Towards a Context-Sensitive User Interaction Framework for Information Systems

Stephan Kölker[1](✉), Felix Schwinger[1,2](✉), and Karl-Heinz Krempels[1,2](✉)

[1] Information Systems, RWTH Aachen University, Aachen, Germany
stephan.koelker@rwth-aachen.de, {schwinger,krempels}@dbis.rwth-aachen.de
[2] Fraunhofer Institute for Applied Information Technology FIT, Aachen, Germany

Abstract. With the rise of mobile devices, users operate applications in a large variety of contexts. In each of these contexts, a user may have different requirements and preferences regarding an application's user interface. The context describes the current physical and social environment of the user, his activity and locomotion, as well as the current location and time. Hence, different user interfaces may be more suitable in specific contexts than others. At the moment, a user interface is an integral part of an application, consequently also limiting its usefulness in certain contexts as the information is not presented in the best possible way. Therefore, we propose a way to decouple the user interface from a specific form of representation. Through this decoupling, it is possible to dynamically adapt user interfaces to the user's specific needs in a context, to increase the value of the application for the user. In this paper, we introduce a general framework for the context-dependent adaptation of user interfaces and evaluate it in the specific context of travel information systems. Travel information systems are particularly suited to evaluate such a framework, as they are usually operated in many different contexts – before a trip, during a trip, and after a trip. The adaptation framework transforms between system-oriented messages and user-oriented messages. The user's context, the output device capabilities, and the user's preference, all influence the choice of the actual representation for user-oriented messages. We implemented a prototype of the proposed system and conducted an experimental evaluation focusing on scenarios from the domain of travel information systems.

Keywords: Context · Context-aware applications · User interface · Human-computer interaction · Adaptation

1 Motivation and Introduction

In the last two decades, smartphones have emerged and had since become widely adopted. A survey from 2019 by the Pew Research Center showed that 81% of U.S. adults own a smartphone[1]. With the rise of the adoption rate, the

[1] Pew Research Center, "Mobile Fact Sheet", https://www.pewresearch.org/internet/fact-sheet/mobile/, viewed January 15, 2020.

A. Bozzon et al. (Eds.): WEBIST 2019, LNBIP 399, pp. 195–217, 2020.
https://doi.org/10.1007/978-3-030-61750-9_9

importance of applications for smartphones also grows. People use these mobile devices mainly as a means for communication and entertainment – but also as mobile information systems. Owing to their high portability, users employ devices in diverse contexts and environments, such as at home, at work, or while traveling.

Users of mobile applications may have specific preferences and requirements in different contexts regarding the *user interface (UI)*. For example, the interaction with a *graphical user interface (GUI)* of a navigation system usually distracts the vision of the driver. The consequence is a highly increased crash potential [38]. Hence, to maximize the UI's utility and minimize the distraction level, the inclusion of the user's context is essential. In this paper, we define *context* as a set of information describing the environment, time and location and activity of an entity. It consists of information about the object itself, its physical and social environment, and all other entities that currently exert an influence on it. Such an entity may be a human, an inanimate object, or a place [11,40].

Manufacturers of mobile devices equip their devices with a wide variety of different sensors, including accelerometers, gyroscopes, magnetometers, GPS, thermometers, WiFi and Bluetooth, cameras, microphones, or light sensors. The data from those sensors allow mobile applications to sense parts of the current context of use [24]. Through sensor fusion, the data from the sensors is correlated and verified to increase the accuracy of the context detection. While the discovery of the current context is crucial for the framework, we do not further discuss the actual context detection and elicitation itself in this paper, but instead, require that such a context detection system exists.

In addition to the context derived from the environment through the sensors, users may also have a domain-specific context. In this paper, we primarily examined contexts of trips from one location to another location, as a traveler has a variety of different contexts during such a trip [22,46]. These contexts include, among others, the situations before a trip, during a trip, and after a trip. Each of these contexts require different kinds of UIs [26]. While, for example, a car navigation system is suited very well for the context of not distracting the driver, it is not well suited for other, more general purposes [36]. When regarding public transportation, there are a variety of contexts that favor certain representations of the information system to the users. For example, when walking with luggage in both hands, using a GUI-based information system on the mobile phone is cumbersome. In such a scenario, a voice- and audio-based information system is more suitable.

UIs of nearly all mobile applications, including most mobile information systems, are predesigned in the form of GUIs. With the appearance of smartphones, the input for those GUIs is mostly tactile, while the output is nearly always graphical [15]. However, as illustrated by the examples above, the hard-coding of UIs with tactile input and graphical output impacts the usability of the information system in certain contexts. In specific contexts, the usage of particular UIs may even be dangerous, e.g., interacting with a GUI while driving. Additionally, such an approach cannot be generalized to run the same program on, for example, a smart speaker.

Most mobile devices nowadays do not only contain a variety of sensors, but also different output modalities. These include, among others, the display, the speaker, and notification lights. Specific applications already support separate input and output modalities. Voice Assistants such as Apple Siri, Google Assistant, or Microsoft Cortana already support the microphone coupled with speech-detection as an input modality. Furthermore, the output modality depends on the answer of the system; the system sometimes answers using the speaker, while it displays others graphically on the device. Similarly, an application may choose the output modality based on the current context of the user.

To aid the development of such applications, we propose a framework to decouple an application from the way the application communicates with the user. Hence, our proposed framework can render messages between the user and the system in different modalities. The system chooses the currently best available modality for the current context as the output modality. Additionally, a user may also interact with the system using various input modalities. As a consequence, a method for finding the best available output modality, based on the user's context, their preferences, and device capabilities is required. Such an adaptation framework can increase the number of situations where users can effortlessly access information systems.

With our previous research, we sketched and described a framework for the adaptation of user interfaces to the current context of use, the capabilities of the available devices and the user's preferences [25]. Hereby, we extend this work by introducing the framework itself in more detail and by presenting a first evaluation of the proposed methods.

The remainder of the paper is structured as follows: Sect. 2 introduces the current State-of-the-Art regarding multimodal UIs and adaptation systems. In Sect. 3, we present our approach to the problem, whereas Sect. 4 evaluates this approach. Finally, Sect. 5 concludes the paper with a summary and highlights future work.

2 State of the Art

Currently, most developers use a predesigned UI for travel information systems that are optimized for specific contexts. Applications, such as a car navigation system or Google Maps, are optimized for a specific context of use. For smartphones, GUIs are the predominant form of presenting information to the user, whereas the user commonly interacts with the system using tactile feedback [15]. Car navigation systems, optimized for a single use case, also make use of bidirectional speech-based interaction between system and user as not to distract the driver from the road [36]. While we focus on a framework for the automatic adaptation of user interfaces, it is still essential to look at the wider literature of context-aware computing. As context-aware computing, in general, is already well-studied, we refer the reader to recent surveys in the area for a broader overview of the subject: [5,14,21,23]. The following excerpt of related work either focuses on context-aware travel information systems or the automatic adaptation of user interfaces. This section is also largely adapted from our previous work [25].

