

Adaptive User Interfaces as an Approach for an Accessible Web of Things

Lukas Smirek
Stuttgart Media University
Nobelstraße 10
D-70569 Stuttgart
+49 711 8923-2672
smirek@hdm-stuttgart.de

Gottfried Zimmermann
Stuttgart Media University
Nobelstraße 10
D-70569 Stuttgart
+49 711 8923-2751
gzimmermann@acm.org

Michael Beigl
Karlsruhe Institute of Technology
Vincenz-Prießnitz-Str. 1
76131 Karlsruhe
+49 721 608-41701
Michael.beigl@kit.edu

ABSTRACT

In this paper, we describe ideas on how to realize personalized and adaptive user interfaces in the Web of Things domain. One of the main goals of personalization is to provide accessible user interfaces for everyone. Our proposal is based on an extension of the Universal Remote Console framework by a RESTFUL communication mechanism in conjunction with the pattern based user interface generation approach of the MyUI project.

Keywords

Adaptive User Interfaces; Web of Things; Universal Remote Console; MyUI; Personalization; Accessibility;

1. INTRODUCTION

During the last years, significant progress was made on technologies enabling the Internet of things (IoT). The main purpose of IoT applications is to interconnect all kind of sensors, actuators and computing units in the environment via internet technologies, in order to assist people in their everyday lives. Possible use cases range from fancy and entertaining ones (e.g., video/audio distribution) to such that are helpful for everyone (e.g., energy management [1]), but also ones that enable a more participatory life for the elderly or for people with disabilities [2], [3] e.g., cooking assistance, fall detection etc.

By analyzing typical IoT scenarios as they are described e.g., in [1] from the user perspective at least two concerns are getting obvious. First of all, almost every part of our everyday life will be affected and the IoT will be a central part of our social life. This means, that the user group of the future IoT will reflect the whole heterogeneity of our society (age, disabilities, culture etc.).

Secondly, users will be confronted with an increasingly complex environment of interconnected devices. In typical use cases, users will gain the highest benefit by interacting with multiple devices. However, as mentioned by [4], adding more functionality only, is not sufficient to increase user satisfaction. Concerning interconnectivity problems, Kephart and Chess even talk about the "nightmare of ubiquitous computing" [5].

Summing up, in future IoT scenarios a heterogeneous user group must be able to handle an increasingly complex environment of interconnected devices. Since IoT will have a significant impact on all parts of our everyday live, it must be assured that no one is left behind because of age, disability or culture. Due to the large variety and sometimes even contradictory user needs and preferences, a "one-size-fits-all" approach is likely to fail [6]. Hence an approach for adaptive user interfaces that support device overarching use cases is required.

2. Related Work

Many of the currently available IoT frameworks like AllJoyn [7], OCF [8], Apple Home Kit [9], and Eclipse Smart Home [10] provide abstract descriptions of the functions and states that can be accessed by a user interface. Some of them like OCF and Home Kit provide templates that are reusable and can be used to classify devices or functionalities. Most of these descriptions do not specify more than data formats. Mayer et al. [11] claim that more interaction related semantic information is required to auto-generate user interfaces and present a system accordingly. However, this system does not take user needs and preferences into account. A system that is focused on providing adaptive and accessible user interfaces on all kind of electronic services is the Global Public Inclusive Infrastructure (GPII) [12]. The system enables the transfer of platform-independent user preferences and needs from one device to another via a cloud service.

Frameworks that are concerned with different accessibility issues (cultural background, accessibility, temporary environmental or personal conditions are the MyUI framework [13], Supple [14] or the Universal Remote Console (URC) [15]. MyUI and Supple employ user interface adaptations to support users with different needs. However, MyUI and Supple usually run on a stand-alone system and are not used to control other, external devices. Consequently, their abstraction models are related to the application and not to devices that shall be controlled. A system that provides abstract descriptions of controllable devices is the URC framework [15], but it lacks a performant user interface generator. A specialty of the URC framework is the concept of a resource server. Any controller that wants to control a target can access the target's description and download related user interfaces or user interface components from the resource server. The resource server is open for third-party contributions. Doing so, it gives user interface developers, assistive technology experts or artists the possibility to inject their expert knowledge [16]. Also, Peissner et al. point to the necessity to make expert knowledge available when accessibility shall be widely supported [6]. Hence, MyUI also supports an open repository for interaction patterns.

In [17] it is argued that appropriate user interfaces are not only required to overcome accessibility issues but also to increase acceptance of ambient intelligence.

3. PROPOSAL

When connecting different things and building applications that are based on standard web technologies, such as HTTP, URL, XML, JSON etc., frequently the expression Web of Things [18] is used instead of Internet of things.

Two technologies that deal with personalization of user interfaces and are based on standard web technologies are the Universal

Remote Console framework (URC) [15], [19] and the MyUI project [13]. We propose to use concepts from these two technologies to enable personalized user interfaces in the Web of Things domain.

