

Research Article

A Method for Designing Physical User Interfaces for Intelligent Production Environments

Pierre Taner Kirisci ^{1,2} and Klaus-Dieter Thoben ^{1,2}

¹University of Bremen, Faculty of Production Engineering, Bagdasteiner Str. 1, 28359 Bremen, Germany

²Bremer Institut für Produktion und Logistik GmbH (BIBA), University of Bremen, Hochschulring 20, 28359 Bremen, Germany

Correspondence should be addressed to Klaus-Dieter Thoben; tho@biba.uni-bremen.de

Received 5 March 2018; Revised 17 June 2018; Accepted 25 July 2018; Published 16 August 2018

Academic Editor: Thomas Mandl

Copyright © 2018 Pierre Taner Kirisci and Klaus-Dieter Thoben. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Physical user interfaces with enhanced interaction capabilities are emerging along with intelligent production environments. In this manner, we pose the question if contemporary design methods and tools are sufficient for the design of this new breed of user interfaces, or if there is rather a need for more efficient design methods and tools. The paper is initiated with a discussion about the need for more sophisticated physical user interfaces with enhanced capabilities for interacting in intelligent production environments. Based on this idea, we derive several functional and nonfunctional requirements for a suitable design method, supporting the conceptualisation of physical user interfaces in the early phases of product development. Hence, we suggest a model-based design method, which incorporates a comprehensive context model and modelling tool, applicable to intelligent production environments. In order to demonstrate the feasibility of the design method, we further conduct a validation and evaluation of the functional modelling tool, based on an industrial use case, in cooperation with design experts. In the final section of the paper, we critically discuss the key characteristics of the design method and thus identify potential issues for future improvement.

1. Introduction

There exists a need for design methods that support the conceptualisation phase of physical user interfaces. This particular early design phase requires techniques that are capable of ensuring that user context and requirements are adequately integrated into the entire design process from the very beginning [1]. The challenge is particularly evident when considering interaction within complex environments such as intelligent production environments, as this would require physical user interfaces with more sophisticated technical properties. These can include advanced input and output functionalities, wireless communication, and sensing techniques. The challenge of conceptualising physical user interfaces with suitable design methods becomes clearer when discussing intelligent production environments focussing on service processes as a case study in more detail, as outlined in Figure 1. In intelligent production environments, physical artefacts such as machinery, control devices, products, and vehicles increasingly possess enhanced technical capabilities.

These can extend from multimodal interaction possibilities to powerful processing capabilities. From a technical point of view, this is due to the integration of sensors, microcontrollers, and telemetric units as the enabling technologies. In this way physical artefacts such as control devices, tooling equipment, vehicles, and robotic systems are upgraded with enhanced technical properties, i.e., transforming from passive objects to interactive products with enhanced capabilities [2–4]. Humans interacting in these environments are therefore exposed to a variety of complex systems.

In combination with mobile interaction devices, humans are empowered with new interaction opportunities. New interaction opportunities include multimodal and situated interaction where an appropriate combination of specific interaction channels (e.g., acoustic, haptic, visual, and tactile) is activated according to the intermediate context of the user (see Figure 1).

Conclusively, there is a need for highly customised mobile interaction devices, which go beyond the capabilities of contemporary mobile devices. Thus, input, output, and

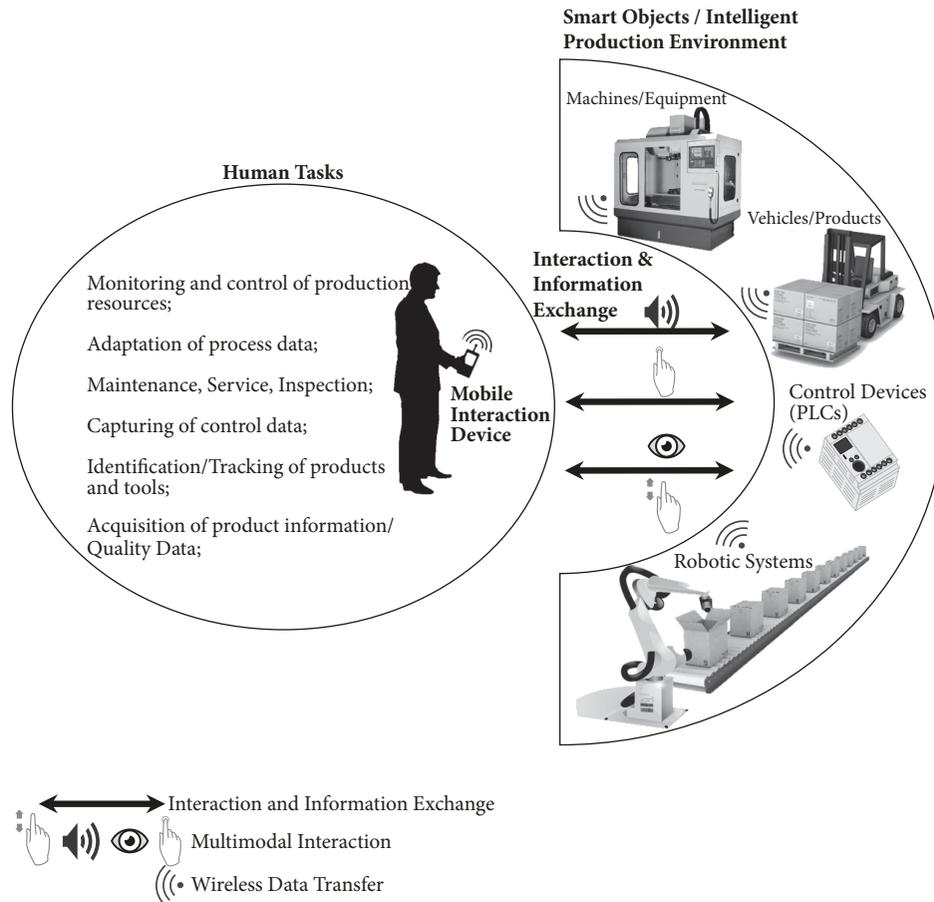


FIGURE 1: Multimodal interaction and information exchange in an intelligent production environment focussing on service processes.

communication devices have to fulfil a wide range of interaction requirements. The reason lies not only in the fact that they are applied in different locations and situations but also because they have to be able to seamlessly integrate with the working context of the user. As such, devices must not only enable a seamless integration in the technical environment but also integrate with the activity of the user. As illustrated in Figure 1, mobile interaction devices are intermediary physical user interfaces acting as a means of interaction between the human and the environment [5]. The mobile interaction device supports the users accomplishing their primary tasks through explicit and implicit interaction. Typical tasks where a support through mobile interaction devices is possible are service processes. Service processes represent preventive or troubleshooting tasks which are performed in order to support the production process [6]. Apart from this, service processes are characterised through activities with an increased extent of mobility and thus often are performed in varying locations [7, 8].

Due to the need of enhanced capabilities of mobile interaction devices, the design methods for mobile interaction devices will have to consider a wide spectrum of possible features as early as in the conceptualisation phase [9]. It becomes obvious from the industrial norm DIN EN ISO

9241-210:2011 (Human-centred design processes for interactive systems/Ergonomics of human-system interaction) as illustrated in Figure 2 (left side) that the early phases of design processes during conceptualisation particularly include the first 2 phases of the user-centred design process. These formerly represent the task and user analysis, i.e., the analysis of user context and specification of requirements. Concurrently these represent the first 2 phases of the product development guideline VDI 2221 [10] as illustrated in Figure 2 (right side). The use of VDI 2221 as a sector-independent procedural guideline is recognised and recommended for development and engineering tasks of technical systems and products. Explicitly in these early design phases, there is a lack of methods and tools which allow a sufficient documentation, analysis, and communication of context of complex work situations [5]. In relation to intelligent production environments, a major challenge for an appropriate design method is the sufficient incorporation of novel interaction concepts [11, 12] and user/usage requirements in the design process. Additionally, it has to be kept in mind that contemporary interaction devices such as control panels and operating device do not fully support potential interaction opportunities provided in intelligent environments (e.g., wireless and multimodal interfaces). Further on to guideline support,

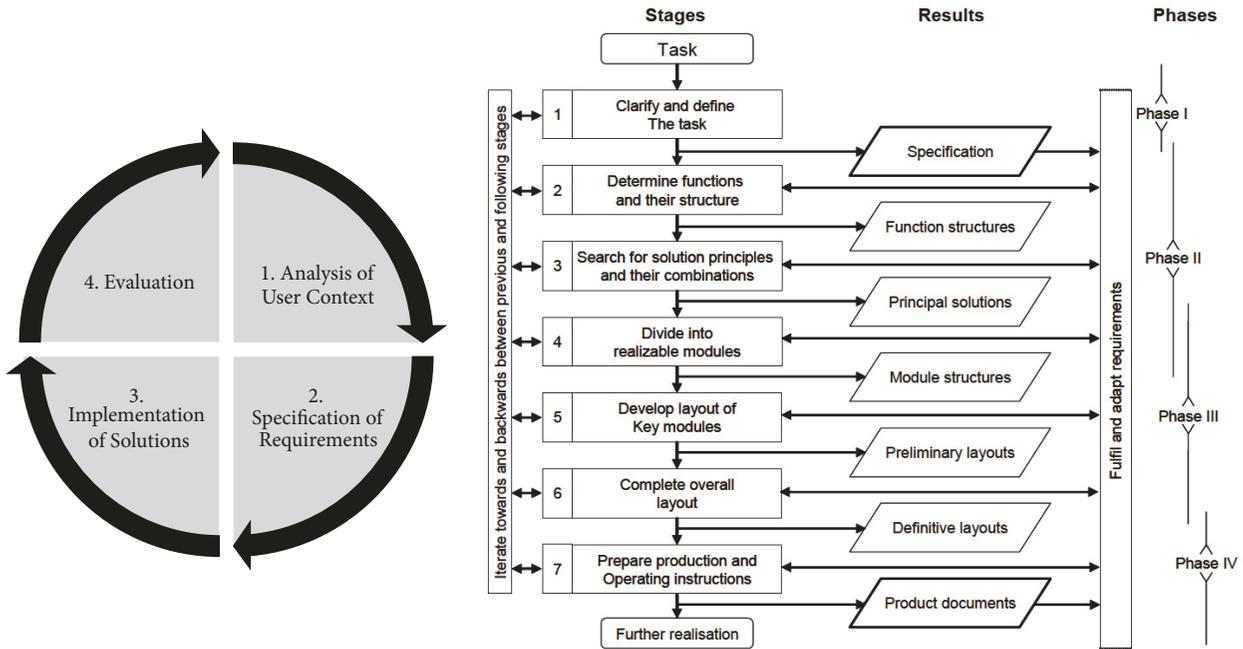


FIGURE 2: User-Centred Design Process according to DIN EN ISO 9241-210:2011 (left) and the product development process according to the guideline VDI 2221 (right).

existing guidelines and standards for user-centred design as various ISO guidelines such as the ISO 9241-210:2011 or the ISO TR 16982:2002 only provide a rough qualitative framework but do not consider the evolutionary steps of the product emergence [13, 14].

As a consequence, there is a need for appropriate methods and tools which not only support the efficient usage of the context in the conceptual design process but also facilitate the systematic description of the context in intelligent production environments [5]. Due to the high degree of complexity in intelligent production environments, it is not feasible for the designer to consider all contextual aspects of the working situation, without supportive methods and tools [15]. The specification of the contextual aspects can elementarily be described through models as suggested in model-based approaches for conceptual design of products and user interfaces [16–18, 18–20]. However, the essential and sufficient model elements that characterise an intelligent production environment are not adequately standardised. Further, if models were to include the different contextual elements necessary for describing intelligent production environments, it is still a contemporary issue to determine effective criteria and rules for mapping the model elements with one another [18]. Mapping of model elements refers to establishing descriptive or logical relations between the model elements as a basis for constructing rules how contextual elements are dependent upon each other. This is particularly relevant when the model is to be used as a data basis for a tool that supports the design process, as this would justify a connection between context and design information. Concurrently, the mapping of model

elements is crucial in the course of a tool for suggesting design recommendations for mobile interaction devices. Although the mapping of model elements has primarily been applied to challenges related to software user interface design (e.g., graphical user interfaces) it can be adopted to physical user interface design. This becomes obvious when logical relations between functionalities of user interfaces and the context of working situations are constructed. Under these circumstances, a relation to the “mapping problem” within the model-based design of software user interfaces can be established [21]. The mapping problem is described as the number of transformation rules which are required to transfer an abstract model to a concrete model [22]. While an abstract model can relate to the representation of user tasks, a concrete model represents the target platform where the user interface is to be implemented. The mapping problem itself originates from the modality theory and was described by Bernsen as “the general mapping problem” [23]: “For every information which is exchanged between the user and a system while performing a task, the input and output modalities are to be determined which offers the best solution for the representation and exchange of this information”. When transferred to the context of physical user interfaces it can be interpreted that the technical functionalities of a mobile interaction device (concrete model) have to be in line with the context of the working situation (abstract model). In conjunction with our area, this means that the design recommendations of the functionalities of a mobile interaction device should be in line with the context of the working situation, the environment, and respective user group.