Mitrevska et al. developed a context-aware in-car information system [29]. Users can interact with particular entities in the environment of the car, e.g., a restaurant. They can either retrieve additional information about these entities or access associated services for this entity, e.g., reserving a table in a specific restaurant. Users can interact with the system via speech, gesture, and displays. However, the UI is not dynamically adapted to the context of use and focuses on being used while driving a car.

Vanderdonckt et al. introduced a language for describing multimodal UIs on different levels of abstraction [42]. The XML-based language allows supporting a device- and modality-independent development of UIs. Furthermore, it can transform a UI from one representation to another one. Lemmelä et al. proposed a manual iterative process for designing multimodal mobile information systems [28]. During the design process, UI developers create multiple UIs, where each of these UIs are manually adapted to different contexts of use. The automated synthesis or transformation of UIs for the different contexts is, however, not examined.

Falb et al. proposed a system to automatically synthesize UIs using a discourse model [17]. This discourse model is a meta-model for the description of interactions. It allows representing human-computer interaction with communicative acts. The system can automatically generate UIs for various devices based on the interaction description. However, it does not take the usage context into account.

Currently, researchers are experimenting with novel device classes for travel information systems. One of these device classes are, for example, *smartwatches* [32,35,45]. Smartwatches have a reasonably small display but are more easily accessible compared to a smartphone. Additionally, Pielot et al. examined the suitability of *vibration* for communicating the next navigational instructions to travelers [31]. The main advantage here is that the traveler is not distracted by this communication form. Eis et al. introduce a travel navigation system with *smart glasses* [16]. Smart glasses have the advantage that information can be overlayed onto a view of the real-world and thus creating an augmented reality view. Finally, Rehman et al. compare navigation on hand-held devices with navigation on smart glasses [33].

Moreover, there is also literature on the dynamic adaptation of UIs to some external circumstances. Baus et al. presented a pedestrian navigation system in 2002 [3]. Their system automatically provided the user with context-depending information, while also adapting the presentation form to the capabilities of the employed hardware and the current information needs of the user.

Christoph et al., in turn, described a process for the dynamic adaptation of UIs to hardware capabilities and user preferences [6,7]. They represent UIs as a sequence of so-called elementary interaction objects *(eIOs)*. A system based on XSLT rules dynamically adapts these XML-based eIOs. Lastly, Criado et al. discussed a model-driven approach for the dynamic adaptation of GUIs [8]. They present a meta-model for the description of GUIs. This meta-model allows the incremental adjustment of GUIs while they are running.

3 Approach

In this section, we present our previous work, the automated *user interface* (UI) adaptation system from Kölker et al. [25], in more detail. This system is responsible for suitably giving the message content from an information system to the user, and for providing an information system with, optionally inferred, user input. Since generalizability is one of the leading design goals of the system, we designed it as an independent service that interacts with various client systems and other system components through specified interfaces.

Figure 1 shows an outline of the adaptation system and its interfaces. In order to present information to the user, a client information system sends its information as a message to the UI adaptation service. The multimodal fission component of the UI adaptation service is then responsible for the transformation of this message into a context-dependent and user-friendly representation. To this end, the multimodal fission component utilizes conversation information from the dialogue manager and information about the context of use from the context manager. The dialogue manager keeps track of the state of all ongoing and past conversations between the client application and the user and thus, is capable of providing information about the conversational context. The context manager interacts with an external context detection service to retrieve information about the current context of use, including user and device profiles and characteristics

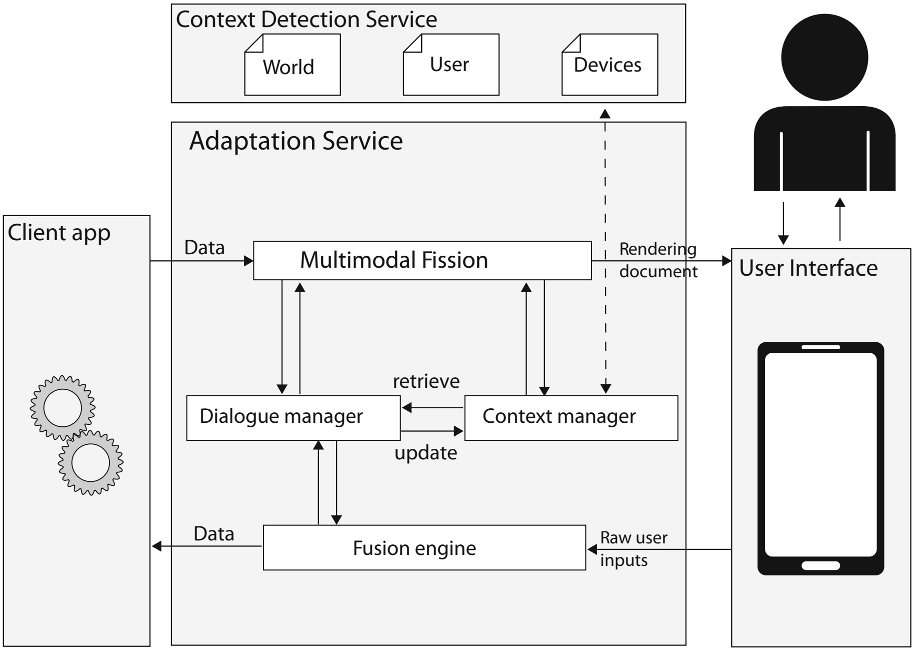

Fig. 1. Overview over the components of the UI adaptation service, from [25].

of the physical and social environment of the user. An external rendering engine then renders the output of the multimodal fission component according to the previously determined appropriate representation to present the information to the user.

The UI adaptation service always listens for user inputs. When user input is registered, the UI adaptation service extracts the information from the user input and provides the targeted information service with that information. The fusion engine component of the UI adaptation service is responsible for integrating the information from multiple sources, such as the user input and conversational information, to interpret the information.

3.1 Service Interfaces

The UI adaptation service acts as a broker between the client information system and the user while utilizing an external renderer to render the user interface. Besides the client information system and the renderer, it interfaces with a context detection service. The context detection service provides context information to the UI adaptation service. This context information is the basis for the UI adaptation.

The Information Service Interface. The interface to the information service treats messages as communicative acts. Communicative acts are messages between intelligent agents. They were formalized in the *Agent Communication Language* (ACL) by the *Foundation for Intelligent Agents* (FIPA) [18].

By performing a communicative act, an agent can execute a specific action (performative). These actions may include informing the recipient about a fact, querying information from the recipient, or requesting the recipient to perform a particular action. Communicative acts are usually part of a conversation between agents. Conversations are sequences of communicative acts between a set of agents. In the framework of the ACL, unique identifiers in messages allow the mapping of communicative acts to conversations. Conversations can adhere to an interaction protocol that imposes rules on this communication.

