A Framework for Context-dependent User Interface Adaptation

Stephan Kölker[1], Felix Schwinger[1,2] and Karl-Heinz Krempels[1,2]
[1] *Information Systems, RWTH Aachen University, Aachen, Germany*
[2] *Fraunhofer Institute for Applied Information Technology FIT, Aachen, Germany*

Keywords: Context, Context-aware Applications, User Interface, Human-computer Interaction, Adaptation.

Abstract: Mobile information systems are operated in a large variety of different contexts – especially during intermodal journeys. Every context has a distinct set of properties, so the suitability of user interfaces differs in various contexts. But currently, the representation of information on user interfaces is hard-coded. Therefore, we propose a dynamic adaptation of user interfaces to the context of use to increase the value of an information system to the user. The proposed system focuses on travel information systems but is designed in a way that it is generalizable to other application domains. The adaptation system works as an independent service that acts as a broker in the communication between an application and the user. This service transforms messages between a user- and a system-oriented representation. The context of use, the device configuration, and user preferences affect the calculated user-oriented representation.

1 MOTIVATION AND INTRODUCTION

In recent years, powerful and flexible, yet small and handy mobile information systems have been developed and became widely available to the public. They do not only provide useful information, but also entertainment and means of communication with other people. Due to their high portability and usefulness, such information systems are used in a diverse set of environments, such as at home, at the working place or during traveling.

A context is a collection of information on the situation of an entity. This includes information on the entity itself, its physical and social environment and all other entities with an influence on it. An entity can be a human, an object or a place (Dey, 2001; Strang and Linnhoff-Popien, 2003).

Most contemporary mobile devices are equipped with a multitude of different sensors, such as accelerometers, cameras or light sensors. The data collected from those sensors give information systems access to a wide range of information on the current context of use (Johnson and Trivedi, 2011).

In this paper, we especially examined the mobility aspect of potential users, as a variety of different contexts can occur during a journey (Hörold et al., 2013). Each context has different properties and makes different demands on the human-computer interaction (Kolski et al., 2011). For example, when carrying luggage with both hands, tactile interaction with an information system is inconvenient because both hands are already occupied. In this situation, the user has to stop walking and put down the luggage to be able to input information into an information system. Likewise, the use of *graphical user interfaces* (GUIs) while driving a car or walking is dangerous. The interaction with a GUI usually distracts the vision of the user. The consequence is a highly increased crash potential (Smith, 2014).

The user interface (UI) of most information systems is hard-coded in the form of GUIs that are controlled over tactile communication means (Edwards and Mynatt, 1994). However, as illustrated in the examples above, this hard-coding of UIs results in restricted access of the user to the information system in certain situations.

To provide the optimal human-computer interaction in each context of use, the user interface must automatically adapt to the current context of use using the available communication resources. Hence, each message between the user and the system is rendered with the currently most suitable modality on the most suitable output device. This increases the number of situations where an information system can be used.

We attempt to sketch a framework for the automatic adaptation of multimodal user interfaces to the current context. The remainder of the paper is structured as

Kölker, S., Schwinger, F. and Krempels, K.
A Framework for Context-dependent User Interface Adaptation.
DOI: 10.5220/0008487200002366
In *Proceedings of the 15th International Conference on Web Information Systems and Technologies (WEBIST 2019)*, pages 418-425
ISBN: 978-989-758-386-5; ISSN: 2184-3252

follows: Section 2 introduces the current State of the Art regarding multimodal UIs and adaptation systems. In Section 3, we present our approach to the problem, whereas Section 4 concludes the paper with a brief summary and highlights future work.

2 STATE OF THE ART

Current widely used travel information systems, such as car navigation systems or Google Maps usually use hard-coded UIs that are optimized for a specific context of use. Most commonly, GUIs are used for the presentation of information to the user and tactile UIs to receive information (Edwards and Mynatt, 1994). Additionally, speech-based interaction is common among travel information systems to present information to drivers. The general topic of context-aware computing is not a recent one, we refer the reader to recent surveys in the area for a more general overview: (Chen and Kotz, 2000; Jaimes and Sebe, 2007; Dumas et al., 2009; Hong et al., 2009)

(Mitrevska et al., 2015) present a context-aware in-car information system that interacts with the user via speech, gesture and displays. Their system enables the user to interact with specific objects in the environment of the car (e.g., a restaurant) to retrieve further information about them or to access associated services (e.g., reserving a table), but the user interaction is not dynamically adapted to the current context.

