

Requirements for the Successful Market Adoption of Adaptive User Interfaces for Accessibility

Matthias Peissner¹, Andreas Schuller¹, Daniel Ziegler¹, Christian Knecht²,
and Gottfried Zimmermann³

¹ Fraunhofer-Institute for Industrial Engineering, Stuttgart, Germany
{matthias.peissner, andreas.schuller,
daniel.ziegler}@iao.fraunhofer.de

² University of Stuttgart, Stuttgart, Germany
christian.knecht@iat.uni-stuttgart.de

³ Media University Stuttgart, Stuttgart, Germany
gzimmermann@hdm-stuttgart.de

Abstract. The concept of adaptive user interfaces is a promising solution for providing users with a wide range of individual needs with accessible technology. Developers only have to implement one generic solution to offer a multitude of individually optimised concrete user interfaces. Whereas a lot of technical functionalities and characteristics of adaptive user interfaces are already solved, there is still no widespread market adoption of adaptive UI technologies. This paper presents a collection of requirements for adaptive user interface systems that can enable widespread market adoption. Furthermore it identifies strategies and individual answers, how these requirements can be addressed and met in future systems building on the Prosperity4all approach. It gives a comparison of existing research solutions and how they compare with the stated requirements.

Keywords: Adaptive user interface, adaptive systems, accessibility, user characteristics, market adoption, requirements.

1 Introduction

In order to provide accessible technology to end-users, adaptive user interfaces are recognized as one established solution (e.g. [1], [2] and [3]). By recognizing the personal and contextual user needs, the system can dynamically react by providing personalized user interfaces that can overcome usage barriers. Although there is a long term history in the research field of adaptive user interfaces there are no significant approaches established on the mainstream market yet. Our proposition is that there are several essential requirements that adaptive user interface systems must comply before being able to reach significant impact on the software market:

Being able to provide wide ranging user interface adaptations may require a huge knowledge base and complex adaptation algorithms. This can hardly be accomplished in a centralized system. Therefore, we have identified the following prerequisite

requirements, which we see as essential to an adaptive user interface system: *Modularity* is necessary to being able to handle the complexity of such a system. In practice, an *extensible* approach is important to allow to focus on a subset of possible adaptive solutions at first and to extend gradually to more elaborated use cases. *Openness* of the implemented approach means that the possibility to integrate external expert knowledge into the adaption system is kept intact. This should also apply after the development has finished.

Ensuring the fulfilment of the previous requirements only represent one necessary prerequisite step toward successful market adoption. Furthermore, we describe three more specific requirements for allowing widespread adoption in software development: *efficiency* in the integration in a general software development processes a *short and easy learning curve* for developers as well as *comprehensible and controllable development processes*. Concluding, an overview and evaluation of existing approaches towards the described requirements is listed and the Prosperity4all concept of reaching these requirements is being described.

2 Prerequisite Requirements

The creation of adaptive user interfaces to lower accessibility barriers for users requires the provision of a broad spectrum of individual solutions. To meet the particular needs of many different users, context variables and target devices, such a system must be able to express and handle user interaction in a great variety of ways. Furthermore, this requires very many adaption rules and complex adaption algorithms. We state the following essential prerequisite requirements that arise from the basics necessities to practically work and handle such an adaptive software system.

2.1 Modularity

Modular approaches are identified by their characteristic to subdivide systems into smaller self-contained elements (modules), by the interchangeable configurability of modules and by clearly specified interfaces between modules. Several modules can be combined in numerous ways, and therefore open the possibility to a large number of possible solutions. Module based solutions with clear interfaces also allow for a facilitated division of work within project teams and for parallel development on different system modules by different project teams or team members.

2.2 Extensibility

When implementing concrete design solutions and corresponding adaption rules, extensibility of all adaption mechanisms should be a key concept. Extensibility means that existing system modules can be scaled within the adaptive system or also added and extended at a later stage. Practically, this can mean to start out the adaptive system behaviour with a subset of user interface elements and to iteratively add more elaborated user interface elements and mechanisms at later project stages. In our

previous experience in the MyUI project, this factor of extensibility has shown to be essential and helpful for being able to handle more complex use cases.

2.3 Openness

In order to provide individual design solutions to the needs of particular users, detailed expert knowledge and the knowledge of different user contexts is needed. Even if developers cover all design solutions that work for the known users, they might lack some particular expert knowledge about certain use cases or the characteristics of previously unknown users. Therefore it is essential that the adaption mechanisms in the adaptive system provide a certain amount of openness. This expert knowledge should be easily integrated also at later stages, regardless of an ongoing or closed development process.