Ontologies ensure that the recipient can correctly interpret the message content of communicative acts. Ontologies are formal representations of specific domains and define concepts and their interrelations [19]. They facilitate knowledge exchange and reuse due to their formality [40,43]. Consequently, the content of communicative acts should be a document that can be parsed and inferred with a pre-defined ontology that all communication participants have access to. Widely used languages for defining ontologies are the *Resource Description Framework* (RDF) [9] and the *Web Ontology Language* (OWL) [37].

Context Interface. A vital factor to consider for the adaptation is the current context in which a user operates an application. This context consists of not only the device-internal context information, such as active applications, available resources, and user profiles but also the broader environmental context of use, such as the current weather, traffic information, the currently used means of transport, the social environment and the constitution of the user itself. Within the UI adaptation service, information on the context of use is retrieved and maintained by a context manager in the form of context models. An external context detection service provides this information.

At the context detection interface, ontologies are used to represent the context of use. Several works have shown that ontologies are especially useful for describing contexts due to their formality and knowledge exchange capabilities [40,43]. The formality also allows the automated deduction of implicit knowledge from explicitly provided knowledge by inference rules. Commonly used ontology languages already contain a specific set of built-in inference rules [20]. However, in the case that more expressivity is needed, logical rules can be used to infer implicit knowledge based on ontologies because ontologies are a specialization of logic programs [2,43].

The Renderer Interface. The UI adaptation service uses an abstract representation of UIs during the adaptation process. The framework translates the abstract UI representation into a language that can be rendered by an external rendering engine. This process allows the utilization of standard renderers to present the information to users. This language is usually highly platform-specific.

3.2 Transformation of Information into a User-Friendly Representation

The transformation of information into a user-friendly representation is a function $\texttt{trans}_{out}$ that maps a tuple of documents in the language of *communicative acts* ($\mathcal{D}_{\mathcal{L}CA}$), the *context detection service* ($\mathcal{D}_{\mathcal{L}CDS}$), *user preferences* ($\mathcal{D}_{\mathcal{L}UP}$), and *device profiles* ($\mathcal{D}_{\mathcal{L}DP}$) into documents in a language that the *renderer* can render ($\mathcal{D}_{\mathcal{L}out}$) [6,25]:

$$\texttt{trans}_{out} : \mathcal{D}_{\mathcal{L}CA} \times \mathcal{D}_{\mathcal{L}CDS} \times \mathcal{D}_{\mathcal{L}UP} \times \mathcal{D}_{\mathcal{L}DP} \rightarrow \mathcal{D}_{\mathcal{L}out},$$

This transformation process consists of several distinct steps. Hence, $\texttt{trans}_{out}$ can be implemented as a composition of the following functions:

- $\texttt{trans\_pui} : \mathcal{D}_{\mathcal{L}CA} \rightarrow \mathcal{D}_{\mathcal{L}PUI}$: transforms communicative acts to *prototypical UIs* (PUI),
- $\texttt{adapt} : \mathcal{D}_{\mathcal{L}PUI} \times \mathcal{D}_{\mathcal{L}CDS} \times \mathcal{D}_{\mathcal{L}UP} \times \mathcal{D}_{\mathcal{L}DP} \rightarrow \mathcal{D}_{\mathcal{L}PUI}$: adapts the prototypical UI according to the context, the device profile and the user profile
- $\texttt{inst\_ui} : \mathcal{D}_{\mathcal{L}PUI} \rightarrow \mathcal{D}_{\mathcal{L}out}$: instantiates the output document

Instead of directly representing the UI in a platform-specific UI language, an abstract platform-independent language is chosen for this task. Here, this language is referred to as "*prototypical UI*" and is further explicated in the next subsection. Figure 2 gives a visual summary of the implementation of the function $\texttt{trans}_{out}$.

Representation of Prototypical User Interfaces. Christoph et al. describe user interfaces in terms of *elementary interaction objects* (eIOs) [6]. We define elementary interaction objects as "non-decomposable objects that enable a user to interact with a system" with the following properties [25]:

- *Type* of the eIO, 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
- Optionally, a *description* of the interaction object
- Optionally, a *content* of the object
- A *content representation*, consisting of
  - *Output device*: on which device the eIO is presented to the user,
  - *Presentation medium*: which medium is used to present the eIO to the user, e.g., display or speaker,
  - *Modality*: the way the information is represented [4], e.g., text, image, or speech, and
  - *Modality properties*: specific modality-dependent representation parameters, e.g., font color, size, or voice
- *Natural language* of the content, e.g., English (en), German (de), Chinese (zh), ...
- *Unique identifier* of the element for reference

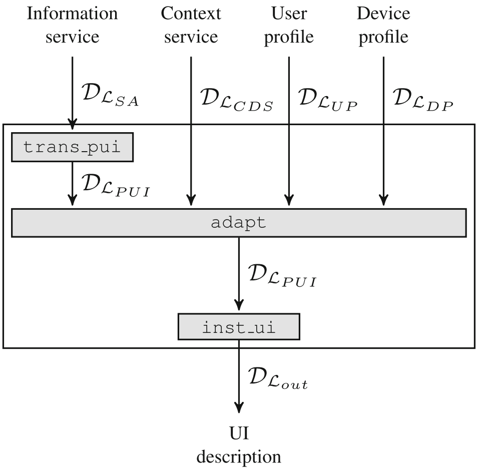
Information service
Context service
User profile
Device profile
$\mathcal{D}_{\mathcal{L}_{SA}}$
$\mathcal{D}_{\mathcal{L}_{CDS}}$
$\mathcal{D}_{\mathcal{L}_{UP}}$
$\mathcal{D}_{\mathcal{L}_{DP}}$
trans_pui
$\mathcal{D}_{\mathcal{L}_{PUI}}$
adapt
$\mathcal{D}_{\mathcal{L}_{PUI}}$
inst_ui
$\mathcal{D}_{\mathcal{L}_{out}}$
UI description

Fig. 2. Implementational view on $\texttt{trans}_{out}$ as a composition of partial transformations. This visualization is from our previous work [25].

For the task of adaptation, the eIO type and content representation are of particular interest. Therefore, the focus of the following descriptions lies on these two properties.

Transformation of Communicative Acts into Prototypical User Interfaces. Communicative acts from the client have to be transformed into prototypical user interfaces to generate an adaptable user interface. The transformation depends on the performative, the direction of the communication, and the corresponding interaction protocol. For this transformation, only two different classes of performatives need to be considered: *informative* and *requesting* performatives. The former class of performatives consists of all performatives that are used to inform other agents about a given subject without requesting any specific action. Conversely, all performatives that are used to request an action from other agents are assigned to the latter class.

