MyUI Individualization Patterns for Accessible and Adaptive User Interfaces

Matthias Peißner, Doris Janssen, Thomas Sellner
Competence Center Human-Computer Interaction
Fraunhofer IAO
Stuttgart, Germany
{matthias.peissner, doris.janssen, thomas.sellner}@iao.fraunhofer.de

***Abstract*—User groups for interfaces to interact with smart environments are highly heterogeneous. Furthermore, such user interfaces often run on different devices, ranging from smartphones to interactive TVs and others. Dynamic user interfaces, which adapt to different devices and diverse user needs at the same time in order to provide a universal access have received little attention so far. Our MyUI system follows a design pattern based approach to adaptive and accessible user interfaces. In this paper, we present the MyUI individualization patterns. They build on information about the user, the context and the devices, which is gathered through system interaction and sensors. The individualization patterns determine global characteristics of the user interface to adapt interaction mechanisms and presentation formats to achieve an optimal fit for the user's abilities, situation and equipment. Through this work, an understandable and human-readable way for editing accessibility rules for a system is achieved.**

Keywords - Design Patterns; adaptive systems; accessibility; user characteristics

I. INTRODUCTION

Modern sensor and agent technologies in smart homes aim at interpreting the user's behavior and intention as well as contextual information of the environment and the used devices. Through these pervasive technologies new appliances for elderly or handicapped persons due to physical/cognitive restriction can be developed for smart environments. For example, a smart TV offers several services such as weather, email or cognitive and physical exercises [6]. Especially, a user group of people with certain disabilities needs a lightweight and unrestricted access to all smart multimedia devices and services through an individual interface [10]. Therefore, it is necessary that these smart multimedia interfaces are self-learning and self-adaptive to offer great opportunities for accessibility and improved usability for a broad range of diverse users and devices.

In this paper, we present the MyUI individualization patterns for adaptive user interfaces. Within the MyUI project, an infrastructure is defined for accessible user interfaces in smart environments, which self-adapt during runtime in order to cover individual needs and limitations. We will show how this infrastructure collects user, device and context information and generates within an ongoing process a current user profile, device profile and user interface profile. The MyUI adaptation relies on a repository of user interface design patterns, which are selected and put together according to the information within the current profiles. A major objective is to reduce the need for configuration or user enrolment.

In this paper, we will set the focus on the MyUI individualization patterns, which define rules for transforming the collected data from the infrastructure to an individual and adapted user interface. We show related work, then explain the concept and the pattern template, and we will show two detailed examples of such interaction patterns.

II. RELATED WORK

For the personalization of user interfaces, mainly two approaches exist. Adaptable user interfaces provide the user with mechanisms to actively customize the interface, whereas adaptive user interfaces initiate and perform dynamic adjustments autonomously ([4][7]).

Both approaches have their pros and cons. The most important advantage of adaptable systems is that the users are in total control of the individual appearance of their user interface. This supports the understandability and traceability of modifications from the user's perspective ([5]). However, this advantage is at the same the main shortcoming of adaptable user interfaces – especially when personalization aims at increasing the accessibility. Users with disabilities and lower levels of ICT (information and communication technology) literacy would benefit most from personalized user interfaces. They often have severe problems with standard configurations. Customization dialogues, however, are a significant barrier – even for abled users [8]. For that reason the system shall provide both functionalities of adaptable and adaptive systems. A good approach for this has been developed within the mixed-initiative system MICA, where the user gets system generated recommendations and decides, if he wants to accept them [3].

But, if the system should be more proactive (e.g., when it comes to adapt a user interface to reach a higher accessibility), it requires understanding of the application state, the current environment and user context as well as the specification of the current device at runtime. Blumendorf et al. [1] describe an approach how to combine models (user, device etc.) and the current application state in order to build adaptive user interfaces. Therefore, three abstract executable types of models – definition elements, situation elements and executable elements – were introduced. These elements

 ISBN: 978-1-61208-225-7

contain static and dynamic variables, which are set at application start or dynamically during runtime. The utilization of these models makes the system aware of context information as well as knowledge about the current application state. Through the connection between context models and application state, the system can dynamically adapt to changes in the environment or user behaviors. However, the knowledge about generating user interfaces is hidden within the system and not visible for developers.

