concerning such systems were originally based on Weiser's vision, where systems are aware of the context that they are used in, and intelligently adapt to this context at runtime (Weiser, 1991). While parts of this vision still remain a vision, a large part of it has already been realized and seamlessly integrated in real-life, everyday systems. Currently realized applications and systems are numerous and include, among others, office automation procedures (*e.g.,* Song et al., 2012), location-based mobile applications (*e.g.,* Gerpott and Berg, 2011), mobile customer relationship management (CRM) systems (*e.g.,* Silberer and Schulz, 2012), and personalized and contextualized advertising systems (*e.g.,* Bauer and Spiekermann, 2011; Zhang and Katona, 2012).

However, little is known about how people actually design such systems (Bauer et al., 2014a, 2014b). Early findings indicate that the design of intelligent systems is complex (Floch et al., 2013) and predominantly technology-driven (Friedewald and Raabe, 2011; Oulasvirta, 2004; Rossi et al., 2005); designers often use the so-called i-methodology (Oudshoorn et al., 2004) and typically come up with *ad-hoc* solutions (Rossi et al., 2005; Serral et al., 2009). In short, system design in the field of intelligent systems tends to be unsystematic. However, the use of unsystematic approaches to system design makes maintenance and later adaptation of systems extremely difficult (Serral et al., 2009). Further, we must also question whether such *ad-hoc* solutions can best address a company's or users' needs. Current research does not go far beyond these initial findings that indicate a mismatch between design practices and needs. However, a deep understanding about these design practices is a prerequisite for being able to evolve and make systematic improvements. Knowledge about the design processes currently undertaken by system designers of intelligent systems is missing.

To systematize the design of intelligent systems, the research community has proposed approaches, frameworks and toolkits. Serral et al. (2009) suggest a model-driven development approach

* Corresponding author. Tel.: +43 1 31336 4420; fax: +43 1 31336 90 4420,
E-mail addresses: chris.bauer@wu.ac.at (C. Bauer), anind@cs.cmu.edu (A.K. Dey).
URL: http://www.christinebauer.eu (C. Bauer)

http://dx.doi.org/10.1016/j.jss.2015.10.041
0164-1212/

for intelligent systems. The MUSIC framework (Floch et al., 2013) provides support by describing typical context and adaptation features relevant for the design of intelligent systems. Further work has proposed a systematic technology selection method for the design of intelligent systems (Razali et al., 2012). Dey et al. (2001) proposed a conceptual framework and a toolkit for supporting the rapid prototyping of intelligent systems. However, the scientific contributions of this work to the field actually address different phases of system design rather than holistically: While some approaches support the ideation phase by proposing typical adaptation features (*e.g.,* Floch et al., 2013), other contributions assume that ideation is already complete and support the requirements determination phase instead (*e.g.,* Sitou and Spanfelner, 2007; van der Zanden, 2008), while further work assumes that requirements concerning context features are already defined and instead support technology selection for implementation (*e.g.,* Razali et al., 2012), *etc.* In other words, existing contributions concentrate on individual aspects in the process of system design. Taking a holistic perspective on the design of intelligent systems (*i.e.,* considering the entire design process) would, however, allow the creation of better functioning designs, as has been demonstrated in various fields of system design (*e.g.,* Charnley et al., 2011; Nilsson et al., 2010).

Against this background, we identify two major research gaps: (i) a deeper understanding about the design processes currently undertaken by system designers of intelligent systems is missing; (ii) there is no systematic design process that supports system designers in their design task in order to serve an organization's, industry's, and/or users' needs.

To address this, we conducted a set of 16 interviews with designers of intelligent systems, hereafter referred to as intelligent systems, in research and in industry in the United States of America, Europe, and Asia. We learned in detail about the procedures that designers followed when designing intelligent systems in their real-world projects. Through our analysis, we identified five different process archetypes, which we will present, compare, and evaluate in this paper.

Further, we propose a novel process for the design of intelligent systems that aims to serve an envisaged purpose, addressing business and user needs. In an evaluation of the suggested process via a further set of 11 interviews with designers of intelligent systems, this process was reported to be valuable for system design; experts expressed that they would use the novel process particularly for achieving incremental innovation. They also said that the different process aspects could be used as a kind of "checklist" to guide the design process. Less experienced designers highly valued the new process because it would help them avoid making the same mistakes as the experienced designers did in their initial projects. Overall, our interviews particularly highlighted that the new process supported the different phases of system design at a coarse (*e.g.,* support in deciding the order of what should be done in the design process) and fine-grained level (*e.g.,* for specifying which context elements should be considered by a system), providing a systematic, yet flexible process that can be applied to a wide variety of design situations.

This paper proceeds as follows: In Section 2 we provide a brief overview of related work on the design of intelligent systems. In Section 3 we describe the qualitative research approach that we employed for eliciting and analyzing the designers' processes. Section 4 presents our findings on the designers' process archetypes and our analysis of these archetypes. In Section 5 we propose an ideal process for the design of intelligent systems, discuss its relation to designers' currently applied processes as presented in the previous section, and demonstrate its utility based on the designers' feedback. Section 6 presents the application of the 'ideal process' in a case study and Section 7 demonstrates how this process may be applied in practice. Section 8 concludes with a summary of the contributions of this work.

2. Background and related work

2.1. Context

The term "context-awareness" in ubiquitous computing was introduced by Schilit et al. (1994). Since then, the context-aware computing community has grown rapidly. With huge advances in technology, we can observe a trend toward increasingly sophisticated systems, termed "intelligent", "context-aware", "adaptive", "situated", *etc.* The joint pivotal element in these systems is known as 'context'.

Although research in context-awareness goes back to the 1990s, there is still a lack of an explicit, single, unified definition of context (Chen and Atwood, 2007). Early attempts to define 'context' in the computer science domain resemble enumerations of examples (*e.g.,* Dey, 1998; Schilit and Theimer, 1994) or synonyms for context (*e.g.,* Brown et al., 1997). More recent concepts are either highly specific to a certain application domain (*e.g.,* Bauer and Spiekermann, 2011) or very generic (*e.g.,* Han et al., 2008). A widely accepted definition was provided by Dey and Abowd (2000), stating that "context is any information that can be used to characterize the situation of an entity". However, characterizing a situation is no easy task: "How are dimensions of context identified, quantified, and interrelated for each situational purpose?" (Bradley and Dunlop, 2005). Given the complexity, variety, and multi-dimensionality of context, system designers have difficulties in identifying and specifying which context is relevant for a system (Choi, 2008). Nevertheless, system designers need to anticipate the relevant combinations and characteristics of context before an intelligent system is implemented in the real world, and decide which context to include in their designs. As context is a crucial element that defines the functionality of an intelligent system and shapes the system's behavior, context selection is a significant task in the design of intelligent systems.

2.2. Processes for system design

In recent years, much effort has been devoted to orient system design toward the situation and needs of a system's potential users (*e.g.,* Iivari and Iivari, 2011; Kohler et al., 2011). In the 1980s, there was increased awareness that the somewhat abstract data that is typically gathered from surveys and focus groups cannot provide the detailed information that designers need about how people carry out their tasks, as Holzblatt (2009) retrospectively points out in her work. This awareness gave rise to the approach known as 'user-centered design' (UCD) (Norman and Draper, 1986), which is intended to support system designers in understanding the context of use from the perspective of future users of a system. Based on UCD, the concept of contextual design (Beyer and Holtzblatt, 1998) provides a set of methods that "tells people what to do at each point so that they can move smoothly through the design process" (Holzblatt, 2009). As ubiquitous, context-aware, and mobile computing requires systems to seamlessly integrate into users' environments (Chen and Atwood, 2007), particular attention has to be paid to the context, in which a system is placed. The framework called 'context-centered design' brings attention to the context in which people will use a system (Chen and Atwood, 2007). Still, this framework does not provide support for designing intelligent systems that automatically adapt their behavior to the situation at run-time.

Irrespective of the design approach selected, the design of a system is always embedded in a more or less structured system design process. Explicit models addressing system evolution date back to the 1950s. The reason these models were introduced was to provide a scheme that allows for managing the design of systems. This scheme

constitutes the basis for planning, organizing, staffing, coordinating, budgeting, and directing software development activities (Scacchi, 2001). The main goal in creating and improving such models has been to find repeatable, predictable processes that improve productivity and quality (Dowson, 1993). While some approaches attempt to systematize or formalize the rather 'unruly' task of programming, others apply project management techniques to the design process to deliver systems on time and within budget. The international standard ISO/IEC 12207:2008 (2008) was introduced to define the tasks required for developing and maintaining systems.

The system development life cycle (SDLC) is a widely accepted information systems design theory (Walls et al., 1992) that led to the creation of several process models for system design. The process models usually comprise some variants of the following phases: requirements determination, design, construction, implementation, and operation (Boehm, 1984; Mantei and Teorey, 1989), while the borders between phases are blurred (Walls et al., 1992). Note that many process models summarize activities that are performed in an early phase of the process known as the 'design phase', while the entire process is referred to as the 'design process'; the latter of which is the focus of our work. Most process models in use today have evolved from three primary approaches: *Ad-hoc* development, waterfall model, and the iterative process. The V-model, for instance, builds on the waterfall model. Instead of moving down the design process in a linear way, the process steps are bent upward after the coding phase to form a 'V shape' in order to demonstrate the relationships between each process step of development/definition and its associated process step of testing. Agile process variations, in contrast, use iterative development as a basis but advocate a lighter and more user-centric viewpoint than the traditional approaches, using continuous feedback and successive refinement of a system.

Each of these process models has its advantages and disadvantages and may more or less suit a particular project. What the process models have in common, though, is that they strive to improve productivity and quality in the design of information systems.

2.3. Approaches to support system design of intelligent systems

As we have outlined in the previous section, there is a variety of well-accepted approaches to system design. However, intelligent systems have a number of properties, which distinguish them from traditional information systems (Kolos-Mazuryk et al., 2005). For instance, flexibility and variability are indispensable both at design time as well as runtime (Fortier et al., 2010) and context has to be dealt with in addition to other design issues such as business requirements elicitation, system coupling, coding, system architecture design, or acceptance testing with potential users (Bauer, 2014; Castelli et al., 2009; Kolos-Mazuryk et al., 2005; Sitou and Spanfelner, 2007). Traditional testing approaches cannot be applied to the application logic of intelligent systems, which contain context-aware parts typically in the middleware (Lu et al., 2006). So-called 'data cleaners' may mistakenly clean up important context information instead of random noise in data (Lu et al., 2008). In addition, a difficult challenge in the design of intelligent system concerns quality issues throughout the design process: with regard to the modeling of an intelligent system (*e.g.,* Sama et al., 2010b; Xu et al., 2012), testing the system (*e.g.,* Lu et al., 2006, 2008; Wang et al., 2007), verification of the system (*e.g.,* Liu et al., 2013; Sama et al., 2010a), and the quality of system environment (*e.g.,* Xu and Cheung, 2005; Xu et al., 2010). As a result, the design of intelligent systems requires a completely new approach to requirements engineering (Kolos-Mazuryk et al., 2005; Sitou and Spanfelner, 2007) and to product development as a whole (Kolos-Mazuryk et al., 2005).

Interestingly, although the concept of intelligent systems has existed since the 1990s, few works have investigated requirements engineering approaches for such systems (Cheng et al., 2009). Omasreiter and Metzker (2004) proposed a context-driven use case creation process to describe the behavior of a system's functions that use context. This process is intended to bridge the gap between critical context situations and user goals by foregrounding these use cases. Kolos-Mazuryk et al. (2005) presented an approach to requirements engineering for pervasive systems that helps to obtain requirements for intelligent systems under development. Choi (2007) introduced a process for requirements analysis, including the use of context-aware use case diagrams, context-switching diagrams, and dynamic service models for intelligent systems. Their process splits business logic and context logic in the requirements analysis phase to reduce the complexity of such systems. Sitou and Spanfelner (2007) presented a model-based requirements engineering approach to systematically specify the basic and adaptive system behavior that should be supported based on user and business needs. In addition, some research approaches use goal models, which resemble a hierarchy of goals that relate the high-level goals for an envisaged system to low-level system requirements. In an early requirements engineering phase such a goal model may be used to specify the autonomic behavior (Lapouchnian et al., 2006) and requirements (Goldsby et al., 2008) of intelligent systems.

Also, with respect to the design of systems, little research has focused particularly on the design of intelligent systems. One approach to support the design of such systems is to use design patterns (Chung et al., 2004; Landay and Borriello, 2003; Rossi et al., 2005) or reusable context and adaptation features (Floch et al., 2013). Still, these pattern-based approaches are based on a literature review and do not provide insights into designers' observed practices. Dow et al. (2006), in contrast, investigated design practices for intelligent systems by interviewing designers. This work, though, focuses on the development of tools to support designers and does not address the issue of how to shape design processes.

Further, for the development of intelligent systems, Serral et al. (2009) suggested a model-driven development approach. Dey et al. (2001) proposed a toolkit to support rapid prototyping of intelligent systems. Razali et al. (2012) provided support for systematically selecting technology for such systems.

As can be seen from Table 1, the presented approaches target challenges in different phases of the system design process.

2.4. Approaches to context identification and modeling for intelligent systems

Despite the difficulty in deciding what context to include in a system design, few works exist that propose how this task should be carried out (Table 2). Kofod-Petersen and Cassens (2006) build their work on an extended version of activity theory (called CHAT), which is based on the assumption of division of labor. Their work aims at supporting humans in modeling context, whereby their approach takes a knowledge-level perspective on the modeling task and analyses the entire sociotechnical system. Also Huang and Gartner (2008) base their approach on activity theory. Their approach determines how to identify context that is relevant for a set of specified activities. An important aspect in their work is that they speak of relevance; whereby they assume that relevance is shaped by the requirements specification of the respective system to be designed. Castelli et al. (2009) take a different approach. They suggest a specific format for organizing and specifying context for system design for context elicitation and specification. They propose a taxonomy of context and a scheme, which is both used as a kind of ad hoc questionnaire when interviewing a client about the system to be built. The scheme is then also used for representing the context specification. Sandkuhl and Borchardt (2014), in contrast, propose an approach for how to perform context modeling, whereby they consider both, the system

Table 1
Overview of approaches to support system design of intelligent systems.

Phase in the design process	Approach	Article	Remarks
Requirements engineering	Use cases	Omasreiter and Metzker (2004)	Context-driven use case creation process
		Choi (2007)	Context-aware use case diagrams, context-switching diagrams, and dynamic service models
	Requirements specification	Kolos-Mazuryk et al. (2005)	
		Lapouchnian et al. (2006)	Autonomic behavior specification
		Sitou and Spanfelner (2007)	
		Goldsby et al. (2008)	Goal-based modeling for requirements specification
Design and modeling	Design patterns	Landay and Borriello (2003)	
		Chung et al. (2004)	
		Rossi et al. (2005)	
		Floch et al. (2013)	Reusable context and adaptation features
	System modeling	Sama et al. (2010b)	
		Xu et al. (2012)	
Development and implementation	Prototyping	Dey et al. (2001)	Rapid prototyping toolkit
	Development	Serral et al. (2009)	Model-driven development
	Technology selection	Razali et al. (2012)	
Testing and verification	Testing	Lu et al. (2006)	Data flow approach for testing context-aware middleware-centric programs
		Lu et al. (2008)	Testing in the presence of context inconsistency
		Wang et al. (2007)	Testing the system
		Xu and Cheung (2005)	Quality of system environment
		Xu et al. (2010)	Quality of system environment
	Verification	Sama et al. (2010a)	Verification of the system
		Liu et al. (2013)	Verification of the system

Table 2
Overview of approaches to context identification and modeling.

