An Environment for Designing and Sharing Adaptation Rules for Accessible Applications

Raúl Miñón
School of Informatics,
University of the Basque Country
Manuel Lardizabal 1, E-20018
Donostia, Spain
raul.minon@ehu.es
+34 943015590

Fabio Paternò
ISTI-CNR
Via Moruzzi 1, 56124 Pisa, Italy
fabio.paterno@isti.cnr.it
+39 050 3153066

Myriam Arrue
School of Informatics,
University of the Basque Country
Manuel Lardizabal 1, E-20018
Donostia, Spain
myriam.arrue@ehu.es
+34 943015590

ABSTRACT
In this work we present a design space for adaptation rules for applications accessible to people with special needs, and an environment supporting the sharing of such rules across various applications. The adaptation rules are classified according to the target user disabilities, as well as other relevant criteria useful to ease their integration in other design tools.

Author Keywords
Accessibility; Adaptation Rule; Model-Based UI; Design Space;

ACM Classification Keywords
H.5.2 [Information Interfaces and Presentation]: User Interfaces—User-centered design; interaction styles; theory and methods

INTRODUCTION
Besides satisfying different countries' legislations and expanding the target user groups, the development of accessible applications must be addressed in order to allow people with special needs to enjoy the same experience as everyone else. For this purpose, accessibility guidelines, such as the WCAG 2.0 [14], must be fulfilled. Unfortunately, there are many developers that do not consider these rules and, in consequence, many users cannot access the resulting applications. Additionally, some accessibility guidelines are adequate for a specific group of users, while they are not very appropriate for others.

One promising approach to deal with these problems is Model-Based User Interfaces design, since it enables the possibility of generating different Final User Interfaces (FUI) from logical descriptions independent of the programming language and the user platform. In addition, this approach also allows modifications of the FUI with limited effort. Thus, the FUI can be easily adapted to the specific user's capabilities. The required standards and accessibility guidelines can be included in the generation process of the FUI. The integration of accessibility requirements at design time in the UI development process of the Cameleon Reference framework [1] was proposed in [7]. However, that approach does not consider context-aware adaptation rules. The MyUI Toolkit [10] provides design patterns for building adaptive UIs. Nevertheless, this toolkit neither provides a mechanism for sharing these patterns to other MBUI design tools nor consider explicitly key aspects of the users with disabilities, such as the assistive technology required.

A Global Public Inclusive approach [13] has been proposed for automatically providing disabled users with a solution able to enhance their interaction with different public services. In contrast, in our solution the designer takes part in the adaptation process in order to identify the most relevant adaptation rules.

General design spaces for considering multiple dimensions of context-aware adaptations and a reference framework for classifying adaptation rules in different domains have already been proposed [2] [4]. In our case, we focus on interface adaptations for improving overall accessibility, which is not sufficiently covered in the previously mentioned research works. We also propose an environment to support sharing of adaptation rules across various design tools, so that accessibility issues can be considered in a more convenient manner.

More specifically, the objective of this paper is twofold: to define a design space able to include a comprehensive set of adaptation rules for accessible applications and to design a repository for compiling many different adaptation rules devoted to people with special needs. For this purpose, we have classified the gathered adaptation rules according to the user disability and other relevant aspects. The resulting repository will enable third-party designers that use model-based UI approaches to easily integrate the available adaptation rules with their tools, either at design time or at run-time.

The paper is organized as follows; section 2 discusses the considered key concepts that structure the design space; subsequently, section 3 describes the design of the repository; afterwards, section 4 illustrates the architecture of the environment and explains the integration process; and finally, section 5 provides conclusions and indications for future work.

DIMENSIONS OF THE DESIGN SPACE

There are some important concepts that can influence the adaptation process, such as, the target user, the granularity level, whether the adaptation aims to generate an adapted or an adaptive UI, and the considered abstraction level of the UI. In this work, these concepts are used to structure a design space able to indicate the different types of adaptations for enhancing the user interfaces devoted to people with disabilities. Figure 1 illustrates the proposed design space populated with different examples, which cover all possible values of the dimensions proposed. For some dimensions it is possible to define an ordering (Granularity Level and Abstraction Level), while for others (User Disability and Adaptation Type) we just indicate the set of possible values.