An XML-based language for describing multimodal UIs on different levels of abstraction is introduced by (Vanderdonckt et al., 2004). This language supports the device- and modality-independent development of multimodal UIs and provides transformations between different forms of UIs. An iterative process for the design of mobile information systems with multimodal interfaces is proposed by (Lemmelä et al., 2008). The result of the design process is a set of UIs for different contexts of use designed by UI developers. The automated synthesis of UIs for different contexts is not examined in their work.

A system for the automated synthesis of UIs is proposed by (Falb et al., 2006). They represent the human-computer interaction by communicative acts and introduce a meta-model for the description of interactions. The UI is automatically generated for various possible output devices based on an interaction description, but does not take the context of use into account.

Besides car navigation systems and PDAs/smartphones, further device classes have been evaluated with regard to their suitability for travel navigation. Travel navigation systems on the basis of *smartwatches* are proposed by (Pielot et al., 2010), (Zargamy et al., 2013), and (Samsel et al., 2015). The suitability of *vibration* for presenting navigational instructions to travelers is examined by (Pielot et al., 2012). (Eis et al., 2017) introduce a travel navigation system with *smart glasses* and (Rehman and Cao, 2016) compare handheld-based navigation with navigation on smart glasses.

(Baus et al., 2002) present a pedestrian navigation system that provides the user with context-dependent information while adapting the presentation of this information to the capabilities of the employed hardware and the information needs of the user. A process for the adaptation of UIs to hardware capabilities and user preferences is described by (Christoph and Krempels, 2007; Christoph et al., 2010). They represent UIs as a sequence of elementary interaction objects. Those interaction objects are represented in XML and adapted via XSLT. The dynamic adaptation of GUIs during runtime is discussed by (Criado et al., 2010). They present a meta-model for the description of GUIs that allows incremental adaptation.

3 APPROACH

In this section, we are introducing our automated *user interface* adaptation system. The UI adaptation system is designed as an independent service that acts as a broker in the communication between the user and an information system. An information system is the client of this adaptation service. The UI adaptation service is responsible for the context-dependent generation of a UI, the reception of user inputs and, optionally, the interpretation of user inputs.

An overview of the system is given in Figure 1. Instead of generating a UI, an information system sends a message to the UI adaptation service. The UI adaptation service calculates a suitable representation for this message based on the context of use. The service forwards this representation to the rendering engine which renders the UI. The UI adaptation service directly receives user inputs and extracts useful information from it. Afterwards, it sends the information to the target application.

3.1 Service Interfaces

For the context-dependent adaptation of a UI, the UI adaptation service interacts with a client information service, a context detection service, and a renderer. In this section, a representation of data on those interfaces is proposed.

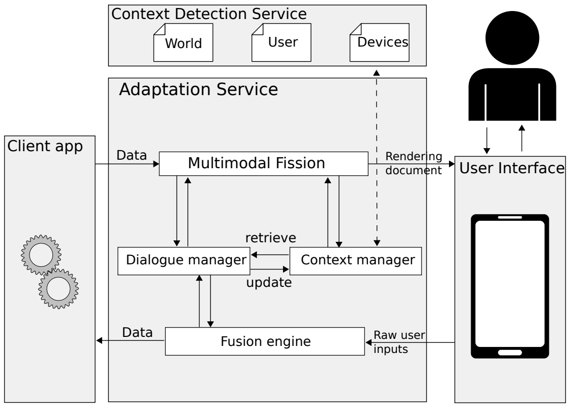

Figure 1: Overview over the components of the UI adaptation service.

The Information Service Interface. The human-computer interaction can be modeled by the *Agent Communication Language* (ACL) proposed by the *Foundation for Intelligent Physical Agents* (FIPA, 2002a). In the framework of FIPA ACL, each message between two agents is a communicative act which contains information about the communicants, the content of the message and the conversation (e. g., interaction protocol and conversation ID). A conversation is a sequence of communicative acts. The interaction protocol restricts the number of those sequences (Odell et al., 2001). In (FIPA, 2002b), several types of communicative acts are proposed, such as:

Inform: the sender provides the recipient with information,

Query: the sender requests the recipient to perform a specific action and to provide the sender with the result of this action, and

Request: the sender requests the recipient to perform a specific action.