Article	Roots of the approach	Use of the approach	Remarks
Kofod-Petersen and Cassens (2006)	Based on an extended version of activity theory (called CHAT)	Supports humans in modeling context from a knowledge-level perspective	Assumptions: most important context are activity and performer of the activity; division of labor
Huang and Gartner (2008)	Based on activity theory	Supports the identification of relevant context for each activity in a sociotechnical system	Assumption: relevance is shaped by requirements specification
Castelli et al. (2009)		Provides a specific scheme/form for context elicitation and specification	Proposes a context taxonomy; uses ad hoc questionnaires in interviews with clients
Sandkuhl and Borchardt (2014)		Provides an approach for context modeling	For system development and system application
Bauer (2014)	Uses a variety of context models from literature	Suggests an approach for context conceptualization for a system	Procedural approach
Jaffal et al. (2014)		Provides a formal methodology for analyzing the impact of context information on the user's action	Formal concept analysis from data

development as well as the system application. Bauer (2014) suggests a framework for conceptualizing context for intelligent systems taking a process-oriented approach. Jaffal et al. (2014) suggest a methodology for analyzing the impact of context information on a user's action. Their intent is to set up a knowledge base of user activities and make a formal concept analysis from data.

In summary, related work provides contributions to different phases of system design. Still, anticipating relevant context – ahead of the actual situation being reality – is a key challenge. Thereby, the challenge for the designers of intelligent systems is not whether to conceptualize (and, thus, to simplify, reduce and objectify) context but how to do this (Chalmers, 2004). However, although there are approaches for systematization (see Sections 2.3 and 2.4), and for context identification and modeling in particular (see Table 2), there is a lack of research that takes a holistic perspective on the design of intelligent systems. Against this background, our approach takes the entire design process into account and considers the various phases in the design process that shape the system idea and implementation. Thereby we take a procedural perspective that guides designers through the process phases, rather than proposing schemes for modeling or providing methodological advice (*e.g.,* preferences for questionnaires, interviews, or document analysis). As a result the approaches are not mutually exclusive as the above described approaches for context modeling and context elicitation may be integrated into our novel approach.

3. Methodology

3.1. Interviews for data collection on the current practice of designing intelligent systems

To understand the design processes that system designers of intelligent systems currently follow, we adopted a qualitative research approach. We conducted 16 interviews with experienced system designers in the field of intelligent, context-aware, adaptive, situated system or application design. We chose this wide framing in order to capture projects that consider some kind of 'context' in run-time, including those projects for which the designers did not strive to meet the technical definition of "context-aware" (as defined by Schilit et al., 1994). This is in line with Abowd (2012), who postulates that "there is a difference between the intellectual area of ubiquitous computing and the community of people who identify themselves as ubicomp researchers". As concerns the requirements for the identification and selection of relevant context in system design, we consider this variety of systems (*i.e.,* intelligent, context-aware, adaptive, situated) as being sufficiently similar. In our view, context-aware means that a system is 'aware' of its context and, for example, may represent or visualize it. Context-adaptive systems are also context-aware and, additionally adapt their behavior to the context that they are aware of. The term 'situated' is frequently interchangeably used for context-aware and context-adaptive. Such systems may be called intelligent if

Table 3
Overview of designers and reported projects.

Designer id	Project id	Project context	Project location	Kind of system designed	Project goal	System's target users
D1	P1	University	Asia	Context-adaptive application icons rearrangement on smartphone	Software solution	Public at large
D2	P2	University	USA	Bridge inspection system	Software solution	Bridge inspectors
D3	P3	University	USA	Sharing activity recognition on smartphones	Software solution	Public at large
D4	P4	University	USA	Contextually protected access system	Software solution	Public at large
D5	P5a	University	USA	Routine enhanced user interface on smartphones	Software solution	Public at large
	P5b	Industry	USA	Occupancy-based indoor location prediction	Validation of new concepts	Facility managers
D6	P6a	University	USA	Autonomous robot balancing on a ball	Validation of new concepts	First step toward autonomous robots that can assist people
	P6b	University	USA	Context-adaptive smartphone ringtone profiling (mute versus ringtone)	Software solution	Public at large
D7	P7	Industry	Europe	Personalized mobile recommendation system for smartphone	Software solution	Public at large
D8	P8 = P9	University	USA	Movement assessments for the rehabilitation of stroke patients	Software solution	Stroke patients
D9						
D10	P10a	Industry	USA	Context-adaptive application icons rearrangement on smartphone	Software solution	Public at large
	P10b	Industry	USA	Adaptive (personalized) smoking-cessation application on smartphone	Software solution	People that want to quit smoking
D11	P11	University	Europe	Controlling devices and appliances in a home	Validation of new concepts	Public at large
D12	P12	University	Europe	System analyzing patient's physiotherapeutic movements for their accuracy	Software solution	Physiotherapists
D13	P13	University	USA	Intelligent water use monitoring system	Software solution	Elderly people in assisted living homes
D14	P14	University	USA	Navigation-system for blind people using smartphone	Software solution	Visually impaired people
D15	P15	University	USA	Emotion-based music playlist generator	Software solution	Public at large
D16	P16	Industry	Asia	In-car system that prevents dangerous situations	Software solution	Public at large

the representation and/or adaptation are considered to be smart. The borders between the variants are definitely easily blurred and debatable, for which we take a broad perspective.

We recruited designers via (i) postings in professional social networks (LinkedIn, XING), (ii) by searching for publications reporting on actual system design, which goes beyond prototyping, and contacting the authors, and (iii) through the personal network of the research team—globally and both in research as well as in industry. In total we conducted 16 interviews with system designers from the United States, Europe, and Asia, from university and industry. The interviews lasted between 30 and 45 min each. In addition to audiotaping the interviews, the interviewer took notes.

Following the approach of prior work on design practices (Dow et al., 2006), we asked designers to report on a particular project that they worked on in the recent past. With this approach, we tried to counteract the tendencies that designers may have to report on an idealized process that they would have liked to follow, because we were interested in their real practices. While most designers reported on one project, some walked us through a second project as well. In total, we collected data on the design practices used in 18 projects. Project fields were varied, including an in-car system that prevents dangerous situations, an emotion-based music playlist generator, a navigation-system for blind people, a system analyzing patient's physiotherapeutic movements for their accuracy, a smoking-cessation support system, an autonomous robot system balancing on a ball, and a contextually protected access system. Table 3 provides an overview of the reported projects.

Each of the interviews started with a short introductory explanation about our research endeavor where we assured participants that our intent was not to judge their processes; instead we wanted to know what designers actually did in practice. Then the interview followed a semi-structured outline.

By the time we had conducted 16 interviews, we experienced saturation (Bowen, 2008), as the last interviews did not reveal additional insights but rather repeated what we had already learned from earlier interviews.

Table 3 provides an overview of the designers and the reported projects. D5, D6, and D10 reported on two projects each. D8 and D9 – although interviewed separately – reported on the same project. D2 represents two designers who were jointly interviewed on their joint project. Most projects (13) reported were university projects, while 5 came from industry projects. The projects P5b, P6a, and P11 involved the validation of new concepts; all other projects had the goal of developing a software solution.

3.2. Interview analysis

After each interview, the researchers reflected on the essential meanings of the designers' reported activities and developed an understanding of the implications. The identified activities were then structured to represent the respective designer's process. Thereby, information was visualized using graphical models (Miles and Huberman, 1994). Displaying data in graphs is simultaneously an activity of data reduction in the data analysis process (Miles and Huberman, 1994). In this phase of analysis, the researchers did not adhere to the requirements of any modeling standard, in order to remain flexible and to allow for a good overview of the complex interview results.

In the next step, we drew from the approach of 'pattern matching', as described by Donald T. Campbell (cf. Yin, 1989), and applied it to the different designer's processes. This technique allowed us to

identify similar and identical patterns in the reported processes. For this purpose, we revisited our interview notes to ensure that the original (semantic) context was not misinterpreted and important cues were not overlooked. For instance, revisiting the original notes made clear that the process component termed 'brainstorming' (to be introduced in Section 4.1) had to be split into three component categories, since one kind of brainstorming served to come up with an idea while other brainstorming sessions were targeted toward the fine-tuning of ideas or evaluating constraints.

Having identified overarching patterns, new graphs were drawn to reflect the identified process components. In a final step, the graphs were reduced to present the essential structures. Overall, the reported processes could be aggregated to five different process archetypes.

After analysis and comparison of the processes, we went back to the interviewees. We showed each interviewee the resulting graphical representation of his or her process(es) to discuss and accordingly validate it (them) (Miles and Huberman, 1994).

3.3. The design of an 'ideal' system design process

Designing a prescriptive system design process (*i.e.,* an artifact) is an inherently iterative process that follows continuous cycles of generating and evaluating design alternatives. Therefore, we built on existing literature in the research domain and on insights gained from the interviews (Section 3.1). We gradually improved a draft version of the artifact and generated enhanced design alternatives. The research team reflected as a group on the draft versions to collectively converge them into a coherent process.

3.4. Expert interviews for utility evaluation

In the interviews that we used for validating the graphical representations of the archetype processes (Section 3.2), we also evaluated the utility of our novel process artifact (described in Section 5). We showed each interviewee our novel process artifact (Fig. 7) and walked them through the process to ensure that they had a reasonable understanding of the artifact. Then we asked them (1) whether the artifact was useful for the design of intelligent systems, (2) whether they could see potential for embedding the artifact in their existing process, and (3) whether they could imagine replacing their 'old' process for the suggested new one.

After each interview, the research team reflected upon the insights from the evaluation interviews, in analytical group sessions.

4. Findings on current design process practices

Among the 18 reported projects, we were able to identify process components that were present in each of them (*e.g.,* technology selection, implementation), some that were frequently used (*e.g.,* brainstorming of an idea, fine-tuning), and some sporadically used components (*e.g.,* data analysis for patterns). Clustering allowed us to generalize the individual processes to five archetype processes. Section 4.1 will outline the underlying constructs and the notations that are then (Section 4.2) used to describe the identified archetype processes. Section 4.3 discusses the findings on the design processes; and Section 4.4 focuses on the challenges that the designers encountered while performing their design processes.

4.1. Constructs

First we describe the underlying constructs that we use as labels in our notation of the archetypes as presented in Section 4.2.

- Goal: A goal is an objective that determines in which direction a project should go and sets the field to which the project should contribute. An example for a goal is: The system should help blind people in orienting in space. While a goal may be explicitly defined for a project, it can also be implicitly set in the designers' minds without them necessarily being aware of it.
- Idea: An idea is a concrete scenario or a set of scenarios outlining system behavior (a clear-cut concept). Within the frame of the above example for a goal, an idea could be the following: The envisaged system is an audio guide, which a blind person uses when moving in a room, that warns the user before running into a wall using an audio signal about 30 cm in front of the wall.
- Technology selection: In the scope of this work, technology refers to any hardware and software (including sensors) that is used to implement an envisaged idea. Examples are a smartphone with its embedded sensors or a depth camera and an accelerometer mounted in a car.
- State-of-the-art analysis: State-of-the-art analysis refers to knowing and understanding existing work in a certain field. It includes a review of (recent) literature in the field (including applications as well as technologies and algorithms), prototype specification analysis, models (for instance, context models or routines in human behavior), and other kinds of concepts, *etc.*
- Constraints evaluation: Within the scope of this paper, constraints evaluation refers to the consideration of various limitations that may either shape the idea or contributes to narrowing down the technological possibilities for the envisaged system's purpose. The evaluation is typically performed in an unstructured and non-guided way. Examples of constraints include time, budget, technical feasibility, availabilities, laws, *etc.*
- Brainstorming for ideas: Brainstorming is an unstructured and unguided technique (performed in a group) that involves the spontaneous contribution of ideas from all members of a group to encourage creativity. It includes the mulling over of ideas in an attempt to devise or find a solution to a problem. In the scope of this work, the main purpose of 'brainstorming for an idea' is to come up with an idea (or set of ideas) for a system.
- Fine-tuning: Fine-tuning takes the identified idea and provides more details and specifics for the idea. It may consider learnings (for instance, about constraints or user needs) from prior stages in the process. As reported by the interviewees, this consideration is performed in an unstructured and unguided way through brainstorming sessions.
- Data analysis for patterns: Data analysis for patterns aims at discovering patterns in (large) data sets. In the scope of this paper, this term refers to the application of any kind of data analysis methods or techniques that allow for pattern discovery. Examples include machine learning, data mining, network analysis, pattern recognition, and many others. The appropriate choice of data-analysis method or technique depends both on the nature of data and on the goals of the analysis. For details see, for instance, Bishop (2006).

The notation we use for presenting the archetypes in the next subsection conveys several dimensions of information: (i) sequence of phases, (ii) deviations in the sequence across reported individual processes of the respective archetype nature, (iii) option richness, and (iv) degree of ambiguity.

(i) The sequence in which the process phases are carried out is depicted by the vertical sequence of the items in the graphics. (ii) As archetypes represent several individual processes as reported in the interviews, slight deviations in the process sequence are possible, which do not change the nature of the respective archetype; such deviations are indicated by dotted lines, which represent alternative sequences of items in the archetypes. Labels for items presented in dotted lines are also connected to items with a dotted line.

(iii) The option richness in terms of whether the number of options considered is rather stable, increasing, or decreasing, is

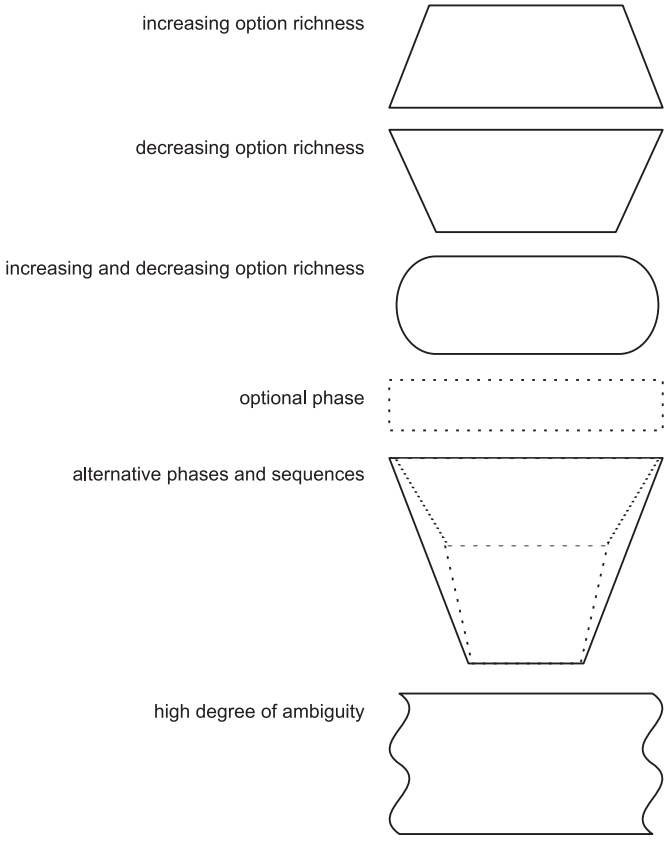

Fig. 1. Elements of notation.

indicated by the breadth of the item. For instance, when performing state-of-the-art analysis, the number of options potentially considered is typically increasing. When selecting a technology or considering constraints, the options decrease. In a brainstorming process new options arise but at the same time options are discussed and some are rejected, for which the graphical item for brainstorming is broadening and narrowing on its vertical sidelines. (iv) As some phases in the design processes appeared to be ambiguous in execution, we wanted to depict this also in the archetype representations, for which we used wavy sidelines as indication for the ambiguity of the phase. For an overview of the notation elements see Fig. 1.

4.2. Archetype processes

From the interviews, we identify five archetypes of design processes. The categorization can be seen in Table 4. Note that the design processes are iterative in nature, allowing one to return to a previous phase, from any phase of the processes. The visualizations of the archetype processes (Figs. 2–6), though, do not represent iterations, which are typically of a sequential nature, in order to keep the complexity of the graphs to a manageable level.

Archetype I, which we term 'informed idea-generating process', was followed in five projects. It refers to a process that is characterized by an informed brainstorming to come up with an idea, and for which a suitable technology is carefully selected to implement the idea (Fig. 2). Designers enter the brainstorming phase only after reviewing prior work (state-of-the-art analysis), which can be found in academic literature as well as in prototype descriptions. While some designers search for information and literature within a specific area, as they have a defined goal (*e.g.,* supportive technology to be used in cars), others have a wide scope and take inspiration by the information they come across. For instance, D16 reported that the design team reviewed papers in the field of in-car applications and

Table 4
Overview of the identified archetypes and the related projects.