The idea of the URC framework is that each thing provides an abstract description of its operational user interface (so-called "user interface sockets", or short "sockets"). This description contains all information a user interface developer needs to design a personalized user interface, as well as all semantic information a controller device requires to control a thing. Currently a protocol is under development that enriches REST APIs with semantic information similar to the ones contained in URC user interface sockets. This shall enable a more standardized access to things connected to the web.

MyUI is a framework to generate adaptive user interfaces based on design patterns [13]. The base for every application is an Abstract Application Interaction Model. Abstract Application Models are based on UML 2.0 state diagrams. Every state of an Abstract Application Interaction Model represents a certain interaction situation of an Application (e.g., choose 1 of N or user input required). At runtime, the MyUI adaptation engine renders a personalized user interface. Therefore, it chooses for every interaction situation contained in an Abstract Application Interaction Model a specific interaction pattern that is fitting best the current context of use (user, environment, technical platform). After executing an interaction, the application moves on to the next interaction situation, depending on the transitions defined in the Abstract Application Interaction Model.

So far, Abstract Application Interaction Models had to be prepared by a developer, putting some extra effort on him. Furthermore, methods for abstract user interface modeling are not yet well established in current user interface development processes.

Our idea is to combine concepts from RESTFUL design [20] [21], Universal Remote Console and MyUI to enable personalized interaction with the Web of Things.

From our perspective, there is a close relationship between the state diagram-based Abstract Interaction Models of MyUI and RESTFUL web services. Every REST endpoint can be seen as a certain interaction situation. Furthermore, well-designed REST APIs use hyperlinks to connect resources and to guide the user through the application (hypermedia as the engine of application state). This is also similar to the transitions from one interaction situation to another in Abstract Application Interaction Models.

Hence, with a modified MyUI adaptation engine it would be possible to render personalized user interfaces for REST services. Instead of using an Abstract Application Interaction Model as input, it would be possible to use the URL of a REST service. Then, the adaptation engine could discover the whole application graph by following the hyperlinks from one resource to another.

However, for every resource (~ interaction situation) the engine requires some semantic information regarding the kind of interaction that can take place. Just by checking the applicable

SAMPLE: Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Conference '10, Month 1–2, 2010, City, State, Country.
Copyright 2010 ACM 1-58113-000-0/00/0010 ...\$15.00.
DOI: <http://dx.doi.org/10.1145/12345.67890>

HTTP verbs gives some hints on the kind of allowed actions on a resource. Still, in order to render a personalized user interface, some more information is required. For this purpose, every resource could provide some semantic information that is similar to the one contained in URC user interface sockets.

Such an approach would also make the creation of an Abstract Application Interaction Model by a user interface developer unnecessary.

So far, MyUI applications run on one specific device with a local user profile. By connecting the MyUI adaption engine with the Global Public Inclusive Infrastructure that provides platform independent user preferences via a cloud service [12] users could be enabled to approach any device of interest and use it with a user interface adjusted to their needs.

4. CHALLENGES

Our research on the described approach above includes the following research questions:

1. What kind of semantic information is required to identify an interaction state? Can URC Sockets satisfy these requirements? Is an extension necessary?

Note to 1: This question addresses issues that are discussed in Mayer et al. [11]. From the authors' point of view it is very likely that the URC framework must be extended by further information. Mayer et al. argue that it is not sufficient that only datatypes are provided in socket descriptions as it is today.

2. Is the hyperlink structure of a REST web service appropriate to provide guidance regarding possible interactions paths through an application? If so, how can this approach serve as a basis for automatic generation of adaptive user interfaces?

Note to 2: This question relates to the previously introduced idea of inferring possible interaction paths by looking at the REST endpoints of a Web service and their basic operations (POST, GET, PUT, DELETE). However, the semantic of these operations is rather limited and should probably be extended when it comes to inferring suitable user interfaces based on interaction paths.

3. What role do task models play in a system that generates adaptive user interfaces based on REST? Should we integrate features from task models (e.g. pre- and post-conditions) in our approach for the description of possible interaction sequences?

Note to 3: As discussed in [22], providing descriptions of the controllable device only is not sufficient to infer an appropriate user interface. Task models are a potential candidate to fill this gap.

4. How can the GPII framework with its notion of personal needs and preferences integrate with our approach for automatic user interface generation based on RESTful web services? Which aspects of personalization should the automatic generation engine deal with, and which should the runtime engine (i.e. web browser) take care of?

Note to 4: Some individual needs and preferences are rather dynamic than static, referencing contextual conditions under which they become active or inactive. Such needs and preferences should be rather dealt with by the runtime engine when the actual context of use is known.

5. ACKNOWLEDGMENTS

The research leading to these results has received funding from the European Union Seventh Framework Program (FP7/2007-2011) under grant agreement no. 610510, Prosperity4All (“Ecosystem infrastructure for smart and personalized inclusion and PROSPERITY for ALL stakeholders”). This publication reflects only the authors’ views and the European Union is not liable for any use that may be made of the information contained herein.