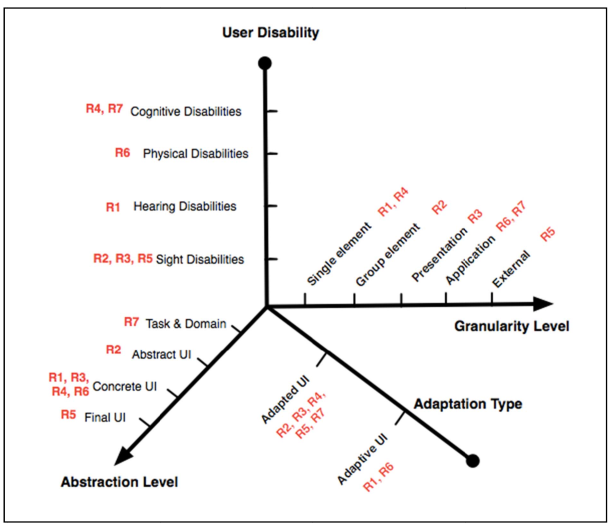


Figure 1. The design space proposed with the adaptation rules exemplified related with each concept.

The concepts considered in the design space are discussed in the rest of this section.

Target Users. It is well known that people with different capabilities require different types of interactions and, as consequence, different types of adaptations. Thus, users are classified in terms of disabilities in order to select the most appropriate adaptation rules for them. In this paper, some disabilities have been considered to test the design space. The disabilities are mainly based on the WCAG 2.0 guidelines [14] and belong to four general categories: Sight Disabilities including low vision, blindness, colour blindness, photosensitivity and eyestrain; Hearing Disabilities including deafness and mild-deafness; Physical Disabilities including limited-movement, key-board only users, Parkinson and paraplegia; and Cognitive Disabilities including decline in maintaining attention, learning disabilities, language disabilities and reduced memory capacity. With this disabilities classification, the adaptation rules can be related to a general category such as the rule 4 or to a specific disability such as the rule 7.

The information for defining relationships between adaptation techniques and considered disabilities has been collected from various specific research works: adaptations techniques devoted to users with motor impairments were collected from [3], techniques focused on cognitive and sensory impairments were taken from [5], [8] and [11], and the information in [6] led us to techniques for visual impairments.

The following examples define an adaptation rule for some specific disabilities: Rule 1 is related to mild-deafness and Rule 2 to blindness. The rules are structured in terms of events, conditions, and actions describing their effects.

Rule 1 (deaf target group):

Event: the noise of the environment changes to a value higher than 25 decibels.

Condition: the user has a mild-deafness disability.

Action: every video has to be changed to a video with subtitles.

Description: when the noise level in the environment is increased over 25 decibels, subtitles are added to each video element.

Rule 2 (blind target group):

Event: the user accesses an application with many interaction elements.

Condition: the user is blind

Action: an application table of content is created to easily access each interaction element.

Description: if a UI contains many interaction elements, a blind user needs a mechanism to identify and easily access each element.

Granularity levels. Adaptation can have an impact at different granularity levels of the user interface: Application, Presentation, Group of Elements and Single Element. Another level denominated External is also considered to refer to adaptations that go further the current application. For example, some cases require that an assistive technology be launched.

These granularity levels are helpful to identify the most suitable order of execution for the adaptation rules required, since it is possible to follow a policy indicating to apply first the adaptation rules related to higher granularity levels and then the others. Thus, for example, if there is one rule changing a group of elements, and a rule changing only one of such elements, following this policy the change specific to the single element is not lost.

The following examples define an adaptation rule for specific granularity levels: Rule 3 is related to presentation level, Rule 4 to single element level and Rule 5 to external level.

Rule 3 (presentation granularity level):

Event: the user interface is activated.

Condition: the user is colour-blind.

Action: change the foreground colour to black and the background colour to white in order to provide a high-contrast UI.

Description: This rule is triggered when the UI is activated; then, it checks if the user has the colour-blind disability; finally, the action part changes the UI background and foreground colours to colours more appropriate for the user disability.

Rule 4 (single element granularity level):

Event: the UI contains an element with a timeout.

Condition: the user has a cognitive disability.

Action: remove the timeout or increase the time limit considerably if it is really necessary.

Description: this rule satisfies the WCAG 2.0 [14] success criterion 2.2.1: *"Cognitive limitations may require more time to read content or to perform functions such as filling out on-line forms. If Web functions are time-dependent, it will be difficult for some users to perform the required action before a time limit occurs"*

Rule 5 (external granularity level):

Event: the user interface is activated.

Condition: the user has low vision.

Action: activate a screen magnifier.

Description: this rule activates an assistive technology when the user has low vision.