The conceptual basis of the MyUI adaptation Framework is a design pattern repository that includes proven design solutions for optimal usability and accessibility. In [9] we defined requirements on adaptive user interfaces to improve accessibility and describe a modular approach to user interface development, which relies on the combination of specific user interface design patterns. Borchers [2] suggested a structure of design patterns. This structure is used for the MyUI patterns. The patterns are related to other patterns of different types and levels of abstraction and linked with software components that can be extended for developing adaptive user interfaces. The MyUI design pattern repository [12] is publically available for reviewing, refining and contributing by other experts in the field.

III. PARAMETERS INFLUENCING THE ADAPTATION OF ACCESSIBLE USER INTERFACES

For the User Interface generating process it is necessary to gather environment, user and device information. This information is collected by several external and internal sensors, which are installed in the user's smart environment and within the used devices. The MyUI Context management system interprets and updates all this information and categorized them in two characteristics: User characteristics and device characteristics. The transformation of the raw sensor data follows a specific ontology, which follows the Open Ambient Assisted Living (AAL) framework [11].

A. User Characteristics

User Characteristics give information about the user, his abilities, disabilities, preferences, and about his current environment (see Table I). All these values are stored in a User Profile, which is continuously updated by the MyUI context manager. The value of the user profile is specified as a numeric value between zero and four. For instance, if a user has a visual acuity of 0.5 that means he has a nearly perfect visual ability. In contrast, a user with a value of 3.8 in hearing is nearly deaf. These values are defined within the MyUI pattern repository.

TABLE I. MYUI USER PROFILE VARIABLES

Name of variable	Description
Visual Acuity	Ability to perceive what is displayed on the screen
Field of Vision	Describes how limited the field of vision of the given user is.
Ambient Light	The amount of ambient light at the users place.
Ambient Noise	The amount of ambient noise at the users place.
Hearing	Describes how limited the user's ability to hear sounds is.
Language Reception	Ability to understand written or spoken language
Language Production	Ability to speak and write language
Understanding Abstract Signs	Ability to understand abstract signs and pictograms
Attention	Ability to handle multiple things at the same time, resp. focusing on something.
Processing Speed	Ability to process information fast.
Working Memory	Ability to remember an exact sequence of steps in a process and the ability to orientate in this process.
Long Term Memory	Ability to learn and remember information for a long time.
ICT Literacy	Ability to use modern information technology.
Hand-Eye Coordination	Ability to coordinate the movement of the hands with things seen.
Speech Articulation	Ability to speak
Finger Precision	Ability to move the fingers precisely.
Hand Precision	Ability to move the hand precisely.
Arm Precision	Ability to move the arms precisely.
Contact Grip	Ability to control things by touching them.
Pinch Grip	Ability to press single buttons.
Clench Grip	Ability to hold object.
First Name	The first name of the user.
Last Name	The last name of the user.
Email Address	The email address of the user.
Preferred Language	The language the user prefers to use.
Successful Interactions	The number of successful interactions with the system.
State transitions	The number of state transitions the user carried out.
MyUI Experience	The experience with the MyUI system.
PreferenceTonalOutput	Selects whether the user prefers output enhanced with sounds.
PreferenceSpeechOutput	Selects whether the user prefers speech-output in addition to text.

Furthermore, not only sensors influence information within the user profile, but also the interactions the user conducts on the system. The interactions are tracked and evaluated automatically. The user is suggested to play certain games on his system in order to find out about abilities like hand precision or processing speed.

B. Device Characteristics

The device profile is created and updated by the device manager via device-specific patterns. The following Table II shows how the device profile looks like.