6. REFERENCES

- [1] W3C WoT group, “Use Cases and Requirements for the Web of Things,” *Use Cases and Requirements for the Web of Things*, 26-Oct-2016. [Online]. Available: <http://w3c.github.io/wot/wot-ucr.html>.
- [2] M.-R. Tazari, “An open distributed framework for adaptive user interaction in ambient intelligence,” in *International Joint Conference on Ambient Intelligence*, 2010, pp. 227–238.
- [3] P. L. Emiliani and C. Stephanidis, “Universal access to ambient intelligence environments: opportunities and challenges for people with disabilities,” *IBM Systems Journal*, vol. 44, no. 3, pp. 605–619, 2005.
- [4] C. Rich, “Building Task-Based User Interfaces with ANSI/CEA-2018,” *Computer*, vol. 42, no. 8, pp. 20–27, Aug. 2009.
- [5] J. O. Kephart and D. M. Chess, “The vision of autonomic computing,” *Computer*, vol. 36, no. 1, pp. 41–50, 2003.
- [6] M. Peissner, A. Schuller, D. Ziegler, C. Knecht, and G. Zimmermann, “Requirements for the Successful Market Adoption of Adaptive User Interfaces for Accessibility,” in *Universal Access in Human-Computer Interaction. Design for All and Accessibility Practice*, C. Stephanidis and M. Antona, Eds. Springer International Publishing, 2014, pp. 431–442.
- [7] AllSeen Alliance, “AllJoyn Framework,” 2016. [Online]. Available: <https://allseenalliance.org/framework>.
- [8] OPEN CONNECTIVITY FOUNDATION (OCF), 2016. [Online]. Available: <https://openconnectivity.org/>. [Accessed: 26-Oct-2016].
- [9] “iOS - Home - Apple,” *The smart home just got smarter.*, 2016. [Online]. Available: <http://www.apple.com/ios/home/>. [Accessed: 26-Oct-2016].
- [10] “Eclipse SmartHome - A Flexible Framework for the Smart Home,” 2016. [Online]. Available: <http://www.eclipse.org/smarthome/>. [Accessed: 26-Oct-2016].
- [11] S. Mayer, A. Tschofen, A. K. Dey, and F. Mattern, “User Interfaces for Smart Things,” *Institute for Pervasive Computing, Zurich*, 2014.
- [12] “GPII,” *Global Public Inclusive Infrastructure*, 2016. [Online]. Available: www.gpii.net.
- [13] Matthias Peißner, “Entwurfsmusterbasierter Ansatz zur Überwindung von Nutzungsbarrieren,” Fraunhofer IAO, Universität Stuttgart, 2014.
- [14] K. Gajos and D. S. Weld, “SUPPLE: automatically generating user interfaces,” in *Proceedings of the 9th international conference on Intelligent user interfaces*, 2004, pp. 93–100.
- [15] G. Zimmermann *et al.*, “Universal Remote Console Standard - Toward Natural User Interaction in Ambient Intelligence,” in *Extended abstracts of the 2004 conference on Human factors and computing systems - CHI '04*, New York, New York, USA, 2004, p. 1608.
- [16] G. Zimmermann and B. Wassermann, “Why We Need a User Interface Resource Server for Intelligent Environments,” in *Workshops Proceedings of the 5th International Conference on Intelligent Environments*, 2009, vol. 4, pp. 209–216.
- [17] S. Piva, C. Bonamico, C. Regazzoni, and F. Lavagetto, “A flexible architecture for ambient intelligence systems supporting adaptive multimodal interaction with users,” *Ambient Intelligence*, pp. 97–120, 2005.
- [18] D. Guinard, V. Trifa, F. Mattern, and E. Wilde, “From the internet of things to the web of things: Resource-oriented architecture and best practices,” in *Architecting the Internet of Things*, Springer, 2011, pp. 97–129.
- [19] ISO/IEC, “ISO/IEC 24752-1:2014. Information Technology - User Interfaces - Universal Remote Console - Part 1: Framework.” International Organization for Standardization (ISO), Dec-2014.
- [20] R. T. Fielding, “Architectural styles and the design of network-based software architectures,” University of California, Irvine, 2000.
- [21] Leonard Richardson, Mike Amundsen, and Sam Ruby, *RESTful Web APIs*. O’Reilly, 2015.
- [22] L. Smirek *et al.*, “Accessible Control of Distributed Devices Supporting Persons with disabilities by Providing Adaptive Interaction,” presented at the The First International Conference on Universal Accessibility in the Internet of Things and Smart Environments, Nice, 2016.
- [23] G. Zimmermann and G. Vanderheiden, “A dream... The universal remote console,” *ISO Focus+*, pp. 11–13, 2010.