2. Materials and Methods

2.1. Related Work. When user requirements are to be integrated into the early design phases, often a mixture of qualitative and quantitative methods is common such as user studies, field and user trials, market studies, or interviews [24]. Since the early product development steps as promoted during the sketch phase where user context is analysed and specified are often characterised by loose creativity and innovation, software tools that provide a systematic design support in the sketch phase are rarely in use [25]. The reason is that loose creativity and innovation are more commonly supported by conceptual methods such as 635 Method, brainstorming, or storyboards than with specific software tools [26]. Moreover, information technology is limited to a more indirect role, such as providing structures for a quick idea exchange as in mind-maps or the preparation of sketches and initial drawings like in graphic design software such as Adobe Illustrator. On the other hand, for idea generation creative techniques like brainstorming or 635 Method are well known and used in product development; thus they still require a great deal of subjective interpretation to allow the practitioner to translate the results into tangible design features [27]. Regarding product development methods with a special focus on integrating customer requirements, it has been confirmed that methods which enable an active customer integration, in comparison to methods where customers are integrated only passively and are more suitable for attaining customer knowledge within innovation development [28].

Having these aspects in mind, we discuss several design methods and tools in the following section. Notably, we have chosen methods and tools, which are in alignment with the criteria; they provide support in the early design phases and are fully or at least partially capable of integrating and considering user context. Thus, our intention was to uncover their strengths and limitations in accordance with the application to work situations and user interfaces in intelligent production environments.

2.1.1. Design Guidelines and Standards. Beside creative techniques, design guidelines and standards are well-established supportive tools for the early product development phases such as within the sketch phase and are nowadays in use in industry. As we have mentioned in the previous chapter, some of the most notorious guidelines are the ISO guidelines such as DIN EN ISO 9241-210:2011. This design standard is based on an iterative approach and runs through an unspecified number of design cycles until the final product is established. The decisions about which design techniques and tools are to be applied in the individual stages are among other aspects (e.g., company internal guidelines and checklists) dependent upon the preferences of the designer. The drawback of this kind of guidelines is however that they fail to consider all product variations and features [29]. Concerning mobile interaction devices, typical product features are represented through interaction features. This means that a guideline could be applicable to a mobile interaction device with conventional interaction possibilities. However, guidelines and standards are not sufficient in cases

when more specific and advanced devices are required, such as devices supporting multimodal interaction. In this respect, ISO guidelines are usually much generalised and rarely of quantitative nature and thus are only sufficient for a concrete technology design up to a certain extent. Moreover, they are often described in descriptive texts and tables, which is not very much in line with preferences of product developers for the presentation or visualisation of design recommendations. Often these guidelines are complemented by company internal guidelines, such as user interface style guides and reference lists [30]. Due to the close conceptual relevance of some mobile interaction devices to mobile smartphones, design and user studies of modern smartphones and interaction concepts can be complementarily considered [31–35].

Complimentary to the abovementioned aspects and limitations, integrating the user context in the design process as early as possible is a primary aim in any respect in order to obtain mobile interaction devices which are fully in line with the aspects of the situation where the envisaged device is to be used.

2.1.2. Model-Based User Interface Development (MBUID) Tools. When regarding interaction design practices in human-computer interaction (HCI), the majority of approaches focus upon supporting the design process of software user interfaces [36, 37]. Thus, supporting the design process in this domain means to provide methods, tools, or frameworks, which support the user interface developer to implement software code on predefined platforms. Significant efforts are necessary to implement the user interface since the user interface must maintain compatibility to a variety of different platforms and support different modes of interaction. Having this in mind, the consideration of reusability, flexibility, and platform independence in user interface development has led to the proposal of model-based user interface development (MBUID) methods and tools [38]. These have been extensively discussed during the past 20 years for various individual aspects of software systems and for different application domains [39]. An established reference framework for model-based user interface development was developed in the European CAMELEON Project [40]. The reference framework is based on 4 layers, while the upper layers consist of the abstract and concrete models and the lower layer represents the software code of the user interface [41]. Another model-based development tool, focussing on the implementation of wearable user interfaces (WUI), called WUI-Toolkit, was proposed in [42]. Further well-known techniques and tools in the MBUID community were proposed by Puerta, Vanderdonck, Luyten, Clerckx, and Calvary [38, 43–45]. However, in spite of the vast amount of research conducted in this area, only limited efforts have been spent so far in advancing model-based methods and tools for designing physical interaction components of industrial user interfaces [37]. Different from traditional mobile HCI where the physical platform or client often represented a GUI-based device such as a smartphone or tablet-PC, the physical platform can be any interactive physical object in

the environment with processing capabilities and support different modes of interaction. This means that physical user interfaces supporting information exchange with the environment must be tailored to the specific interaction needs. A good example is represented by human-computer interaction scenarios, where the need for systematically conceptualising adequate technologies for interaction support becomes obvious.

2.1.3. Tools for Prototyping Physical Interaction. Prototyping of physical user interfaces is highly relevant to understanding physical interaction between the prototyping subject, i.e., the hardware component that can be any type of input/output device, embedded system or sensor/actuator, and the human. The design task itself is likely to be successful if the designer has sound technical understanding of the physical user interface and at the same time the required interaction procedures. For this purpose, a number of toolkits have emerged supporting different facets of physical prototyping of user interfaces [46–52]. Most well-known physical prototyping toolkits in the HCI domain include “Shared Phidgets” a toolkit for prototyping distributed physical user interfaces and “iStuff mobile” a rapid prototyping framework for exploring sensor-based interfaces with existing mobile phones [49]. Complimentarily there exist conceptual frameworks such as for tangible user interfaces, a paradigm of providing physical form to digital information, thus making bits directly manipulable [46]. Independent of the embedded characteristics of these toolkits and frameworks, prototyping physical interaction is specifically dominated by challenges such as programming interactions among physical and digital user interfaces, implementing functionality for different platforms and programming languages, and building customised electronic objects [52]. While the abovementioned toolkits are more or less subject to these challenges, more recent efforts such as the “ECCE Toolkit” successfully address these issues by substantially lowering the complexity of prototyping small-scale sensor-based physical interfaces [52]. In spite of these approaches, it must be noted that these tools usually support prototyping of a specific type or category of physical user interfaces (e.g., sensors). More important is the fact that these tools do not sufficiently consider processes and task descriptions and their concurrent interrelations to appropriate user interface functionalities.

2.1.4. Inclusive Design Toolkits. An approach to support the design process of physical interaction devices (e.g., keyboards, displays) was developed by the Engineering Design Centre of the University of Cambridge [29, 46]. The approach is based on a web-based toolkit called “Inclusive Design Toolkit,” which can be considered as an online repository of design knowledge and interactive resources, leading to a proposal of inclusive design procedures and further more specific inclusive design tools, which designers may consult to accompany them through their product development process. In accordance with the definition of the British Standards Institution from 2005, inclusive design as such can be considered as a general design approach for products, services, and environments that include the needs of the

widest number of people in a wide variety of situations and to the greatest extent possible [53, 54]. Generally, inclusive design approaches are currently present in the consumer product sector. Although these approaches are more commonly applied to special user groups, from a technical point of view there exist no limits regarding its application to industrial products such as physical user interfaces. In this respect, inclusive design can be seen as progressive, goal-oriented process, an aspect of business strategy and design practice [46]. However, an application to industrial use cases is not feasible as the focus is upon fulfilling only the requirements of users but less upon the entire context of the application. Apart from this shortcoming, offered tools for inclusive design rely on the designers’ assumptions; thus assumptions have a risk of not being accurate, which can drive to incorrect assessments [29]. For supporting the early design phase of user interfaces the University of Bremen developed a set of design tools which support designers of physical user interfaces from the sketch to the evaluation phase with a virtual user model [55]. The results of the related EU project VICON (<http://www.vicon-project.eu>) can be obtained from the open source platform Sourceforge and from the project website. Although the focus was primarily set upon users with special accessibility needs, the approach extends inclusive design principles by considering not only the user requirements but also complementarily other contextual aspects such as environmental context. Additionally, the approach implies model-based options for an adaptation upon other context domains beyond the consumer product sector.

2.1.5. Design Patterns. Another design approach which can be applied to physical user interfaces is the design pattern approach as described by [56]. The theoretical background of design patterns was proposed as early as in the 1960s by Christopher Alexander [57]. In the 1970s Alexander developed a pattern language for an architectural design where 253 design patterns were proposed with universal character [58]. These allow a catalogue style description of patterns based on their properties. In later years the design pattern theory was adopted to the area of software development [59, 60] as well as extended to other subdomains of computer science such as ubiquitous computing and interaction design [61]. More recently design pattern approaches have been applied to the context of adaptive user interfaces [56, 62], although up until now there exist no complete collection of design patterns regarding the reusability of physical interaction components.

Through the analysis of the abovementioned design methods, approaches, and tools, we have unveiled a number of limitations and insufficiencies regarding their application to designing physical user interfaces for intelligent production environments. These are summarised in Table 1.

In Figure 3 the functional requirements are summarised in relation to the limitations identified in Table 1.

Although the definition of an appropriate context model will be discussed in a later section, it is reasonable to consider the role of emotional awareness in designing physical user interfaces. Integrating emotions in context for developing emotional aware systems have been particularly investigated in several research papers [63–68]. In [63] an approach is

TABLE I: User interface design methods and the limitations.

User Interface Design Methods	Limitations
Design Guidelines and Standards	(i) Descriptive and qualitative nature, not sufficient for detailed technology design (ii) Fail to consider all product variations and features (iii) Not in line with preferences of product developers for the presentation or visualisation of production processes design recommendations
Model-Based User Interface Development Tools	(i) Main focus upon software user interfaces (ii) Physical platform is predefined
Tools for Prototyping Physical Interaction	(i) No support in defining, configuring and integrating user context in the prototyping process (ii) Support of only certain categories of physical user interfaces and components
Inclusive Design Toolkits	(i) Considers only user context (ii) Major focus on consumer products Strong reliance on the designers' assumptions
Design Patterns	(i) No complete collection of design patterns regarding the reusability of physical interaction components

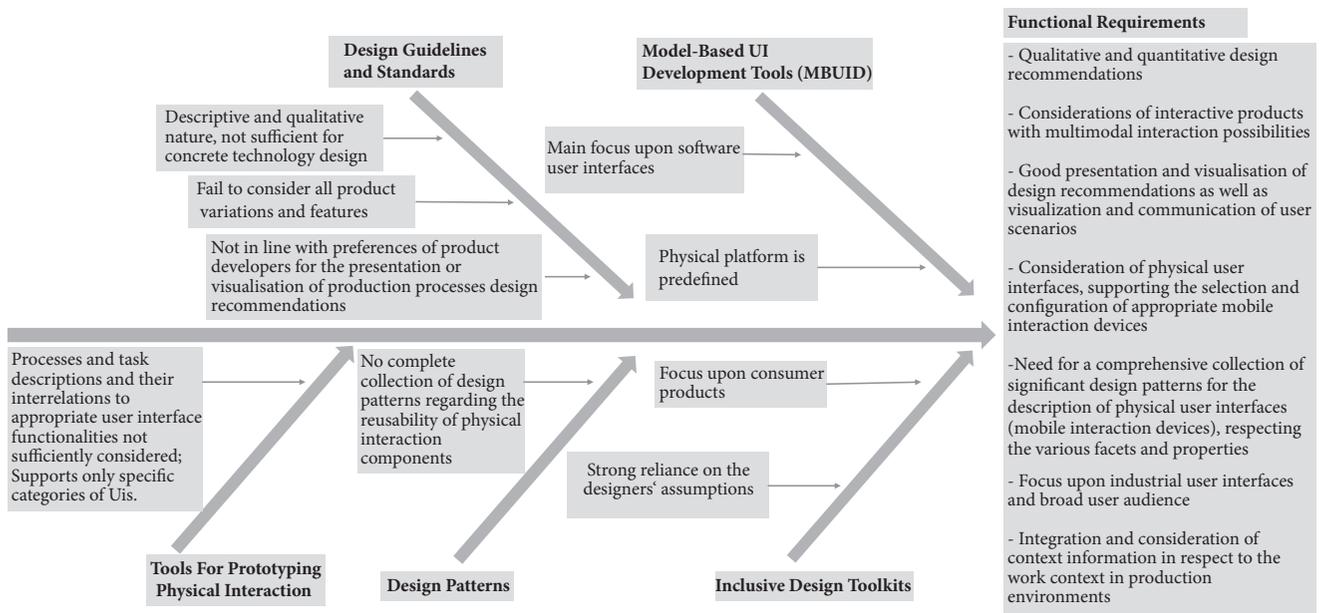


FIGURE 3: Limitations of user interface design methods and resulting functional requirements of an appropriate design method.