The content of a communicative act can be represented by an ontology. Established ontology languages are the *Resource Description Framework* (RDF) (Cyganiak et al., 2014) and the *Web Ontology Language* (OWL) (Schreiber and Dean, 2004).

Context Interface. The context manager is the component within the UI adaptation service that retrieves and maintains context models. An external context detection service provides the context manager with a description of the current context of use.

A context of use can be described by ontological models because ontologies provide a formal model that facilitates knowledge sharing and reuse across different entities (Strang and Linnhoff-Popien, 2003; Wang et al., 2004). Ontologies are a tool for the formal modeling of concepts and their interrelations (Gruber, 1993). A predefined set of inference rules allows to infer implicit knowledge from an ontology. As a specialization of logic programs, ontologies can also be translated into logic programs and extended with logical rules (Baader, 2010; Wang et al., 2004).

The Renderer Interface. At the renderer interface, the UI adaptation service provides a UI renderer with a description of the final UI. The format of this description highly depends on the targeted platform.

3.2 Transformation of Information into a User-friendly Representation

We formally describe the transformation of information from an application into a user-friendly representation as a function $\mathtt{trans}_{out}$ that maps from the input documents to an output document:

$$\mathtt{trans}_{out} : \mathcal{D}_{\mathcal{L}_{SA}} \times \mathcal{D}_{\mathcal{L}_{CDS}} \times \mathcal{D}_{\mathcal{L}_{UP}} \times \mathcal{D}_{\mathcal{L}_{DP}} \to \mathcal{D}_{\mathcal{L}_{out}},$$

where $\mathcal{D}_{\mathcal{L}_{SA}}$ denotes the set of documents in the language of *communicative acts*, $\mathcal{D}_{\mathcal{L}_{CDS}}$ the set of documents in the language of the *context detection service*, $\mathcal{D}_{\mathcal{L}_{UP}}$ the set of documents in the language of the *user preferences*, $\mathcal{D}_{\mathcal{L}_{DP}}$ the set of documents in the language of the *device profile*, and $\mathcal{D}_{\mathcal{L}_{out}}$ the set of documents in the language that is understood by the *renderer*. The function $\mathtt{trans}_{out}$ is a composition of the following functions:

- $\mathtt{trans\_pui} : \mathcal{D}_{\mathcal{L}_{SA}} \to \mathcal{D}_{\mathcal{L}_{PUI}}$ (transformation from speech acts to prototypical UIs),
- $\mathtt{adapt} : \mathcal{D}_{\mathcal{L}_{PUI}} \times \mathcal{D}_{\mathcal{L}_{CDS}} \times \mathcal{D}_{\mathcal{L}_{UP}} \times \mathcal{D}_{\mathcal{L}_{DP}} \to \mathcal{D}_{\mathcal{L}_{PUI}}$ (context-dependent adaptation of prototypical user interfaces), and
- $\mathtt{inst\_ui} : \mathcal{D}_{\mathcal{L}_{PUI}} \to \mathcal{D}_{\mathcal{L}_{out}}$ (instantiation of the output document),

where $\mathcal{D}_{\mathcal{L}_{PUI}}$ denotes the set of documents in the language of the *prototypical user interfaces*. A prototypical UI is a system- and implementation-independent description of a UI that is subject to adaptation. An overview over the composition of the functions is given in Figure 2.

Representation of Prototypical User Interfaces. In this work, we represent the prototypical user interface as a set of *elementary interaction objects* (eIOs), as introduced by (Christoph and Krempels, 2007). Elementary interaction objects are non-decomposable objects that enable a user to interact with a system.