TABLE II. THE MYUI DEVICE PROFILE

Device variable	Description

category	Category of the device, on which the application is running
titleFontSize	Font size of title elements
bodyTextFontSize	Font size of all body text elements
complementaryTextFontSize	Font size of all complementary text elements, e.g., additional descriptions or comments of minor importance for the user
cellSize	The variable grid describes how many pixels are needed to build one cell of a grid.
cellCount	The variable cellCount describes how many cells are needed to fill the display.
titleBarArea	The Coordinates and layout properties of the title bar
contentArea	The Coordinates and layout properties of the content area
contentControlArea	The Coordinates and layout properties of the content control area
functionsArea	The Coordinates and layout properties of the functions area
adaptationArea	The Coordinates and layout properties of the adaptation area
microphoneAvailable	Whether there is a microphone available
touchAvailable	Whether there is a touch input device available
pointingDeviceAvailable	Whether there is a pointing device available
keyboardAvailable	Whether there is a keyboard available
remoteControlAvailable	Whether there is a remote control available
cameraAvailable	Whether there is a camera available
displayAvailable	whether there is a display available
speakerAvailable	whether there is a speaker available

Those parameters of user profile and device profile are input for the user interface generation process and will be used in order to adapt the user interface optimally to a certain user, his device and his current environment.

IV. PATTERN APPROACH

Within the MyUI interface generation system, design patterns are used to display the knowledge about how to adapt the system to a certain user. These patterns are stored within the design patterns repository, which includes proven design solutions for optimal accessibility and usability. In order to assure flexibility and extensibility, the MyUI design pattern repository is developed on basis of a media wiki (see [12] and Figure I).

The MyUI design patterns within this repository are human-readable without any further learning or knowledge about rule-based systems. They are linked to re-usable software components. Every instance of a MyUI user interface is the result of the composition of a number of single design patterns – or clearly put: reusable software components linked to these design patterns.

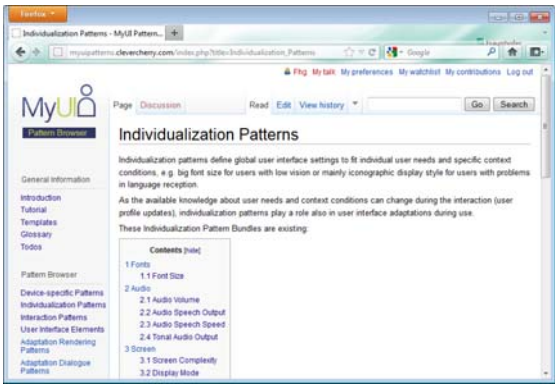

FIGURE I. The MyUI Design Pattern repository can be accessed through the MyUI Pattern Browser

The MyUI design patterns repository includes six categories of design patterns:

Device-specific patterns interprete the information provided by the device. The output is a device profile, which contains standardized information about the device (e.g. hardware provided by the device, suitable font sizes for that device, etc.)

Individualization patterns, which will be shown in detail later on in this paper, take a look on the user and his needs, as well as the context conditions. E.g. they define big font size for users with low vision or a simple, plain surface for people with attention deficits.

Interaction patterns show solutions for given interaction situations, for example, a optimized formular for an interaction situation, where a user should provide input. They use information of the user and the device profile.

User interface elements are basic elements, of which interaction patterns consist. They provide the generic primitives required composing the interaction patterns, e.g., a "selection list" interaction pattern requires "option buttons" as user interface elements.

Adaptation patterns are used, when the user interface changes, due to environmental changes or due to besser knowledge about the user, his needs and his disabilities. They define the mechanisms of switching from one instance of a user interface to another.

V. INTERACTION PATTERNS FOR THE ADAPTATION OF THE INTERFACE TO A CERTAIN USER

Individualization patterns are the core piece of the automatic user interface individualization. They create and update the user interface profile. They are closely related to specific user characteristics as stored in the user profile and relevant device characteristics as stored in the device profile. They process the current user profile and device profile and "translate" the user, environment and device characteristics into user interface features, i.e. global settings.

Each time when changes occur to the user profile, all rules of the individualization patterns will be recalculated and the best fitting individualization patterns will be

 ISBN: 978-1-61208-225-7

activated. This will result in changes to the global settings as defined in the user interface profile. Thus, user interface parameterization is a permanently on-going process during the interaction. Changes in the user profile serve as triggers for a parameterization updates.