introduced for building shared digital spaces where emotional interaction occurs and is expressed according to the WYSIWIF (what you see is what I feel) paradigm. The concept focusses upon interpersonal communication use cases and, as such, ways of affective communication have been demonstrated through TUIs and component Phidgets as a means of a physical emotional communication device. In spite of the general notion that emotional awareness is more relevant to applications for consumers, it is complementarily recognised that emotional aware techniques can also provide value for collaborative work scenarios, e.g., as a means of improving interaction among group members. A comprehensive survey of mobile affective sensing systems has been conducted in [64]. A major focus was on breaking down and understanding the interrelation of the elements of sensing, analysis and application when designing affective sensing

systems. For this purpose, a component model for affective sensing has been proposed, expressing that the most crucial challenges still rely on understanding the link between people's emotions and behaviour in different contexts yielding in the need of more sophisticated emotional models. From a more practical point of view, the aggregation of emotional data across places over time, privacy, and energy awareness is seen as the most significant challenges for implementing affective sensing systems. When viewing contemporary technologies, facial recognition, voice analytics, eye tracking, VR and AR technology, biosensors, and sentiment analysis are considered as key enabling technologies for realising emotional awareness in physical user interfaces [64, 65]. Furthermore, these technologies are necessary for making emotions machine-readable and interpretable. The most recent research efforts from 2015 until today see the greatest

TABLE 2: Dependence between functional and nonfunctional requirements.

Functional Requirements (FR)	Non-Functional Requirements	Non-Functional Requirements for software ISO/IEC 9126-1
FR1-Qualitative and quantitative design recommendations on a more comprehensive level or quantitative design recommendations	Applicability, Inferability, and Universality	Functionality
FR2-Considerations of interactive products with multimodal interaction possibilities; Consideration of physical user interfaces, supporting the selection and configuration of appropriate mobile interaction devices	Inferability and Analysability	Functionality, Maintainability
FR3-Good presentation and visualisation of design recommendations as well as visualisation and communication of user scenarios	Applicability and Analysability	Reliability, Maintainability
FR4-Need for a comprehensive collection of significant design patterns for the description of physical user interfaces (mobile interaction devices), respecting the various facets and properties	Reusability and Extensibility	Usability, Maintainability
FR5-Integration and consideration of context information in respect to the work context in intelligent production environments	Adaptability	Portability
FR6-Focus upon industrial user interfaces and a broad audience of users in mobile working situations	Applicability	Reliability, Usability

benefits of integrating emotional context through affective sensing systems in consumer-based applications such as marketing and consumer applications, health and wellbeing, and entertainment. This tendency unveils that there exists a need for more comprehensive research efforts regarding the value and implications of emotional awareness in professional domains such as in service processes in intelligent production environments.

Finally, the limitations in Table 1 served as an essential basis for us in order to define requirements for a design method appropriate for conceptualising mobile interaction devices in intelligent production environments. As the envisaged design method should not only be declarative but also provide mechanisms, which fulfil certain functions with qualitative criteria, it is reasonable to distinguish between functional requirements and nonfunctional requirements. Likewise, the differentiation between functional and nonfunctional requirements is well-established practice in requirements engineering [26]. In this manner, the identified limitations in Table 1 provided us with a basis for deriving functional and nonfunctional requirements for a respective design method.

2.2. Design Method Requirements. In accordance with the limitations defined in Table 1, it was possible for us to relate a number of functional requirements for an appropriate design method. Concurrently, the identified limitations represent functional limitations of existing design approaches with respect to designing physical user interfaces for intelligent production environments. The functional requirements are therefore directly derived from the respective functional limitation and can be considered as requirements for specific functionalities that an appropriate design method has to fulfil in order to provide a sufficient support for conceptualising mobile interaction devices.

Nonfunctional requirements, on the other hand, can be seen as quality criteria for a set of functional requirements, which can be applied to infer concrete procedures of the design method. When adopting a model-based character for the design method, generally valid requirements for reference models such as reusability, universality, adaptability, and recommending ability can serve as an orientation for defining nonfunctional requirements [57, 58]. Likewise, a set of nonfunctional requirements, which at the same time correspond to the general properties of reference models, can be derived from the above-defined functional requirements. These are illustrated and interrelated in Table 2. Concurrently the identified nonfunctional requirements additionally can be related to some of the major qualities of a software according to ISO/IEC 9126-1 (norm for product quality in software engineering). These are specifically functionality, reliability, usability, maintainability, and portability, which are mapped to the nonfunctional requirements for and functional requirements (FR) of the envisaged design method. Emphasizing the relation of functional and nonfunctional requirements of the design method to quality attributes for evaluating software makes sense as it enables a standardised validation process through well-defined and recognised validation criteria.

Under these circumstances, the nonfunctional requirements in Table 2 illustrate that there exists a connection between the qualitative characteristics of the envisaged design method and the general properties of reference models (middle column). This aspect underpins that the design method should incorporate a model-based character. In other words, the basis for designing mobile interaction devices for intelligent production environments should be represented by a model, which describes and interrelates the context of intelligent production environments. Concurrently, the model can be used in order to infer design recommendations and present these in a comprehensive way. In the next

TABLE 3: Procedural guidelines for the design method.

Non-Functional Requirements	Procedural Guideline	Functional Requirements
Universality	Identification of necessary partial models and their description in the style of working situations within a reference concept	FR5-Integration and consideration of context information in respect to the work context in intelligent production environments
Applicability, Inferability, Universality	Inclusion of a partial model, which incorporates potential design recommendations and interrelates these to remaining contextual elements	FR1-Qualitative and quantitative design recommendations on a more comprehensive level or quantitative design recommendations
Applicability and Analysability	Easy set-up and handling of partial models as well as their visual presentation	FR5-Integration and consideration of context information in respect to the work context in intelligent production environments FR3-Good presentation and visualisation of design recommendations as well as visualisation and communication of user scenarios
Inferability and Adaptability	Integration of a mechanism for the configuration of context elements supported by a modelling tool	FR5-Integration and consideration of context information in respect to the work context in intelligent production environments
Analysability and Inferability	Realisation of a well-defined rule framework between partial models as well as the selection and implementation of an appropriate analytical technique for validating the consistency of logical rules	FR6-Focus upon physically unimpaired users in mobile working situations FR2-Considerations of interactive products with multimodal interaction possibilities; Consideration of physical user interfaces, supporting the selection and configuration of appropriate mobile interaction devices FR3-Good presentation and visualisation of design recommendations as well as visualisation and communication of user scenarios
Inferability	Description of the context model in a semantic language in order to infer new knowledge based on existing knowledge (open world assumption). The interpretation and assessment of rules have to be ensured in order to infer new knowledge.	FR2-Considerations of interactive products with multimodal interaction possibilities; Consideration of physical user interfaces, supporting the selection and configuration of appropriate mobile interaction devices

section of the paper, the identified requirements will be consulted in order to develop concrete procedures and define a conceptional framework in alignment to the design method.

2.3. Conceptual Design Framework. In the preceding chapter, we pointed out that a key property of the design method is model-based, which results in the need for a model-based design method. In respect to the identified functional requirements (FR1-FR6), it is constructive that we consider models or modelling principles as one of the major means for attaining the functional requirements. Modelling principles likewise imply techniques and tools for establishing and combining information. In this case, the information is represented in context elements, which is particularly relevant for working situations in intelligent production environments. Through the qualitative analysis of the functional and nonfunctional requirements, it is possible for us to elaborate several tangible procedures, which can be directly incorporated into procedural guidelines of the design method. These are described in Table 3.

The compilation of the procedures in the table provides further insight regarding scope and structure of necessary context elements. As such, we propose that the sufficient

description of the context which, e.g., includes the working situation, the environment, and potential design recommendations can be described with specific partial models.

The aim at this point is described as identifying the scope and type of partial models that are sufficient for describing a context in intelligent production environments. The identification of relevant context elements is based upon a qualitative analysis regarding the categorisation of context for intelligent production environments. In this analysis, we identified necessary context elements and extended these into a context model for intelligent production environments. The foundation of the model is loosely related to Schmidt's model for context-aware computing [69]. The model of Schmidt proposes a context feature space for context-aware mobile computing that incorporates human and physical environment factors. On a high level, these factors can be interrelated to the functional requirement FR5—integration and consideration of context information in respect to the work context in intelligent production environments. Having this in mind, Schmidt's model was considered as a rudimentary basis for specifying the context in intelligent production environments. Figure 4 illustrates an extended context model for intelligent production environments, where Schmidt's

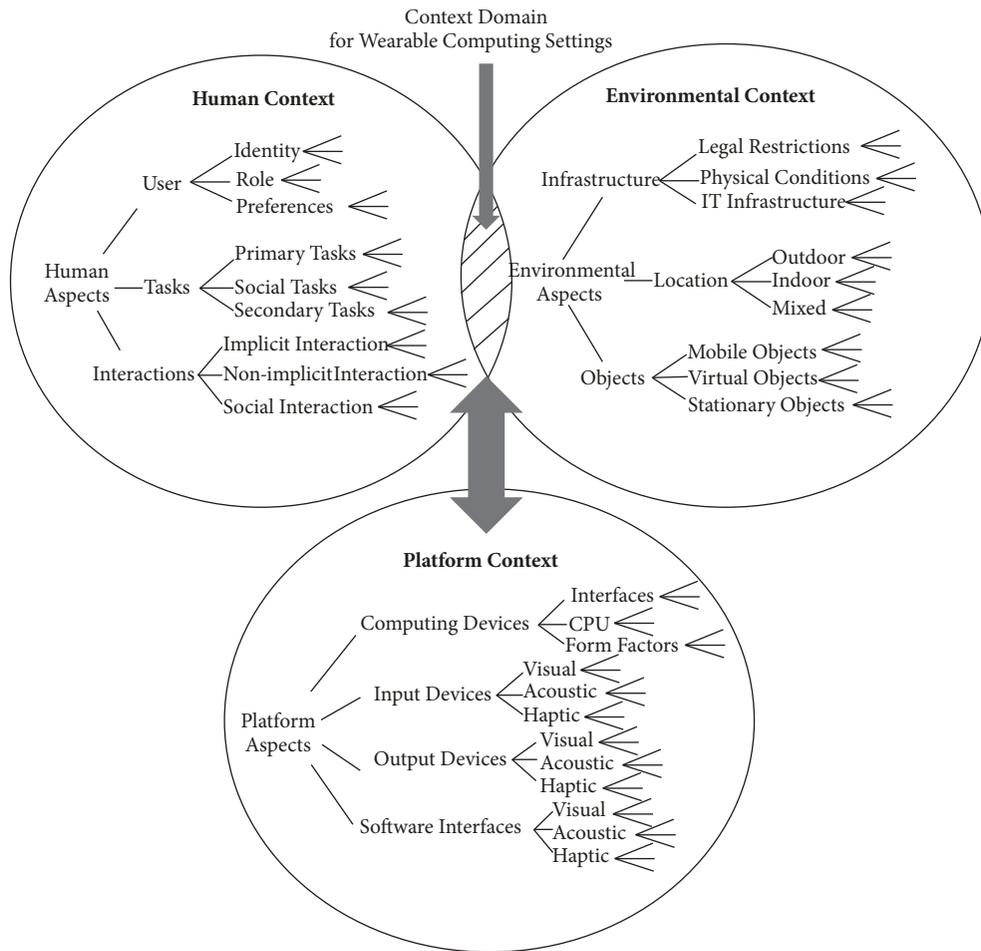


FIGURE 4: An extended model for the context in intelligent production environments.

model was used as a basis [8]. The extended context model focusses towards supporting the design of wearable computing systems in intelligent production environments. Likewise, wearable computing systems consist of configurable input and output devices and offer advanced interaction opportunities to the user. Accordingly, they represent mobile interaction devices in the broader sense. Pertaining to the model of Schmidt, we have maintained the differentiation between human and environmental context. These context elements are concurrently connected to the functionalities and properties of a potential mobile interaction device. In this scheme, human context is directly related to the role of the users, tasks, and interactions with the environment. All situations that can be captured with human senses are relevant here. The environmental context is defined through a type of context that may result from human context and that can be captured with the help of an intermediary system [70]. Thus, an intermediary system can be represented by a technical device that is capable of capturing environment data for instance through sensors. As such, the environmental context is a dynamic context like physical conditions of a working environment (e.g., light conditions, infrastructure,

and temperature), as well as the technical properties and artefacts of the environment.