To be applicable to the problem at hand, we extend the definition of eIOs by (Christoph and Krempels,

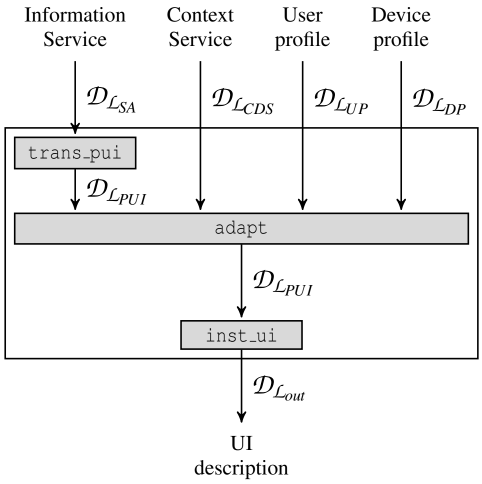

Figure 2: Overview of the transformation function $\texttt{trans}_{out}$ and its components.

2007). In this extended definition, an eIO has the following properties:

- *Type* of the interaction object, one of the following:
  - *Single select*: selection of one single element from a given list
  - *Multiple select*: selection of multiple elements from a given list
  - *Input*: input of information into the system
  - *Inform*: output of information to the user
  - *Action*: initiate an action in the system
- *Description* of the interaction object (optional)
- *Content* of the object, if needed
- *Content representation* (output device, representation medium, modality[1] and modality properties[2])
- *Natural language* of the content
- *Unique identifier* of the element for reference

Transformation of Communicative Acts into Prototypical User Interfaces. A communicative act is transformed into an eIO based on the performative, the identity of the sender and receiver and the associated interaction protocol. The performatives from (FIPA, 2002b) are classified into either one of two classes: *informs* or *requests*. Informs are used to inform the receiver of a message about a given subject without requesting any specific action. The purpose of requests is to invoke a specific behavior in the receiver of the message. Feedback about the results of an action can be part of this behavior. The associated interaction protocol specifies whether feedback is expected.

[1]e. g., text, image, or speech
[2]e. g., font color, size, or voice

As visualized in Table 1, the eIO types can be arranged into a two-dimensional space spanned by the performative class and the direction of the communication. This arrangement and the above-mentioned classification of performatives together define a mapping from communicative acts to eIOs.

Adaptation of Prototypical User Interfaces. Subject to adaptation is the content representation of every eIO. As shown in Figure 2, the adaptation of the eIO document is based on the context of use and the user and device profile information.

To resolve possible ambiguities and contradictions among the input documents, we define a hierarchy of the input documents: In case of a conflict between the device profile and the user profile the system discards the information from the user profile. The reason for this is that the device profile defines the technically possible means of communication, whereas the user profile defines abilities, but also preferences of the user. The implications from the context of use to the UI are assigned to the lowest priority because they are desirable, yet optional properties of the UI.

On a high level, the UI adaptation procedure works as follows:

1. Remove all preferences from the user profile that are in conflict with the device profile (this yields the *adjusted user profile*)
2. From the set of all theoretically possible content representations, determine the set of admissible content representations $R_{admissible}$
3. Find the best content representation(s) cr_{best} from $R_{admissible}$ for the current context C according to an evaluation function $\texttt{eval}$:

$$cr_{best} = \underset{cr \in R_{admissible}}{\operatorname{argmax}} \ \texttt{eval}(C, cr)$$

4. Set the properties of all eIOs according to cr_{best}, the device profile and the adjusted user profile

The adaptation system determines the set of admissible content representations based on the available output devices, their interaction capabilities, the abilities of the user and the message content. The message content imposes restrictions on the available modalities (e. g., image data cannot be represented by vibration).

The evaluation function $\texttt{eval}$ assigns a rating to a context and a content representation.

$$\texttt{eval} : \mathcal{C} \times R \rightarrow \mathbb{R},$$

where $\mathcal{C}$ denotes the set of all possible contexts and R the set of all content representations. A more detailed

Table 1: Characterization of eIOs by the direction of the communication and the interaction type.