Archetype id	Archetype name	Project id	Project context
I	Informed idea-generating process	P1	University
		P2	University
		P4	University
		P15	University
		P16	Industry
II	Idea-driven process	P5b	Industry
		P6a	University
		P10b	Industry
		P11	University
III	Idea-based technology selection process	P8 = P9	University
		P12	University
		P14	University
IV	Technology-driven idea-generating process	P3	University
		P7	Industry
		P10a	Industry
		P13	University
V	Explorative data-driven process	P5a	University
		P6b	University

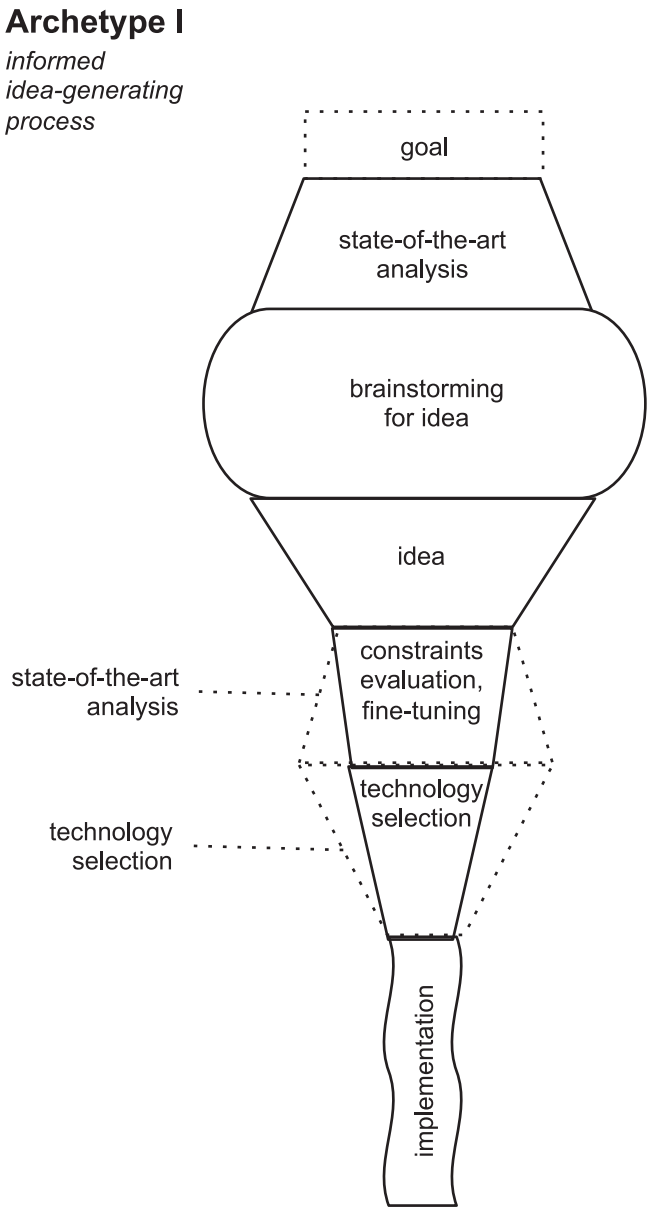

Fig. 2. Archetype I—informed idea-generating process.

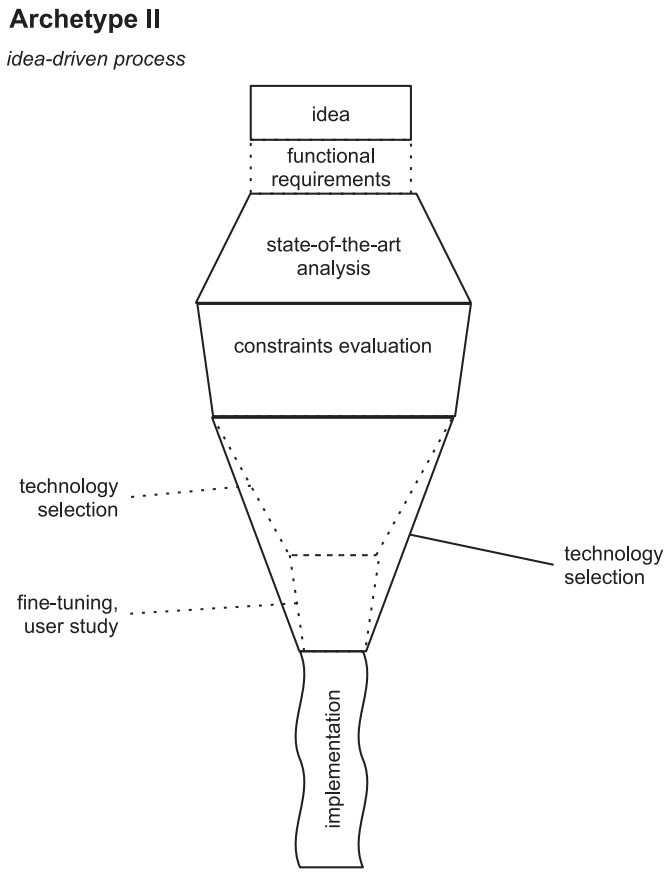

Fig. 3. Archetype II—idea-driven process.

collected information on the diverse issues that could distract drivers while driving (*e.g.,* drowsiness) and, thus, could end in risky situations in traffic. D2 reported on a project where they already had a well-structured storyboard about the current situation for bridge inspections, which is a specific application area, which they wanted to streamline by using information and communication technology to assist bridge inspectors in their tasks. D15, in contrast, only had the rough idea that they wanted to combine the topics of mood and music. The brainstorming task is typically unstructured and unguided, allowing for a free flow of ideas; however, this *unstructuredness* risks that fruitful opportunities for providing intelligence may be overlooked. Between the definition of an idea and the final technology selection, designers use different combinations of means to carefully select the technology.

An important task is the evaluation of constraints in realizing the idea; this task is typically addressed in a brainstorming manner, again unstructured and unguided. D4 reported that he considers technological constraints partly already in the brainstorming for an idea phase because as an experienced designer it seems rather impossible to block thinking processes on knowledge that you have already gained. Furthermore, user studies are used to evaluate the idea and to identify user needs; both may lead to fine-tuning the original idea, which is typically performed in an unstructured brainstorming session. D4 explains that user studies help to shape the project in more detail; for instance, they learned from their user study that the differentiation between 'being at home' and 'being at work' has a tremendous impact on how secure a user feels with a certain adaptive smartphone protection mechanism and how useful or annoying the same mechanism may feel. Defining functional requirements that support the idea is also a means for fine-tuning the original idea as vague ideas get systematically manifested in defined functionalities. Further, some designers go back to the state-of-the-art in the field, considering similar ideas and existing approaches for the technical implementation of the solution. After selecting appropriate technology, the designers follow this archetype process through to the implementation of a prototype. The implementation phase itself is, as the designers report, rarely as straightforward as it may seem, because they frequently have to reconsider decisions due to erroneous earlier decisions or overlooked limitations. For instance, D2 pointed out that physical constraints in technical solutions are often neglected and that sensors being considered may not be used in the final implementation due to their weight or volume. As a result, it is frequently necessary to reconsider prior design decisions as late as in the implementation phase.

Archetype II, which we term the 'idea-driven process', could be identified in four projects. This process archetype is characterized by an idea-driven, informed, and evaluated technology selection (Fig. 3). The starting point for designers following this process is a concrete idea. Some designers first define the functional requirements for the envisaged idea before proceeding with the next step. With or without the functional requirements, the next step is a thorough review of existing work in the field, which includes similar ideas and (conceptual and technical) approaches to the problem. D10 reports that for an adaptive (personalized) smoking-cessation application, he had to delve into medical research (*i.e.*, domain-specific research) as well as to study literature on technological approaches to the problem. Informed by this state-of-the-art in the field, the designers evaluate constraints in addressing the envisaged idea; this evaluation is thereby typically performed in an unstructured and unguided manner. For instance, D6 speaks of a "hardware tradeoff" that has to be considered for an autonomous robot and D5 analyzed a given data set, finding out that the available room occupancy data was not sufficient to support an intelligent microclimate control system and had to be combined with other data sources providing indoor location data. After this evaluation, the designers select an appropriate technology for implementing the idea given the constraints of the project. Having selected the technology, most designers continue with the implementation of a prototype. Depending on the original idea's sophistication and concreteness (*e.g.,* whether or not there is already detailed knowledge about users' needs for the envisaged system or how precisely the idea has already been shaped), user studies for evaluating the idea are performed, which may result in fine-tuning of the idea. As a result, the implementation phase may still be characterized by a high degree of ambiguity.

Archetype III, which we term 'idea-based technology selection process', was reported in 3 projects. It is characterized by an idea-based technology selection (Fig. 4). Like Archetype II, this process also starts with an idea for an envisaged system. In contrast to the other archetypes, designers following this process select the technology that they want to use for addressing the idea in mostly an *ad hoc* or not thoroughly informed way; some designers, though, evaluate constraints before picking a technology for the project (*i.e.,* in some cases, technology selection is performed alone, while in some other cases, technology selection is followed by fine tuning). For instance, D12 argued that for the envisaged system that should replace a physiotherapist, it was apparent to him to use (Kinect) cameras for analyzing whether movements are correctly performed. After this technology selection, he compared diverse solutions that used this kind of technology. D9, on the other hand, reported that an early idea of movement assessments for the rehabilitation of stroke patients had already been implemented as a prototype using a Kinect camera. When thoroughly redesigning the system to become a "virtual coach" using a gamification approach, refining the movement assessment, and introducing suggestions for training improvements, the new team decided to stick to the chosen technology of the prototype. Given the selected technology, the designers proceed with fine-tuning the original idea, so that the idea is feasible with the selected technology; this is typically performed in brainstorming sessions that are unstructured and unguided. D14, for instance, wanted to have a cheap

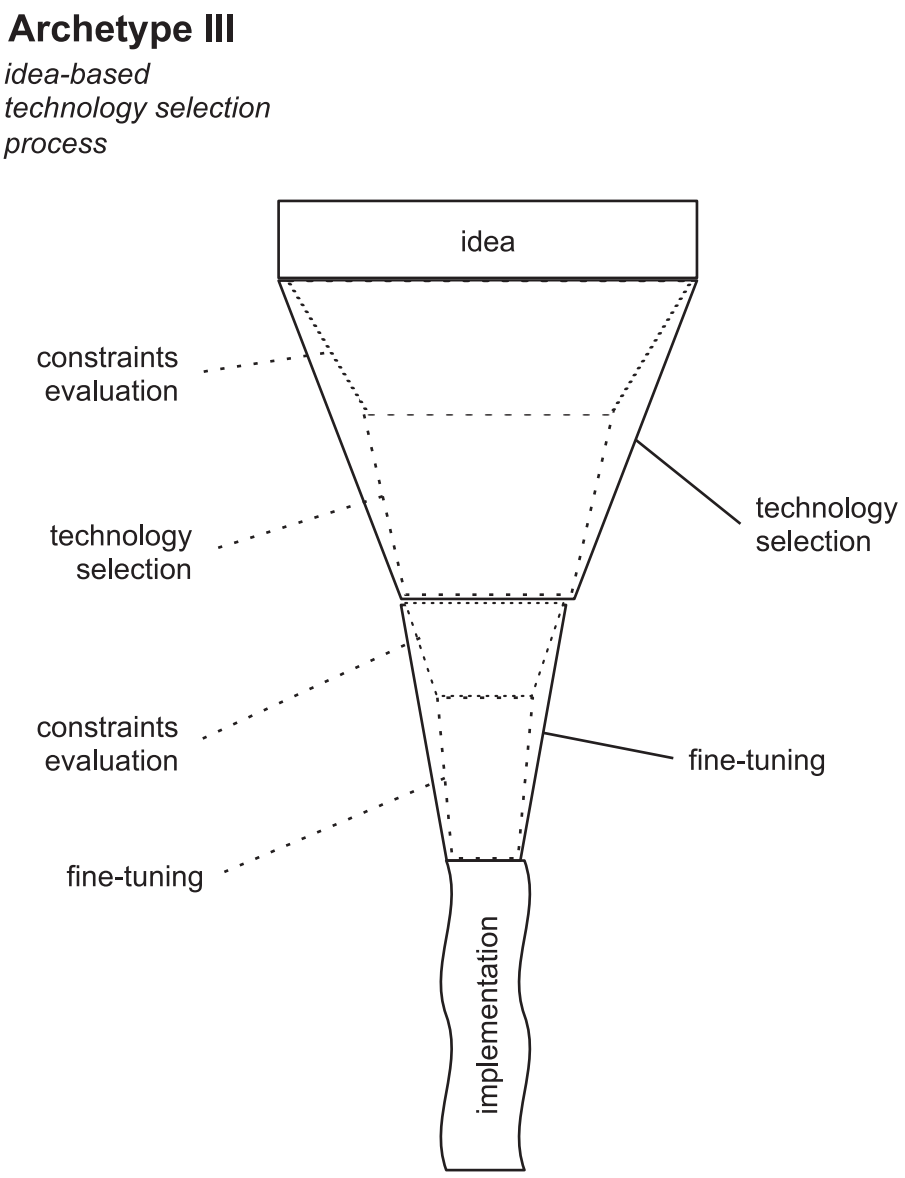

Fig. 4. Archetype III—idea-based technology selection process.

and broadly available solution for a navigation guide for blind people. When she had decided on an iPhone–Kinect combination, she had to fine-tune the originally broad idea to what she could actually implement with the sensors integrated in the chosen hardware. As soon as a solution is found, this archetype continues with the implementation of a prototype. Similar to the previously described archetype processes, the implementation phase in this archetype is also characterized by high ambiguity.

Archetype IV, which could be identified in four projects, is purely technology-driven (Fig. 5); so we termed it 'technology-driven idea-generating process'. Whether within the frame of a set goal or without any creativity-limiting constraints, the first step of this process is the selection of a technology. Whether this is informed or not by the state-of-the-art in the field (state-of-the-art analysis), this process proceeds with brainstorming for an idea, given the selected technology. A typical project following this process is a smartphone-based one (*e.g.,* P10a, P7), where designers first decide to use a smartphone and its embedded (sensor) technology and computing capacities, and then look for creative ideas about how to leverage the device's capacity to provide utility for its users. If designers already know that the idea is implementable, they can start right away on its implementation. In most cases, though, it is essential to evaluate constraints with the selected technology before they can move on to the idea's implementation and later adjustments may be necessary.

Archetype V was found in two explorative research projects (Fig. 6) and is termed 'explorative data-driven process'. Whether idea-based or creatively open, technology is selected for the project (*e.g.,* a smartphone with its integrated sensors in P6b). With the selected technology, designers collect all the data that they can elicit while users are performing a specific task—which may be as broad as "use your smartphone as you normally do". D5 explains that he wanted to try and see what was possible to enhance smartphone user interfaces. After data collection, this data is analyzed for patterns, whereby personal experience and intuition were reported as important factors for this task (D6). Having found patterns, the process continues with the implementation (in the idea-based variant of the process) or, in an unstructured and unguided brainstorming session; the designers strive to come up with an idea for leveraging these patterns, which is then implemented as a prototype. For instance, D6 reports that she iterated on an algorithm for machine learning and repeatedly went back to the sensors and sensor data for refinement. Depending on the robustness of the ideas for leveraging the patterns, the implementation phase is characterized by a lower or higher degree of ambiguity.