Individualization patterns set parameters to the "User Interface Profile". The information within this profile is used by interaction components (e.g., MainMenu) in order to decide, how to show itself (e.g., as textual menu, as graphical menu with a certain amount of entries or as a sound menu).

A. *Format of Individualization Patterns*

Individualization pattern bundles collect patterns that suite the same purpose. The MyUI design patterns repository includes bundle descriptions for individualization patterns in the following format:

TABLE III. Template for individualization pattern bundles

<ID>	**<name of the individualization pattern bundle>**
Problem	< describe a general interaction problem related to a value range(s) of certain aspects/variables of the user profile and the device profile>
sets <variable(s)> as used by <pattern bundle>	<put in, which variables are set by this bundle, connecting individualization patterns with interaction patterns by variables of the user interface profile being the output of the one and the input of the other pattern. e.g., user interface profile variables such as numericNavigation determine, which variant of the interaction pattern bundle »Main Menu« to select>
Patterns	<indicate the patterns of the bundle>

Individualization patterns are described in the following format:

TABLE IV. Template for individualization patterns

<ID>	**<name of the individualization pattern>**
Problem	< describe a general interaction problem related to a value range(s) of certain aspects/variables of the user profile and the device profile>
Pattern Bundle	<pattern bundle>
Context	**IF** <check values of user profile variable(s) and/or device profile variable(s)>
Solution	**THEN** Set <user interface profile variable(s)> = <value(s)>
Code	<if applicable, provide a reference to the **reference** related code to implement the solution>
Diagram	<if applicable, illustrate the design solution in a schematic and concise manner>
Rationale (references)	<explain the principles or rationale behind the pattern and provide references to literature, standards, etc.>
Substitutes	<individualization pattern(s)>: grouping two or more individualization patterns to a pattern bundle
uses <variable(s)> as set by <pattern bundle>	<fill in variables used within the if-Statement of this pattern, reflecting the sequential steps of device profiling and individualization and their dependencies in the UI parameterization process>
requires <variable(s)> as set by <pattern bundle>	<fill in variables used within the then-Statement of this pattern, establishing the connection to variables set by device-specific patterns, which are referred to in the solution statement of a individualization pattern>

B. *Example: Individualization Pattern Bundle "Display Mode"*

To show how individualization patterns are used, the pattern bundle "Display Mode" will be explained. These patterns suit the purpose to set the ratio between textual information and graphical information.

TABLE V. Pattern Bundle "Display Mode"

Individualization Pattern Bundle	**Display Mode**
Problem	These patterns decide about the ratio between textual information and graphical information
sets <variable(s)> as used by <pattern bundle>	**sets** *displayMode* **as used by** MainMenu, FunctionsMenu, NavigationMenu, ListWithAttributes MultiListWithAtrributes, TreeStructure
Patterns	Display Mode – mainly text (default) Display Mode – text and graphics Display Mode – mainly graphics Display Mode – graphics only Display Mode – text only

All patterns of this pattern bundle will set the parameter *displayMode*, which will be used by various Interaction Patterns like MainMenu etc.

Table VI presents an example of a pattern, which determines the ratio of textual and graphical information on the screen in accordance with current setting in the user

profile variables *understandingAbstractSigns* and *languageReception*.

TABLE VI. Pattern "Display Mode – text & graphics"