Depending on the direction of the communicative act, a different set of eIOs is available for the transformation. If the recipient of the communicative act is the user, the communicative act is transformed into an *inform* eIO because the other eIO types are for inputting information into the system. Hence, a communicative act from the user to the system can be either translated into a *select*, *input*, or *action* eIO. The system translates communicative acts with an informative performative into either select or input eIOs depending on the message content. In contrast, communicative acts with a requesting performative are translated into action eIOs. The mapping of communicative acts to eIOs is visualized in Table 1.

Besides generating eIOs for presenting the message content to the user, possible further eIOs are includable to provide the user with means to respond, such as input or action eIOs. The inclusion of these eIOs depends on whether feedback from the user is expected. This inclusion can be checked based on the interaction protocol, the conversation history, and the current communicative act.

Table 1. Characterization of eIOs by the direction of the communication and the interaction type [25].

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

Adaptation of Prototypical User Interfaces. After the translation of communicative acts into eIOs, a suitable user interface representation is to be determined. To this end, the framework adapts the content representation of every eIO to the context of use, device, and user profile.

The implications to the user interface from each of those documents are assigned an importance value to avoid ambiguous or contradictory information.

If information from one document is conflicting with information from another, the information from the more important document is considered relevant. The implications from the device profile are the most important because the device profile defines the technically possible means of communication. The second most important information comes from the user profile. The user profile contains information about how the user can and wants to communicate. The context of use has the least significant impact on the user interface.

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

1. Calculate the *adjusted user profile* by removing all preferences from the original user profile that are conflicting with the device profile,
2. Determine the set of all admissible content representations $R_{admissible}$ from the set of all theoretically possible content representations based on the device profile(s), the user profile, and the message content,
3. Determine the most suitable content representation(s) r_{best} from $R_{admissible}$ for the current context of use C according to an evaluation function `eval`:

$$r_{best} = \underset{r \in R_{admissible}}{argmax} \; \texttt{eval}(C, r)$$

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

The set of admissible content representations is determined based on the available output devices, their interaction capabilities, the abilities of the user, and the message content. The latter restricts the possible modalities. For example, vibration is usually unsuitable to represent textual information.

If multiple different content representations have an equally optimal rating, the message is represented in all optimal content representations concurrently. Thus, one copy of the respective eIOs for each optimal content representation is created.

Representation of Contexts. In this work, we represent context as a consistent knowledge-base with context information. Then, a logical expression α defines a context class consisting of all contexts that entail α. We refer to such logical expressions α as *partial context descriptions* because they are used to define context classes by describing parts of a context [25]. For example, the logical expression $\alpha = weather(rainy) \wedge environment(crowded)$ defines the class of all contexts with rainy weather and a crowded environment. The advantage of this approach over the manual classification of distinct context classes is its flexibility and the possibility to define very fine-grained context classes. These properties are required in this use case due to the complex relationship between contexts of use and user interfaces.

Evaluation Function. Let $\mathcal{C}$ denote the set of all possible contexts, R the set of all content representations and $E \subseteq \mathbb{R}$ the set of all ratings ranging from "unsuitable" to "suitable". Then, the following function assigns a rating to each admissible combination of context and content representation:

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

To combine different ratings into a single one, we define a combination function. Ideally, this combination function $\oplus$ is chosen such that the result of combining different ratings is consistent with the definition and semantics of ratings.

Instantiation of the User Interface. Since the UI description languages differ among different mobile device platforms [10,41], the translation of the prototypical UI into a final UI is highly platform-specific. As an alternative to the native UI description languages of widely used mobile device platforms, developers can also write applications in HTML, CSS, and JavaScript by using techniques like Progressive Web Apps [1] or specific frameworks [44]. Besides, renderers and interpreters for HTML, CSS, and JavaScript are available on all major mobile platforms because they are one of the core components of Web browsers. Therefore, the translation of the prototypical UI into a UI described by HTML, CSS, and JavaScript reduces implementation effort because a broad range of platforms is supported by default. HTML and CSS mainly describe the visual appearance of a UI, while JavaScript code snippets can achieve auditive and tactile outputs. Therefore, a mapping from a prototypical UI in terms of eIOs to a collection of HTML, CSS, and JavaScript documents can be defined.

3.3 Transformation of Information from the User

Assuming that the user will, by default, choose the most suitable communication channel to communicate with an information system, the communication channel for user input does have to be selected by the UI adaptation service. Instead, the UI adaptation service needs to be able to receive input from all possible inbound communication channels simultaneously. Specifically, this means that input, selection, and action eIOs will always be rendered on all possible incoming communication channels concurrently and that only the modality and modality properties need to be selected for them.

From the perspective of the user, one can distinguish between *proactive* and *reactive* communication. Proactive communication is initiated by the user, while the user only responds to messages from the system in a reactive communication. Both communication modes have to be handled differently by the adaptation service. The adaptation service has to provide the user with access to the currently available functions of the client information system to allow the user to initiate a communication. The inclusion of additional interaction objects enables reactive communication that give the user the possibility of reacting to a message from the client information system. The necessity of including these other interaction objects depends on the communicative act, the conversation history, and the interaction protocol of the respective conversation.

Processing User Input. Generally, inputs from the user are received either as GUI interaction or raw data. The user generates GUI events during the interaction with a GUI. Recording the real world directly produces raw data, e.g., by recording audio or video data. The system forwards the data that is input at the UI to the UI adaptation service for further processing. This additional processing depends on whether the data type of the input data matches the expected data type in the interaction protocol of the associated conversation. In case of a match, the data can be directly sent to the client information service. Otherwise, the data has to be interpreted by the UI adaptation service. For example, if the user inputs spoken words, although textual data is required according to the interaction protocol, the adaptation service needs to transcribe the words and send the transcription to the client. Similarly, GUI events, such as touches or clicks, can be mapped to intents. The interpretation procedures are designed as pre-defined building blocks by the developer of the adaptation service with a specific input and output data type. Then, a suitable interpretation pipeline can be generated automatically by assembling several of such building blocks to a pipeline.

According to Dumas et al., the coordination of multiple incoming data streams is the task of a fusion engine [14]. For the proposed adaptation service [25] such coordination is not necessary because it only supports concurrent multimodality [30]. Therefore, the only task of the fusion engine [25] is to link the incoming messages from the user to conversations by storing the corresponding conversation ID in every interaction object.

After the incoming data has been processed successfully, the adaptation service generates a communicative act containing the processed data and sends it to the client information service.

3.4 Dialogue Manager

As stated by Dumas et al., the task of a dialogue manager is to provide and manage information about conversations [14]. Specifically, the dialogue manager maintains the communication history, keeps track of the states of all conversations, and provides information about the associated interaction protocols. Additionally, based on the interaction protocol, the dialogue manager can also check the validity of communicative acts in a conversation. It can also provide the expected data type of the content of the next communicative act in a conversation.