3 Basic Requirements for Market Acceptance

In addition to the prerequisite requirements, there are further requirements that are targeted at developers and designers and therefore essential to potential market success and rate of uptake of an adaptive system. To tackle the scepticism towards automatically adaptive user interfaces (as described in [1]) we have identified three key challenges that also resemble our requirements:

- Efficiency in the development process of adaptive user interfaces,
- Short learning curve for developers and
- Comprehensible and controllable development processes for generating resulting user interfaces

These challenges are mainly caused by the approach of model based user interfaces [4] that are generally used for the creation of adaptive user interfaces [5]. This means that an abstract model of a user interface is used as the fundamental structure for generating all possible variants of the adaptive UI. The abstract model describes the commonalities between all variants and is independent of concrete presentation and interaction mechanisms.

The growing requirement of user interfaces running on a multitude of different target devices and platforms has fostered the need for abstract markup languages for automatically generated user interfaces. Some of the most notable recent approaches include TERESA [6] with CONCUR TASK TREES [7], PERSONAL UNIVERSAL CONTROLLER (PUC) [8] for example HUDDLE [9] and UNIFORM [10], additionally UIML (User Interface Markup Language) and CANONICAL ABSTRACT PROTOTYPES (CAP) with newer enhancements CAP3 ([11], [12]).

The work with abstract models is partially difficult to combine with the current practice of the design and development of user interfaces (see [13]). Therefore, the market success of a newly developed adaptive system is also highly dependent on to which extend the previously described requirements are being fulfilled.

3.1 Efficiency in the Development Process of Adaptive User Interfaces

One important precondition for a successful adoption of new software development approaches is their economic viability and efficient implementation. Theoretically, all model based approaches for automatic software generation promise potentially great possible economic savings. Especially for the generation of user interface variants (e.g. different target devices, see [14]), automated processes can relieve the developers of considerable tasks. The user interface is only specified once in an abstract description that resembles the basis for automatic generation different concrete UI variants. Even retrospective changes on the model can be automatically mapped to the different generated variants [15].

To be able to fully benefit from the potential of these automatically generated user interfaces, the following two requirements are of greatest importance:

— *Conciseness and clarity:*

The adaptive system should keep the manually crafted artefacts (models) in terms of extend and complexity to a minimum (see [16]). Redundancies in the resulting models should be avoided.

— *Reusability:*

The generation and adaption of the user interface should be based on generic and reusable mechanism and components. Therefore reoccurring components do not have to be created for each application, but reused and referenced in the abstract application model. Precondition is a clear distinction between application logic from the adaption mechanisms of the interface.

3.2 Short Learning Curve for Developers

The creation of a model that describes the user interface independently from the concrete design of the user interface and the used input devices requires a certain ability to abstract on the side of the developer. This abstracted and time consuming working mode of model based creation of UIs doesn't necessarily resemble today's practice, even if some model based systems (like TERESA) employ a graphical editor for the model creation [6]. The graphical visualization of a developed user interfaces is nowadays often an integral part of the development process ([17], [18]). Oftentimes editors with WYSIWYG-functionality are being used. The learning of a new model language is often mentioned as one barrier for employing adaptive systems at all [13].

3.3 Comprehensible and Controllable Development Processes

Developers want to control and understand the automated processes to generate adaptive user interfaces. Myers et al. point out that it can be difficult in some cases to make the connection between the specification of a user interface and the final generated output [13]. One reason for the existing desire to further influence the compiled UI output was that until recently generated user interfaces were not able to satisfy a very high aesthetic level compared to manually designed, non-generated UIs [13].

4 Comparison of Related Work

There have been several research projects in the field of adaptive user interfaces. Not all of them had their particular focus on the support of the interface developers and the market impact, therefore their fulfilment of the aforementioned requirements varies naturally. Table 1 contains the comparison of the related projects in regard to our market requirements; the following sections will briefly describe the most relevant characteristics for each project in this context.