Individualization Pattern	Display Mode – text and graphics
Problem	Users with mild language reception impairments will benefit from a balanced display mode with text and graphics.
Pattern Bundle	Display Mode
Context	**IF** (*understandingAbstractSigns* < 2 AND 1 ≤ *languageReception* < 2) OR (*understandingAbstractSigns* ≥ 2 AND *languageReception* ≥ 2)
Solution	**THEN** set *displayMode*: text and graphics
Rationale (references)	See MyUI Document D2.1 (p. 57), URC11 citing ER-FG-01 (2010), National Institute on Aging (NIA) National Library of Medicine (2005) Making Your Website Senior FriendlyL A Checklist. (http://www.nia.nih.gov/HealthInformation /Publications/website.htm, Kurniawan, S. and Zaphiris, P. (2005). Research-Derived Web Design Guidelines for Older People. Proceedings of 7th international ACM SIGACCESS Conference on Computers and Accessibility 20-05 (ASSETS'05), pp 129-135: http://portal.acm.org/ citation.cfm?id=1090810
Substitutes	Display Mode – mainly text Display Mode – mainly graphics Display Mode – graphics only Display Mode – text only
uses <variable(s)> as set by <pattern bundle>	-
requires <variable(s)> as set by <pattern bundle>	-

The other individualization patterns of the pattern bundle "Display Mode" work likewise.

C. Example: Individualization Pattern Bundle Font Size

Another interesting example for a pattern bundle is "Font Size". This individualization patterns identify the adequate font size for a certain user, using a certain device. Beforehand, device-specific patterns would have been executed calculating possible font sizes for the device and his resolution. As it can be seen, the individualization patterns refer to these variables.

TABLE VII. Pattern Bundle "Font Size"

Individualization Pattern Bundle	Display Mode
Problem	For different situations there are different font sizes. Within this pattern bundle, the patterns set the appropriate font size according to user profile. They don't set fixed point sizes, but select relative font size variables as set by device-specific patterns
sets <variable(s)> as used by <pattern bundle>	**sets** *titleFontSize* *bodyTextFontSize* *complementaryTextFontSize* **as used by** several user interface elements
Patterns	Font Size - Philips Standard (Default) Font Size - Medium Font Size - Large Font Size - X Large

Table VIII presents an example of a pattern, which determines the font size in accordance with the current setting in the user profile variable *visualAcuity*. Beforehand, device-specific patterns have set the possible font sizes of this device, as an extra-large font size one a smartphone would probably have another value then an extra-large font size on an iTV.

TABLE VIII. Pattern "Font Size – Xlarge"

Individualization Pattern	Font Size - Xlarge
Problem	Users with significant vision impair-ments need significantly increased font sizes in order to be able to read the dis-played text in a comfortable manner.
Pattern Bundle	Font Size
Context	**IF** 3 ≤ *visualAcuity* < 4
Solution	**THEN** set titleFontSize = titleFontSizes.xlarge set bodyTextFontSize = bodyTextFontSizes.xlarge set complementaryTextFontSize = complementaryTextFontSizes.xlarge
Rationale (references)	Percentage re-sizing based on WCAG 2008. Font resizing standards for CSS suggest a scale between 100-200%
Substitutes	Font Size - Philips Standard (Default) Font Size - Medium Font Size - Large

 ISBN: 978-1-61208-225-7

uses <variable(s)> as set by <pattern bundle>	-
requires <variable(s)> as set by <pattern bundle>	**requires** titleFontSizes bodyTextFontSizes complementaryTextFontSizes **as set by** Device Font Sizes

The other individualization patterns of the pattern bundle "Font Size" work likewise.

Within the pattern repository, for every facet that influences the user interface, individualization patterns decide about how the ideal user interface for this special situation would look like. Further individualization pattern bundles, are for example:

- Audio Speech Speed: a pattern bundle that decides, depending on user characteristics like attention and processingSpeed, about the speed when vocal audio output is needed
- Navigate and Select: A pattern bundle, that considers used input devices as well as user characteristics (e.g., ICTLiteracy, fingerPrecision) in order to detect the method for navigation and selection for this certain usage situation

Detailed information about the shown pattern bundles and a lot of further patterns and pattern bundles can be found at [12].

VI. CONCLUSION AND FUTURE WORK

The MyUI framework enables an advanced development of accessible, adaptive user interfaces through a generic approach including all relevant parameters for UI adaptation, which are user, device and environment (context). Changes are tracked automatically through system interaction and sensors, and even runtime changes on one of the parameters can be treated by a runtime change of the user interface using adaptation patterns.