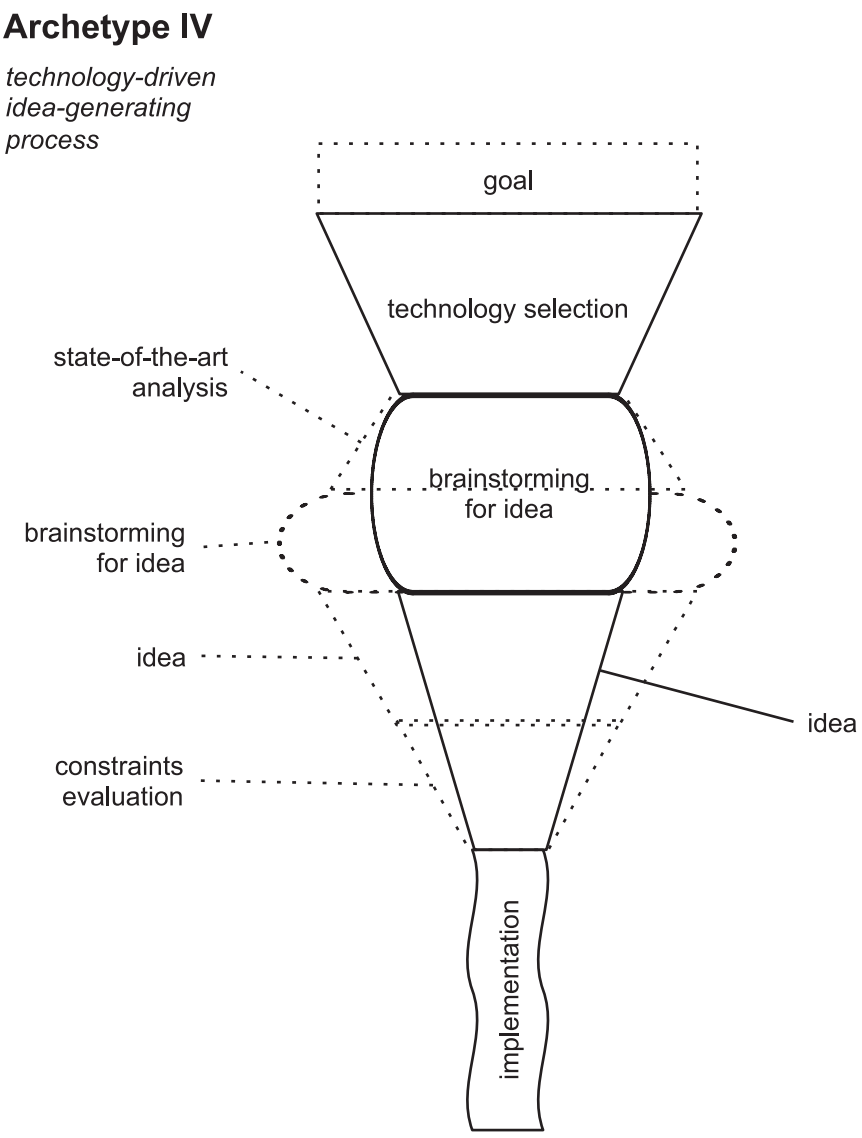

Fig. 5. Archetype IV—technology-driven idea-generating process.

4.3. Discussion of the design processes

Most of the designers we interviewed reported using some sort of brainstorming in their design processes, which tended to be unsystematic and without any guidance or tools. Brainstorming is used for many purposes; for coming up with system ideas, to fine-tune a project, to identify and select context elements, or to consider constraints of any nature (*e.g.,* budget constraints, technological constraints). Interestingly, our interviewees reported a tendency to consider constraints in an early brainstorming phase, when ideating on system ideas. In other words, idea generation, screening, conceptualization, business assessment, and technological assessment are all considered intermingled in a fuzzy way. Thus, compared to how other types of systems are typically designed, we could identify an unsystematic, strongly interwoven approach to system design among the interviews. Particularly the activities concerned with the identification and selection of relevant context elements for the respective systems – *i.e.,* the tasks that have to be performed additionally when designing intelligent systems – appeared intermingled with the other activities throughout the process of system design.

Furthermore, we found that most reported design processes tended to have an explorative character in terms of starting the process with a broad spectrum of opportunities where the idea or goal is only vaguely defined. Interestingly, though, when expressing this observation to the interviewees, some emphasized that they knew exactly what they wanted to do and that their work had not been explorative at all.

Another interesting finding is that, to a large extent, the five identified archetypes contain the same process components, but use them in different orders. The order, though, may have a significant impact on the results. Selecting the technology before fine-tuning or performing constraints evaluation (Archetypes IV and V; partly also Archetype III) narrows down options drastically and may lead to a sub-optimal system. In the worst case, an incorrect (because of being not thoroughly informed) technology selection early in the process could even cause a project to fail, for instance, if the selected technology does not offer a viable technical solution. Narrowing down the options first with a constraint evaluation followed by brainstorming for fine-tuning the project, or other way round, may lead to better results (*i.e.,* systems).

Furthermore, it is interesting to see that some designers start with a relatively specific idea (Archetypes II and III; partly also Archetype V), while others start with a rather broad goal and narrow it down to a more specific idea in a later phase of the process (Archetypes I and IV).

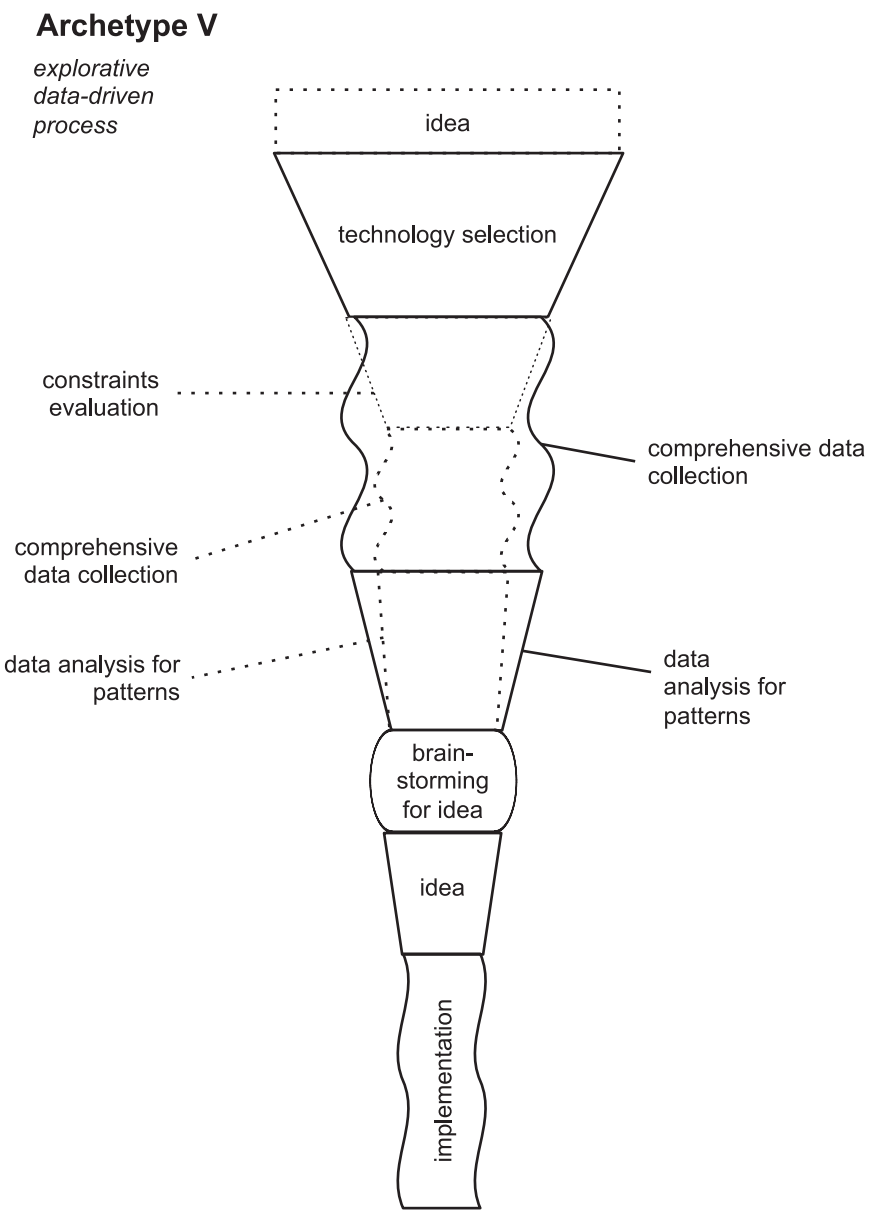

Fig. 6. Archetype V—explorative data-driven process.

The determinant of which type of process is finally followed depends on the purpose of a particular envisaged system and/or how well the specification for the system has been elaborated. The more specific an idea is and the fewer options that are being considered, the more appropriate it would be to select the technology rather early in the process. When the goal is rather vague, in contrast, more effort has to be spent on fine-tuning before it seems fruitful to start technology selection.

In addition, we learned that the large majority of the interviewed designers (all except D1, D7, and D10) have not explicitly used any of the existing context models. This seems surprising, as many context models (*e.g.,* Bradley and Dunlop, 2005; Schmidt et al., 1999; Tarasewich, 2003) were created with the intent to better understand context and to support the selection of context elements (cf. Bauer, 2012). Still, some of the designers indicated that they are well aware of the literature in the field; D4 pointed out that the context models presented in the literature might have implicitly inspired designers' work. However, as Bauer and Spiekermann (2011) also pointed out, our interviews indicate that designers seem to mainly rely on personal intuition and experience, when it comes to identifying and selecting the relevant context elements that best serve an intelligent system.

Compared to the other archetype processes, Archetype 5 stands out from the rest as following a very explorative approach. This process would typically be pursued within performing research or for cutting-edge, innovative new product development for consumers. However, companies installing an intelligent information system for their intra- or inter-organizational use would rarely use this archetype process.

To sum up, designers of intelligent systems follow similar approaches but sometimes in a non-ideal order, include much brainstorming that is unstructured and unsystematic, and frequently build their systems on not thoroughly informed ideas. However, those projects that seem to be informed are frequently not business-oriented in the sense that a system is designed without having a clear understanding of the potential purpose and whether there is a market for it. As a result, there is a need to introduce an 'ideal process' that guides system designers in their design. Particularly the selection of context elements tends to be not thoroughly informed and/or unsystematic, which calls for particular guidance with this task.

4.4. Discussion of the particular challenges in designing intelligent systems

During the process of designing an intelligent system, the designers encountered various problems. For instance, D14 reports of difficulties with testing the software solution since the specification of the software's functionalities was too imprecise in the project. She emphasizes that functionalities should be defined more precisely in an earlier stage of the design process.

In an early stage of the project P8, the designers decided to use the Kinect technology. D9 reports that familiarizing himself with this technology, which had been new to him, had taken a large amount of the time allotted to the project. It was extremely challenging to find out whether and how the team members could implement the ideas and concepts that came up during the project with this relatively new technology. Retrospectively, D9 considers the selected technology as not being the best choice for the goal of the project.

Furthermore, D6 and D13 report for their projects respectively that a main challenge had been to identify what context data was needed and at what level of accuracy. D6 had to go back to the sensors frequently and adjust what context data was collected and the respective accuracy of these to obtain useful data for the project goal. Only later in the project, it becomes clear what context data required what level of accuracy in the project. D13 explains that they had collected too fine-grained context data early in their project, which ended up not being meaningful for their project. In addition, D13 reports that data collection and testing was limited by the hardware (*i.e.,* smartphones), which was not fully considered in an early stage of the project.

D12 noted that it is typically expensive to achieve high accuracy of context recognition. For his project, he reports that high accuracy was important, though, since it seemed to be a necessary requirement for achieving high acceptance of the solution among the users.

A further challenge frequently encountered by the designers was related to constraints, particularly with respect to power constraints (D6, D14), hardware tradeoffs (D6), and legal constraints for data collection and recording (D11, D14, D15). Frequently, the designers had to reconsider their conceptual solutions while already in the implementation phase as they encountered constraints that they had not considered, or not considered fully, earlier.

To sum up, the specification of functionalities should be explicitly addressed in an 'ideal process'. Technology selection should not be addressed too early in the process. Context data requirements, particularly with respect to its accuracy, need to be addressed. Additionally, constraints are an important issue, which should be explicitly dealt with in an 'ideal process'.

5. An ideal process

In this section, we outline the flow through such an 'ideal process' (Section 5.1). This process uses the same constructs that were already defined in Section 4.1. In Section 5.2 we suggest two supportive 'modules' for the process that are designed to support system designers in their context selection task in the fine-tuning phase (Section 5.2.1) as well as in considering constraints for the envisaged system in the constraints evaluation phase (Section 5.2.2). Finally, Section 5.3 presents the results of the utility evaluation of the process and the two modules.

5.1. The goal-driven fine-tuning process

The starting point for our ideal process is the decision to design an intelligent system in a certain domain for a specific purpose.

From a business point of view, it is necessary to specify an intelligent system's goals, before other design actions may be started (Choi, 2007) (Fig. 7). A set goal is typically based on identified business or user needs. Pang et al. (2012) suggest that a "business-technology joint design" is needed to bring successful intelligent system solutions into the market. They emphasize that the creation of business value is essential, and informational as well as technical requirements have to build on and toward this business value, because users will pay for values, not for technologies.

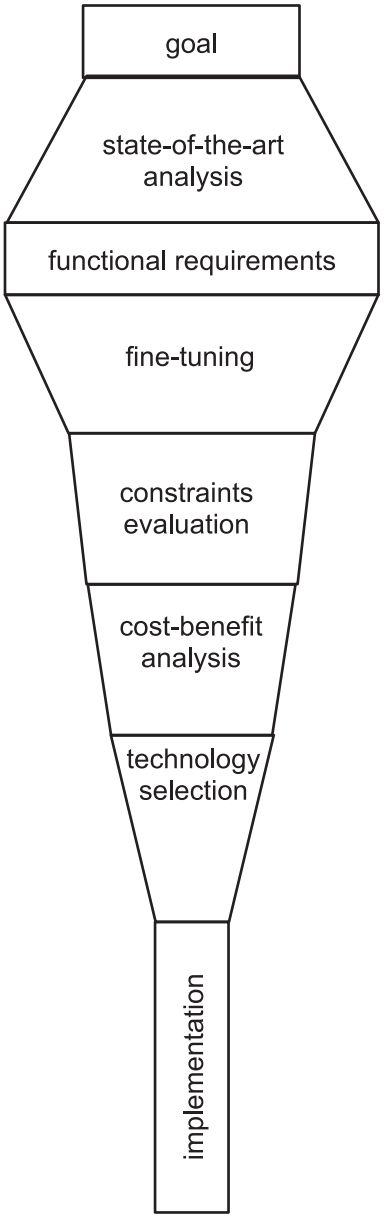

Fig. 7. The ideal process—goal-driven fine-tuning process.

Within the frame of the specified goal, a thorough state-of-the-art analysis of existing work and ideas follows, to be able to build on existing knowledge and avoid "reinventing the wheel". This analysis could include scientific literature analysis, prototype analysis (including in-house prototypes as well as competitors' products and other market players' work), patent analysis, and being informed by experts in the field. The objective of this stage is to embrace the information landscape that may be relevant for the envisioned intelligent system. The central questions are: What context exists that might be relevant to serve the envisioned system's purpose? What solutions to similar projects have been found? What are the main challenges in addressing this problem? After this phase, designers are well informed about the state-of-the-art concerning what is possible to do, where challenges exist, and what has already been done.

Informed by this analysis on a broad scale, system designers should be capable of defining indispensable functional requirements for the envisaged system. Sitou and Spanfelner (2007) emphasize that user and business needs have to be taken into consideration for requirements definition. Essentially, the basic functions need to be specified before being able to define the context elements that are required as input for the envisaged system. Several approaches for use case creation for intelligent systems are available that may further support this task (*e.g.,* Choi, 2007; Omasreiter and Metzker, 2004).