- **DAMASK.** Damask serves as a pattern based modelling approach and tool. Designers can sketch a user interface that can be used for the UI generation on other devices. Accessibility issues are not addressed explicitly. Nevertheless the system's pattern-based approach and practical functions seem to be very beneficial in this context [18].
- **FAME.** Fame, the "Model-based Framework for Adaptive Multimodal Environments", is a model based framework for adaptive multimodal user interfaces. The approach is centered on a behavior matrix, where the developer specifies the system behavior under different conditions. The defined adaptation rules limited to input and output modality. Some interdependencies between adaptable components are not handled which can lead to a potential break of consistence [19].
- **GUIDE.** Guide, "Gentle user interfaces for elderly people", is a European research project. It tries to strengthen the accessibility of interactive TV applications. The user interface adaption is mainly constrained to the augmentation of the original user interface by offering additional input and output modalities. There is no overall abstract application model describing the user interface on a more general level. The augmentation of existing web applications is realized by annotating HTML code ([20], [21] and [22]).
- **MICA.** Mica, the "Mixed-Initiative Customization Assistance" is a system for the intelligent aid for the user to personalize complex user interface. The system is adapting the helping functions that are implemented within a cooperative dialogue with the user (mixed initiative). Nevertheless the user keeps the complete control over the configuration process. Relevant in this context is especially the unique mixed initiative approach in the user dialogue and the focus on controllability of the intelligent system behavior by the user ([23] and [24]).
- **OASIS/ASK-IT.** OASIS "Open architecture for Accessible Services Integration and Standardization") and its predecessor ASK-IT ("Ambient Intelligence System of Agents for Knowledge-based and Integrated Services for Mobility Impaired Users") both have developed and employed a modular, rule based system to adapt the user interface to particular user needs. OASIS developed the so called adaptive widget library which allows for a direct integration in developer tools. However, there is no adaptation of structural aspects of the interaction, i.e. higher abstractions of the interaction flow ([25], [26] and [27]).
- **PUC and HUDDLE.** PUC ("Personal Unified Controller") is a descriptive language that serves for the abstract definition of user interfaces. It can allow for the generation of different UIs for diverse devices and output modalities. HUDDLE

builds upon the PUC to offer automatically generated user interfaces for controlling a home cinema system. The approach is relevant, because it is one of the few validated systems employing an abstract user interface description language, and implemented the idea of so called smart templates, that can adapt themselves according to the chosen target device and platform ([8], [28], [9] and [10]).

- **PLASTIC USER INTERFACES.** The basis for plastic user interfaces lays in ubiquitous computing and the easy integration of different user devices. There is a strong focus on providing conceptual and generic descriptions on many different aspects like the user, the social and physical environment and the technical platform. However, the described adaptive mechanisms refer rather to technical device characteristics than of user characteristics ([29], [30], [31], [32], [33], [34], [35], [36], [37], [38], and [39]).

Table 1. Assessment of relevant research projects

Legend: ● Broad coverage, in-depth concepts ◐ Partial coverage, in-depth concepts
 ◑ Broad coverage, simple concepts ○ Low coverage, simple concepts

Requirements	DAMASK	FAME	GUIDE	MICA	OASIS / ASK-IT	PUC / Huddle	Plastic UI	SUPPLE	UII/ AVANTI	MyUI
Modular and extensible adaptation mechanisms (2.1 and 2.2)		◐	●	○	◐	◐	●	●	◐	●
Public available adaption mechanisms and open to external contributions (2.3)			◐	○	◐	◐	◐	◐	◐	◐
Efficient Development: reusable components and low effort for the specification with sufficient cardinality (3.1)	◐	○	◐		◐	◐	●	●	○	●
Low thresholds for beginners and low demands for developers' skills (3.2)	●	◐	●		●	●	◐	○	○	●
Generation of the adaptive user interface understandable and manageable from the developers point of view (3.3)					●	◐	◐	○	◐	●

- **SUPPLE.** Supple is one of the most influential adaptive systems for accessibility. Based on a cost function that takes into account used devices, current tasks, user preferences as well as motoric and perceptive limitations. However, performance issues limit the set of characteristics taken to account. Supple also offers the user possibilities to influence the output of the generated user interface ([1], [40], [41], and [42]).