The underlying idea of the extended model is based on the assumption that human and environmental context directly affects the type of interaction resource and interaction modality of the envisaged interaction device. Therefore a partial model describing the characteristics of potential interaction devices is necessary which is referred to as “platform context” in the extended context model in Figure 4. Notably, the area where human context and environmental context is connected over different instances can be regarded as the context domain of a mobile interaction device. The reason is that it represents the area where both human and environmental contexts are directly connected to the platform context. As an example, considering solely very bright light conditions in the working environment may lead to the design recommendation of a light adaptive display. However, when also considering the human context, e.g., that the primary task of the user requires full visual attention, the resulting design recommendation would be further constrained and consequently lead to an alternative design recommendation for the related platform. This

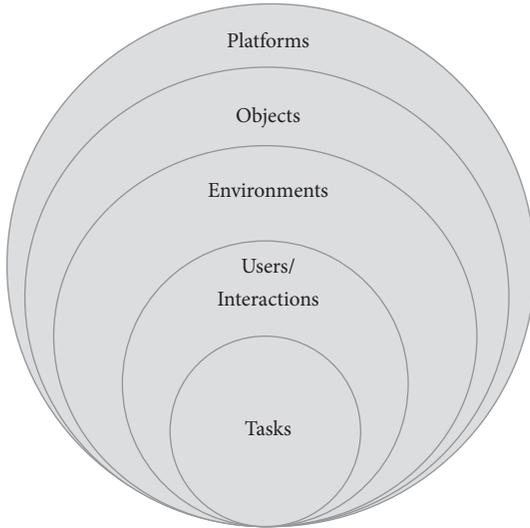


FIGURE 5: The context elements for an appropriate context model.

example shows that considering only human context is not sufficient to acquire a valid recommendation for the most appropriate platform. Moreover, we think it is necessary to consider all relevant contextual aspects and their mutual impacts in order to attain a valid design recommendation for the most appropriate mobile platform. Accordingly, we have consulted the extended context model in Figure 4 for identifying the most significant context elements for setting up an appropriate context model. For this purpose, it is constructive to cluster the context elements of the extended context model according to their correlation. Elements that have a high correlation can be united to a single context element. As a result, six context elements were abstracted which concurrently implies all context main and subelements of the extended context model, as highlighted in Figure 5. With the identified context elements, it is possible to specify the context model more comprehensively. Six context elements can be transferred into six individual partial models. From this perspective, the different context elements provide insight into the scope and type of possible partial models for the design method. However, when considering that an implementation of the context model is foreseen, the complexity of the model should be held as low as possible while maintaining functional requirements. This means that the fulfilment of the requirement universality of the context model yields in describing context elements in a higher level of abstraction. This prevents the consideration of interactions as a means of refinement/detailing of work tasks. Moreover, we propose to rather consider interaction constraints, interaction preferences, and exemplary interactions since these are likely to have a direct impact on the type of interaction resources. When the user interactions are reduced to interaction constraints of the user, the interaction model can be merged into the user model that results to one partial model. In detail, this means that six context elements are the basis for five partial models, namely, task model, user model, environment model, object model, and platform model.

In spite the fact that the six context elements in Figure 5 are sufficient for describing the context in an intelligent production environment, the model does not yet incorporate design recommendations that are necessary in order to gain qualitative design recommendations as a main function of the design method. In alignment with the model-based character, we find it reasonable that the context model should, therefore, include design recommendations as an additional context element resulting to an additional partial model. This additional partial model provides the necessary data basis for inferring design recommendations, which implies that there must be data relations between the design recommendations and all other remaining context elements. Strictly speaking, the recommendation model possesses data relations to all other partial models such as task model, user model, environment model, object model, and platform model. In this way, the extended context model can be leveraged to include seven context elements that are the basis for six partial models. The partial models as highlighted in the upper right side in Figure 6 include task model, user model, environment model, object model, platform model, and recommendation model. The first four partial models, namely, task model, user model, environment model, object model, and platform model, provide context information for potential aspects of a working situation in an intelligent production environment. When these are interlinked with the recommendation model, the initial basis is finally prepared for an overall, functional context model.

Structure and contents of the context model represent a formal collection of terminologies or in other words a terminological reference framework for a specific application domain. This view is in line with the notion of an ontology. Accordingly, it is legitimate to describe the partial models of the context model as an ontology. In this way, the model data is applied as a representation of the context. One of the main advantages of describing partial models in a semantic language like OWL (Web Ontology Language) is that already existing information is not negated when subjoining new information (Open World Assumption). Concurrently this is the necessary condition in order to be able to infer new knowledge. As a consequence, the six partial models can be realised on the basis of ontology classes.

Figure 7 highlights the data interrelations described in the data properties between the recommendation model and the task model of the overall context model. The example describes an inspection process (class) where testing (instance) is required. The description of the inspection task within the ontology implies the description of the required component and functionality of the interaction device as a data property. Thus, the functionality is described as input, output, and communication functionality, while the relevant component is described as input, output, and a communication unit. On this level, data relations to the recommendation model are established through the matching terms included in the descriptions of the data properties. Considering the recommendation class “implicit_interaction_identification_techniques”, the data relation to the task model that means the description of the required functionality and components for the corresponding recommendation are defined. The data

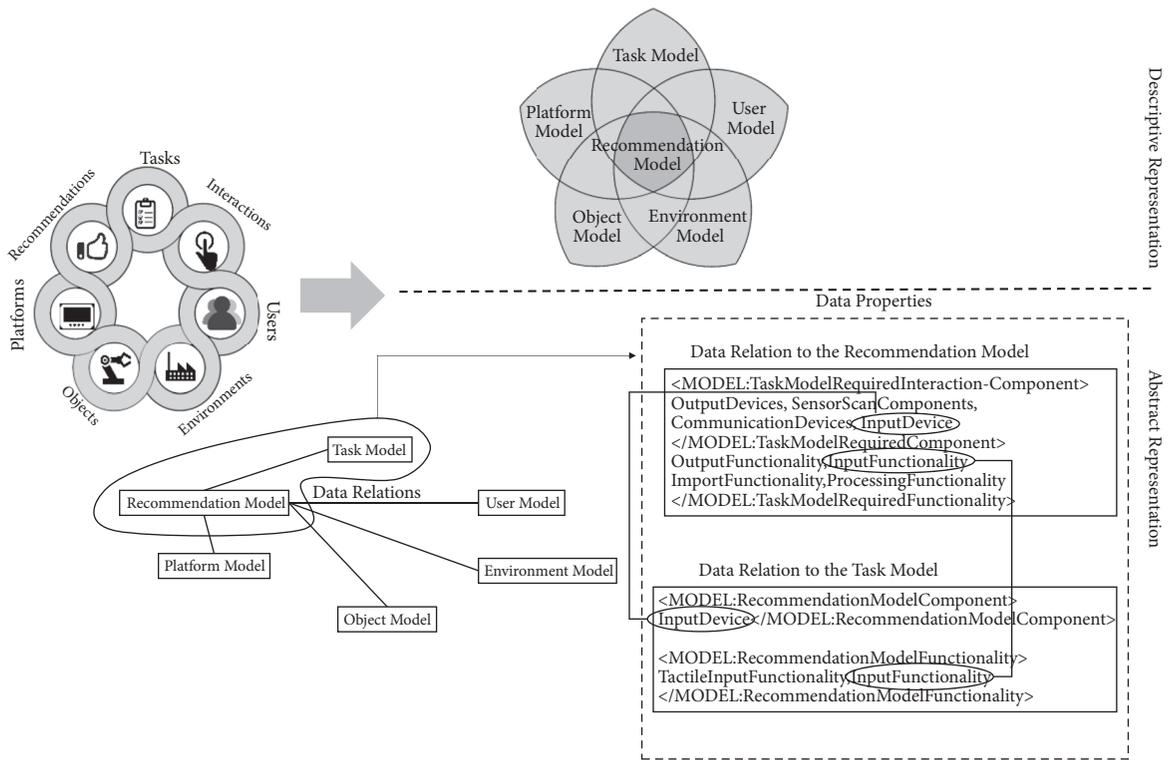


FIGURE 6: Ontology classes as an initial basis for the partial models and data relations between the recommendation model and remaining partial models.

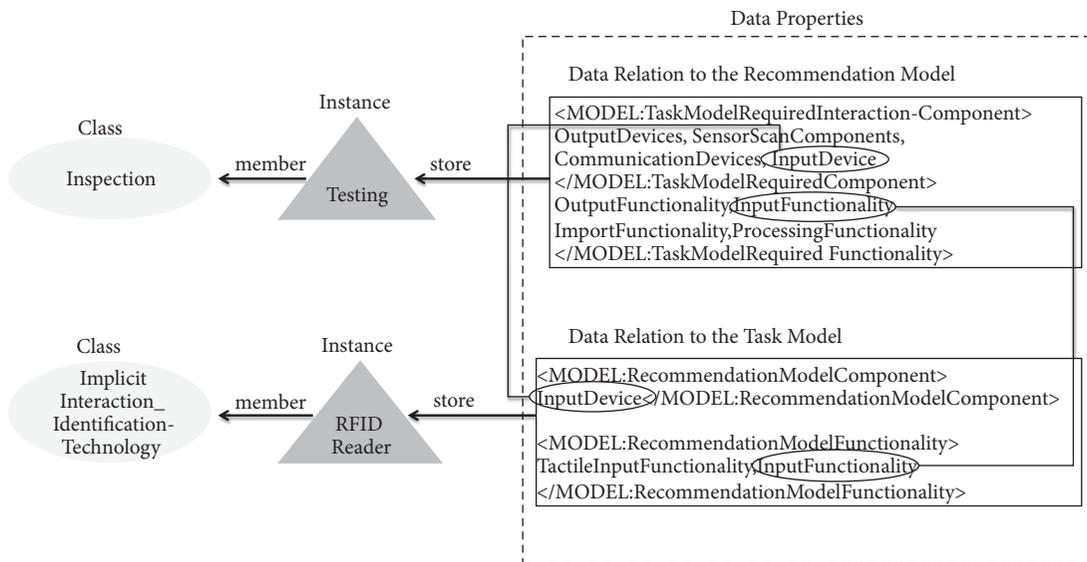


FIGURE 7: Example of data relations between the recommendation model and task model.

interrelation represents the basis for specifying logical rules with the support of a reasoning engine while a reasoner is able to identify data properties between the partial models that correspond to one another and infer a logical rule. As a result, an input device as a communication device, which supports implicit interaction (RFID reader), is recommended. The

relations of the remaining ontology classes such as user, environment, object, and platform also include descriptions of the data relations to the recommendation model. Vice versa, the recommendation model is defined by data relations to the remaining partial models. As such, these are handled in the same manner as described with the task model. As