Thereafter, designers go through several phases to narrow down and substantiate their ideas. For the design of intelligent systems, the order of steps performed in this narrowing-down process is crucial, because sorting out fruitful options too early could result in non-ideal solutions or even in not being able to solve a problem. As Dobson (2005) points out, "the essence of pervasive computing lies in synthesizing data from a range of sources to extract and utilize the maximum amount of available information." A rather drastic step of narrowing-down options is, for instance, the selection of a specific technology. When, for example, deciding to use an accelerometer for solving a task, then it will be impossible to consider the context element 'emotion' in the project, even when it may be highly relevant to addressing the problem at hand. As a consequence, it is key in the fine-tuning process to keep options that could be relevant for solving the addressed problem within scope for as long as possible. Once a context element has been removed from consideration, system designers will not consider them anymore ('out of sight is out of mind'); and even if considered again, it would mean going back in the process to an earlier stage, resulting in an inefficient solution-finding process.

As a result, we suggest fine-tuning as the next step to detail broad ideas and shortlist them. While the current practice of doing so tends to be in a rather unsystematic way through brainstorming, we propose a systematic procedure, which we outline in detail in Section 5.2.1. The objective of fine-tuning is to sort out options that are not relevant for the envisaged system. The result is a shortlist of relevant ones for a good solution. We emphasize that constraints of any nature should not be addressed yet at this process stage, as this could lead to filtering out elements that may be relevant. Furthermore, trying to evaluate constraints already at this stage would mean doing this for context elements that are probably not relevant to a good solution, which is inefficient.

Accordingly, a constraints evaluation stage follows the fine-tuning stage. The objective of this stage is to determine whether the shortlisted context elements can be further considered in the project in terms of feasibility (*e.g.,* patents for the methodology are available, context information is accessible, technology is sufficiently accurate). While some context elements will be filtered out at this stage, others will be identified as readily feasible. As a result of this design stage, a subset of opportunities for a final solution will again be short-listed. While Section 4 showed that the current practice of a constraints evaluation is rather unstructured and frequently performed in an earlier stage of system design, we suggest a more systematic approach (see Section 5.2.2) and emphasize the importance of not performing this evaluation too early in the process.

The objective of the cost–benefit analysis is to evaluate the relevant and feasible context elements concerning their costs and benefits, to be finally able to come up with suggestions for system design and implementation. Each context element is evaluated concerning its cost involved to obtain the required data and its benefit regarding its contribution to make the intelligent system support the desired set of functionality. Costs and benefits are thereby not solely to be interpreted in monetary terms (tangible) but also include non-monetary (intangible) values such as image, social responsibility, cross-selling, first-mover, *etc.* (Fleisch and Tellkamp, 2003). For each context element for which various ways of implementation are feasible, typically the implementation with the best cost–benefit ratio is selected. Still, strategic consideration may also influence the selection. For instance, one may invest in seminal methods and techniques. The result of this design stage is a cost–benefit assessment for each of the technical implementations considered, and within those, for each considered context element, in the design of the respective intelligent system.

Finally, based on a thorough cost–benefit analysis, the best option for technology selection can be identified. Razali et al. (2012), for instance, suggest a systematic technology selection method that may be used for this task. The result of this stage is the decision about which technologies to use for considering the respective context elements in the system. This ultimately leads to the implementation of the intelligent system. A toolkit that supports rapid prototyping (Dey et al., 2001) may be useful for this task.

5.2. Supportive guides for the process

As already indicated, the interviews revealed that there were two tasks in particular that are approached rather unsystematically; both refer to the identification and selection of relevant context elements for a system: fine-tuning and constraints evaluation. As a result, we focus in on these two phases and provide guidance for these tasks.

In Section 5.2.1, we suggest a systematic procedure for selecting the relevant context elements for an envisaged system in the fine-tuning phase of a project. Section 5.2.2 deals with the constraints evaluation and provides a checklist-like taxonomy that should prevent designers from missing important kinds of constraints.

5.2.1. Fine-tuning procedure for context selection

We suggest a fine-tuning procedure that is structured in four phases: (1) relevance, (2) combination, (3) precision and accuracy, and (4) level of relevance. The numbers (1)–(4) represent the logic sequence of the phases (Fig. 8).

- Relevance: As emphasized, for instance, by Huang and Gartner (2008) some context elements are relevant for a system while others are not.
- Combination: Some context elements are highly dependent on each other for a specific purpose. A typical combination is, for instance, spatiotemporal context that combines location and temporal information, as for many systems it is essential that two entities are in the same place at the same time. Consequently, for such systems the context element's location and time have to be combined.
- Precision and accuracy: A system does not always require perfect precision and accuracy (Dobson, 2005). For instance, for a location tracking system the Cartesian coordinates may be required as context, while for a home appliance control system it may suffice to know whether the owner is at home. Accordingly, the precision and accuracy required for context elements has to be specified, as each technical solution will deliver different results concerning accuracy and precision; and even the readings of the very same sensor may vary widely over time (Dobson, 2008).
- Level of relevance: While some context elements are crucial for fulfilling a system's purpose, other context elements may only add value, for instance, by enhancing the convenience of the user. The level of relevance will support the making of well-informed decisions about which context elements have to be included in the system and which may be optionally implemented depending on their relative costs and benefits.

In a first step, the context elements that appear relevant for an intelligent system solving the specific problem are identified. Thereby, relevant context is identified at a high level of context abstraction (Bettini et al., 2010). The central question is: What context elements – out of the enormous context universe – are relevant for the identified problem? This results in a condensed taxonomy of relevant context information. Possible sources for the identification of relevant

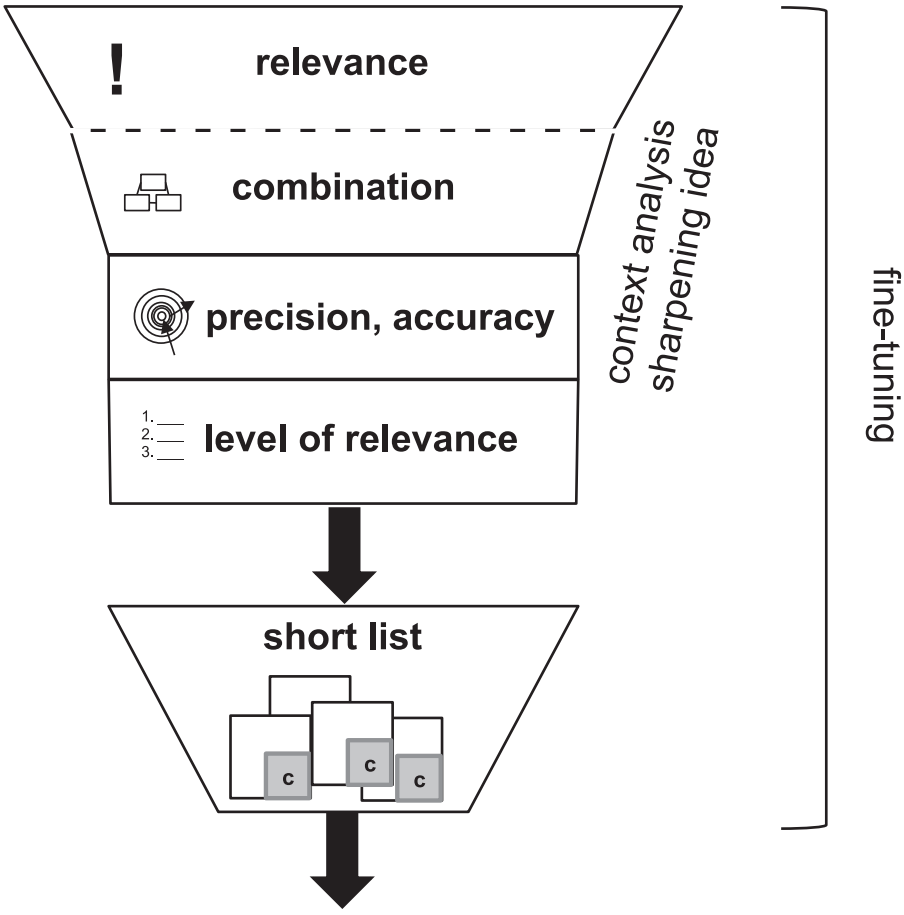

Fig. 8. Fine-tuning context elements.

context are, for instance, document analysis (*e.g.,* Huang and Gartner, 2008), interviews with stakeholders (including potential users) (*e.g.,* Castelli et al., 2009), or so-called "speed dating" for rapid prototyping (*e.g.,* Davidoff et al., 2007).

In a second step, context elements that can be processed in combination with another one are combined. The central question: What context elements are dependent on each other to be valuable for the designated problem? For instance, when building a system that references the location of places via social context (*e.g.,* "the place you met your fiancée for dinner last time") (Kjeldskov and Paay, 2010), both spatial and social context are relevant and make sense in combination only. The result of the second step is, hence, a condensed list of context elements. Sometimes the steps 'relevance' and 'combination' may be performed concurrently. Still, first the relevance of a context element is determined, and in a second step the necessity of the combination with another context element can be specified.

The third step is dedicated to specifying the necessary level of precision and accuracy of context elements. The central question: What level of precision of accuracy is required for a given context element? If, for instance, temperature is identified as relevant, we have to determine the level of precision and accuracy. Is it sufficient to know the average temperature in a region (as, for example, reported by weather stations)? Or is it necessary to know the temperature in front of a certain store? Or is it crucial to know the temperature exactly at the coordinates where a user is situated (for example, in front of a refrigerated display case in a particular shop)? Or is it the change of temperature (for example, from hot to cool) that is of interest for an application? While for some context elements, a high grade of precision will be necessary, for others rather broad (*e.g.,* average, approximated), valid information will suffice. Sometimes, all we need to know is whether someone is present or absent (Dobson, 2005). For location, for instance, some systems will require the use of precise Cartesian coordinates, while for other systems approximating spatial context to specific places or regions will suffice (Dobson, 2005; Kjeldskov and Paay, 2010). Often location information is implicit in other information sources, although in an imprecise way. For instance, one may infer a diary-owner's location from the information provided in the diary (Dobson, 2005). We emphasize that there are a lot of different sources that can supply context information, going beyond context collected via physical sensors (*e.g.,* user name, entry in an electronic agenda for remembering a meeting, age of the user, number of computers in a room, the ingredients for a recipe). Also for such information, precision and accuracy are relevant (*e.g.,* exact time and location of meeting versus estimate for meeting time, age of the user in days, years, or between 30 and 40, correct number of computers, an approximation of the number or between 4 and 6 computers, white cabbage for a recipe versus providing the information 'vegetable'). The result of the third step is a hierarchical structure: For each context element, the levels of precision and accuracy (freshness) are determined and described. An alternative approach is to specify context elements in terms of "quality requirements" in a wider sense as proposed by Hoyos et al. (2011). This approach includes precision and accuracy, as well as other criteria such as coverage location range, format, believability, and availability. When using that approach, we suggest, however, considering only criteria that are independent from technology and/or source. For instance, according to our proposed process, the criteria 'data format', 'data location' for acquisition, and 'availability' should only be assessed later in the cost–benefit analysis where various approaches for technology selection and implementation are compared. For instance, temperature may be measured by a

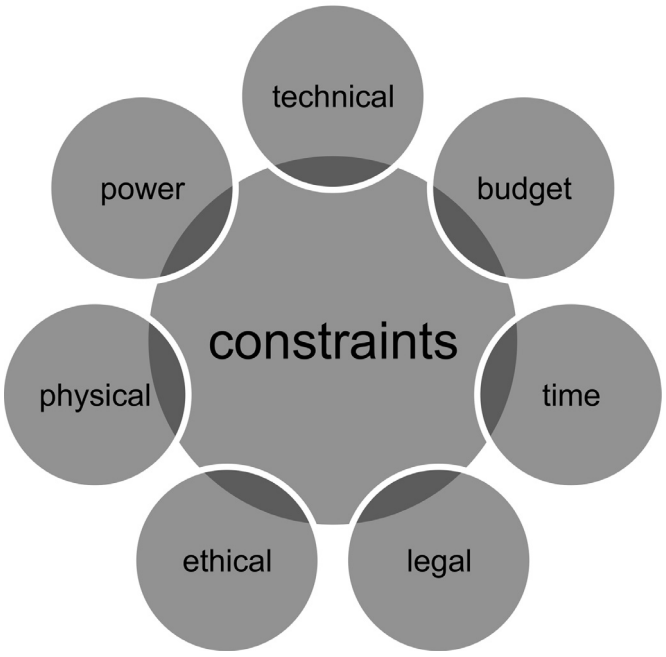

Fig. 9. Constraints evaluation module.

sensor that belongs to the system to be built or may be retrieved from an open or commercial weather data provider. The decision about which solution to take should not be intermingled with context identification and selection, which is the purpose of the procedure described here.

The fourth step is for assessing the level of relevance of the qualified context elements. While some elements or element combinations may be crucial for a decent solution (must have), some may be valuable but not crucial (should have), and others may be nice to have. The central questions: Which are crucial context elements? Which elements would 'just' enhance the service? The purpose of this step is to describe what context elements are needed for the system to fulfill the goal and distinguish the essential context elements (must have) from ones that would substantially improve the system (should have) and additions that are nice but not substantial (nice to have). If a lot of context elements are identified, this step helps to narrow down the scope of context elements, to focus on the essential ones. For small-scale projects with a narrowly defined scope and goal, this step may be minor and could also lead to the result that all previously identified context elements are crucial for the system. Still, it may be important information for the subsequent analyses to know which context elements are crucial and which are not. For the latter, perhaps less time and/or budget could be spent on identifying possible solutions than for the more important context elements. The result of this fourth step is simultaneously the result of the fine-tuning stage.

Accordingly, the result of the fine-tuning stage is the identification of information that is (1) relevant, (2) combined as needed, (3) specified with the necessary level of precision and accuracy and (4) assessed by level of relevance.

5.2.2. Constraints evaluation (feasibility analysis)

In our first set of interviews, we identified four types of constraints that have to be considered before being able to assess possible solutions in a cost–benefit analysis: technical, budget, time, and legal constraints. As a result of the expert interviews for utility evaluation (Section 5.3), three constraints were added: ethical constraints, physical constraints (*i.e.,* size, volume, weight), and power constraints (Fig. 9).

Whatever the nature of a constraint, the goal of the constraints evaluation is to determine whether the considered context elements are obtainable in the required grade of precision and accuracy. This evaluation includes the formats, quality, and the reliability of the methodologies and sources. The central questions are: Can the desired context information be acquired on time and within the budget? What ways of realizing the context information acquisition exist? At what quality? How reliable is the method? Are there legal barriers involved? Is it possible to implement the method on time and within budget? Can the resulting complexity of sensor information be processed within the given time and budget constraints? Is it ethical to collect the desired data? Are there appropriate workarounds? What are the limits concerning size, volume, and weight of potential physical sensors? Is there access to a power supply within reach of the system?

Gray and Salber (2001) describe six quality attributes for context: coverage, resolution, accuracy, repeatability, frequency, and timeliness. Ranganathan et al. (2004) provide a categorization of three different kinds of quality metrics: resolution, confidence, and freshness.

It may happen that a required context element is not accessible at the desired level of quality (Sheng and Benatallah, 2005). If it is a 'nice to have' context element, a lesser quality may be sufficient or the element (and the functionality, in which it is embedded) may have to be entirely omitted. If it is a crucial context element for solving the problem, though, it indicates that the entire envisaged solution is (to date) not realizable for this project.

Determining the feasibility of the solution also requires thinking about the design and implementation of the system. We have to bear in mind that information acquired from physical sensors is almost always low-level context (Bettini et al., 2010). With interpretation and transformation, we then achieve a higher level of context abstraction at the specified level of precision (Broll et al., 2007). For instance, Loke (2006) describes six different ways to specify the high-level context abstraction "in a meeting now" based on low-level contextual cues. If only the high-level abstraction is relevant for the particular application, the various ways to obtain and determine this situation would be important to consider. The degree of reliability of the various methods to acquire the information "in a meeting now" will vary. The same is true for the quality of information.

As a result of the constraints evaluation, the various context elements and methods to collect and process them are identified, which is represented as a set of alternative solutions.