		Direction	
		System → User	User → System
Performative class	Inform	Inform	Input, selection
	Request (without feedback)	Inform	Action
	Request (with feedback)	Inform + inputs and selections	Action

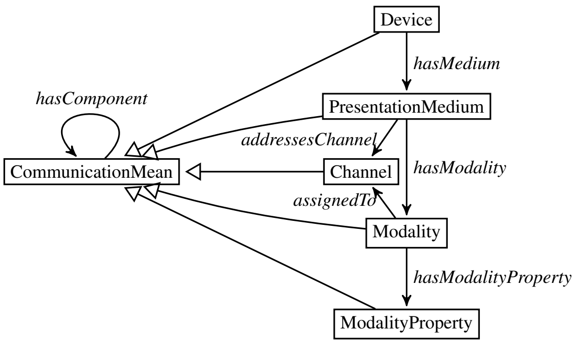

Figure 3: Communication infrastructure ontology.

description of `eval` is given in Section 3.3. If multiple content representations have an optimal rating, the renderer renders the UI using all optimal content representations concurrently.

Implementation of Content Representation Search. For the search of the best content representations, a compact representation of all admissible combinations of output devices, modalities and modality properties is needed. In the following, we refer to this representation as *communication infrastructure*. The communication infrastructure is modeled by an *RDF ontology*. Figure 3 shows the corresponding *RDF schema*.

A communication infrastructure has a hierarchical structure: on the first levels are available devices. Every device can usually access several presentation media. Each presentation medium addresses a specific communication channel and has a set of available modalities. Each modality is assigned to exactly one communication channel and has a set of possible modality properties. Due to this structure, a path from a device to a modality property describes a valid content representation. Besides, the abstract concept of communication means with `hasComponent` relations is included. `CommunicationMean` is as a superclass of all aforementioned communication means and `hasComponent` is a generalization of all other relations prefixed with "`has`". This allows for a more generic definition of the search algorithm.

The search algorithm is a uniform-cost search on the RDF graph of the communication infrastructure (Russell and Norvig, 2018). For the search, only `hasComponent` relations are considered. The search starts on all nodes with no incoming edge (root nodes). The cost of an edge is determined by the evaluation of the target node of the edge in the current context. The result of the search is the set of all best-rated paths from one of the root nodes to a leaf node. The rating for a path is the combination of the ratings of all edges on that path.

This algorithm is applicable to a wide variety of different device configurations because it relies on a generic graph structure for the representation of the communication infrastructure. Consequently, this algorithm is well-adapted to use cases with automated recognition of connected devices and easily customizable preferences.

Instantiation of the User Interface. The instantiation of the UI is highly platform-specific. Each mobile device platform provides a unique UI description language (Thornsby, 2016; D'areglia, 2018). Common to all platforms is the possibility to write applications with HTML, CSS and JavaScript, e. g., through Progressive Web Apps (Ater, 2017) or frameworks (Wargo, 2012). To support a broad range of platforms, it is sensible to use HTML, CSS and JavaScript for the definition of the final UI. It is possible to define a mapping from graphical eIOs to HTML and CSS and from auditive and tactile eIOs to JavaScript code snippets.

3.3 Suitability of User Interfaces in Contexts

The evaluation function `eval` (cf. Section 3.2) determines the suitability of a UI in a given context. In this section, we propose an implementation for the evaluation function. This implementation is targeted towards journey contexts.

Classification of Journey Contexts. Instead of classifying journey contexts into distinct context classes, such as in (Hörold et al., 2013), journey contexts are identified by *partial context descriptions* because many context aspects have to be considered in the evaluation. A partial context description is an expression that describes parts of a context. Partial context descriptions identify the class of all contexts that match the description and allow for very fine-grained context classes. For example, the partial context description "rainy and crowded" refers to the class of all contexts with rainy weather and a crowded environment. A partial context description can be implemented as a query in predicate logic.

Representation of an Rating. A rating can be represented by a real value $r \in [0,1] \subset \mathbb{R}$, where $r = 1$ stands for "suitable" and $r = 0$ for "unsuitable". The combination function $\oplus$ should meet the following requirements:

- $([0,1], \oplus)$ is a monoid with "suitable" (i.e., 1) as identity element.
- $\oplus$ is monotonically increasing, i.e., if $(r_1, r_2) \leq (r'_1, r'_2)$, then $r_1 \oplus r_2 \leq r'_1 \oplus r'_2$.
- The result of the combination of two ratings satisfies the rating semantics, i.e., $r_1 \oplus r_2 = 1$ means "suitable" and $r_1 \oplus r_2 = 0$ means "unsuitable".
- The result of a combination with "unsuitable" is always "unsuitable", i.e., $r \oplus 0 = 0$.