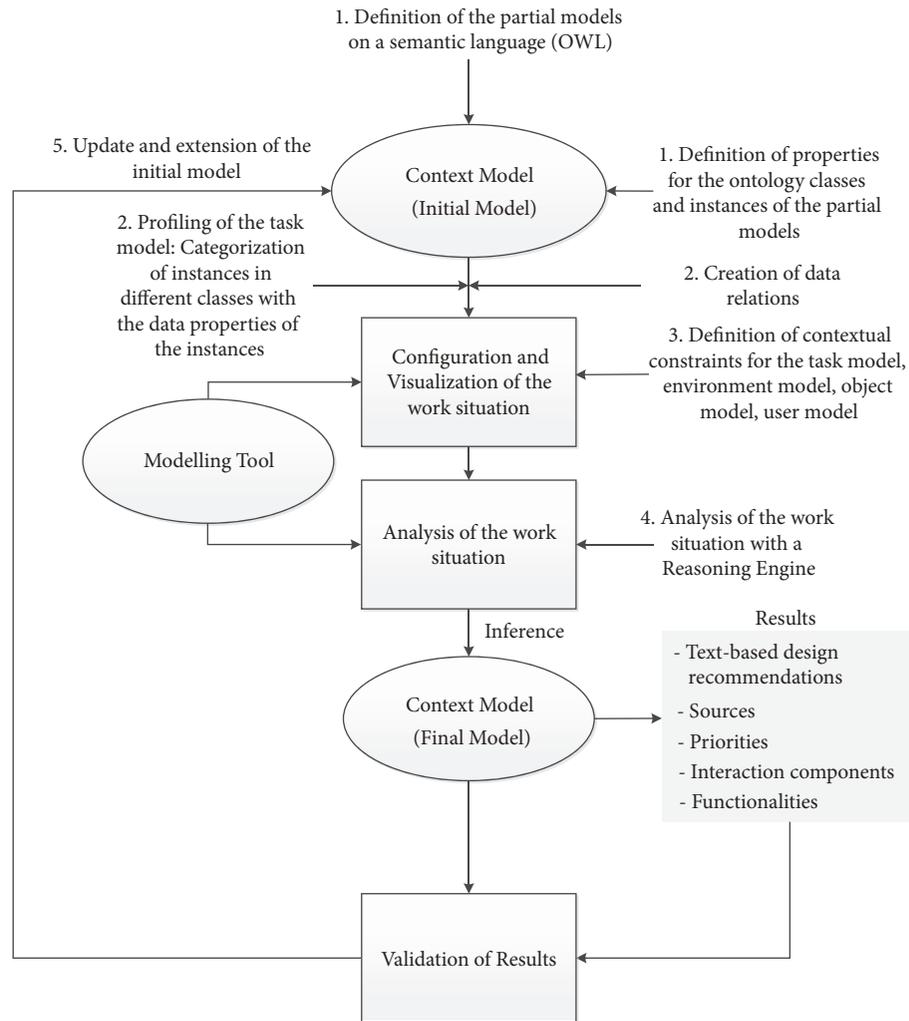


FIGURE 8: Procedural model for the design method.

a consequence, each recommendation (E1..En) is defined through data properties of the instances of the ontology classes.

2.4. Procedural Model for the Design Method. According to the procedural guidelines identified in the preceding section, we propose the corresponding procedure model in Figure 8. The procedure model is initiated by the procedure of constructing six partial models, yielding to a basic context model that combines all six partial models into an initial model. The requirement for setting up the initial model is described stepwise as follows:

- (1) Definition of partial models in a semantic language and definition of data properties (i.e., name, real value, and category) for all six ontology classes and instances. This implies manually defining the properties of the ontology classes and partial models.
- (2) Profiling of the task model and creation of data relations.

The task model should follow the predefined task definitions, which are configured to match mobile working situations in production environments. For this purpose, we have chosen a generic and standardised task description based upon the task catalogue for maintenance according to the DIN 31051 (German industrial standard for the fundamentals of maintenance). In this respect, profiling is a necessary step in order to pursue a task and activity selection. To achieve this, the instances (activities) are grouped into different classes (tasks) according to their data properties. At the same time, data relations between the instances and ontology classes are defined as a foundation for the generation of rules. Through completing these two steps, the requirement for the configuration of the working situation is prepared and the manual configuration of a working situation is conducted, which leads to the third step in the procedural model.

- (3) The configuration of the working situation. This step takes place manually with the support of a modelling tool.

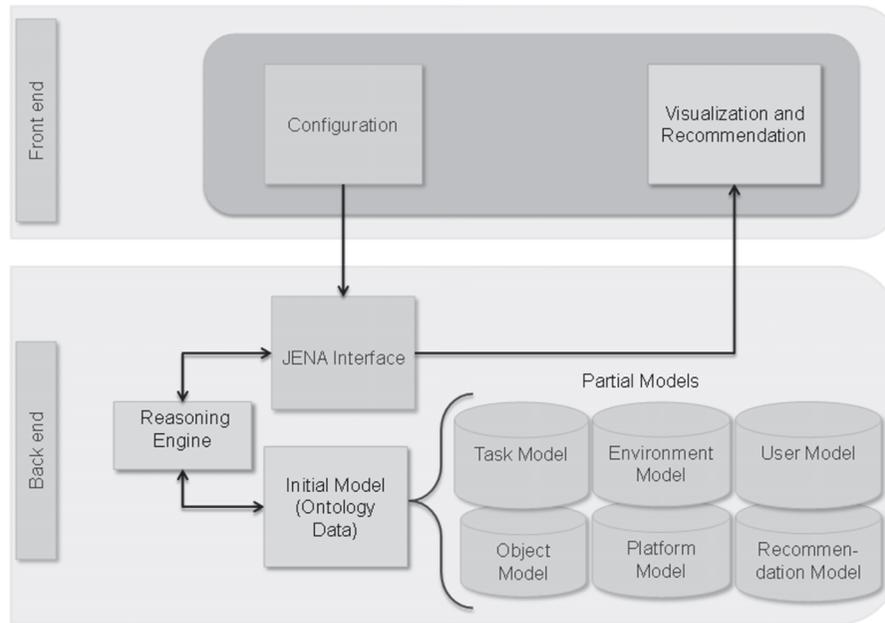


FIGURE 9: System architecture of the modelling tool.

For this purpose, the task of the user is specified with the task model by selecting predefined tasks and activities in accordance with the work situation. As such, the basis for the modelling of the work situation is the database of the initial model. In order to enable and visualise the configuration process, we propose to apply a modelling tool, which shall be described in a later section. After the tasks of the working situation are selected, a number of text-based design recommendations are presented. Subsequently, further contextual constraints such as environmental aspects, interaction and user constraints, user preferences, and object aspects are defined with the modelling tool.

- (4) Analysis of the work situation with a reasoning engine as an integral part of the modelling tool.

This is carried out automatically through the principle of inference. Through this approach new knowledge is inferred from existing knowledge (within the initial model), which finally is incorporated in the final model. Practically this implies that the amount of context information is reduced to the specific context-dependent information. The result is a list of text-based design recommendations in accordance with platform components, functionalities, and variations, which refer to a specific production context of an intelligent production environment.

- (5) Validation of the results.

In this final manual step, potential contradictions and inconsistencies are uncovered, which might lead to an adaptation of the initial model such as the introduction of new data interrelations. Thus, an adaptation or extension of partial models is required in order to add new expert knowledge and can be practically performed through an ontology.

2.5. Implementation of the Modelling Tool. The implementation of the modelling tool as seen in Figure 9 is technically regarded as an implementation of the system architecture of the required modelling tool. At the same time it is important to note that the implementation of the modelling tool can be considered as a major part of formalizing the design method. This is because the modelling tool incorporates and connects the context models on the basis of an ontology as discussed in the preceding sections. Based on the requirements and guidelines elaborated in the last sections, Figure 9 presents the system architecture of a prototypical modelling tool that is composed of a front end (user interface) and backend (reasoning engine and ontology model). As illustrated in the third and fourth step of the procedural model in Figure 8, the modelling tool should allow the configuration, visualisation, and analysis of the work situation, as well as the output of design recommendations. In order to fulfil these requirements, the configuration of the work situation is accomplished with the ontology data of the partial model (initial model) in the backend sphere. Thus, to process the ontology data, we have applied the Apache Jena Framework. Jena as such provides a collection of tools and Java libraries to develop Java applications. As a rule-based inference system, Jena includes a reasoning engine. The reasoning engine is responsible for analysing the consistency of ontology data and assessing the rule framework. Due to this inference process, the final model is created. Finally, the created ontology data of the partial models are imported and exported through the Jena interface.

Figure 10 provides an overview of the initial model, which is represented by the ontology data of the six partial models.

A more comprehensive overview of the database of the partial models is made visible by applying the prototypical

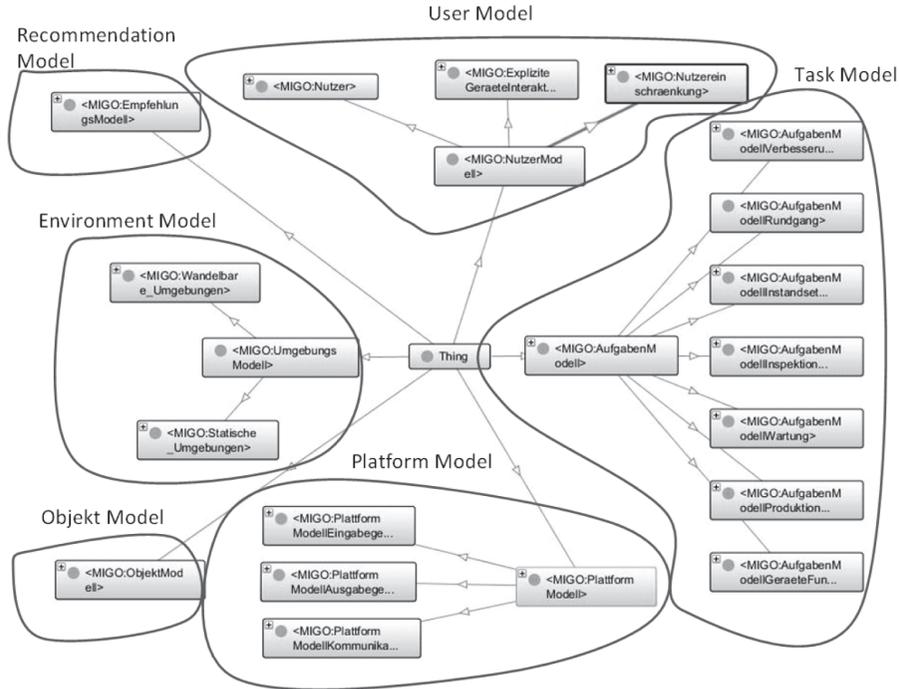


FIGURE 10: Visualisation of the initial model as an ontology.

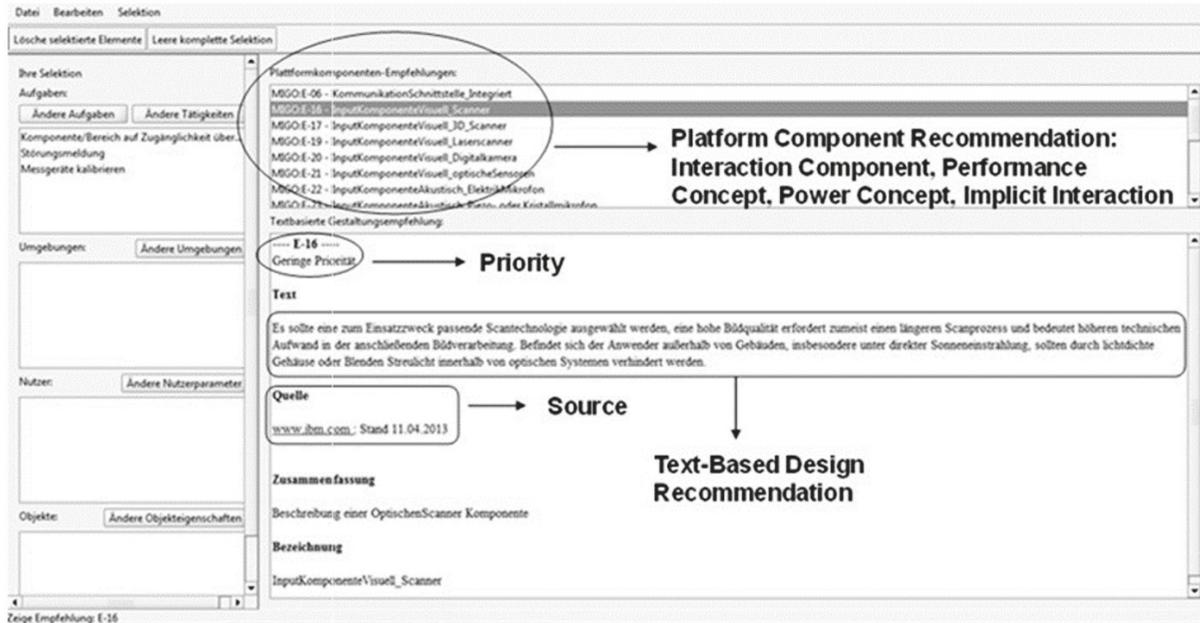


FIGURE 11: The graphical user interface of the prototypical modelling tool.

modelling tool. The focus of the modelling tool is restricted to the data basis of the partial models, and as such to the sequential configuration of a mobile work situation within a production scenario in order to acquire appropriate physical platform components and design recommendations for conceptualising mobile interaction devices. These are to be in line with the predefined context of a mobile work situation.