5.3. Results of the utility evaluation

The utility of the proposed design process was evaluated by drawing on the experience of system design experts in the field (see Section 3.4). The results attest to a high perceived utility of the design process for real-world projects.

Mainly, the designers differentiated between two types of applications: (i) For projects, where the idea is more or less clear-cut, the functional requirements can be defined right after the state-of-the-art analysis (see Fig. 7) because the idea for the envisaged system is sufficiently clear. The designers mentioned particularly that the process is therefore useful for incremental innovations. (ii) For innovative projects, where the goals are fuzzy, in contrast, it is difficult to define the functional requirements as a third step right after the state-of-the-art analysis. The designers suggested addressing the requirements definition in this case after the fine-tuning phase because the envisaged system has to be sufficiently clear to perform this task.

Furthermore, D11 suggested that a fuzzy goal should be made more concrete before the state-of-the-art analysis because he felt that it would be difficult to perform a state-of-the-art analysis for a very broad field. D14 explained that performing a state-of-the-art analysis always includes some kind of concretizing of ideas and narrowing down the scope because one cannot study the entire field and instead continuously decides at what point to delve deeper into a topic throughout the entire analysis process.

D8 and D13 explained that it would be difficult for them not to think in terms of technology until the technology selection phase. "I have an engineering background. I always think about technology," D8 indicated. However, both D8 and D13 agreed that there was an advantage of leaving options open until the very end before implementation.

The detailing of the fine-tuning stage on how to consider or filter out context elements for an envisaged system was well accepted among all interviewed system designers. They welcomed this guidance and affirmed its usefulness for the design of intelligent systems. Most of the interviewees (11) indicated explicitly that they would use it in future projects. While some were thinking about using the fine-tuning process as a guide in brainstorming sessions with their project teams, others indicated that they would use it in more structured settings. The designers shared the opinion that the guide would be particularly useful for "newbies" for designing intelligent systems. Furthermore, the designers emphasized that iteration may be required in this process if something important was forgotten in an earlier step. D4 highlighted that he would iterate from 'relevance' to 'precision' for each context element identified.

For the constraints evaluation, we initially considered time, budget, legal, and technological constraints. D2 pinpointed the significance of physical constraints with respect to volume and weight of the technical solutions being considered. Also power constraints, which were mentioned by four system designers, add to volume and weight, in particular for mobile solutions. Further, some interviewees indicated that it is important to consider ethical constraints, which may vary with respect to culture. D1 pointed out that the cost–benefit analysis could be integrated into the constraints evaluation phase of the process.

In the interview with D4, there arose an interesting issue of differentiating between strict and 'controllable' constraints. While, for instance, legal constraints are strict, companies may increase (or decrease) other constraints such as budget or time for a project.

Overall, the interviewed system designers concurred that the taxonomy of constraints would be valuable for designers that are new in the field of designing intelligent systems. D13 emphasized that it would prevent beginners from making the same mistakes that most people do in their initial projects. For more experienced designers, in contrast, the constraints should already be clear from learning-by-doing in earlier projects. Still, the majority of the interviewees agreed that having a reminder for legal or ethical constraints could also be useful for experienced designers because these are constraints that technology-oriented professionals tend to forget. D11 explained that the rather abstract taxonomy would not increase his knowledge or experience, but he would greatly appreciate a checklist that details the "to-be-regarded issues" for each of the constraints.

Furthermore, almost all interviewees emphasized that iterations are crucial in system design and borders between the process stages are fuzzy. Particularly, fine-tuning and constraints evaluation are easily blurred.

Overall, most of the system design experts indicated that there will never be a 'one fits all process' as different purposes will require different courses of action, which will result in variations of the proposed process when applied in practice. Some, however, added that the proposed process allows enough flexibility for such variations and that it constitutes a good guide. D13 explained that the process would be a good documentation of practices in ubiquitous computing and could even imagine it to be included in the main textbooks about ubiquitous computing fundamentals.

Based on the interviews, the benefits of the process for its application in practice may be summarized as follows: First, the process systematizes the design efforts, which makes work less prone to errors in practice. Second, it helps designers to not forget about important design steps and issues such as considering physical constraints of sensors. Third, the process allows for a steeper (or faster) learning curve, since novices to designing intelligent systems are less likely to make mistakes that are covered by the process guide. Fourth, it counteracts predominant technology thinking, as the process pinpoints and guides a designer through significant design steps prior to technology selection. Fifth, the fine-tuning procedure provides a step-by-step guide for the identification, assessment, and selection of context elements for an intelligent system, which is a task that has to be dealt with in addition to other design issues such as business requirements elicitation, system coupling, coding, system architecture design, or acceptance testing with potential users (cf. Bauer, 2014; Castelli et al., 2009; Kolos-Mazuryk et al., 2005; Sitou and Spanfelner, 2007). Overall, the process was considered a good guide, based on feedback from the interviews, that provides sufficient flexibility allowing for variations that are required for the complex challenge of designing intelligent systems (cf. Fortier et al., 2010).

6. Application of our 'ideal process' to the design of an intelligent supportive mobile system for diabetics: a case study

We apply our ideal process to the design of a representative intelligent system: an intelligent supportive mobile system for diabetics. For the sake of simplicity, the tables and diagrams we provide only partially represent the artifacts created throughout the process. Note that we assume the design team working on this project is interdisciplinary and consists of people with business management backgrounds, health experts with respect to diabetes, and IT specialists. The project, we draw from, was carried out in a university context. The research objective was to evaluate the applicability and utility of a guide, containing a wide range of potentially relevant context elements, in the process of designing an intelligent supportive mobile system for diabetics.

The goal of the envisaged system is to support diabetics in everyday life in dealing with this disease. The challenge at this stage is that it may be difficult to precisely define the goal. Particularly when a stakeholder, who is not the future user, initiates the project, this might result in some misunderstandings or vague specifications. As the goal specification for the envisaged system is rather vague, a formative user study would be useful to better understand the problems that occur in a diabetic's daily life. Based on these initial interviews, two major subgoals can be identified: (1) reminding diabetic users of common tasks such as measuring blood sugar levels or injecting insulin, in a flexible and non-intrusive manner and (2) (partial) automation of documentation of these tasks' results.

After defining the goals, the next stage in the process is a state-of-the-art analysis in the field. If the goal is rather broad, the analysis needs to be performed on a broad level as well, as various characteristics of the problem need to be explored, which may also require more time. On the other hand, for very specific goals, the state-of-the-art analysis may also be performed on a slightly broader basis when the goal represents a novelty in the field. In any case, the analysis needs to cover domain-specific aspects as well as topics that concern IT needs. For the envisaged system, the addressed topics in the domain of diabetes have to be explored (*e.g.,* kinds of diabetic tasks, the required frequencies, the desired and undesired effects of treatments) as well as existing systems that address similar issues in the relevant domain (*i.e.,* systems for diabetics) as well as similar types of systems (*i.e.,* reminders and automated documentation). Note that potential users might be an additional source of useful knowledge, as they are likely to know the hints and tips as well as gadgets that they find useful.

After tapping the knowledge acquired from the state-of-the-art-analysis, functional requirements need to be defined. As some designers (D1, D5, and D11) pointed out in interviews, this task may be reasonably performed for a precise goal and proper state-of-the-art in the field (*e.g.,* for incremental innovation). However, it might impose some challenges on designers for radically innovative projects, as these break completely new ground. The envisaged system for

Table 5
Excerpt of functional requirements for the intelligent, mobile system for supporting diabetics.

/F11/	The system sends reminders for measuring blood sugar.
/F21/	The system sends reminders for insulin injecting.
/F31/	The system sends reminders for eating.
/F42/	If a specified period after a reminder elapses, the reminder is sent again.
/F43/	A maximum of a defined number of reminders is sent.
/G11/	If the system receives data input for measuring blood sugar before the reminder for measuring blood sugar is sent, the reminder is deactivated.
/G12/	If connectivity from glucometer to mobile user device is established, the measurement data is transmitted to the system including a time stamp.
/H23/	The system calculates the dose of insulin according to the insulin/carbohydrate units (CU) ratio based on the CU input.

diabetics may be regarded as incremental innovation, as it can build on prior work in the areas of context-adaptive reminders (*e.g.,* Johnson et al., 2014; Tsai and Liu, 2014) and syndication of several mobile devices used in the medical domain via wireless connections (*e.g.,* Corchado et al., 2010; Morón et al., 2014). An excerpt of the functional requirements for the system is given in Table 5.

Depending on how elaborate the goal and the functional requirements are, the fine-tuning stage will take more or less time. In this stage, the challenge for people with an IT background is to abstract from technology (*i.e.,* not considering technology constraints in this stage). A further hurdle is to think creatively without being too visionary, which may result in ideas that are by far not (yet) implementable in any form. Using the fine-tuning guide for identifying and evaluating context elements, Table 6 illustrates an excerpt of the process and the results of each step for the envisaged system. Thereby, the guide (Fig. 8) is used as a structural element for the brainstorming session by the design team.

In the constraints evaluation, the shortlisted context elements are evaluated for possible constraints. While the constraints evaluation guide may provide inspiration for first-time designers (particularly for stakeholders without an IT background), no further specific guide is – to the best of our knowledge – currently available. The "obstacle analysis" as proposed by Ruiz-López et al. (2013) may be used, although their approach is dedicated to requirements engineering. Still, it offers structure to the constraints evaluation task, which may be eligible for application. For our envisaged system for diabetics, the main constraint is on the hardware side, because to be useful, all involved devices have to be mobile. Devices such as a glucometer for measuring blood sugar, insulin pens (*i.e.*, a device for injecting insulin), smart watches that are also capable of measuring heart rate, and smart scales that can transfer weight data wirelessly are available on the market. Sensing the presence of people may be challenging with a small-scale device and issues concerning accuracy have to be considered. In the subsequent cost–benefit analysis, each approach for operationalizing whether people are present has to be analyzed to see which offers the best solution, at acceptable costs.

For the cost–benefit analysis, a thorough set of technological alternatives, which allows for gathering the selected context inputs (either via manual user input or sensing), has to be elaborated, in order to estimate the benefits and costs of each alternative. As such, it is also important to consider non-monetary benefits and costs. For the envisaged diabetes system, the perceived benefit of automatically detecting demographics such as gender and age may not be high compared to relying on manual input of these data, as these are one-time inputs. The detection and/or prediction of dynamic context such as current activity and stress level may have particular value for users, as such information does not only determine the required insulin intake amount, but also whether the user is ready for insulin intake or whether this task should be postponed. Generally, a major challenge for the cost–benefit analysis lies in the difficulty in anticipating the benefits, particularly when it comes to highly innovative systems. For this and other reasons, it might occur that valuable, innovative projects are not pursued further because the benefits are underestimated; however, overestimation is bad as well.

Based on the cost–benefit analysis, technology may be selected. Taking this approach, a 'technology-follows-business' strategy is followed. A challenge in this stage is to select the most appropriate operationalization for every situation; however, the choice for one or another technology may be driven by, for instance, the 'fanciness' of a given sensor, its ubiquitous availability, a designer's particular expertise with it, *etc.*, which may not represent the most appropriate technology. Therefore, we suggest having an interdisciplinary team for decision-making because the technological alternatives should be evaluated from various perspectives including monetary assessments as well as non-monetary ones (*e.g.,* corporate image, first-mover advantages, possibility for pervasive embedding, user acceptance). Besides having a technical and a financial expert, we suggest, for the diabetes system, to draw on the expertise of a diabetes specialist (*i.e.,* physician and/or therapist) and a diabetic. In particular, it would be beneficial to consider a diabetic who is very accepting of new technology and one who is less so.

Finally, having selected the technology, the team can proceed with the implementation. Typically, and also confirmed by our interviewed designers, this stage of the process is not as linear as one might desire. Accordingly, going back to earlier stages in the process for corrective actions may be necessary.

7. Application of our 'ideal process' in practice

To manage the high level of complexity involved in designing intelligent systems, we proposed the 'ideal process' as a guide and framework for application in practice to plan for, design, and implement intelligent systems. It divides the design project into multiple stages, thus, simplifying the whole process. Thereby, the proposed process highlights the important phases that are essential to consider for designers of intelligent systems. The phases serve as a guide to the design activity and provide a flexible but consistent way to conduct design projects at the level matching the scope of the project.

In this section, we walk through the considerations of how to use the 'ideal process' for a concrete project example, which we then use to generalize from. The first question to address is what the project is about. Our simple – but non-trivial – example is to develop a messenger that sends out notifications to all members of a particular institute that are currently located in a specific lab on campus. A concrete example for such a message is, "Please tell me the phone number that I wrote down in the upper right corner of the whiteboard in the lab". The goal of the project seems to be more or less clear, but still needs to be refined to allow for a manageable scope for a state-of-the-art analysis. For this reason, the goal of the project needs to be streamlined first. Questions to be answered include: Why it is important that the message recipients are at the particular location? Is it the location or some other parameter that defines the addressees? For our example project, the notifications should only go to 'present' users (i) for reaching those that can fulfill a task on campus, and (ii) for not disturbing other institute members for which the message is not relevant. Based on this statement, we can derive that it is not specifically the users' location but their potential capability to fulfill a certain task that is relevant for the system. The ideal addressee for the whiteboard example above has access to the whiteboard and is capable of reading the requester's handwriting. This information is critical for the requirements specification and is also important for the fine-tuning stage of the process.

Probably, the following questions would also come up in a discussion of the designers and project owners: Should the system be a mobile solution? And do the addressees currently use specific

Table 6
Excerpt for context elements fine-tuning.

Step in the process	Context element	Judgment	Comments
Relevance	Time	Relevant	Therapy is scheduled by time
	Location	Maybe relevant	At home, at work, out of home will need different considerations for therapy reminders
	Identity	Maybe relevant	Only authorized people may be advised by the system
	Demographics	Relevant	Sex and age for calculation of required insulin
	Psychological predisposition	Not relevant	Some people are afraid of needles; but the system may not in a position to consider this
	Legal circumstances	Not relevant	Maybe needles cannot be disposed in normal rubbish pin; system will not consider that
	Degree of formality	Relevant	Insulin therapy in public or in business environment is not widely accepted
	Presence of people	Relevant	Unknown persons should not be present when taking insulin
	Biophysical conditions	Relevant	Influences lifestyle and, thus, therapy schedule indirectly; heart rate is related to activity level
	Habits	Maybe relevant	Influence therapy schedule
	Preferences	Maybe relevant	Lifestyle determines schedule, routines, and variations of these
	Connectivity	Maybe relevant	For up- and download of information; connection to other devices required for therapy (*e.g.,* glucometer)
	Task	Relevant	Important for alert frequency for reminders
	Social environment	Maybe relevant	Related to social acceptance of therapy treatments
Combination	Presence of people + degree of formality + social environment habits + preferences		
Precision + accuracy	Presence of people + degree of formality + social environment	Yes/no Very formal, formal, somewhat formal, not formal	
	Habits + preferences	May be acquired by questionnaire once	
	Time	Up to 30 min precision	
	Location	At home, at work, out of home	
	Identity	Estimation sufficient	
	Demographics	Male/female, age in 5-years scale may be acquired by questionnaire once	
	Biophysical conditions	Estimation in real-time for current situation for reminder	
	Connectivity	Yes/no	
	Task	Blood sugar measurement, insulin injection, documentation, other	
Level of relevance	Presence of people + degree of formality + social environment habits + preferences	Should have	
	Time	Must have	
	Location	Nice to have	
	Identity	Nice to have	
	Demographics	Must have	
	Biophysical conditions	Must have	
	Connectivity	Should have	
	Task	Must have	
= Shortlist	Time	Must have	
	Demographics		
	Biophysical conditions		
	Task		
	Presence of people + degree of formality + social environment	Should have	
	Habits + preferences		
	Connectivity		
	Location	Nice to have	
	Identity		

related devices? However, as the project goal is clear and manageable and the required resources of the project are rather minor, we suggest postponing answering these questions to a later stage in the process (*i.e.*, constraints evaluation and/or technology selection).