- **UUI/AVANTI.** Avanti is based upon the Unified User Interface (UUI) approach. It offers alterations of the context and user profile both during initialization phase and during runtime. Basis of the adaptations are design patterns that resemble solutions to different user and context profile data. The “Unified User Interface Design Method“ describes the process to develop an adaptive application. The model is focused on user characteristics, only motoric disabilities and blindness are mentioned. The different application states are designed to have an easily extensible rule base to mention other user needs ([43], [44], [45], [46], and [47]).
- **MYUI.** MyUI (Mainstreaming Accessibility through Synergistic User Modelling and Adaptability) integrates all aspects of adaptive user interfaces for accessibility in one system [12]. It includes the multimodal design patterns repository, the runtime environment with the adaption engine enabling user interface generation and dynamic adaptations during run-time, the specification of an abstract user interface model and a development tool. MyUI mostly complies with all the described requirements (see Table 1).

5 Prosperity 4 All Approach

Even though MyUI theoretically fulfils most of the requirements for the successful market adoption of adaptive user interfaces for accessibility [48], in practical, it was never used for developing commercial products. Besides having the focus of the patterns and the reference implementation on interactive TV (iTV) the main reason for the absence of market acceptance is probably the missing infrastructure and community.

This is the starting point of Prosperity4All. MyUI and other promising approaches from scientific research will be refined and integrated to a multi-sided-platform, the Global Public Inclusive Infrastructure (GPII). International project participation, the involvement of industry enterprises, cross-platform development techniques and the open source strategy will help adaptive user interface approaches to reach a better market penetration [49]. GPII allows commercial product developers and service providers as well as external technical contributors to easily access and use existing solutions, extend or adapt them in order to fit specific requirements. Additionally crowdsourcing can help making new ideas accessible for diverse end users.

A more concrete plan for integrating and improving the MyUI components is described in the following subsections:

5.1 User and Context Management

The MyUI Context Manager holds a user profile based on variables expressing user capabilities and environmental conditions. The variables are expressed by numeric values an interval common to all of them. Based on collected sensor data the profile variables are updated in a permanent process [12].

The architecture of the GPII as developed by the Cloud4all project uses an ontology of user needs and preferences that are directly linked to user interface and

interaction aspects [50]. To enable the integration of MyUI concepts into the GPII architecture a mapping between the stored user preferences and the capabilities based user profile will be established.

Special attention will be directed to MyUI runtime events triggered by sensors and how this information can be used to determine the need for runtime adaptations by refining the users set of preferences.

5.2 Runtime Environment

The MyUI runtime environment will be used to interpret and process information about user needs and environmental factors to generate and adapt accessible user interfaces. Therefor flexible layouts and CSS variations with the newest web technologies will be used similar to those in the “Responsive Design” approach for device specific adaption. If applicable, runtime environments for additional platforms will be implemented. Additionally the MyUI runtime environment will be extended to work with Assistive Technologies (AT) as plugged and configured by AsTerRICS (Assistive Technology Rapid Integration & Construction Set). AsTerRICS allows the creation of flexible solutions for people with disabilities using a large set of sensors and actuators. Possible applications are Computer input, Environmental Control, Toys and Games, Brain/Neural computer interfaces, Android Phone support [51]. In addition the Universal Remote Console (URC) will be integrated which allows users to interact with networked devices and services in their environments in universal and natural ways, utilizing technologies like natural language interaction [52].

5.3 Pattern Repository

The idea and the base of the open and extensible MyUI Pattern Repository will mainly be taken over. Each pattern describes a rule which is interpreted by the adaption engine. They have a defined structure and can be related to other patterns. There will be a machine-readable and a human-readable part in order to allow the contribution from developers and accessibility experts. To preserve platform and technology independence the machine-readable part is written in pseudocode and needs to be transferred to the adaption engine manually. In Prosperity4All the transformation process will be made easier for example by providing generative stubs. Furthermore the repository will be extended with patterns for further applications and devices. The open source character of the project ensures that new patterns will be available to all the interested developers. Additionally quality assurance issues will be addressed in order to guarantee the usefulness of new patterns.

5.4 Development Toolkit

To comply with the in section 3 described basic requirements for market acceptance a development toolkit is an essential component and therefor will also be part of the Prosperity4All approach. It supports industrial user interface developers and designers to quickly create abstract models of adaptive user interfaces which are needed for

run-time adaptations. Developers and decision-makers of software enterprises gave positive feedback for the MyUI toolkit in regard to efficiency, learning curve, comprehensibility and controllability [48]. Prosperity4All will also address the mentioned concerns for example regarding more complex projects.

The URU utilizes a User Interface Socket Description to define the abstract interaction possibilities. This approach will be examined for integration of translation possibilities with the MyUI Abstract Application Interaction Model.