3.5 Implementation

For the evaluation of the proposed UI adaptation system, we implemented a prototype for the adaptation procedure. This prototype transforms messages from client information systems into descriptions of the output representation together with a suitability rating using the previously described function $\texttt{trans}_{out}$. Device profiles, user profiles, and descriptions of the context of use influence the adaptation process. In the following, some implementational details are presented.

Evaluation Function. In the prototype, we represent a rating as a real value in $[0, 1]$ [25]. A rating of $\texttt{eval}(C, r) = 1$ means that the representation r is suitable in context C, whereas $\texttt{eval}(C, r) = 0$ means that it is unsuitable. For the selected range of ratings, the multiplication function is a reasonable choice for the combination function [25].

Evaluation of User Interfaces in Contexts. The evaluation function, as listed before, assigns a rating to a tuple of context and content representation. However, since the number of possible contexts is generally infinite and the number of content representations high, such an assignment is impossible to define manually. As a simplification, only a rating for the combination of a specific context class and a component of a content representation is set, while all other combinations are rated as "suitable" $(r = 1)$ by default. Therefore, the evaluation function is decomposable as follows:

$$\texttt{eval}(C, r) = \bigoplus_{\gamma : C \models \gamma} \bigoplus_{cm \in r} \texttt{eval}'(\gamma, cm).$$

Here, $\texttt{eval}'(\gamma, cm)$ is the evaluation of a single communication mean cm in a context class defined by γ. In our prototype, this evaluation is defined by lookup tables.

In the prototype, three different tables are used: a table for evaluating devices, communication channels, and modalities, respectively. The evaluation of modality properties is omitted here due to a large number of possible modality properties. Table 2, Table 3, and Table 4 show the lookup tables. The values in the lookup tables are based on common knowledge and thought experiments.

According to the communication infrastructure, a presentation medium addresses a specific communication channel. Since presentation media with the same addressed communication channel share essential properties concerning their suitability during a journey, presentation media are evaluated according to the channel that they address.

In the experimental evaluation, we consider five major classes of mobile devices: tablets, smartphones, smart watches and smart glasses. The evaluation of devices in Table 2 is mainly based on their accessibility in the given contexts. For example, when transporting luggage, hand-held devices are far less accessible than wearables and, thus, are assigned a considerably worse rating. Furthermore, the general accessibility determines the ranks of the devices in all contexts.

Table 3 introduces the evaluation of communication channels. We chose the visual, auditive and the tactile communication channel because they are the main communication channels in contemporary information systems [13]. Various environmental factors have a considerable impact on the human-device communication. Consequently, their existence results in a reduction of the rating. For example, a noisy environment halves the rating of all content representations using the auditive channel. Thus, it becomes unlikely for the system to choose such a content representation in that situation.

Table 2. Evaluation of information devices.

Context	Tablet	Smart-phone	Smart watch	Smart glasses	Board computer
All contexts	0.96	0.97	0.98	1.0	0.99
One-handed luggage transport	0.1	0.25	1.0	1.0	1.0
Two-handed luggage transport	0.1	0.1	0.5	1.0	1.0
Standing inside a moving vehicle	0.1	0.25	1.0	1.0	1.0
Standing inside a moving vehicle $\wedge$ one-handed luggage transport	0.1	0.1	0.5	1.0	1.0
Active movement $\wedge$ smartphone not in visual field	0.0	0.0	0.75	1.0	1.0
Active movement $\wedge$ smartphone in visual field	0.0	1.0	0.75	1.0	1.0

Table 3. Evaluation of communication channels.

Context	Visual	Auditive	Tactile
All contexts	1.0	1.0	1.0
High ambient light level	0.5	1.0	1.0
No sight contact	0.0	1.0	1.0
Vibrating environment	0.75	1.0	0.75
No body contact	1.0	1.0	0.0
Noisy environment	1.0	0.5	1.0
"Do not disturb"	1.0	0.0	0.5
Active movement $\wedge$ device not in visual field	0.9	1.0	1.0

Table 4 presents the evaluation of modalities. We distinguish between nine major classes of modalities: texts, structured texts, diagrams, pictograms, images, lamps, speech, sounds, and vibration. Texts are sequences of symbols and words that express information in a natural language. Structured texts consist of short portions of text that are structured in a two-dimensional space (e.g., tables). Diagrams are arrangements of various geometric shapes, pictographs and short portions of text to communicate messages or thoughts or to illustrate information (e.g., a map). Pictograms are simplified, symbolic graphical representations of objects and concepts (e.g., application icons). Images are naturalistic graphical representations of objects and scenes in the real world and lamps are simple binary or multi-valued light emitters. Speech is the audible representation of human language and all non-speech auditive signals are referred to as “sounds”.

The ratings for those modalities were determined based on multiple aspects: the accessibility of a modality in a given context, the acceptance of them by the social environment, and the compatibility of the modality with the current activity of the user. For example, when the user is watching a movie, the information representation should preferably be short, concise, and silent. Otherwise, the information might harm the activity of the users by interfering with the visual or auditive channel of the movie Consequently, the modalities (structured) text and vibration are assigned to a high rating in this context.

Table 4. Evaluation of modalities.

Context	Text	Structured text	Diagram	Pictogram	
All contexts	0.98	0.99	1.0	0.0	
Active movement	0.0	0.25	1.0	1.0	
Visual entertainment	0.99	1.0	0.5	0.5	
Context	Image	Lamp	Speech	Sound	Vibration
All contexts	0.75	1.0	1.0	0.99	0.99
Active movement	0.5	1.0	1.0	1.0	1.0
Auditive entertainment	1.0	1.0	0.98	0.99	1.0
Visual entertainment	0.5	1.0	1.0	1.0	1.0
Noisy environment	1.0	1.0	0.25	0.75	1.0
Hearing impaired	1.0	1.0	0.1	0.8	1.0
No headphones in the public	1.0	1.0	0.5	1.0	1.0
Display on	1.0	0.0	1.0	1.0	1.0

Implementation of the Content Representation Search. We implemented the search for suitable content representations as a uniform-cost search [34] on a graph representation of the space of all admissible content representations [25]. In this graph, each node represents a component of a content representation, such as an output device, a presentation medium, a modality, or a modality property. The algorithm assigns a suitability rating $\texttt{eval}(C, b)$ for the current context C to each edge (a, b). The graph structure follows the *RDF schema* depicted in Fig. 3.

The RDF schema for the communication structure has a class for each component of a content representation. Naturally, a device can have one or multiple presentation media. A presentation medium can output information in one or various modalities, and a modality can have one or multiple properties. Specializations of the `hasComponent` relationship in the schema represent these relationships. Additionally, a presentation medium can address various communication channels, and a modality can use a specific communication channel to represent information.