For being able to specify other functional requirements, a state-of-the-art analysis seems essential. What has already been solved? Are solutions available (*e.g.,* commercially, as open source, from internal projects)? Are reusable modules or code available? Source for information may be research documentation, patents, solutions on the market, or prior or concurrent internal work. If additional information is necessary (*e.g.,* about user preferences, user behavior, intentions to use), running a user study or expert interviews may also a viable source of information. As part of the state-of-the-art analysis of the example project, no existing system was found that fulfilled the

exact goal. However, work from the field of attention-aware systems was found to be useful as the relation between the primary task (*i.e.,* what a user is doing when receiving a message) and the secondary task (*i.e.,* reading the message) strongly impacts the outcome of both tasks (*e.g.,* Bailey et al., 2000; Fox et al., 2009). Furthermore, from the field of location-aware computing, a variety of approaches on how location may be determined (*e.g.,* Dobson, 2005; Loke, 2006) were found to be useful. Such an analysis could lead to the decision not to build a system, if there is already an existing solution. However, in this case, no such solution existed.

Inspired by the state-of-the-art analysis in combination with the specified project goal, the next step is to specify functional requirements. If something turns out not to be sufficiently clear at that stage, stepping back to the state-of-the-art analysis and/or running a supplementary study, may be required. For many projects, including our example project – as we will show later – the specified functional requirements will need adjustments later in the process.

As the 'ideal process' proposes, the next stage of the process is the fine-tuning stage, for which a fine-tuning procedure for context selection exists (see Fig. 8). Since 'relevance of a task' is still poorly defined at this stage of the project and various intentions for use among notification senders and acceptability among message receivers are conceivable, we recommend including potential users in this procedure. We suggest interviewing potential notification senders about how they would intend to use the notification system and iterate for all context elements that they come up with (*e.g.,* interruptibility of user, capacity for fulfilling the task, knowledge for fulfilling the task, location, congruence of tasks, importance of tasks, urgency of tasks), first for their relevance, then potential combination requirements, the precision and accuracy necessary for the element, and then categorize those for the level of relevance. The same procedure may be run through for potential message receivers, also interviewing them about their attitude toward receiving messages that might not be relevant to them (*e.g.,* high filtering costs due to being 'spammed' by unrelated tasks) or being missed out for notifications that would have been relevant (*e.g.,* feeling ignored by a manager or feeling privileged by not having to perform certain tasks). As an additional step, we suggest consolidating all context elements identified by users (*e.g.,* role of the message's sender, frequency of notifications) and iterating with individual users through the four steps of context analysis (*i.e.,* relevance, combination, precision and accuracy, as well as level of relevance) for each of the context elements. The rationale behind this additional step is that other users' ideas might inspire users and they may even come up with additional ideas for relevant context elements. If this additional step of context analysis seems to bring up a lot of additional ideas, repeating this step may be advisable. The result of the fine-tuning process may, as already mentioned, lead to refining the functional requirements. Maybe it is even necessary to go back to state-or-the-art analysis to see if comparable work exists on crucial context elements that had not considered before the fine-tuning procedure. In our example project, opening up for task relevance and presence at the lab instead of focusing on tracking users' location inspires for considering various additional and/or alternative context elements, such as the current availability of a user, which leads to a refinement of the functional requirements to allow for this aspect (instead of focusing on location).

In our example project, the constraints evaluation is performed in a sequence of informal brainstorming sessions of the design team and more formal enquiries to responsible people in the organization such as the consultation of the legal department for privacy-related issues for tracking context elements such as the users' location. The initially considered technical constraint to go for a mobile solution only is later dropped, since the inspection of the lab clarified that desktops are available on site. As a result, for our example project, the cost–benefit analysis should later in the process clarify whether a desktop solution is a viable alternative to a mobile one. The constraints evaluation stage, however, reminds us to discuss all kinds of constraints. In our example project, the design team – inspired by the constraints evaluation module overview (Fig. 9) – discusses constraints related to (a potential lack of) user acceptability. The team decides to run user studies with various prototypes to test the users' attitude toward the respectively implemented technologies (*e.g.,* pressure sensors on chairs to know whether a user is at his or her desk, tracking what software is currently active on a user's desktop computer, sensors on keyboards for measuring a user's skin conductance level to determine his or her stress level). For the purpose of shortlisting potential prototypes, the team performs a cost–benefit analysis, for which they have to make several assumptions for the benefits, as there are no results yet on the users' attitudes toward the respective technologies. For the analysis, the team decides to use a matrix on the costs and benefits and quantify both, costs and benefits, on a scale from 1 to 10, because due to the assumptions made, financial methods (*e.g.,* Net Present Value calculations) do not seem viable for the example project at this stage. Based on the matrix, the solution using software tracking stands out because of its high evaluation. As a result, the example project continues with a lightweight implementation of this one solution only to be tested among the users, along with a conjoint analysis that allows for comparing with other solutions on a conceptual level. The analysis shows that the software tracking solution is still the best among the considered solutions, although some users expressed privacy concerns. In a situation where there was a less clear result of the matrix evaluation, which is likely to occur for most projects, we would suggest a comparison of several lightweight prototype implementations – if feasible given constraints.

In the example project, the design team does not want to leave out other potentially good and more privacy-friendly solutions and decides to collect additional insights for potential later improvements of the system. A user acceptance study reveals that users in the lab consider the measurement of their skin conductance levels far less intrusive than tracking their software use behavior. Hence, in this case the user acceptance study inherently dictates the technology selection decision. A more formal financial assessment and planning of the final solution are, though, necessary to guarantee that the project is definitely within time and budget constraints. For other projects, other criteria (*e.g.,* first mover advantage, time to market) may be applicable.

The walkthrough the application of the 'ideal process' in the example project illustrates one of the various ways in which the process may be used for the design of an intelligent system. It demonstrates the flexibility of the process to be adjusted to the specific requirements of a project.

As each phase of the 'ideal process' uses the results of the previous one, the phases are interdependent. Depending upon the size and complexity of the project as well as the chosen development model framework, phases may be combined, iterated, or may even overlap.

For instance, when adopting the waterfall model, the phases of the 'ideal process' will be executed sequentially, requiring one phase to be completed, before the subsequent phase may start. The phase of fine-tuning context elements is likely to follow the guide rigidly for context identification and assessment, either in meeting with the design-team or by integrating responsible members of the project's client team. Thereby the design team may iterate the fine-tuning procedure for each context element identified as relevant one after the other, or each step of the procedure may be executed for all context elements at once, before proceeding with the next step of the procedure for all context elements.

When taking an iterative, team-based approach to development, the phases of the 'ideal process' may be considered when defining the smaller-scale tasks for project planning. Thereby, the guide for fine-tuning context elements may, for instance, be used in informal brainstorming sessions in team meetings. Alternatively, more structured settings may be established for fine-tuning.

8. Conclusion

The design of intelligent information systems is a complex task that goes beyond the capacities of existing system design process models. As a result, current practices are more or less unstructured and tend to use lots of brainstorming and *ad-hoc* approaches. However, as we know from traditional system design, systematic processes help to improve quality and productivity of both the process and the resulting system.

In this article, we have proposed a novel process for the design of intelligent systems to support system designers in making well-informed decisions in their design processes, going through the process efficiently, and coming up with better elaborated and designed intelligent systems. The process components of this novel 'ideal process' are mostly the same as designers reported to use in their current practice for the design of intelligent systems. The order of the components, though, is now more goal- and business-oriented, rather than explorative and/or technology-oriented. Interestingly, the ideal process shows only small deviations from the identified Archetype I of current design practices. This deviation, though, is crucial to allow for a business-value orientation. In an evaluation with experts in the field, the novel process was perceived as being very useful and was reported as being particularly applicable when the idea of the to-be-built system is sufficiently clear. Still, most of the system design experts that we interviewed argued that there will never be a 'one fits all process' and design practices will adopt variations of the proposed process to fit the requirements of different system purposes. Accordingly, we expect the proposed process and the guides to be used in a variety of ways, for instance, with respect to iterations or how it is being adapted to the design model, in which they are embedded. A specific example is that industry projects frequently involve designing a system for a specific technology; in such cases the technology and its constraints have to be considered at earlier stages in the process.

There are numerous advantages of adopting the proposed process. By following the process, system design of intelligent systems becomes more systematic and more efficient. The goal-orientation allows design teams to come up with systems that have a particular purpose that fit users' and/or businesses' needs compared to 'playful' projects that do not fulfill a particular purpose. Less experienced designers may have a steeper learning curve, as they have a guide to orient themselves and are less likely to replicate mistakes; for instance, the constraints analysis guide will help them in considering the major constraints and not missing crucial constraints which may lead to a dead end in the implementation. Our utility evaluation suggests using the proposed process particularly for incremental innovations in the realm of the design of intelligent systems. The scope of applicability of our proposed process is quite wide. The scenarios used to generate and generalize the process include context-adaptive smartphone applications (*e.g.,* icon rearrangement, ringtone profiling, activity recognition sharing), remote monitoring and controlling devices for facility management (*e.g.,* water monitoring, bridge inspections, location predictions) and prevention of dangerous events (*e.g.,* situation analysis in-car systems), health-supporting systems (*e.g.,* movement assessment for rehabilitation, smoking-cessation support), navigation systems (*e.g.,* support for visually impaired), and recommender systems (*e.g.,* music playlist generator).

This work contributes to the information systems and ubiquitous computing field in several ways. First, we analyzed and documented system designers' current approaches to the design of intelligent systems. Analysis reveals that these practices can be aggregated to a set of five process archetypes. In addition, we identified that the approaches to system design are rather unstructured, characterized by ideating and brainstorming, and rely to a large part on intuition and prior experience in the field. Second, our work provides system designers with a novel design process for intelligent systems. We have demonstrated that this process will be valuable for supporting system designers working on real-life projects in the field. Third, we proposed a procedural guide that allows systematizing the process of considering context elements in system design. While experienced system designers in the field tend to more or less follow this guide intuitively, experts indicated that this guide documents the process well and could therefore be considered as part of the 'fundamentals in ubiquitous computing' taught at universities.

A limitation of our work is rooted in our interview approach. The projects selected for the study may not be representative, for instance, because designers from the USA are more strongly represented in the sample than designers from other parts of the world; furthermore, there is a dominance of designers from universities compared to industry among the interview participants. Potentially the order of questions asked in the semi-structured interviews might have influenced the answers of the designers. For instance, presenting the 'ideal process' representations before the representation of the respective designer's project might have led to slight biases about the usefulness of the respective processes. In the future, implementation of the process designs in various projects and a comparison of resulting systems as well as of the designers' experiences may deliver in-depth insights into the process.

While it may seem that the overall 'ideal process' depicted in Fig. 7 could be applied to the design of any system, the design process was particularly elaborated for intelligent systems. Compared to the design of other systems, context as core functionality has to be dealt with additionally in the system design of intelligent systems. As we showed in our study, the approach to design activities concerned with context is currently unsystematic and such activities are strongly intermingled with other activities in the design process. Outside of intelligent systems, system design processes and models have existed for years and system designers are experienced with both the application of the processes and models as well as with the nature of those systems. However, similar experience with the nature of systems and the design of such is lacking for intelligent systems, as intelligent systems have a number of properties, which distinguish them from traditional systems. The purpose of the proposed 'ideal process' depicted in Fig. 7 is therefore the systematization of the process of designing intelligent systems. While the process may be applicable to the design of any system, its application to systems other than intelligent systems (as discussed in this paper) may not achieve similar quality results compared to adopting traditional system design approaches.

We encourage researchers to further evaluate the 'ideal process' in longitudinal studies, accompanying system design processes of real-world projects in various domains employing different developing approaches. Such efforts will add to the generalizability of our work's utility in various fields of application.

Acknowledgements

We are immensely grateful to Sandra Paulhart-Hebenstreit who supported our work by designing the intelligent supportive mobile system for diabetics.

References

Abowd, G.D., 2012. What next, Ubicomp?: Celebrating an intellectual disappearing act. In: UbiComp'12. ACM, Pittsburgh, PA.

Bailey, B.P., Konstan, J., Carlis, J.V., 2000. Measuring the effects of interruptions on task performance in the user interface. In: IEEE International Conference on Systems, Man, and Cybernetics. Nashville, TN. IEEE, pp. 757–762.

Bauer, C., 2012. A comparison and validation of 13 context meta-models. In: ECIS 2012. AIS, Barcelona Paper 17.

Bauer, C., 2014. A framework for conceptualizing context for intelligent systems (CCFIS). J. Amb. Intell. Smart Environ. 6, 403–417.

Bauer, C., Spiekermann, S., 2011. Conceptualizing context for pervasive advertising. In: Müller, J., Alt, F., Michelis, D. (Eds.), Pervasive Advertising. Springer, London, pp. 159–183.