6 Conclusions and Outlook

The basic requirements for market acceptance in section 3 as well as the prerequisite requirements in section 2 are described on a rather high level. From a technical point of view to gain acceptance in the real world software market it will be necessary to determine which common software engineering practices have to be supported. Based on the results there may be implications for the structure of the runtime infrastructure and components as well as for the development environment and tools.

A more human-centered perspective brings the developer into focus. It is not sufficient only making developers realize that they have to implement adaptive user interfaces for their software products. Instead it seems necessary to provide an environment where they actually want to do so. One major key may be the user experience of developers working with the components and tools needed.

From the commercial point of view the goal is not to provide special solutions for a distinct market for accessibility solutions but to demonstrate and communicate the potential of adaptive user interfaces for one mainstream market including all users regardless of any kind of temporary or permanent special needs.

Acknowledgments. The research leading to these results has received funding from the European Union's Seventh Framework Program under FP7 Grant #610510 and #248606. The opinions herein are those of the authors and not necessarily those of the funding agency.

References

1. Gajos, K.Z., Weld, D.S., Wobbrock, J.O.: Automatically generating personalized user interfaces with Supple. *Artificial Intelligence* 174(12-13), 910–950 (2010)
2. Ringbauer, B., Peissner, M., Gemou, M.: From “Design for all” towards “Design for one” – A modular user interface approach. In: Stephanidis, C. (ed.) *HCI 2007*. LNCS, vol. 4554, pp. 517–526. Springer, Heidelberg (2007)
3. Savidis, A., Stephanidis, C.: Unified user interface design: Designing universally accessible interactions. *Int. J. Interacting with Computers* 16(2), 243–270 (2004)
4. Szekely, P.: Retrospective and Challenges for Model-Based Interface Development. In: *Proceedings DSV-IS 1996*. Eurographics, pp. 1–27. Springer, Heidelberg (1996)

5. Cockton, G.: Some critical remarks on abstractions for adaptable dialogue managers. In: Proceedings of the 3rd Conference of the British Computer Society on People and Computers, pp. 325–343. Cambridge University Press, Cambridge (1987)
6. Paterno, F., Santoro, C., Mäntyjärvi, J., Mori, G., Sansone, S.: Authoring pervasive multimodal user interfaces. *International Journal of Web Engineering and Technology* 4(2), 235–261 (2008)
7. Paterno, F.: *Model-Based Design and Evaluation of Interactive Applications*. Springer, London (1999)
8. Nichols, J., Myers, B.A.: Creating a lightweight user interface description language: An overview and analysis of the personal universal controller project. *ACM Transactions on Computer-Human Interaction* 16(4), Article 17, 37 pages (2009)
9. Nichols, J., Myers, B.A., Rothrock, B.: UNIFORM: automatically generating consistent remote control user interfaces. In: Proceedings CHI 2006, pp. 611–620. ACM, New York (2006)
10. Nichols, J., Rothrock, B., Chau, D.H., Myers, B.A.: Huddle: automatically generating interfaces for systems of multiple connected appliances. In: Proceedings UIST 2006, pp. 279–288. ACM, New York (2006)
11. Van den Bergh, J., Luyten, K., Coninx, K.: CAP3: context-sensitive abstract user interface specification. In: Proceedings EICS 2011, pp. 31–40. ACM, New York (2011)
12. Peissner, M., Häbe, D., Janssen, D., Sellner, T.: MyUI: generating accessible user interfaces from multimodal design patterns. In: Proceedings EICS 2012, pp. 81–90. ACM, New York (2012)
13. Lin, J., Landay, J.A.: Employing patterns and layers for early-stage design and prototyping of cross-device user interfaces. In: Proceedings CHI 2008, pp. 1313–1322. ACM, New York (2008)
14. Myers, B., Hudson, S.E., Pausch, R.: Past, present, and future of user interface software tools. *ACM Transactions on Computer-Human Interaction - Special Issue on Human-Computer Interaction in the New Millennium, Part 1* 7(1), 3–28 (2000)
15. Meskens, J., Vermeulen, J., Luyten, K., Coninx, K.: Gummy for multi-platform user interface designs: shape me, multiply me, fix me, use me. In: Proceedings AVI 2008, pp. 233–240. ACM, New York (2008)
16. Trewin, S., Zimmermann, G., Vanderheiden, G.: Abstract user interface representations: how well do they support universal access? *ACM SIGCAPH Computers and the Physically Handicapped* (73-74), 77–84 (2002)
17. Newman, M.W., Landay, J.A.: Sitemaps, storyboards, and specifications: a sketch of Web site design practice. In: Proceedings DIS 2000, pp. 263–274. ACM, New York (2000)
18. Lin, J., Landay, J.A.: Employing patterns and layers for early-stage design and prototyping of cross-device user interfaces. In: Proceedings CHI 2008, pp. 1313–1322. ACM, New York (2008)
19. Duarte, C., Carriço, L.: A conceptual framework for developing adaptive multimodal applications. In: Proceedings of the 11th International Conference on Intelligent user Interfaces (IUI 2006), pp. 132–139. ACM, New York (2006)
20. Biswas, P., Langdon, P., Duarte, C., Coelho, J.: Multimodal adaptation through simulation for digital TV interface. In: Proceedings EuroITV 2011, pp. 231–234. ACM, New York (2011)
21. Coelho, J., Duarte, C.: The Contribution of Multimodal Adaptation Techniques to the GUIDE Interface. In: Stephanidis, C. (ed.) *Universal Access in HCI, Part I, HCII 2011*. LNCS, vol. 6765, pp. 337–346. Springer, Heidelberg (2011)