The multiplication function meets these requirements for the chosen value range. Consequently, the combination of two ratings r_1 and r_2 is their multiplication $r_1 \cdot r_2$.

Mapping from Contexts to User Interfaces. A content representation is a combination of multiple communication means. Due to the large variety of possible content representations, the assignment of a suitability value to each single content representation is hard. Let a context C be represented as a set of all possible partial context descriptions that hold in C. Then, the evaluation function can be represented as

$$\texttt{eval}(C, cr) = \bigoplus_{c \in C} \bigoplus_{cm \in cr} \texttt{eval}'(c, cm),$$

where $\texttt{eval}'$ is the evaluation of a single communication mean in a context.

The value of $\texttt{eval}'(c, cm)$ is retrieved from lookup tables. These lookup tables contain suitability evaluations of communication means assigned to partial context descriptions. Due to the high number of possible partial context descriptions[3] and the fact that most communication means are suitable in most contexts, the lookup table does not need to contain all possible evaluations. Instead, a default value is introduced and only those values are given that deviate from this default value. This default value is the identity element of $([0,1], \oplus)$, i. e., "suitable" or 1.

[3]For example, the following subset of the set of all partial context descriptions is already uncountably infinite: $\{\text{temperature} = x \mid x \in \mathbb{R}\}$.

3.4 Transformation of Information from the User

The proposed UI adaptation service is able to receive information from the user on any available input channel. This enables the user to freely choose the most suitable input channel. This decision is based on the assumption that the user knows best which communication channel is the most suitable one in the current context.

In general, two basic modes of human-computer interaction from the perspective of the user can be distinguished:

Proactive Communication: the user initiates a communication with the information service.

Reactive Communication: the user responds to a previous action of the information service according to an interaction protocol.

To support proactive communication, the client information service has to provide the adaptation service with a set of functions for the current application mode. For reactive communication, the adaptation service has to augment the prototypical UI with additional interaction objects for user input during the adaptation.

Processing User Input. The user input is first registered at the UI. Inputs are either GUI events or raw data (e. g., audio or video data). These inputs are forwarded to the UI adaptation service. If the input data type matches the expected data type in the interaction protocol of the conversation, the data is directly forwarded to the client. Otherwise, the adaptation service interprets the input data. An interpretation, for example, can be the translation of spoken speech into a textual representation, or the mapping of a touch event to an intent. The necessary interpretation procedure is determined based on the input and output data types of predefined interpretation procedures.

Multimodal systems employ a fusion engine for the coordination of multiple incoming data streams (Dumas et al., 2009). The proposed UI adaptation service supports concurrent multimodality (Nigay and Coutaz, 1993), so such coordination is not necessary here. Here, the task of the fusion engine is to link the incoming messages from the user to conversations. This linkage is possible by keeping track of which interaction object corresponds to which conversation.

At last, a communicative act is generated and sent to the client information service. The fields *conversation id*, *sender* and *receiver* are filled with the appropriate information and the raw or interpreted input data is added as the content of the communicative act.

3.5 Dialogue Manager

The dialogue manager provides and manages information about conversations (Dumas et al., 2009). This includes the communication history, the states of all conversations and the interaction protocols. Furthermore, the dialogue manager validates communicative acts in the context of a conversation and determines the expected type of data of a communicative act corresponding to an interaction protocol.

4 CONCLUSION AND FUTURE WORK

In this work, we proposed a context-dependent UI adaptation system. This system is a service that transforms messages between a system- and a user-oriented representation depending on the context of use.

Messages between an application and the adaptation service are represented as communicative acts and translated into a prototypical UI. Elementary interaction objects (eIOs) describe prototypical UIs, whereas an ontology models the context of use. The content representation of the eIOs is adapted to the context of use via a uniform-cost search on a communication infrastructure graph and an evaluation function. Finally, the prototypical UI is transcribed into a document that can be rendered and presented to the user by the targeted rendering engine. Messages from a user are received by the adaptation service, processed in the context of the associated conversation and forwarded to the client application as communicative acts.

In future work, suitable evaluation functions for specific domains should be defined, a functional prototype of the system should be implemented and the system should be tested in real-life environments with potential users, including developers and end-users of the UI. The system and the evaluation function should then be refined based on user feedback and observations from those tests.