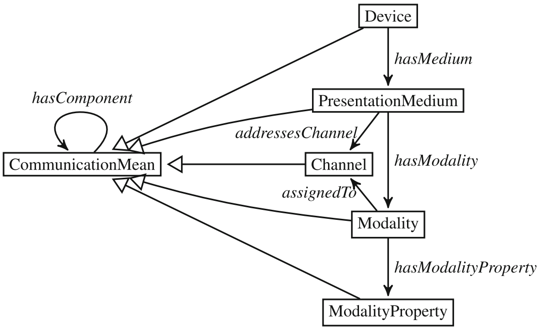

Fig. 3. Visual depiction of an RDF schema for content representations [25].

For simplicity, we introduce a virtual root node that is connected to each instance of `Device` via `hasComponent` relations. The uniform-cost search can then start on this root node. For the search, only on the edges referring to `hasComponent` relations or its specializations are considered. During the search, all best-rated paths from this root node to a leaf node are determined.

The representation of the search space as a graph enables the application of this search method to different device configurations [25]. Therefore, this algorithm is applicable to use cases with automated device recognition and customized user preferences.

4 Evaluation

In this section, we present the conducted experiments and results in detail.

4.1 Evaluation Setup

For the evaluation, the prototype described in Sect. 3.5 was used. To obtain sensible contexts and profiles, we extract them from typical use cases of a travel navigation system. After this, we compare the result of the adaptation to a manually determined optimal user interface.

We derived the following use cases for travel information systems from a study by the Association of German Transport Companies (Verband Deutscher Verkehrsunternehmen e.V., VDV) [39]:

- Navigation for public transport
- Navigation for individual transport
- Navigation for shared vehicles

Furthermore, the VDV defined persona in [39]. We then identified four relevant groups of users among those personas:

- Unimpaired public transport users
- Mobility-impaired public transport users
- Perception-impaired public transport users
- Individual transport users

We selected these user groups because they have different communication needs in several contexts during a journey. They are the basis for the user profiles in the experiments.

4.2 Evaluation Results

This section shows the results of the evaluation grouped by use case. First, the situation of each use case is described in general. More specific situations serve as the basis of the evaluation. For each situation, we list the result of the UI adaptation. Here, the best-rated modalities are shown for each device class to provide an overall result. Hence, no assumption about the availability of specific devices is necessary. Finally, we evaluate the result by comparing it with a manually determined optimal user interface for that situation.

Navigation for Public Transport. The main task of a travel navigation system in public transport is to notify the user about arriving vehicles of interest, the next station, and when to alight from the vehicle. Optionally, it provides the user with the possibility of checking in manually or for signaling the driver to stop at the next station.

When using means of public transport, the user is located either at a public transport station or inside a public service vehicle [22]. The environment in public transport stations and public service vehicles is often noisy due to traffic and other passengers. Inside a public service vehicle, passengers have to either stand or sit. Standing passengers usually need to hold somewhere to avoid falling.

Consider the following situation: an unimpaired public transport user waits at a noisy bus station for the next bus. In this situation, the travel information system's task is to notify the user about the next vehicle of interest. Based on the context information (noisy environment, public service station, unimpaired user), the UI adaptation service selects the following representations for each device class:

- Smartwatch, vibration (rating: 0.9702)
- Smartphone, vibration (rating: 0.96029)
- Tablet, vibration (rating: 0.95039)
- Smart glasses, sound (rating: 0.73507)

To summarize, the system chooses tactile communication means on each of the given devices that support tactile information output. If no tactile information output is available, an auditive signal notifies the user. The noisy environment severely impacts the auditive communication channel between the user and the device. Therefore, the user might not register an auditive notification. Consequently, this situation favors the usage of the tactile communication channel. Furthermore, the visual channel is unsuitable in a situation in which the user does not pay attention to the device. Consequently, the result of the adaptation system is optimal in this situation.

After the vehicle arrived, the user enters the vehicle, but no free seats are available. Therefore, the user is standing inside the vehicle and holds the handrail. The information system's next task is to notify the user about the station where alighting a vehicle is necessary. The UI adaptation service selects the following representations for each device class:

- Smart glasses, sound (rating: 0.9801)
- Smartwatch, vibration (rating: 0.9702)
- Smartphone, sound and vibration (rating: 0.24)
- Tablet, sound and vibration (rating: 0.0950)

In short, easily accessible devices, such as smart glasses and smartwatches, are preferred in this situation, and the system avoids the visual communication channel. This behavior is suitable in the given situation because the user is not already watching the device and has to hold the handrail with one hand. This selection ensures that the user will receive the notification.

Navigation for Individual Transport. In individual transport, the primary use case for travel information systems is navigational instructions. Usually, navigational instructions consist of a notification and further information on the action to be performed by the user. The modalities speech, diagram, or text represent navigational instructions well and provide the user with additional information.

The act of active movement mainly characterizes contexts of individual transport by the user. Consequently, the user has to spend a considerable amount of mental resources on the task of controlling the movement. This limitation holds true especially for the vision of the driver [27]. Apart from that, those contexts can be very diverse. For example, while the car protects the driver from many environmental influences, such as noise and rain, a cyclist usually exposed to those influences.

The context under consideration in this use case is the following: the user is driving the car (active movement). In this context, the UI adaptation service will select the following representations for a navigational instruction:

- Board computer, speech and diagram (rating: 1.0)
- Smart glasses, speech and diagram (rating: 0.99)
- Smartwatch, diagram (rating: 0.882)

- Smartphone, speech and diagram (rating: 0.485)
- Tablet, speech and diagram (rating: 0.24)

The proposed UI adaptation system yields a configuration that is mostly employed in state-of-the-art navigation systems: speech commands with visualization in the form of a diagram (e.g., a map view with short amounts of text in the form of annotations). Again, easily accessible devices are preferred, such as a board computer or smart glasses that enable the user to retrieve information without the need for using a hand-held device. It is important not to distract the driver in this context because the driver has to focus on controlling the movement of the car. However, reading text demands the visual resources of a reader for a considerable amount of time, and consequently, text is generally unsuitable during acts of active movement. While speech commands are a generally acceptable representation of a navigational instruction, it might be easier for a user to map them to the real world when accompanied by a visual representation. Furthermore, map-based navigation systems have been proven to be suitable for this type of context [12]. Consequently, the modality selection of the adaptation system is reasonable in this context.

Navigation for Shared Vehicles. In the next use case [39], the tasks of the travel information system are:

- to announce the vehicle position, properties and access code,
- to provide the user with the possibility to read and agree to the terms and conditions of the data transfer from a mobile device to the board computer, and
- to offer alternatives in case of deviations from the schedule.