22. Coelho, J., Duarte, C., Biswas, P., Langdon, P.: Developing accessible TV applications. In: Proceedings ASSETS 2011, pp. 131–138. ACM, New York (2011)
23. Bunt, A., Conati, C., McGrenere, J.: Mixed-Initiative Interface Personalization as a Case Study in Usable AI. *AI Magazine* 30(4), 58–64 (2009)
24. Bunt, A., Conati, C., McGrenere, J.: Supporting interface customization using a mixed-initiative approach. In: Proceedings IUI 2007, pp. 92–101. ACM, New York (2007)
25. Leonidis, A., Antona, M., Stephanidis, C.: Rapid Prototyping of Adaptable User Interfaces. *International Journal of Human-Computer Interaction* 28(4), 213–235 (2012)
26. Leuteritz, J.-P., Widlroither, H., Mourouzis, A., Panou, M., Antona, M., Leonidis, A.: Development of Open Platform Based Adaptive HCI Concepts for Elderly Users. In: Stephanidis, C. (ed.) UAHCI 2009, Part II. LNCS, vol. 5615, pp. 684–693. Springer, Heidelberg (2009)
27. Ringbauer, B., Peissner, M., Gemou, M.: From “design for all” towards “design for one”—A modular user interface approach. In: Stephanidis, C. (ed.) Universal Access in HCI, Part I, HCII 2000. LNCS, vol. 4554, pp. 517–526. Springer, Heidelberg (2007)
28. Nichols, J., Chau, D.H., Myers, B.A.: Demonstrating the viability of automatically generated user interfaces. In: Proceedings CHI 2007, pp. 1283–1292. ACM, New York (2007)
29. Frey, A.G., Céret, E., Dupuy-Chessa, S., Calvary, G., Gabillon, Y.: UsiComp: an extensible model-driven composer. In: Proceedings EICS 2012, pp. 263–268. ACM, New York (2012)
30. Calvary, G., Serna, A., Coutaz, J., Scapin, D., Pontico, F., Winckler, M.: Envisioning Advanced User Interfaces for e-Government Applications: a Case Study. In: Practical Studies in E-Government: Best Practices from Around the World, pp. 205–228. Springer, Heidelberg (2011)
31. Ceret, C.: Toward a flexible design method sustaining UIs plasticity. In: Proceedings EICS 2011, pp. 307–310. ACM, New York (2011)
32. Dessart, C.-E., Motti, V.G., Vanderdonckt, J.: Showing user interface adaptivity by animated transitions. In: Proceedings EICS 2011, pp. 95–104. ACM, New York (2011)
33. Demeure, A., Calvary, G., Coninx, K.: COMET(s), A Software Architecture Style and an Interactors Toolkit for Plastic User Interfaces. In: Graham, T.C.N. (ed.) DSV-IS 2008. LNCS, vol. 5136, pp. 225–237. Springer, Heidelberg (2008)
34. Vanderdonckt, J., Coutaz, J., Calvary, G., Stanciulescu, A.: Multimodality for Plastic User Interfaces: Models, Methods, and Principles. In: Multimodal user Interfaces: Signals and Communication Technology. LNEE, pp. 61–84. Springer, Heidelberg (2008)
35. Demeure, A., Calvary, G., Vanderdonckt, J.: The COMETS Inspector: Towards Run Time Plasticity Control Based on a Semantic Network. In: Coninx, K., Luyten, K., Schneider, K.A. (eds.) TAMODIA 2006. LNCS, vol. 4385, pp. 324–338. Springer, Heidelberg (2007)
36. Balme, L., Demeure, A., Barralon, N., Calvary, G.: CAMELEON-RT: A Software Architecture Reference Model for Distributed, Migratable, and Plastic User Interfaces. In: Markopoulos, P., Eggen, B., Aarts, E., Crowley, J.L. (eds.) EUSAI 2004. LNCS, vol. 3295, pp. 291–302. Springer, Heidelberg (2004)
37. Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouillon, L., Vanderdonckt, J.: A Unifying Reference Framework for Multi-Target User Interfaces. *Interacting with Computers* 15(3), 289–308 (2003)
38. Calvary, G., Dâassi, O., Balme, L., Demeure, A.: Towards a new generation of widgets for supporting software plasticity: The “comet”. In: Feige, U., Roth, J. (eds.) EHCI-DSVIS 2004. LNCS, vol. 3425, pp. 306–324. Springer, Heidelberg (2005)
39. Coutaz, J.: User interface plasticity: model driven engineering to the limit? In: Proceedings EICS 2010, pp. 1–8. ACM, New York (2010)