Adapted UIs and Adaptive UIs. Adapted users interfaces are interfaces that are adapted at design time and are instantiated at run-time. In the case of people with special needs, they are focused on tailoring the most adequate user interface for the specific capabilities of the user. Thus, when users with disabilities interact with the UI, it is totally adapted to their needs. These UIs are valid when the context of the user is static. However, when the context (considered as user, platform, environment and social relations) is dynamic, for example when the user is walking or the noise of the environment is increased, sometimes the UI interface needs to be adaptive in order to change according to the surrounding context. One example of adaptation performed when the context changes is the change of UI modality. This dimension is necessary to identify in the design space both types of adaptations, the ones that are defined at design-time and those obtained at run-time.

The following example, Rule 6, defines an adaptation rule for adaptive UI.

Rule 6 (adaptive UI):

Event: the user begins to move.

Condition: the user has paraplegia AND the UI is not rendered with the vocal modality.

Action: the user interface is changed to the vocal modality.

Description: this rule is triggered when the user begins to move and a change in the context is detected. It checks if the user has Paraplegia and if the UI modality is not vocal. Finally, an equivalent UI with the vocal modality is provided.

Attributes / Rules	Adapted/ Adaptive	Disability Category	Granularity Level	Abstraction Level	Assistive Technology
Rule 1	Adaptive	Hearing Disabilities	Single element	CUI	NONE
Rule 2	Adapted	Sight Disabilities	Group element	Abstract	NONE
Rule 3	Adapted	Sight Disabilities	Presentation	CUI	NONE
Rule 4	Adapted	Cognitive Disabilities	Single element	CUI	NONE
Rule 5	Adapted	Sight Disabilities	External	FUI	Magnifier
Rule 6	Adaptive	Physical Disability	Application	CUI	Speech Recogniser
Rule 7	Adapted	Cognitive Disabilities	Application	Task & Domain	NONE

Table 1. The meta-information associated with each rule.

Abstraction Level. The consideration of the abstraction level in the design space is useful to provide the designers with the possibility of integrating the required adaptations in their applications at the level of generality preferred. Thus, if there was a specific requirement derived from an adaptation rule, the designer can perform the necessary changes in the first phases of the design process, which require less effort in making modifications. For instance, if an adaptation rule requires splitting the UI for a cognitive impairment person in various UIs, it is better for the designers to know it when they are defining the task model instead of knowing it when all the design of the UI is implemented and then considerable effort is required to update it.

According to the Cameleon Reference Framework [1], there are four different abstraction levels in the development process of model–based User Interfaces: Task and Domain, Abstract User Interface, Concrete User Interface and Final User Interface. The adaptations can be applied at each of these different levels. For instance, adaptation rules related to task sequencing must be considered at the Task & Domain level, whereas adaptation rules related with some specific UI modality must be considered at the Concrete UI level.

The following example, Rule 7, defines an adaptation rule for the task abstraction level.

Rule 7 (task abstraction level):

<u>Event:</u> the application contains too many different interaction elements for performing different tasks at the same time.

<u>Condition:</u> the user has problems in maintaining attention.

<u>Action:</u> the UI is organised in such a way that only one task is shown in every moment.

<u>Description:</u> showing many options and interaction elements can be confusing for users with problems in maintaining attention. This rule aims at simplifying the UI so users can efficiently perform tasks.

DESIGN OF THE ADAPTATION REPOSITORY

The repository stores the different adaptation rules modelled with some meta-information in order to select and classify them adequately. The necessary meta-information is:

Rule ID. A unique identifier.

Source. The adaptation rule's path and name.

Adapted/Adaptive. This value indicates if the adaptation rule is devoted for adapted UI or for adaptive UI.

Disability. This attribute is composed of a set of values that indicate the disabilities addressed with the adaptation rule.

Granularity Level. This value specifies the granularity level.

Assistive Technology. This value indicates if the adaptation rule requires any specific assistive technology to work. Although the assistive technology is not a dimension in the design space, it is useful to include it in the meta-information for practical purposes.

Abstraction Level. This value indicates the abstraction level associated with the adaptation.

Table 1 provides the meta-information associated with each example rule included in the previous section. In order to specify the rules we have used the AAL-DL [12] language since it fulfils our requirements. This language allows modelling of adaptation rules following an ECA approach (Event-Condition-Action). The event part identifies when the adaptation rules have to be applied, such as when the UI is activated or when a change in the context is detected. The condition part filters the rules for specific situations, target groups or contexts. Finally, the action part specifies necessary changes in the UI.