The necessary steps that lead to a design recommendation are dependent upon 4 modelling steps by configuring tasks, environment, user, and objects. Figure 11 shows the original version of the graphical user interface (front end) of the prototypical modelling tool. We have programmed the modelling tool in Java with the support of the Java Development Kit (JDK) and the Java Runtime Environment (JRE). The

TABLE 4: Validation results of the modelling tool according to nonfunctional qualities of software systems.

Non-Functional Qualities according to ISO/IEC 9126-1	Low Fulfilment	Medium Fulfilment	High Fulfilment
Functionality			X
Reliability		X	
Usability	X		
Maintainability	X		
Portability		X	

layout of the user interface and the arrangement of the functional icons follow a classical programming structure.

Figure 11 was at the same time the basis for a technical validation and an evaluation with end users.

3. Results and Discussion

3.1. Validation and Evaluation of the Modelling Tool. We have conducted the validation of the modelling tool through the configuration of a representative working situation and comparison to the internal data relations of the model [5]. As such, the validation provides the evidence that the modelling tool and finally the design method is capable of continuously generating the appropriate design recommendations for mobile interaction devices, which are in line with a specified work situation. As a typical work situation, we have chosen a maintenance process in a production environment where troubleshooting and calibration represent the tasks that are performed. In the work situation, the user is interacting with tools, wearing protective gloves and prefers visual data acquisition. We have further conducted the verification of the design recommendations on basis of the ontology data of the initial model. For every stepwise configuration of the work situation (task, environment, user, and object), the data relation of the design recommendation has been manually read out of the data model and documented. Afterwards, we have compared these results to the results provided by the modelling tool when successively configuring the described work situation. The results confirm that the manually determined recommendations are fully in line with the results determined by the modelling tool. This means the manually determined design recommendations from the initial model are identical to the automatically determined design recommendation by the reasoning engine. The identification of inconsistencies would have been an evidence that there is a bug in the data relations, which might lead to different or inappropriate recommendations.

Afterwards, we have performed the evaluation of the design method in cooperation with design experts from several companies. Among the participants were user interface designers, mobile technology developers, and technical consultants. The purpose was to test the method in practice in order to gather feedback from designers regarding the applicability and usability of the method. The evaluation involved the usage of the modelling tool while configuring a given mobile service process. Subsequently, the designers answered an online questionnaire in LimeSurvey, an open

source online survey tool. As a use case, we have selected an inspection process in a factory plant as this represents a typical process where large amounts of data have to be collected by interacting with the environment with the help of physical user interfaces. To support the participants, we have described the inspection process as a comprehensive text, while highlighting certain keywords (process steps and tasks), which can be easily identified in the modelling tool. A time slot of seven days with three days follow-up was predetermined for the participants. The whole evaluation process, which involved installing the modelling tool, configuring the use case, and answering the online questionnaire, took averagely 45-60 minutes per participant. Although 22 companies confirmed their participation, only the representatives of 11 companies practically conducted the whole evaluation. The results of the evaluation were significantly constructive for the refinement of the modelling tool, which is described in the next chapter.

During the configuration of the use case, the majority of the participants had difficulties to distinguish clearly between the context elements user tasks and user interactions as according to the proposed context model, tasks, and interactions were integrated into one single submodel. While configuring the process some participants expressed the requirement to be able to add individually and more detailed defined context elements, which go beyond the data basis of the initial model. Regarding the quality of the design recommendations, the majority of the participants considered these as comprehensive. Besides, the participants perceived the design recommendations as beneficial since they still left enough space for implementing creative efforts of the designers. However, the participants had a consensus that the visual presentation of the design recommendations should be improved since the modelling tool claims to provide a significant benefit in the early design process. In this respect, the use of examples with images such as illustrations of the platform components was proposed.

Table 4 provides an overview up to what qualitative extent the modelling tool fulfils the nonfunctional qualities of a software according to ISO/IEC 9126-1. Classifications between low, medium, and high fulfilment are based on the feedback of the evaluation results.

Usability and maintainability were considered as being fulfilled to a low extent since; e.g., the presentation and visualisation of design recommendations as well as the limitation regarding extension were unsatisfactory to the majority of the participants. In addition, the user interface

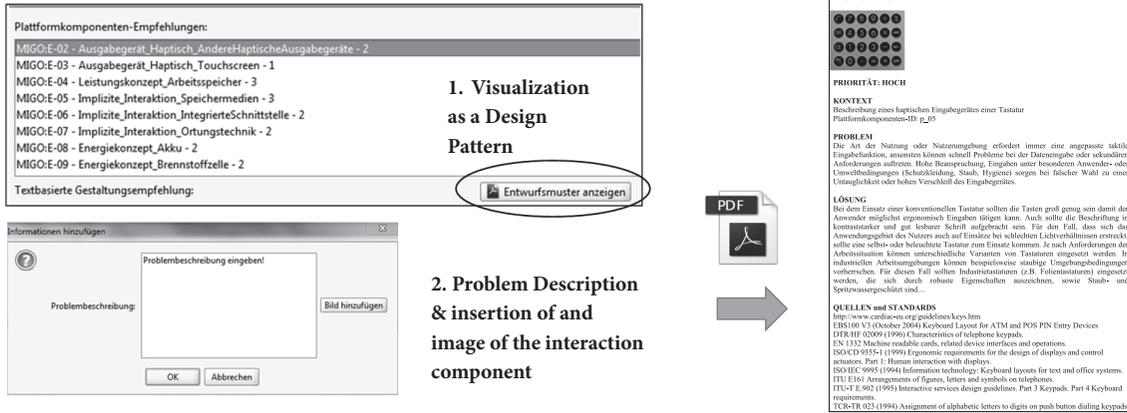


FIGURE 12: Extension of the modelling tool with a functionality for an automatic generation of design patterns.

for configuring the work situations was not implemented in a self-explaining manner and thus still has room for further improvement. On the other hand, functionality in the sense of general applicability and inferability, namely, obtaining design recommendations from context, was perceived very well by the participants.

3.2. *Refinement of the Modelling Tool.* The refinement of the modelling tool focussed upon two major aspects identified during the evaluation: the extension of the context model and the improvement of the visual representation of the design recommendations.

For this purpose, we suggested incorporating a technique that improves the visual presentation of design recommendations and at the same time ensures capturing and integration of expert knowledge in a straightforward way. When considering design pattern techniques as described by [56, 61], it appears reasonable to apply this approach in the design method. The advantages of this form of information presentation lie particularly in the standardised structuring and processing of design information. Concurrently, this may lead to a simplification of design information exchange and reuse between design teams. Besides, a better visualisation and communication of design recommendations can be achieved by applying a mechanism that compiles text-based design information to a PDF file, which follows the structure of a design pattern. The objective from a designer perspective would be to have a tool that automatically generates design patterns from acquired design recommendations, which originate from the data of the initial model. This means that instead of relying on a collection of existing design patterns and matching these to a given context, the approach here pursues to create the design patterns from the obtained platform components and design recommendations.

In this manner, we decided to extend the modelling tool with a functionality that enables the automatic generation of physical design patterns as highlighted in Figure 12. In order to demonstrate the feasibility of this approach, we found it reasonable to realise twenty exemplary design patterns, consisting of input, output, and communication devices.

Subsequently, we have implemented an automatic mechanism in the modelling tool that enables an automatic generation of design patterns based on the design recommendations of the tool. The design patterns can then further be grouped according to their affiliation to ontology subclasses and instances with the aim of validating the consistency of design patterns for a certain working situation. This means design patterns belonging to the same ontology subclasses and instances can be grouped by a certain category, while design patterns with overlapping ontology subclasses and instances can be considered interrelated. For example, two design patterns describing a keyboard and a flexible display, although mainly possessing different ontology subclasses and instances, could both belong to the task instance “maintenance” and therefore be regarded as interrelated to one another. Subsequently, for a configured work situation, although design patterns belonging to different subclasses and instances are suggested, it can nevertheless be assumed that there exists a relation between these individual design patterns. Finally, related design patterns are more likely to be appropriate for a certain work situation than design patterns, which are not related in any sense. Thus, the property of relations between design patterns represents one of the main requirements of a design pattern language, which is a good basis for an extension of this work.

The solution space of the platform component consists of a text-based recommendation, a description of the functionality, of the platform component, the variations, and the sources as illustrated exemplarily in Figure 13.

In alignment with the introduction of design patterns, we believe it is a reasonable path to facilitate the creation of additional design patterns based on the expertise of the designers. The creation of additional design patterns would fulfil the requirement of ensuring a continuous update of the design recommendations and platform components. Vice versa, extending design patterns by the users of the modelling tool indirectly leads to an extension of the context model as suggested during the evaluation process. This becomes obvious when the considering that the data from the additional design patterns can be seamlessly integrated into the

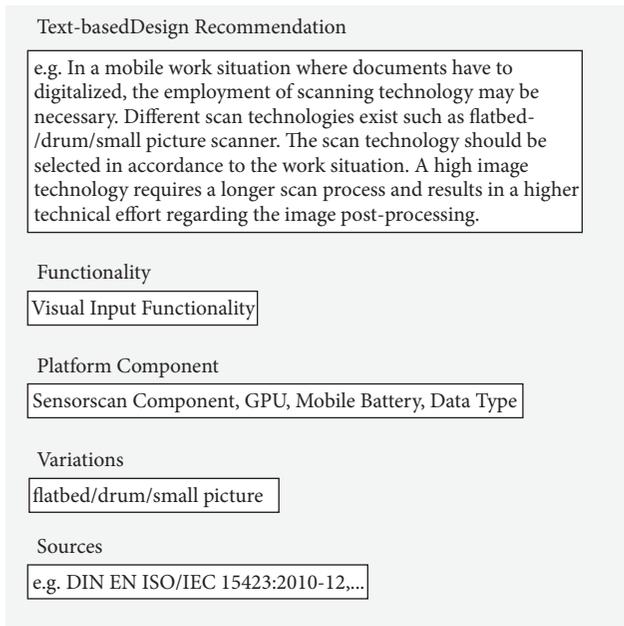


FIGURE 13: Solution space for the design recommendation.

platform and recommendation models. At the same time, this approach would grant dynamic properties to the context model.

Under the premise that these approaches are integrated into the procedural model in Figure 8, the model is refined and extended with respect to the procedural steps. Figure 14 illustrates the procedural model of the design method, which is extended by the feature of visualising and generating design patterns. The lower part of the model shows the extended procedural steps of the design method. From a practical point of view, the functionality of creating design recommendations is employed in the modelling tool by a designated functionality that automatically retrieves the design pattern, which corresponds to the particular design recommendation. As mentioned, the design pattern follows the composition of HCI design patterns, which incorporates a description of the context, problem, solution, and an image of the platform component. However, the design recommendation and the platform component information do not per default include a description of the problem and an image of the mobile interaction device.

Therefore, it is necessary to complement this information a priori to the design pattern generation in order to bring the design pattern in line with the HCI design pattern structure. Finally, the emerged design patterns are crosschecked and validated by an expert and subsequently incorporated in the initial model. In this manner, it is ensured that the data of the initial model is continuously updated and thus upgraded from a nonfunctional perspective with dynamical properties.

3.3. Potentials of the Modelling Tool. In the first version of the modelling tool as described in Section 2.5, the design recommendations are prioritised regarding their relevance, based upon the approval of the applied user interface technologies

in practice. However, the prioritisation is limited by a fixed predefinition in the initial model (high, medium, and low), which does not provide a mechanism to further distinguish between same priority levels of design recommendations. Practically, this means that two or more design recommendations having the same priority levels and belonging to the same type of interaction component are not further distinguished. This leads to the fact that the designer has to decide which design recommendation is most suitable for the work context. Through dynamically assigning priority levels to the design recommendations, it would be possible to differ, e.g., between relevant and optional design recommendations. This could be achieved by employing an intelligent algorithm, which may keep track of how often a design recommendation has been proposed and employed for a certain work context.

A further aspect was the implication of the design recommendations upon the overall user interface concept when concurrently implementing multiple design recommendations. Likely, the implementation of a quantitative design recommendation may have a negative implication (undesired effect) on other interaction components; i.e., design recommendations may affect each other when these are all implemented. As an example, a recommendation of a discrete distance between the keys of a keyboard may lead to the situation that the arrangement and form factor of nearby interaction components are affected accordingly. In order to minimise this effect, it is necessary to investigate the contextual relations between design recommendations more thoroughly. A major step could be to define dependencies between design patterns, which may result in developing a pattern language. Conclusively, we believe that the conception of a design pattern language for mobile interaction devices in intelligent production environments should lead to a significant contribution to the advancement of design pattern languages. However, at this point, the necessity for the development of a design pattern language is less an issue for qualitative design patterns because of the broader design space and a higher freedom of design. As qualitative recommendations are more suitable for the realisation of new user interface concepts, quantitative recommendations are more efficient for the refinement and adaptation of existing user interface concepts. For this reason, we advise considering a balance between qualitative and quantitative design recommendations.