40. Gajos, K.Z., Wobbrock, J.O., Weld, D.S.: Improving the Performance of Motor-Impaired Users with Automatically-Generated, Ability-Based Interfaces. In: Proceedings CHI 2008, pp. 1257–1266. ACM, New York (2008)
41. Gajos, K.Z., Weld, D.S.: SUPPLE: Automatically Generating User Interfaces. In: Proceedings IUI 2004, pp. 93–100. ACM, New York (2004)
42. Weld, D., Anderson, C., Domingos, P., Etzioni, O., Lau, T., Gajos, K., Wolfman, S.: Automatically personalizing user interfaces. In: Proceedings IJCAI 2003, pp. 1613–1619. Morgan Kaufmann Publishers Inc., San Francisco (2003)
43. Savidis, A., Antona, M., Stephanidis, C.: A Decision-Making Specification Language for Verifiable User-Interface Adaptation Logic. *International Journal of Software Engineering and Knowledge Engineering* 15(6), 1063–1094 (2005)
44. Savidis, A., Stephanidis, C.: The Unified User Interface Software Architecture. In: *User Interfaces for All – Concepts, Methods and Tools*, pp. 389–415. Lawrence Erlbaum Associates Inc., Mahwah (2001)
45. Savidis, A., Stephanidis, C.: Unified user interface design: designing universally accessible interactions. *Interacting with Computers* 16(2), 243–270 (2004)
46. Kobsa, A., Koenemann, J., Pohl, W.: Personalised hypermedia presentation techniques for improving online customer relationships. *The Knowledge Engineering Review* 16(2), 111–155 (2001)
47. Stephanidis, C., Paramythi, A., Akoumianakis, D., Sfyarakis, M.: Self-adapting web-based systems: Towards universal accessibility. In: *Proceedings of the 4th ERCIM Workshop on ‘User Interfaces for All’*, 17 pages (1998)
48. Peissner, M.: *Entwurfsmusterbasierter Ansatz für adaptive Benutzungsschnittstellen zur Überwindung von Nutzungsbarrieren*. Dissertation. Universität Stuttgart, Stuttgart (2014)
49. Vanderheiden, G., Trevisanus, C., Chourasia, J., The, A.: global public inclusive infrastructure (GPII). In: *ASSETS 2013*, pp. 1–3 (2013)
50. Madrid, J., Peinado, I., Koutkias, V.: Cloud4all Priority applications and User Profile Ontology (D101.1). Public Deliverable of the Cloud4all Project (2012), <http://cloud4all.info/render/binarios.aspx?id=90>
51. Ossmann, R., Thaller, D., Nussbaum, G., Pühretmair, F., Veigl, C., Weiß, C., Diaz, U.: AsTeRICS, a Flexible Assistive Technology Construction Set. *Procedia Computer Science* 14, 1–9 (2012)
52. LaPlant, B., Trewin, S., Zimmermann, G., Vanderheiden, G.: The universal remote console: a universal access bus for pervasive computing. *IEEE Pervasive Computing* 3(1), 76–80 (2004)