Regarding the introduction of smartwatches, smart glasses, and smart speakers in the recent past, it is likely that users will interact with multiple devices on a regular basis. The proposed system can provide a framework for the seamless interaction between applications and users across different devices.

REFERENCES

Ater, T. (2017). *Building Progressive Web Apps: Bringing the Power of Native to the Browser*. O'Reilly Media.

Baader, F. (2010). *The Description Logic Handbook*. Cambridge University Press.

Baus, J., Krüger, A., and Wahlster, W. (2002). A resource-adaptive mobile navigation system. In *Proceedings of the 7th International Conference on Intelligent User Interfaces*, IUI '02, pages 15–22, New York, NY, USA. ACM.

Chen, G. and Kotz, D. (2000). A survey of context-aware mobile computing research. Technical report, Department of Computer Science, Dartmouth College, Hanover, NH, USA.

Christoph, U. and Krempels, K.-H. (2007). Automatisierte Integration von Informationsdiensten. *PIK - Praxis der Informationsverarbeitung und Kommunikation*, 30(2):112–120.

Christoph, U., Krempels, K.-H., von Stulpnagel, J., and Terwelp, C. (2010). Automatic context detection of a mobile user. *Proceedings of the International Conference on Wireless Information Networks and Systems (WINSYS), 2010*, pages 1–6.

Criado, J., Vicente Chicote, C., Iribarne, L., and Padilla, N. (2010). A model-driven approach to graphical user interface runtime adaptation. *CEUR Workshop Proceedings*, 641.

Cyganiak, R., Wood, D., and Lanthaler, M. (2014). Rdf 1.1 concepts and abstract syntax. Technical report, World Wide Web Consortium (W3C). Accessed on January 03rd, 2019 at http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/.

D'areglia, Y. (2018). *Learning iOS UI Development*. Packt Publishing.

Dey, A. K. (2001). Understanding and using context. *Personal and Ubiquitous Computing*, 5(1):4–7.

Dumas, B., Lalanne, D., and Oviatt, S. (2009). Multimodal interfaces: A survey of principles, models and frameworks. In *Lecture Notes in Computer Science*, pages 3–26. Springer Berlin Heidelberg.

Edwards, W. K. and Mynatt, E. D. (1994). An architecture for transforming graphical interfaces. In *Proceedings of the 7th annual ACM symposium on User interface software and technology - UIST '94*. ACM Press.

Eis, A., Klose, E. M., Hegenberg, J., and Schmidt, L. (2017). Szenariobasierter prototyp für ein reiseassistenzsystem mit datenbrillen. In Burghardt, M., Wimmer, R., Wolff, C., and Womser-Hacker, C., editors, *Mensch und Computer 2017 – Tagungsband*, volume 17, pages 203–214, Regensburg, Germany. Gesellschaft für Informatik e.V., Gesellschaft für Informatik e.V.

Falb, J., Rock, T., and Arnautovic, E. (2006). Using communicative acts in interaction design specifications for automated synthesis of user interfaces. In *21st IEEE/ACM International Conference on Automated Software Engineering (ASE 2006)*. IEEE.

FIPA (2002a). FIPA ACL Message Structure Specification. Technical Report SC00061G, Foundation for Intelligent Physical Agents (FIPA), Geneva, Switzerland.

FIPA (2002b). FIPA Communicative Act Library Specification. Technical Report SC00037J, Foundation for Intelligent Physical Agents (FIPA), Geneva, Switzerland.

Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220.

Hong, J., Suh, E., and Kim, S. (2009). Context-aware systems: A literature review and classification. *Expert Systems with Applications*, 36(4):8509–8522.

Hörold, S., Mayas, C., and Krömker, H. (2013). Analyzing varying environmental contexts in public transport. In Kurosu, M., editor, *Human-Computer Interaction. Human-Centred Design Approaches, Methods, Tools, and Environments*, pages 85–94, Berlin, Heidelberg. Springer Berlin Heidelberg.

Jaimes, A. and Sebe, N. (2007). Multimodal human–computer interaction: A survey. *Computer Vision and Image Understanding*, 108(1-2):116–134.