The knowledge about best practices for accessible and adaptive user interfaces is stored within an open, easy readable design pattern repository [12]. Different categories of patterns address different steps within the user interface adaptation process. The individualization patterns we have shown in detail within this paper determine global characteristics of the user interface, which are constant within the whole adapted user interface.

Today, the MyUI design pattern repository includes pattern for use cases like email and instant messaging. In the future, the patterns will be refined and improved stepwise while enlarging the MyUI system to develop more applications for smart environments and devices. The repository is open for changes by the community and presents a complete overview about knowledge and intelligence of the system. As the barrier for changes on the patterns is low, new scientific findings or practical insights can easily be included and will improve the adaptation process further on.

VII. ACKNOWLEDGEMENTS

The authors acknowledge the help of other partners in the MyUI consortium in this work: Phillips Consumer Lifestyle, Ingenieria Y Soluciones Informaticas Del Sur (ISOIN), CleverCherry.com, Universidad Carlos III de Madrid, The University of Nottingham, Forschungszentrum Informatik, Birmingham City Council, Semmelweis Egyetem, and Ayuntamiento De Getafe.

The MyUI project is funded by the European Union (FP7-ICT-2009-4-248606).

REFERENCES

[1] M. Blumendorf, G. Lehmann, and S. Albayrak (2010). Bridging Models and Systems at Runtime to Build Adaptive User Interfaces. InProceedings of the 2nd ACM SIGHCHI symposium on Engineering interactive computing systems (EICS '10). ACM, New York, NY, USA, 9-18.

[2] J. Borchers. A pattern approach to interactive design. UK, John Wiley & Sons Ltd., 2001.

[3] A. Bunt, C. Conati, and J. McGrenere. (2009). A Mixed-Initiative Approach to Interface Personalization. AI Magazine 30(4).

[4] H. Dieterich, U. Malinowski, T. Kühne, and M. Schneider-Hufschmidt, (1993). State of the Art in Adaptive User Interfaces. In: M. Schneider-Hufschmidt, T. Kühne & U. Malinowski (Eds.): Adaptive User Interfaces: Principles and practice. Amsterdam: North-Holland, 13–48.

[5] G. Fischer, (1993). Shared knowledge in Cooperative Problem-Solving Systems – Integrating Adaptive and Adaptable Components. In: M. Schneider-Hufschmidt, T. Kühne & U. Malinowski (Eds.): Adaptive User Interfaces: Principles and practice. Amsterdam: North-Holland, 49–68.

[6] M. C. Huebscher and J. A. McCann (2004). Adaptive middleware for context-aware applications in smart-homes. Proceedings of the 2nd workshop on Middleware for pervasive and ad-hoc computing. ACM New York, NY, USA, 111-116.

[7] A. Kobsa, J. Koenemann, and W. Pohl (2001). Personalised hypermedia presentation techniques for improving online customer relationships. The Knowledge Engineering Review, Vol. 16:2, 111–155.

[8] W. E. Mackay (1991). Triggers and barriers to customizing software, in: S. P. Robertson, G. M. Olson & J. S. Olson (Eds.). Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '91), New York: ACM. 153–160.

[9] M. Peissner, A. Schuller, and D. Spath (2011). A Design Patterns Approach to Adaptive User Interfaces for Users with special Needs. In: HCII'11 Proceedings of the 14th international conference on Human-computer interaction: design and development approaches - Volume Part I. Springer-Verlag Berlin, Heidelberg, 268-276.

[10] F. Portet, M. Vacher, C. Golanski, C. Roux, and B. Meillon (2011). Design and evaluation of a smart home voice interface for elderly: acceptability and objection aspects, in: Personal and Ubiquitous Computing, London, vol. 15, 1-18.

[11] FZI Living Lab Ambient Assisted Living FZI Forschungszentrum Informatik, Germany, http://aal.fzi.de/, last access on 2012/03/08

[12] MyUI Design Patterns Repository. Available at http://myuipatterns.clevercherry.com, last access on 2012/03/08

 ISBN: 978-1-61208-225-7