Bauer, J.S., Newman, M.W., Kientz, J.A., 2014a. Thinking about context: design practices for information architecture with context-aware systems. In: iConference 2014. Berlin, Germany, pp. 398–411.
Bauer, J.S., Newman, M.W., Kientz, J.A., 2014b. What designers talk about when they talk about context. Hum.-Comput. Interact. 29, 420–450.
Bettini, C., Brdiczka, O., Henricksen, K., Indulska, J., Nicklas, D., Ranganathan, A., Riboni, D., 2010. A survey of context modelling and reasoning techniques. Pervasive Mobile Comput. 6, 161–180.
Beyer, H., Holtzblatt, K., 1998. Contextual Design: Defining Customer-Centered Systems. Morgan Kaufmann, San Francisco, CA.
Bishop, C.M., 2006. Pattern Recognition and Machine Learning. Springer, New York, NY.
Boehm, B.W., 1984. Software engineering economics. IEEE Trans. Softw. Eng. 10, 4–21.
Booch, G., 1982. Object-oriented design. Ada Lett. I, 64–76.
Bowen, G.A., 2008. Naturalistic inquiry and the saturation concept: a research note. Qual. Res. 8, 137–152.
Bradley, N.A., Dunlop, M.D., 2005. Toward a multidisciplinary model of context to support context-aware computing. Hum.-Comput. Interact. 20, 403–446.
Broll, G., Hußmann, H., Prezerakos, G.N., Kapitsaki, G., Salsano, S., 2007. Modeling context information for realizing simple mobile services. In: Mobile and Wireless Communications Summit 2007. Budapest, pp. 1–5.
Brown, P.J., Bovey, J.D., Chen, X., 1997. Context-aware applications: from the laboratory to the marketplace. IEEE Pers. Commun. 4, 58–64.
Castelli, V., Thomas, P., Bertone, R., 2009. Requirement engineering for context aware applications: a proceeding for context elements identification and representation. In: XV Congreso Argentino de Ciencias de la Computación (CACIC 2009), San Salvador de Jujuy. Argentina.
Chalmers, M., 2004. A historical view of context. Comput. Supported Cooperat. Work 13, 223–247.
Charnley, F., Lemon, M., Evans, S., 2011. Exploring the process of whole system design. Des. Stud. 32, 156–179.
Chen, Y., Atwood, M.E., 2007. Context-centered design: bridging the gap between understanding and designing. In: Jacko, J.A. (Ed.), Human-Computer Interaction, Part 1 (HCII 2007). Springer, Berlin, pp. 40–48.
Cheng, B.H.C., Lemos, R.d., Giese, H., Inverardi, P., Magee, J., Andersson, J., Becker, B., Bencomo, N., Brun, Y., Cukic, B., Serugendo, G.D.M., Dustdar, S., Finkelstein, A., Gacek, C., Geihs, K., Grassi, V., Karsai, G., Kienle, H.M., Kramer, J., Litoiu, M., Malek, S., Mirandola, R., Müller, H.A., Park, S., Shaw, M., Tichy, M., Tivoli, M., Weyns, D., Whittle, J., 2009. Software engineering for self-adaptive systems: a research roadmap. In: Cheng, B.H.C., de Lemos, R., Giese, H., Inverardi, P., Magee, J. (Eds.), Software Engineering for Self-Adaptive Systems. Springer, Berlin/Heidelberg, Germany, pp. 1–26.
Choi, J., 2007. Context-driven requirements analysis. International Conference on Computational Science and its Applications (ICCSA 2007). Springer, San Francisco, CA.
Choi, J., 2008. Context: from birth to design. ALPIT 2008. Dalian, Liaoning, China, pp. 347–352.
Chung, E.S., Hong, J.I., Lin, J., Prabaker, M.K., Landay, J.A., Liu, A.L., 2004. Development and evaluation of emerging design patterns for ubiquitous computing. In: DIS 2004, Cambridge, MA.
Corchado, J.M., Bajo, J., Tapia, D.I., Abraham, A., 2010. Using heterogeneous wireless sensor networks in a telemonitoring system for healthcare. IEEE Trans. Inf. Technol. Biomed. 14, 234–240.
Davidoff, S., Lee, M.K., Dey, A.K., Zimmerman, J., 2007. Rapidly exploring application design through speed dating. In: 9th International Conference on Ubiquitous Computing (Ubicomp 2007). Innsbruck, Austria. Springer, pp. 429–446.
Dey, A.K., 1998. Context-aware computing: the CyberDesk project. In: AAAI'98, Palo Alto, CA, pp. 51-54.
Dey, A.K., Abowd, G.D., 2000. Towards a better understanding of context and context-awareness. In: Workshop on The What, Who, Where, When, and How of Context-Awareness, Part of CHI 2000. The Hague, The Netherlands.
Dey, A.K., Abowd, G.D., Salber, D., 2001. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. Hum.-Comput. Interact. 16, 97–166.
Dobson, S., 2005. Leveraging the subtleties of location. In: 2005 Joint Conference on Smart Objects and Ambient Intelligence (sOc-EUSAI 2005). Grenoble, France. ACM, pp. 189–193.
Dobson, S., 2008. Co-design for context awareness in pervasive systems. In: Delaney, K. (Ed.), Ambient Intelligence with Microsystems: Augmented Materials and Smart Objects. Springer, New York, NY, pp. 297–307.
Dow, S., Saponas, T.S., Li, Y., Landay, J.A., 2006. External representations in ubiquitous computing design and the implications for design tools. In: 6th Conference on Designing Interactive systems (DIS 2006). University Park, PA. ACM, pp. 241–250.
Dowson, M., 1993. Software process themes and issues. In: 2nd International Conference on the Software Process. Berlin, Germany. IEEE, pp. 54–62.
Fleisch, E., Tellkamp, C., 2003. The challenge of identifying value-creating ubiquitous computing applications. In: Workshop on Ubiquitous Commerce, part of 5th International Conference on Ubiquitous Computing (UbiComp 2003). Seattle, WA.
Floch, J., Frà, C., Fricke, R., Geihs, K., Wagner, M., Lorenzo, J., Soladana, E., Mehlhase, S., Paspallis, N., Rahnama, H., Ruiz, P.A., Scholz, U., 2013. Playing MUSIC: building context-aware and self-adaptive mobile applications. Softw.: Pract. Exp. 43, 359–388.
Fortier, A., Rossi, G., Gordillo, S.E., Challiol, C., 2010. Dealing with variability in context-aware mobile software. J. Syst. Softw. 83, 915–936.
Fox, A.B., Rosen, J., Crawford, M., 2009. Distractions, distractions: does instant messaging affect college students' performance on a concurrent reading comprehension task? Cyberpsychol. Behav. 12, 51–53.
Friedewald, M., Raabe, O., 2011. Ubiquitous computing: an overview of technology impacts. Telematics Inform. 28, 55–65.
Gerpott, T.J., Berg, S., 2011. Determinants of the willingness to use mobile location-based systems: an empirical analysis of residential mobile phone customers. Bus. Inform. Syst. Eng. 3, 279–287.
Goldsby, H.J., Sawyer, P., Bencomo, N., Cheng, B.H.C., Hughes, D., 2008. Goal-based modeling of dynamically adaptive system requirements. In: 15th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS 2008). Belfast, Northern Ireland. IEEE, pp. 36–45.
Gray, P., Salber, D., 2001. Modelling and using sensed context information in the design of interactive applications. In: Little, M.R., Nigay, L. (Eds.), 8th IFIP Working Conference on Engineering for Human-Computer Interaction (EHCI 2001). Springer, Toronto, Canada, pp. 317–335.
Han, L., Jyri, S., Ma, J., Yu, K., 2008. Research on context-aware mobile computing. In: International Conference on Advanced Information Networking and Applications. GinoWan, Okinawa. IEEE, Japan, pp. 24–30.
Holzblatt, K., 2009. Contextual design. In: Sears, A., Jacko, J.A. (Eds.), Human-Computer Interaction: Development Process. CRC Press, Boca Raton, pp. 941–963.
Hoyos, J.R., Preuveneers, D., García-Monina, J.J., Berbers, Y., 2011. A DSL for context quality modeling in context-aware applications. In: Novais, P., Preuveneers, D., Corchado, J.M. (Eds.), 2nd International Symposium on Ambient Intelligence (ISAmI 2011). Springer, Salamanca, Spain, pp. 41–49.
Huang, H., Gartner, G., 2008. Using activity theory to identify relevant context parameters. In: Gartner, G., Rehrl, K. (Eds.), 5th International Conference on Location Based Services and TeleCartography. Springer, Salzburg, Austria, pp. 35–45.
Iivari, J., Iivari, N., 2011. Varieties of user-centredness: an analysis of four systems development methods. Inform. Syst. J. 21, 125–153.
ISO/IEC 12207:2008, 2008. Systems and software engineering: software life cycle processes.
Jaffal, A., Kirsch-Pinheiro, M., Le Grand, B., 2014. Unified and conceptual context analysis in ubiquitous environments. In: 8th International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM 2014). Rome, Italy. IARIA, pp. 48–55.
Johnson, K., Jimison, H.B., Mandl, K.D.M., 2014. Consumer health informatics and personal health records. In: Shortliffe, E.H., Cimino, J.J. (Eds.), Biomedical Informatics: Computer Applications in Health Care and Biomedicine. Springer, London, UK, pp. 517–539.
Kjeldskov, J., Paay, J., 2010. Indexicality: understanding mobile human-computer interaction in context. ACM Trans. Comput.-Hum. Interact. 17, 14:11–14:28.
Kofod-Petersen, A., Cassens, J., 2006. Using activity theory to model context awareness. In: Roth-Berghofer, T.R., Schulz, S., Leake, D.B. (Eds.), 2nd International Workshop Modeling and Retrieval of Context (MRC 2005). Springer, Edinburgh, UK, pp. 1–17.
Kohler, T., Fueller, J., Matzler, K., Stieger, D., 2011. Co-creation in virtual worlds: the design of the user experience. Manag. Inform. Syst. Q. 35, 773–788.
Kolos-Mazuryk, L., Poulisse, G.-J., van Eck, P., 2005. Requirements engineering for pervasive services. In: 2nd Workshop on Building Software for Pervasive Computing (part of OOPSLA 2005), San Diego, CA, pp. 18–22.
Landay, J.A., Borriello, G., 2003. Design patterns for ubiquitous computing. Computer 36, 93–95.
Lapouchnian, A., Yu, Y., Liaskos, S., Mylopoulos, J., 2006, October. Requirements-driven design of autonomic application software. In: Erdogmus, H., Stroulia, E., Stewart, D.A. (Eds.), 2006 Conference of the Center for Advanced Studies on Collaborative research (CASCON 2006), Toronto. Ontario, Canada, pp. 16–19.
Liu, Y., Xu, C., Cheung, S.C., 2013. AFChecker: Effective model checking for context-aware adaptive applications. J. Syst. Softw. 86, 854–867.
Loke, S.W., 2006. On representing situations for context-aware pervasive computing: six ways to tell if you are in a meeting. In: 4th Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops 2006). Pisa, Italy. IEEE, pp. 39–43.
Lu, H., Chan, W.K., Tse, T.H., 2006. Testing context-aware middleware-centric programs: a data flow approach and an RFID-based experimentation. In: 14th ACM SIGSOFT Symposium on the Foundations of Software Engineering SIGSOFT (FSE 2006). Portland, Oregon. ACM, pp. 242–252.
Lu, H., Chan, W.K., Tse, T.H., 2008. Testing pervasive software in the presence of context inconsistency resolution services. In: 30th International Conference on Software Engineering (ICSE 2008). Leipzig, Germany. ACM, pp. 61–70.
Mantei, M.M., Teorey, T.J., 1989. Incorporating behavioral techniques into the systems development life cycle. Manag. Inform. Syst. Q. 13, 257–274.
Miles, M.B., Huberman, A.M., 1994. Qualitative Data Analysis: An Expanded Sourcebook, 2nd ed. Sage Publications, Thousand Oaks, CA/London, United Kingdom/New Delhi, India.
Morón, M.J., Luque, R., Casilari, E., 2014. On the capability of smartphones to perform as communication gateways in medical wireless personal area networks. Sensors 14, 575–594.
Nilsson, S., Johansson, B., Jönsson, A., 2010. A holistic approach to design and evaluation of mixed reality systems. In: Dubois, E., Gray, P., Nigay, L. (Eds.), The Engineering of Mixed Reality Systems. Springer, London, UK, pp. 33–55.
Norman, D.A., Draper, S.W., 1986. User-Centered System Design: New Perspectives on Human-Computer Interaction. Lawrence Earlbaum Associates, Hillsdale, NJ.
Omasreiter, H., Metzker, E., 2004. A context-driven use case creation process for specifying automotive driver assistance systems. In: 12th IEEE International Requirements Engineering Conference (RE 2004). Kyoto, Japan. IEEE, pp. 334–339.

Oudshoorn, N., Rommes, E., Stienstra, M., 2004. Configuring the user as everybody: gender and design cultures in information and communication technologies. Sci., Technol. Hum. Values 29, 30–63.
Oulasvirta, A., 2004. Finding meaningful uses for context-aware technologies: the humanistic research strategy. In: SIGCHI Conference on Human Factors in Computing Systems (CHI 2004). Vienna, Austria, pp. 247–254.
Pang, Z., Chen, Q., Han, W., Zheng, L., 2012. Value-centric design of the internet-of-things solution for food supply chain: value creation, sensor portfolio and information fusion. Inf. Syst. Front. 17, 289–319.
Ranganathan, A., Al-Muhtadi, J., Chetan, S., Campbell, R., Mickunas, M.D., 2004. Middlewhere: a middleware for location awareness in ubiquitous computing applications. In: 5th ACM/IFIP/USENIX International Conference on Middleware. Toronto, Canada. Springer.
Razali, R., Zin, A.M., Mokhtar, M.R., Mirisaee, S.H., 2012. A systematic technology selection method for context-aware applications development. Res. J. Appl. Sci., Eng. Technol. 4, 3368–3374.
Rossi, G., Gordillo, S., Lyardet, F., 2005. Design patterns for context-aware adaptation. In: 2005 Symposium on Applications and the Internet Workshops (SAINT-W'2005). Trento, Italy. IEEE, pp. 170–173.
Ruiz-López, T., Noguera, M., Rodríguez, M.J., Garrido, J.L., Chung, L., 2013. REUBI: a requirements engineering method for ubiquitous systems. Sci. Comput. Program. 78, 1895–1911.
Sama, M., Elbaum, S., Raimondi, F., Rosenblum, D.S., Wang, Z.M., 2010. Context-aware adaptive applications: fault patterns and their automated identification. IEEE Trans. Softw. Eng. 36, 644–661.
Sama, M., Rosenblum, D.S., Wang, Z.M., Elbaum, S., 2010. Multi-layer faults in the architectures of mobile, context-aware adaptive applications. J. Syst. Softw. 83, 906–914.
Sandkuhl, K., Borchardt, U., 2014. How to identify the relevant elements of "context" in context-aware information systems? In: 13th International Conference on Perspectives in Business Informatics Research (BIR 2014). Lund, Sweden. Springer, pp. 290–305.
Scacchi, W., 2001. Process models in software engineering. In: Marciniak, J.J. (Ed.). Encyclopedia of Software Engineering, 2nd ed. Wiley, New York, NY.
Schilit, B.N., Adams, N., Want, R., 1994. Context-Aware Computing Applications, WMCSA 1994. IEEE, Santa Cruz, CA, pp. 85–90.
Schilit, B.N., Theimer, M.M., 1994. Disseminating active map information to mobile hosts. IEEE Netw. 8, 22–32.
Schmidt, A., Beigl, M., Gellersen, H.-W., 1999. There is more to context than location. Comput. Graph. 23, 893–901.
Serral, E., Valderas, P., Pelechano, V., 2009. Towards the Model Driven Development of context-aware pervasive systems. Pervasive Mobile Comput. 6, 254–280.
Sheng, Q.Z., Benatallah, B., 2005. ContextUML: a UML-based modeling language for model-driven development of context-aware web services. In: 4th International Conference on Mobile Business (ICMB 2005). Sydney, Australia, pp. 206–212.
Silberer, G., Schulz, S., 2012. Mobile Customer Relationship Management (mCRM): Constraints and Challenges, E-Marketing: Concepts, Methodologies, Tools, and Applications. Information Resourcces Management Association, Hershey, PA, pp. 324–340.
Sitou, W., Spanfelner, B., 2007. Towards requirements engineering for context adaptive systems. In: COMPSAC 2007, Beijing, pp. 593-600.
Song, W., Zheng, P., Zhang, L., 2012. Improve working efficiency by office automation system. Leadersh. Manag. 49, 9721–9723.
Tarasewich, P., 2003. Towards a comprehensive model of context for mobile and wireless computing. In: AMCIS 2003, Tampa, FL, pp. 114-124.
Tsai, P.-H., Liu, J.S., 2014. Intelligent tools for reducing medication dispensing and administration error. In: Gibbons, J., MacCaull, W. (Eds.), Third International Symposium, FHIES 2013 (Revised Selected Papers). Springer, Macau, China, pp. 1–21.
van der Zanden, G, 2008. Requirements engineering for context-aware applications. In: 9th Twente Student Conference on IT. Enschede, The Netherlands.
Walls, J.G., Widmeyer, G.R., El Sawy, O.A., 1992. Building an information system design theory for vigilant EIS. Inform. Syst. Res. 3, 36–59.
Wang, Z., Elbaum, S., Rosenblum, D.S., 2007. Automated generation of context-aware tests. In: 29th International Conference on Software Engineering (ICSE 2007). Minneapolis, MN. IEEE, pp. 406–415.
Weiser, M., 1991. The computer for the 21st century. Sci. Am. 265, 94–104.
Xu, C., Cheung, S.C., 2005. Inconsistency detection and resolution for context-aware middleware support. In: 13th ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE 2005). Lisbon, Portugal. ACM, pp. 336–345.
Xu, C., Cheung, S.C., Chan, W.K., Ye, C., 2010. Partial constraint checking for context consistency in pervasive computing. ACM Trans. Softw. Eng. Methodol. 19, 1–61.
Xu, C., Cheung, S.C., Ma, X., Cao, C., Lu, J., 2012. ADAM: identifying defects in context-aware adaptation. J. Syst. Softw. 85, 2812–2828.
Yin, R.K., 1989. Case Study Research: Design and Methods, 5th ed. Sage Publications, Newbury Park, CA.
Zhang, K., Katona, Z., 2012. Contextual advertising. Mark. Sci. 31, 980–994.

Christine Bauer is an Assistant Professor at the Department of Information Systems & Operations at Vienna University of Economics and Business, Austria. She holds a Doctoral degree in Social and Economic Sciences and a Master's degree in International Business Administration, both from University of Vienna, Austria, and a Master's degree in Business Informatics from Vienna University of Technology, Austria. Her research focuses on ubiquitous computing and human–computer interaction.

Anind K. Dey is a Full Professor and Director of the Human-Computer Interaction Institute at Carnegie Mellon University, and is director of the Ubicomp Lab. He conducts research at the intersection of human–computer interaction, machine learning and ubiquitous computing, and has published over 100 papers in these areas. He serves on the editorial board of IEEE Pervasive and the Personal and Ubiquitous Computing Journal, among others.