Johnson, D. A. and Trivedi, M. M. (2011). Driving style recognition using a smartphone as a sensor platform. In *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 1609–1615.

Kolski, C., Uster, G., Robert, J.-M., Oliveira, K., and David, B. (2011). Interaction in mobility: The evaluation of interactive systems used by travellers in transportation contexts. In *Human-Computer Interaction. Towards Mobile and Intelligent Interaction Environments*, pages 301–310. Springer Berlin Heidelberg.

Lemmelä, S., Vetek, A., Mäkelä, K., and Trendafilov, D. (2008). Designing and evaluating multimodal interaction for mobile contexts. In *Proceedings of the 10th international conference on Multimodal interfaces - IMCI '08*. ACM Press.

Mitrevska, M., Moniri, M. M., Nesselrath, R., Schwartz, T., Feld, M., Korber, Y., Deru, M., and Muller, C. (2015). SiAM - situation-adaptive multimodal interaction for innovative mobility concepts of the future. In *2015 International Conference on Intelligent Environments*. IEEE.

Nigay, L. and Coutaz, J. (1993). A design space for multimodal systems. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '93*, pages 172–178, Amsterdam, The Netherlands. ACM Press.

Odell, J., Parunak, H. V. D., and Bauer, B. (2001). Representing agent interaction protocols in uml. In Ciancarini, P. and Wooldridge, M., editors, *Agent-Oriented Software Engineering*, pages 121—140. Springer.

Pielot, M., Poppinga, B., Heuten, W., and Boll, S. (2012). PocketNavigator: Studying tactile navigation systems in-situ. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 3131–3140, New York, NY, USA. ACM.

Pielot, M., Poppinga, B., Vester, B., Kazakova, A., Brammer, L., and Boll, S. (2010). Natch: A watch-like display for less distracting pedestrian navigation. In Ziegler, J. and Schmidt, A., editors, *Mensch & Computer 2010: Interaktive Kulturen*, pages 291–300, München. Oldenbourg Verlag.

Rehman, U. and Cao, S. (2016). Augmented-reality-based indoor navigation: A comparative analysis of handheld devices versus google glass. In *IEEE Transactions on Human-Machine Systems*, volume 47, page 1–12. Institute of Electrical and Electronics Engineers (IEEE).

Russell, S. and Norvig, P. (2018). *Artificial Intelligence: A Modern Approach, Global Edition*. Addison Wesley.

Samsel, C., Dudschenko, I., Kluth, W., and Krempels, K.-H. (2015). Using wearables for travel assistance. *Proceedings of the 11th International Conference on Web Information Systems and Technologies*.

Schreiber, G. and Dean, M. (2004). OWL web ontology language reference. https://www.w3.org/TR/owl-ref/. Accessed on November 6th, 2018.

Smith, A. (2014). More than half of cell owners affected by 'distracted walking'. http://www.pewresearch.org/fact-tank/2014/01/02/more-than-half-of-cell-owners-affected-by-distracted-walking/. Accessed on 15.09.2018.

Strang, T. and Linnhoff-Popien, C. (2003). Service interoperability on context level in ubiquitous computing environments. In *International Conference on Advances in Infrastructure for Electronic Business, Education, Science, Medicine, and Mobile Technologies on the Internet*.

Thornsby, J. (2016). *Android UI Design*. O'Reilly Media.

Vanderdonckt, J., Limbourg, Q., Michotte, B., Bouillon, L., Trevisan, D. Q., and Florins, M. (2004). USIXML: a user interface description language for specifying multimodal user interfaces. In *Proceedings of W3C Workshop on Multimodal Interaction WMI*.

Wang, X. H., Zhang, D. Q., Gu, T., and Pung, H. K. (2004). Ontology based context modeling and reasoning using OWL. In *IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second*. IEEE.

Wargo, J. M. (2012). *PhoneGap Essentials: Building Cross-platform Mobile Apps*. Addison-Wesley Professional.

Zargamy, A., Sakai, H., Ganhör, R., and Oberwandling, G. (2013). Fußgängernavigation im urbanen Raum - Designvorschlag. In Boll, S., Maaß, S., and Malaka, R., editors, *Mensch & Computer 2013: Interaktive Vielfalt*, pages 365–368, Munich, Germany. Oldenbourg Verlag.