ARCHITECTURE OVERVIEW AND INTEGRATION PROCESS

As previously mentioned, the repository allows the integration of adaptation rules devoted for people with disabilities in Model-Based User Interface tools. These adaptation rules can be integrated both at design time and at run-time. At design-time it allows designers to obtain the necessary adaptation rules for specific groups of people with disabilities at the different abstraction levels, whereas the run-time approach provides support for context-aware UIs. Figure 2 illustrates the adaptation process, divided into several steps. We include a normalization step because we want to provide the possibility of supporting various model-based languages and their supporting tools, but we cannot implement transformations for all of them. Thus, we have a normalization step that transforms them to the MARIA language. MARIA [9] was selected as underlying language, because it implements every levels of the Cameleon Framework both at design-time and at run-time. Additionally, it is compliant with the language AAL-DL [12], which was selected for the implementation of the adaptation rules as mentioned in the previous section. Below these steps are explained in detail:

Step1: the Model-Based UI (MBUI) design tool performs a query to the *Adaptation environment* sending two parameters, the MBUI specification and the filter. The filter contains the user disability. The MBUI design tool can access the repository through a web service integrated in the system.

Step 2: the *Adaptation environment* receives the query and sends it to the *Request Parser* module. This module identifies the User Interface Description Language (UIDL) used, the **abstraction level of the UI** and the **disability** selected by the designer. The *Request Parser* module sends this information to the *UIDL Normalization* module.

Step 3: the *UIDL Normalization* module transforms the received UI to the MARIA language.

Step 4: after the UIDL normalization, the *Adaptation Manager* module selects the adaptations rules dedicated to the disability selected and the UI abstraction level. As mentioned in the previous section, the adaptation rules are stored in the *Adaptation Repository*, therefore, this module establishes communication with it to obtain the rules. It is worth to point out that the designer can access the *Adaptation environment* when considering every abstraction level of the UI development process to obtain the available adaptation rules of the corresponding level. Once the adaptation rules are selected, the module orders the adaptation rules to solve conflicts between them. For this purpose, as previously stated, the **granularity level** of the adaptation rule is a key concept. Firstly, the ones with the granularity level with the *external* value will be applied; then, the rules with the *application* value; after that, the rules with the *presentation* value; subsequently, the ones with the *group of elements* value; and finally, the rules with the *single element* value.

Step 5: having the rules selected with the adequate order, the *Adapter* module applies the adaptation rules to the UI specification. The result of this step provides a UI specification adapted to the user disabilities.

Step 6 and Step 7: the UI adapted is transformed into the original model-based language by means of the *UIDL Inverse Normalization* module and it is returned to the corresponding MBUI design tool. Then**,** the MBUI design tool is able to generate FUIs adapted to the user disabilities.

Step 8: as previously mentioned, the repository also provides support for context-aware UIs. The designer of the application will be able to decide whether enable this support or not. If he/she does not want to enable it, the process finishes in the previous step. Otherwise, the MBUI design tool sends to the *Adaptation environment* the MBUI specification and the context information in order to apply the context-aware adaptation rules. The MBUI design tool needs to integrate a specific code to detect changes in the context. Then, each time the context changes the FUI will be adapted if necessary.

Step 9: the repository receives the new query and sends it to the *Request Parser* module.

Step 10: the *UIDL Normalization* module transforms again the received MBUI specification to the MARIA language.

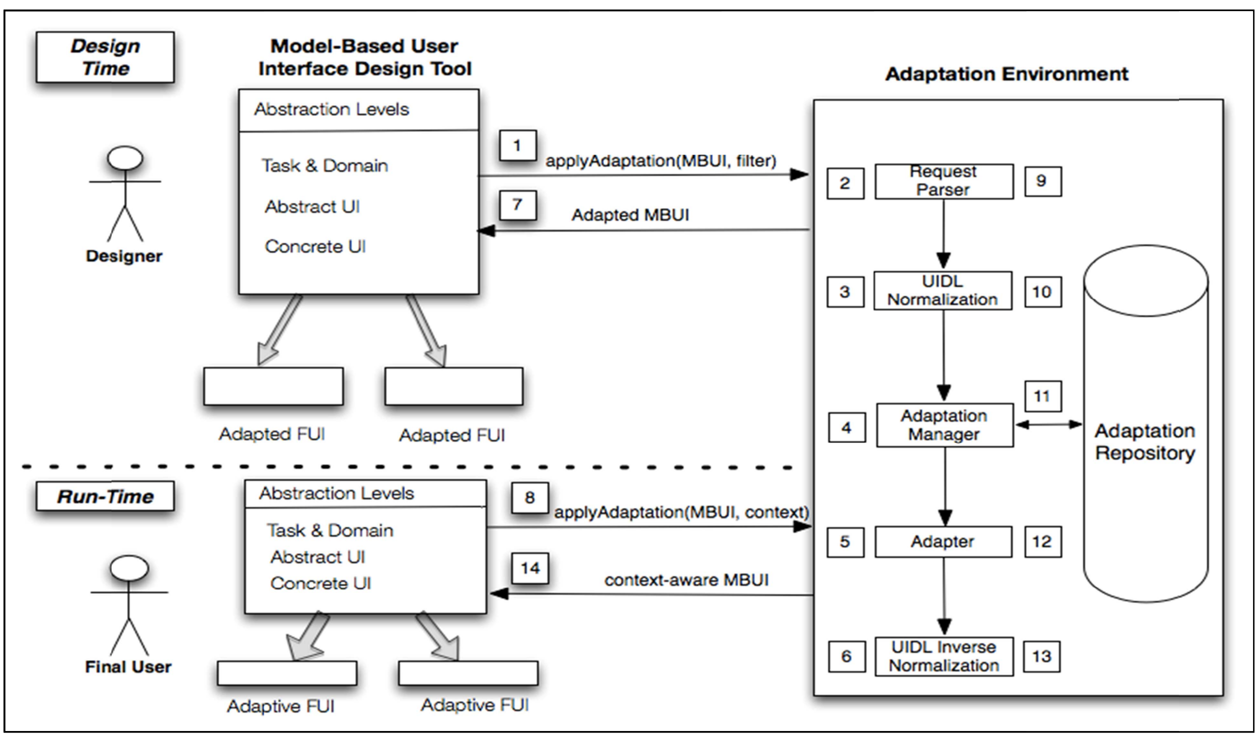
Step 11: the context-aware adaptation rules related to the context information are selected and, as in the Step 4, the adaptation rules selected are ordered.