Similar to the use case "individual transport", the user is notified about an event and provided with further information. Furthermore, the set of possible contexts in this use case can be very diverse and could be possibly intersecting with the earlier mentioned use cases (e.g., a user receives a notification about the position of an ordered shared vehicle while sitting on the bus).

Consider the following situation: a user just ordered a shared vehicle and is now standing near the vehicle. The travel information system needs to announce the vehicle position and properties to the user. According to the UI adaptation system, the following representations for each device class are optimal:

- Smart glasses, speech and diagram (rating: 0.99)
- Smartwatch, diagram (rating: 0.98)
- Smartphone, speech and diagram (rating: 0.97)
- Tablet, speech and diagram (rating: 0.96)

Here, the UI adaptation system selects the modalities "speech" and "diagram" for the representation of the information. This selection is generally useful in the given context because the information to be communicated to the user is complex and cannot be represented by modalities with a low information capacity, such as vibration. Besides, the environment is not restricting the communication, and therefore, the system can choose the best matching modalities.

Conclusion. As shown in the examples above, the proposed UI adaptation system can flexibly provide reasonable user interfaces for a diverse set of contexts based on the given evaluation rules. However, this pen-and-paper evaluation of the adaptation service should be complemented by a user study in the future to cover more aspects of real-life contexts and to be able to adapt the evaluation function to the diverse contexts in the real world.

5 Conclusion and Future Work

In this work, we extended our previous work of a context-dependent UI adaptation system [25]. It is helpful for the UI to regard the current context of the user, the available device capabilities, and the user's personal preferences to increase the usefulness of applications in various situations. We sketc.hed a service that transforms messages between a system- and a user-oriented representation and adapts them according to the factors mentioned earlier.

We modeled the messages between the application and the sketc.hed adaptation service as communicative acts. These messages can then be translated into a prototypical UI. An ontology models context of use, whereas elementary interaction objects (eIOs) describe prototypical UIs. A uniform-cost search on the communication infrastructure graph adapts the content representation of the eIOs to the current context of use with the help of evaluation functions. Lastly, the targeted rendering engine renders the document that the system generated from the prototypical UI. As the framework is also applicable to bidirectional communication, it also needs to consider the messages from the user to the system. The adaptation service receives these messages and processes them according to the current state of the associated conversation and forwards them to the client application as communicative acts. This work additionally shows a first evaluation of the adaptation system that indicates that a dynamic adaptation of UIs is feasible. Future work should define suitable evaluation functions for the adaptation system. Test cases with real-life environments and potential users must be described. The user groups consist of application developers on the one hand, and end-users of the UI, on the other hand. After obtaining the first evaluation results, the evaluation function can be refined based on the user's feedback.

Currently, the adaption rate of smart speakers and smartwatches is rising. Potentially, also smart glasses will become more widespread. In such a scenario, it is most likely that users will regularly interact with multiple devices. Here, it is not sensible to have a distinct information state on each device but to regard the interaction as communicative acts that are shared between all devices. Such a seamless interaction between multiple devices of the user is particular important for travel information systems, as the user may use hand-held devices or maybe in-vehicle devices. The proposed system provides a framework for seamless interaction across different devices.

References

1. Ater, T.: Building Progressive Web Apps: Bringing the Power of Native to the Browser. O'Reilly Media, Sebastopol (2017)
2. Baader, F.: The Description Logic Handbook. Cambridge University Press, Cambridge (2010)
3. Baus, J., Krüger, A., Wahlster, W.: A resource-adaptive mobile navigation system. In: Proceedings of the 7th International Conference on Intelligent User Interfaces, IUI 2002, pp. 15–22. ACM, New York (2002). https://doi.org/10.1145/502716.502723
4. Bernsen, N.O.: A reference model for output information in intelligent multimedia presentation systems. In: Proceedings of the ECAI 1996 Workshop on: Towards a Standard Reference Model for Intelligent Multimedia Presentation Systems (1996)
5. Chen, G., Kotz, D.: A survey of context-aware mobile computing research. Technical report, Department of Computer Science, Dartmouth College, Hanover, NH, USA (2000)
6. Christoph, U., Krempels, K.H.: Automatisierte Integration von Informationsdiensten. PIK - Praxis der Informationsverarbeitung und Kommunikation **30**(2), 112–120 (2007). https://doi.org/10.1515/piko.2007.112
7. Christoph, U., Krempels, K.H., von Stulpnagel, J., Terwelp, C.: Automatic context detection of a mobile user. In: Proceedings of the International Conference on Wireless Information Networks and Systems (WINSYS), 2010, pp. 1–6 (2010)
8. Criado, J., Vicente Chicote, C., Iribarne, L., Padilla, N.: A model-driven approach to graphical user interface runtime adaptation. In: CEUR Workshop Proceedings, vol. 641 (2010)
9. Cyganiak, R., Wood, D., Lanthaler, M.: Rdf 1.1 concepts and abstract syntax. Technical report, World Wide Web Consortium (W3C) (2014). http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/. Accessed 3 Jan 2019
10. D'areglia, Y.: Learning iOS UI Development. Packt Publishing, Birmingham (2018)
11. Dey, A.K.: Understanding and using context. Pers. Ubiquit. Comput. **5**(1), 4–7 (2001). https://doi.org/10.1007/s007790170019
12. Dingus, T.A., Hulse, M.C., Antin, J.F., Wierwille, W.W.: Attentional demand requirements of an automobile moving-map navigation system. Transp. Res. Part A General **23**(4), 301–315 (1989). https://doi.org/10.1016/0191-2607(89)90013-7
13. Varela, E., Gabriel, I., Quiroz, A., Báez, L.A., Salazar, H., Villaseñor, L.: Artificial neural networks for the study of cosmic rays. In: Ruiz, P.H., Agredo-Delgado, V. (eds.) HCI-COLLAB 2019. CCIS, vol. 1114, pp. 113–123. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-37386-3_9
14. Dumas, B., Lalanne, D., Oviatt, S.: Multimodal interfaces: a survey of principles, models and frameworks. In: Lalanne, D., Kohlas, J. (eds.) Human Machine Interaction. LNCS, vol. 5440, pp. 3–26. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-00437-7_1
15. Edwards, W.K., Mynatt, E.D.: An architecture for transforming graphical interfaces. In: Proceedings of the 7th Annual ACM Symposium on User Interface Software and Technology - UIST 1994. ACM Press (1994). https://doi.org/10.1145/192426.192443
16. Eis, A., Klose, E.M., Hegenberg, J., Schmidt, L.: Szenariobasierter Prototyp für ein Reiseassistenzsystem mit Datenbrillen. In: Burghardt, M., Wimmer, R., Wolff, C., Womser-Hacker, C. (eds.) Mensch und Computer 2017 - Tagungsband, vol. 17, pp. 203–214. Gesellschaft für Informatik e.V., Gesellschaft für Informatik e.V., Regensburg, Germany (2017). https://doi.org/10.18420/muc2017-mci-0164