4. Conclusions

In this paper, we dealt with the realisation of a design method for conceptualising physical user interfaces, namely, mobile interaction devices for intelligent production environments. The analysis of the state of the art had the aim of identifying the shortcomings of existing design methods, tools, and techniques. This enabled us to derive a series of functional and nonfunctional requirements. These were interrelated to one another while construing (at this point loosely defined) procedural guidelines for a suitable design method, which yielded in a comprehensive procedural model. In alignment with the procedural model, we proposed a model-based conceptual framework, which considers the scope, structure,

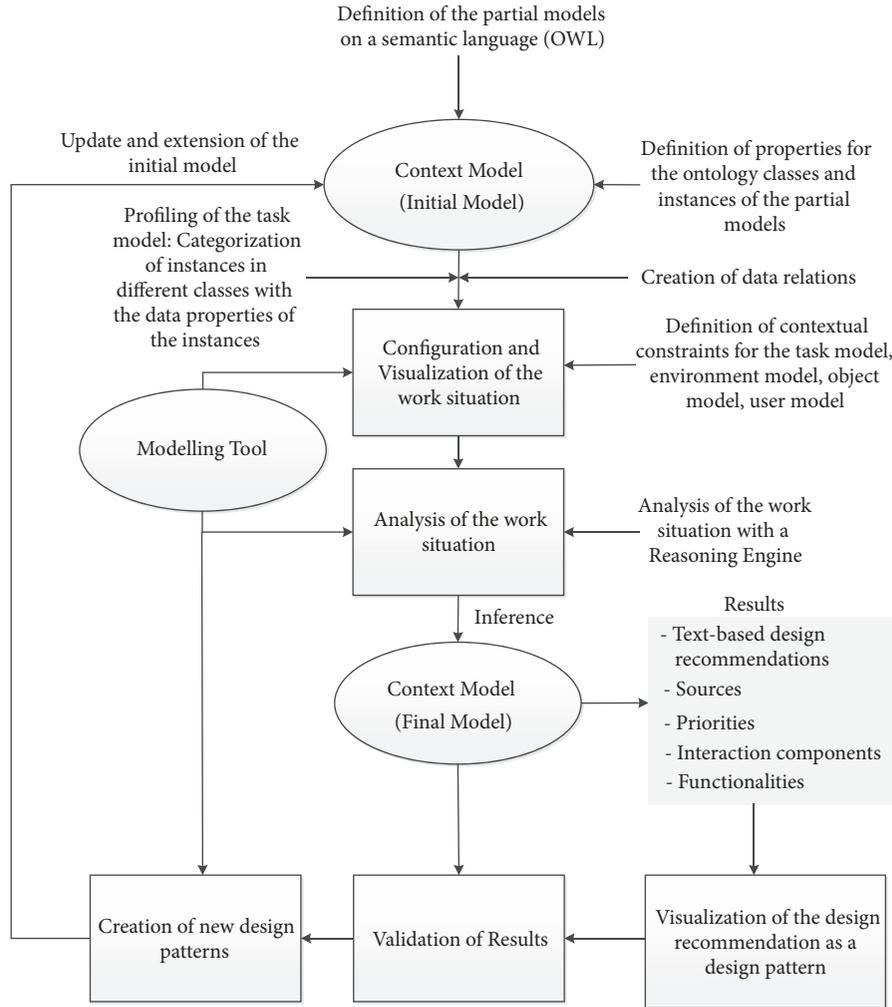


FIGURE 14: An extended procedural model of the design method.

and modelling techniques for describing the features of mobile interaction devices. Thus, in line with the notion of a model-based framework, we further proposed an ontology-based context model, which consists of six interrelated partial models that provide the means for describing intelligent production contexts. Correspondingly, we suggested the need for a modelling tool, which enables not only the configuration of work situations but also the inference of design recommendations through the utilisation of a reasoning technique. This allows the elaboration of text-based design recommendations in relation to platform components and their priority level for mobile interaction devices.

The implementation of a modelling tool as a part of the design method was achieved through the implementation of a system architecture, which includes all modules required for the applying of the design method. Within the prototypical version of the modelling tool, we managed to demonstrate that a series of design recommendations are inferred, including also those recommendations that are not explicitly excluded. Due to this fact, the design recommendations were categorised context-dependently according to “high,” “medium,” and “low” priority.

We have then enhanced the prototypical modelling tool with the additional functional requirement of considering an improvement of the visualisation of design recommendations and an extension of the context model. In this respect, we have introduced a mechanism that enables the automatic generation of design patterns from the context-dependent design recommendations and platform components. In close correlation to this feature, we have recognised that there is a need for a feature that ensures that the underlying context model is continuously held up to date. For this purpose, we have employed a feature that enables an on-demand creation of additional design patterns while integrated into the initial model. In summary, this led to a revised procedural model for the design method.

We note that the challenges incorporated with the early design phases in user interface conceptualisation are not yet entirely exploited. The fact that intelligent production environments are characterised by cyberphysical systems will increase the need for specialised mobile interaction devices, which support the interaction with distributed information in the cyberphysical work environment. As a consequence, methods and tools, which consider and efficiently utilise the

context within the product development process, will play a significant role. Moreover, it will eventually be recognised that emotional awareness is an equally important context element, which cannot be neglected in the early design phases. Examples include maintenance processes and collaborative working in intelligent production environments where emotions as fear, inconvenience, fatigue, boredom, and distraction can have a severe impact on safety conditions and quality of work of the human.

Conclusively, in economic terms, within the design organizations and divisions, we consider that supporting methods and tools should be able to seamlessly integrate into already existing development processes. Particularly, methods and tools which are least-disruptive regarding existing and well-established design processes will possess the potential to be accepted by product-developing companies in the long run.

5. Outlook and Future Work

In this paper, we have set the focus on the early design phase (sketch phase) of mobile interaction devices, particularly suitable for work situations concerning intelligent production environments. Beyond the sketch phase, the proposed context model definitely possesses the potential to be integrated and utilised in the CAx design phase. Hence, the advantage of this possibility lies in the realisation of (data) annotations to priority existing 3D conceptual designs of product data models. Accordingly, in the EU research project VICON (<http://www.vicon-project.eu>) first approaches were developed for realising annotations between product data models and design recommendations from the sketch phase. This approach and the underlying techniques were demonstrated with quantitative design recommendations for user interfaces of mobile phones and washing machines. It is conceivable that these results may be used effectively for leveraging the introduced design method to be integrated into the CAx design phase. Next to the integration in the CAx design phase, we definitely see a great potential for integrating our modelling approach with prototyping tools for physical interaction such as for tangible user interfaces and phidget components. In this respect, the formal presentation of design recommendations of our modelling tool can be construed as XML-defined design patterns in order to achieve an interoperability with physical UI prototyping tools such as the ECCE toolkit and vice versa [52]. In this way, process and tasks configurations for work situations and their interrelation to design recommendations can be seamlessly integrated on an abstract level, complementing the contemporary features of physical UI prototyping tools.

Apart from the challenges related to the subsequent design phases, we expect that software user interfaces will converge with physical user interfaces. An appropriate example which highlights this paradigm is to consider touch screens, as hardware and software elements are directly entangled with one another. Thus, design methods in this area should be capable of considering both development strands. Additionally, due to the continuous advancement of mobile interaction devices including wearable technology, we anticipate that adaptive hardware concepts will play an increasingly

vital role in cyberphysical production environments. Further, we believe that it is likely that adaptive physical user interfaces are capable of coping with a wide range of situations, as it is currently possible with discrete mobile interaction devices. Thereby, new requirements and possibilities will emerge within user interface design. It will be less the case to uncover relations between interaction concepts and work situations. Moreover, the point will be to describe the spectrum of configuration possibilities of adaptive interaction devices in relation to different work situations. When we consider the continuous emergence of the fourth industrial revolution, it is reasonable to predict that adaptive hardware concepts will be a well-established interaction concept in the future. This tendency will concurrently foster new paradigms in product development, which possess the potential of dissolving the limitations between the design process and the designed artefact.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

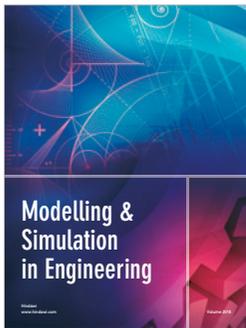
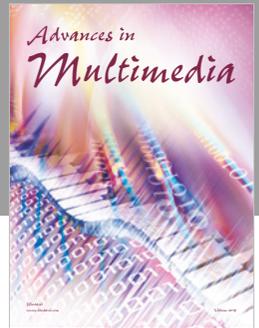
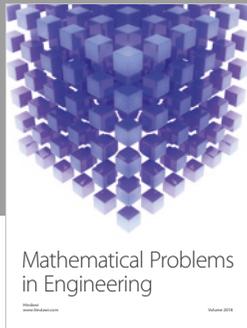
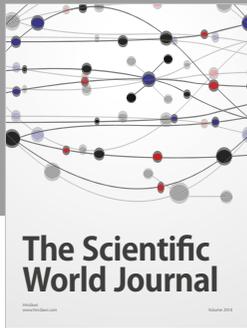
The authors declare that there are no conflicts of interest regarding the publication of the paper.

References

- [1] C. Courage and K. Baxter, *Understanding Your Users: A Practical Guide to User Requirements Methods*, Morgan Kaufmann, 2005.
- [2] B. Buxton, *Sketching User Experiences: Getting the Design Right and the Right Design*, Morgan Kaufmann, 2010.
- [3] M. E. Porter and J. E. Heppelmann, "How Smart, Connected Products Are Transforming Companies," *Harvard Business Review*, vol. 93, no. 10, pp. 96–114, 2015.
- [4] R. Schmidt, M. Möhring, R.-C. Härting, C. Reichstein, P. Neumaier, and P. Jozinovic, "Industry 4.0-potentials for creating smart products: empirical research results," in *Proceedings of the International Conference on Business Information Systems*, pp. 16–27, 2015.
- [5] P. T. Kirisci, *Gestaltung mobiler Interaktionsgeräte: Modellierung für intelligente Produktionsumgebungen [Design of mobile interaction devices: Modelling for intelligent production environments]*, 1 Auflage. Springer Vieweg, 2016.
- [6] S. Biege, G. Lay, and D. Buschak, "Mapping service processes in manufacturing companies: Industrial service blueprinting," *International Journal of Operations and Production Management*, vol. 32, no. 8, pp. 932–957, 2012.
- [7] T. W. Zängler, *Mikroanalyse des Mobilitätsverhaltens in Alltag und Freizeit*, Springer, Berlin, Germany, 2000.
- [8] P. Kirisci, E. Morales-Kluge, E. Angelescu, and K.-D. Thoben, "Design of Wearable Computing Systems for Future Industrial Environments," in *Handbook of Research on Mobility and Computing: Evolving Technologies and Ubiquitous Impacts and Computing*, pp. 1226–1246, IGI Global, 2011.