17. Falb, J., Rock, T., Arnautovic, E.: 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 (2006). https://doi.org/10.1109/ase.2006.71
18. FIPA: FIPA ACL message structure specification. Technical report SC00061G, Foundation for Intelligent Physical Agents (FIPA), Geneva, Switzerland (2002)
19. Gruber, T.R.: A translation approach to portable ontology specifications. Knowl. Acquis. **5**(2), 199–220 (1993). https://doi.org/10.1006/knac.1993.1008
20. Hitzler, P., Krotzsch, M., Rudolph, S.: Foundations of Semantic Web Technologies. Taylor & Francis Ltd. (2009)
21. Hong, J., Suh, E., Kim, S.: Context-aware systems: a literature review and classification. Expert Syst. Appl. **36**(4), 8509–8522 (2009). https://doi.org/10.1016/j.eswa.2008.10.071
22. Hörold, S., Mayas, C., Krömker, H.: Analyzing varying environmental contexts in public transport. In: Kurosu, M. (ed.) HCI 2013. LNCS, vol. 8004, pp. 85–94. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39232-0_10
23. Jaimes, A., Sebe, N.: Multimodal human–computer interaction: a survey. Comput. Vis. Image Underst. **108**(1–2), 116–134 (2007). https://doi.org/10.1016/j.cviu.2006.10.019
24. Johnson, D.A., Trivedi, M.M.: Driving style recognition using a smartphone as a sensor platform. In: 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC), pp. 1609–1615 (2011). https://doi.org/10.1109/ITSC.2011.6083078
25. Kölker, S., Schwinger, F., Krempels, K.H.: A framework for context-dependent user interface adaptation. In: Proceedings of the 15th International Conference on Web Information Systems and Technologies. SCITEPRESS - Science and Technology Publications (2019). https://doi.org/10.5220/0008487204180425
26. Kolski, C., Uster, G., Robert, J.-M., Oliveira, K., David, B.: Interaction in mobility: the evaluation of interactive systems used by travellers in transportation contexts. In: Jacko, J.A. (ed.) HCI 2011. LNCS, vol. 6763, pp. 301–310. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21616-9_34
27. Lansdown, T.C., Brook-Carter, N., Kersloot, T.: Distraction from multiple in-vehicle secondary tasks: vehicle performance and mental workload implications. Ergonomics **47**(1), 91–104 (2004). https://doi.org/10.1080/00140130310001629775
28. Lemmelä, S., Vetek, A., Mäkelä, K., Trendafilov, D.: Designing and evaluating multimodal interaction for mobile contexts. In: Proceedings of the 10th international conference on Multimodal interfaces - IMCI 2008. ACM Press (2008). https://doi.org/10.1145/1452392.1452447
29. Mitrevska, M., et al.: SiAM - situation-adaptive multimodal interaction for innovative mobility concepts of the future. In: 2015 International Conference on Intelligent Environments. IEEE (2015). https://doi.org/10.1109/ie.2015.39
30. Nigay, L., Coutaz, J.: A design space for multimodal systems. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI 1993, Amsterdam, The Netherlands, pp. 172–178. ACM Press (1993). https://doi.org/10.1145/169059.169143
31. Pielot, M., Poppinga, B., Heuten, W., Boll, S.: PocketNavigator: studying tactile navigation systems in-situ. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2012, pp. 3131–3140. ACM, New York (2012). https://doi.org/10.1145/2207676.2208728

32. Pielot, M., Poppinga, B., Vester, B., Kazakova, A., Brammer, L., Boll, S.: Natch: a watch-like display for less distracting pedestrian navigation. In: Ziegler, J., Schmidt, A. (eds.) Mensch & Computer 2010: Interaktive Kulturen, pp. 291–300. Oldenbourg Verlag, München (2010)
33. Rehman, U., Cao, S.: Augmented-reality-based indoor navigation: a comparative analysis of handheld devices versus google glass. IEEE Trans. Hum.-Mach. Syst. **47**, 1–12 (2016). https://doi.org/10.1109/thms.2016.2620106
34. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach, Global Edition. Addison Wesley, Boston (2018)
35. Samsel, C., Dudschenko, I., Kluth, W., Krempels, K.H.: Using wearables for travel assistance. In: Proceedings of the 11th International Conference on Web Information Systems and Technologies (2015). https://doi.org/10.5220/0005481306350641
36. Samsel, C., Thulke, D., Beutel, M.C., Kuck, D., Krempels, K.H.: In-car intermodal travel assistance using mobility service platforms. In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), pp. 800–805. IEEE (2018)
37. Schreiber, G., Dean, M.: OWL web ontology language reference (2004). https://www.w3.org/TR/owl-ref/. Accessed 6 Nov 2018
38. Smith, A.: More than half of cell owners affected by 'distracted walking' (2014). http://www.pewresearch.org/fact-tank/2014/01/02/more-than-half-of-cell-owners-affected-by-distracted-walking/. Accessed 15 Sept 2018
39. Steinert, T., et al.: VDV-Mitteilung 7046: Definition und Dokumentation der Nutzeranforderungen an eine offene Mobilitätsplattform. Technical report, Verband Deutscher Verkehrsunternehmen (VDV) e. V., Köln, Germany (2018)
40. Strang, T., Linnhoff-Popien, C.: 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 (2003)
41. Thornsby, J.: Android UI Design. O'Reilly Media, Sebastopol (2016)
42. Vanderdonckt, J., Limbourg, Q., Michotte, B., Bouillon, L., Trevisan, D.Q., Florins, M.: USIXML: a user interface description language for specifying multimodal user interfaces. In: Proceedings of W3C Workshop on Multimodal Interaction WMI (2004)
43. Wang, X.H., Zhang, D.Q., Gu, T., Pung, H.K.: Ontology based context modeling and reasoning using OWL. In: Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshop. IEEE (2004). https://doi.org/10.1109/percomw.2004.1276898
44. Wargo, J.M.: PhoneGap Essentials: Building Cross-platform Mobile Apps. Addison-Wesley Professional, Boston (2012)
45. Zargamy, A., Sakai, H., Ganhör, R., Oberwandling, G.: Fußgängernavigation im urbanen Raum - Designvorschlag. In: Boll, S., Maaß, S., Malaka, R. (eds.) Mensch & Computer 2013: Interaktive Vielfalt, pp. 365–368. Oldenbourg Verlag, Munich (2013)
46. Zauner, F.J., Radermacher, B.: VDV-Mitteilung 7047: itcs-Nutzungsfälle/Betriebsfälle in der praktischen Anwendung - Methodik und Praxisbeispiele. Technical report, Verband Deutscher Verkehrsunternehmen (VDV) e. V., Köln, Germany (2019)