- [9] D. Holman, A. Girouard, H. Benko, and R. Vertegaal, "The design of organic user interfaces: Shape, sketching and hyper-context," *Interacting with Computers*, vol. 25, no. 2, pp. 133–142, 2013.
- [10] J. Jansch and H. Birkhofer, "The development of the guideline VDI 2221 - The change of direction," in *Proceedings of the 9th International Design Conference, DESIGN 2006*, pp. 45–52, Croatia, May 2006.
- [11] K. Luyten, C. Vandervelpen, and K. Coninx, "Task modeling for ambient intelligent environments: Design support for situated task executions," in *Proceedings of the 4th International Workshop on Task Models and Diagrams, TAMODIA '05*, pp. 87–94, Gdansk, Poland, September 2005.
- [12] E. Riedenklau, *Development of Actuated Tangible User Interfaces: New Interaction Concepts and Evaluation Methods*, Bielefeld University, 2016.
- [13] N. Bevan, J. Carter, and S. Harker, "Iso 9241-11 revised: What have we learnt about usability since 1998?" *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 9169, pp. 143–151, 2015.
- [14] N. Bevan, "International standards for HCI and usability," *International Journal of Human-Computer Studies*, vol. 55, no. 4, pp. 533–552, 2001.
- [15] P. T. Kirisci and K.-D. Thoben, "The role of context for specifying Wearable computers," in *Proceedings of the 3rd IASTED International Conference on Human-Computer Interaction, HCI 2008*, pp. 248–253, Austria, March 2008.
- [16] M. M. Andreasen, "Modelling—The Language of the Designer," *Journal of Engineering Design*, vol. 5, no. 2, pp. 103–115, 1994.
- [17] G. Maarten Bonnema and F. J. A. M. van Houten, "Use of models in conceptual design," *Journal of Engineering Design*, vol. 17, no. 6, pp. 549–562, 2006.
- [18] M. Modzelewski, M. Lawo, P. Kirisci et al., "Creative Design for Inclusion Using Virtual User Models," in *Computers Helping People with Special Needs*, vol. 7382 of *Lecture Notes in Computer Science*, pp. 288–294, Springer Berlin Heidelberg, Berlin, Germany, 2012.
- [19] P. Biswas, P. Robinson, and P. Langdon, "Designing inclusive interfaces through user modeling and simulation," *International Journal of Human-Computer Interaction*, vol. 28, no. 1, pp. 1–33, 2012.
- [20] E. Castillejo, A. Almeida, and D. López-de-Ipiña, "Ontology-Based Model for Supporting Dynamic and Adaptive User Interfaces," *International Journal of Human-Computer Interaction*, vol. 30, no. 10, pp. 771–786, 2014.
- [21] T. Clerckx, K. Luyten, and K. Coninx, "The mapping problem back and forth: Customizing dynamic models while preserving consistency," in *Proceedings of the 3rd Annual Conference on Task Models and Diagrams, TAMODIA '04*, pp. 33–42, Czech Republic, November 2004.
- [22] A. Puerta and J. Eisenstein, "Towards a General Computational Framework for Model-Based Interface Development Systems," in *Proceedings of the Intelligent User Interfaces, IUI 99*, pp. 171–178, 1999.
- [23] N. O. Bernsen, "Modality theory in support of multimodal interface design," in *Proceedings of the Spring Symposium on Intelligent Multi-Media Multi-Modal Systems*, pp. 37–44, 1994.
- [24] P. Kirisci, K.-D. Thoben, P. Klein, and M. Modzelewski, "Supporting Inclusive Design of Mobile Devices with a Context Model," *Advances and Applications in Mobile Computing*, pp. 65–88, 2012.
- [25] P. Kirisci, K.-D. Thoben, P. Klein, and M. Modzelewski, "Supporting inclusive product design with virtual user models at the early stages of product development," in *Proceedings of the 18th International Conference on Engineering Design (ICED 11), Impacting Society through Engineering Design*, vol. 9, pp. 80–90, 2011.
- [26] C. Ebert, *Systematisches Requirements Engineering*, Heidelberg: dpunkt.verlag, 4th edition, 2012.
- [27] C. Barnes and S. P. Lillford, "Decision support for the design of affective products," *Journal of Engineering Design*, vol. 20, no. 5, pp. 477–492, 2009.
- [28] S. Zogaj and U. Bretschneider, "Customer Integration in New Product Development: a Literature Review Concerning the Appropriateness of Different Customer Integration Methods to Attain customer Knowledge," in *Proceedings of the ECIS 2012: The 20th European Conference on Information Systems (ECIS'12)*, vol. 2, pp. 1–14, 2012.
- [29] E. Zitkus, P. Langdon, and J. Clarkson, "Accessibility Evaluation: Assistive Tools for Design Activity," in *Proceedings of the 1st International Conference Sustainable Intelligent Manufacturing*, pp. 659–670, IST Press, Leiria, Portugal, 2011.
- [30] M. Schickler, *Entwicklung mobiler Apps im Business und E-Health Bereich*, 2015.
- [31] J. Agar, *Constant touch: A global history of the mobile phone*, Icon Books Limited, 2013.
- [32] M. S. Al-Razgan, H. S. Al-Khalifa, M. D. Al-Shahrani, and H. H. AlAjmi, "Touch-Based Mobile Phone Interface Guidelines and Design Recommendations for Elderly People: A Survey of the Literature," in *Neural Information Processing*, vol. 7666 of *Lecture Notes in Computer Science*, pp. 568–574, Springer, Berlin, Germany, 2012.
- [33] P. Beigl, F. Schneider, and S. Salhofer, "Takeback systems for mobile phones: review and recommendations," *Proceedings of the Institution of Civil Engineers - Waste and Resource Management*, vol. 165, no. 1, pp. 25–35, 2012.
- [34] N. Gedik, A. Hanci-Karademirci, E. Kursun, and K. Cagiltay, "Key instructional design issues in a cellular phone-based mobile learning project," *Computers & Education*, vol. 58, no. 4, pp. 1149–1159, 2012.
- [35] Y. S. Park and S. H. Han, "Touch key design for one-handed thumb interaction with a mobile phone: effects of touch key size and touch key location," *International Journal of Industrial Ergonomics*, vol. 40, no. 1, pp. 68–76, 2010.
- [36] E. Goodman, E. Stolterman, and R. Wakkary, "Understanding interaction design practices," in *Proceedings of the the 2011 annual conference*, p. 1061, Vancouver, BC, Canada, May 2011.
- [37] P. T. Kirisci and K. Thoben, "Vergleich von Methoden für die Gestaltung Mobiler Endgeräte [Comparison of methods for the design of mobile devices]," *icom - Zeitschrift für interaktive und Koop. Medien*, vol. 8, no. 1, pp. 52–59, 2009.
- [38] A. Puerta, "A model-based interface development environment," *IEEE Software*, vol. 14, no. 4, pp. 40–47, 1997.
- [39] E. Yigitbas, H. Fischer, and S. Sauer, "Model-Based User Interface Development for Adaptive Self-Service Systems," in *Design, User Experience, and Usability. Theories, Methods, and Tools for Designing the User Experience*, vol. 8517 of *Lecture Notes in Computer Science*, pp. 206–213, Springer International Publishing, Cham, 2014.
- [40] G. Calvary, J. Coutaz, D. Thevenin, Q. Limbourg, L. Bouillon, and J. Vanderdonck, "A unifying reference framework for multi-target user interfaces," *Interacting with Computers*, vol. 15, no. 3, pp. 289–308, 2003.

- [41] G. Meixner, "Modellbasierte Entwicklung von Benutzungsschnittstellen," *Informatik-Spektrum*, vol. 34, no. 4, pp. 400–404, 2011.
- [42] H. Witt, T. Nicolai, and H. Kenn, "The WUI-Toolkit: A Model-Driven UI Development Framework for Wearable User Interfaces," in *Proceedings of the 27th International Conference on Distributed Computing Systems Workshops (ICDCSW'07)*, pp. 43–43, Toronto, ON, Canada, June 2007.
- [43] J. Vanderdonck and M. Florins, "Model-based design of mobile user interfaces," in *Proceedings of the Mobile HCI2001: Third International Workshop on Human Computer Interaction with Mobile Devices*, 2001.
- [44] T. Clerckx, F. Winters, and K. Coninx, "Tool support for designing context-sensitive user interfaces using a model-based approach," in *Proceedings of the 4th international workshop on Task models and diagrams*, ACM International Conference Proceedings Series, pp. 11–17, Gdansk, Poland, September 2005.
- [45] G. Calvary, J. Coutaz, and D. Thevenin, "A Unifying Reference Framework for the Development of Plastic User Interfaces," in *Engineering for Human-Computer Interaction*, vol. 2254 of *Lecture Notes in Computer Science*, pp. 173–192, Springer Berlin Heidelberg, Berlin, Germany, 2001.
- [46] B. Ullmer and H. Ishii, "Emerging frameworks for tangible user interfaces," *IBM Systems Journal*, vol. 39, no. 3–4, pp. 915–930, 2000.
- [47] S. Greenberg and C. Fitchett, "Phidgets: Easy development of physical interfaces through physical widgets," in *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology (UIST'01)*, pp. 209–218, USA, November 2001.
- [48] H. Gellersen, G. Kortuem, A. Schmidt, and M. Beigl, "Physical prototyping with smart-its," *IEEE Pervasive Computing*, vol. 3, no. 3, pp. 74–82, 2004.
- [49] R. Ballagas and F. Memon, "iStuff mobile: rapidly prototyping new mobile phone interfaces for ubiquitous computing," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1107–1116, 2007.
- [50] S. Houben and N. Marquardt, "WATCH connect: A toolkit for prototyping smartwatch-centric cross-device applications," in *Proceedings of the 33rd Annual CHI Conference on Human Factors in Computing Systems, CHI 2015*, pp. 1247–1256, Republic of Korea, April 2015.
- [51] B. Hartmann, S. R. Klemmer, M. Bernstein et al., "Reflective physical prototyping through integrated design, test, and analysis," in *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology 2006, UIST'06*, pp. 299–308, Switzerland, October 2006.
- [52] A. Bellucci, I. Aedo, and P. Díaz, "ECCE toolkit: Prototyping sensor-based interaction," *Sensors*, vol. 17, no. 3, 2017.
- [53] R. Herriott, "Are inclusive designers designing inclusively? An analysis of 66 design cases," *The Design Journal*, vol. 16, no. 2, pp. 138–158, 2013.
- [54] P. John Clarkson and R. Coleman, "History of inclusive design in the UK," *Applied Ergonomics*, vol. 46, pp. 235–247, 2015.
- [55] S. Matiouk, M. Modzelewski, Y. Mohamad et al., "Prototype of a Virtual User Modeling Software Framework for Inclusive Design of Consumer Products and User Interfaces," in *Universal Access in Human-Computer Interaction. Design Methods, Tools, and Interaction Techniques for eInclusion*, vol. 8009 of *Lecture Notes in Computer Science*, pp. 59–66, Springer Berlin Heidelberg, Berlin, Germany, 2013.
- [56] J. O. Borchers, "A pattern approach to interaction design," *AI & Society*, vol. 15, no. 4, pp. 359–376, 2001.
- [57] C. Alexander, *Notes on the Synthesis of Form*, Harvard University Press, Cambridge, Mass, USA, 1964.
- [58] C. Alexander, *A Pattern Language: Towns, Buildings, Construction*, Oxford University Press, 1977.
- [59] D. C. Schmidt, "Using Design Patterns to Develop Reusable Object-Oriented Communication Software," *Communications of the ACM*, vol. 38, no. 10, pp. 65–74, 1995.
- [60] D. Riehle and H. Züllighoven, "Understanding and using patterns in software development," *TAPOS*, vol. 2, no. 1, pp. 3–13, 1996.
- [61] J. A. Landay and G. Borriello, "Design patterns for ubiquitous computing," *The Computer Journal*, vol. 36, no. 8, pp. 93–95, 2003.
- [62] S. Imtiaz, "User Centered Design Patterns and Related Issues A Review," *International Journal of Human-Computer Interaction*, vol. 4, no. 1, pp. 19–24, 2013.
- [63] A. Neyem, C. Aracena, C. A. Collazos, and R. Alarcón, "Designing emotional awareness devices: what one sees is what one feels," *Ingeniare. Revista chilena de ingeniería*, vol. 15, no. 3, pp. 227–235, 2007.
- [64] E. Kanjo, L. Al-Husain, and A. Chamberlain, "Emotions in context: examining pervasive affective sensing systems, applications, and analyses," *Personal and Ubiquitous Computing*, vol. 19, no. 7, pp. 1197–1212, 2015.
- [65] A. Mcstay, *Empathic Media: The Rise of Emotional AI*, SAGE Publications Ltd, 1st edition, 2018.
- [66] D. Cernea and A. Kerren, "A survey of technologies on the rise for emotion-enhanced interaction," *Journal of Visual Languages and Computing*, vol. 31, pp. 70–86, 2015.
- [67] D. Lottridge, M. Chignell, and A. Jovicic, "Affective interaction: understanding, evaluating, and designing for human emotion," *Reviews of Human Factors and Ergonomics*, vol. 7, no. 1, pp. 197–217, 2011.
- [68] A. Alibage, A. Jetter, and A. Alibage, *PDXScholar Drivers of Consumers' Emotional Engagement with Everyday Products: An Intensive Review of the Literature and an Attempt to Conceptualize the Consumer-Product Interactions Within the Emotional Design Process Engineering and Technology Management*, 2017.
- [69] A. Schmidt, M. Beigl, and H.-W. Gellersen, "There is more to context than location," *Computers & Graphics*, vol. 23, no. 6, pp. 893–901, 1999.
- [70] E. Reponen and K. Mihalic, "Model of primary and secondary context," in *Proceedings of the the international workshop in conjunciton with AVI 2006*, pp. 37–38, Venice, Italy, May 2006.



Hindawi

Submit your manuscripts at
www.hindawi.com