Step 12: the necessary context-aware adaptations are applied to the MBUI specification through the *Adapter* module.

Step 13: the MBUI specification is transformed into the original model-based language.

Step 14: the context-aware MBUI specification generated is returned to the MBUI design tool, which will then be able to generate the context-aware FUI.

Thus, when the process is finished, users with specific disabilities will interact with a FUI adapted to their needs.



This FUI will also be able to support changes in the surrounding context improving the interaction and the experience of the user through an accessible context-aware FUI.

CONCLUSIONS AND FUTURE WORK

The main contribution of this work is to provide a design space structured with the key concepts that must be exploited when considering adaptations for people with disabilities. In addition, we present examples covering the different dimensions of the design space in order to better understand it and ease the integration of different types of adaptations in other model-based approaches.

Moreover, we describe the design of a repository and the necessary meta-information to share and access these adaptation rules, and describe an architecture to support the interaction with the repository, both at design-time and at run-time. This repository facilitates the integration of adaptation rules in other applications.

In future work, we will explore adaptation to support combination of different disabilities. Furthermore, we will investigate further mechanisms in order to solve possible conflicts among the rules.

ACKNOWLEDGMENTS

This research work has been partly funded by the Spanish Ministry of Science and Innovation ModelAcces project (grant TIN2010-15549), the Department of Education, Universities and Research of the Basque Government (grant IT395-10) and by the EU FP7 STREP SERENOA project (http://www.serenoa-fp7.eu). In addition, Raúl Miñón holds a Ph.D. scholarship and a scholarship for performing a research stay in a foreign country from the Research Staff Training Programme of the Department of Education, Universities and Research of the Basque Government.

REFERENCES

1. Calvary, G., Coutaz, J., Bouillon, L., Florins, M., Limbourg, Q., Marucci, L., Paternò, F., Santoro, C., Souchon, N., Thevenin, D., Vanderdonckt, J. 2002. The CAMELEON reference framework, 2002. Deliverable 1.1, CAMELEON Project. http://www.w3.org/2005/Incubator/model-based-ui/wiki/Cameleon reference framework
2. Coutaz, J., Calvary, G. HCI and Software Engineering for User Interface Plasticity. In *Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications*, Third Edition, Julie A. Jacko (Ed.), 2012, 1195-1220
3. Gajos, K.Z., Weld, D.S., Wobbrock, J.O. Automatically Generating Custom User Interfaces for Users with Physical Disabilities. In *J. Artificial Intelligence*, 2010, Elsevier, Vol. 174, N.12-13, 910-950.
4. Genaro Motti, V., A computational framework for multi-dimensional context-aware adaptation. In *Procs. of the 3rd ACM SIGCHI symposium on Engineering interactive computing systems*, 2011. ACM, 315-318.
5. Kurniawan, S. H., King, A. D., Evans , G., Blenkhorn, P. L. Personalising web page presentation for older people. In *Interact. Comput*., 2006, ACM, 18, 3, 457-477.
6. Lunn, D., Bechhofer, S., Harper, S. The SADIe transcoding platform. In *Proceedings of the 2008 international cross-disciplinary conference on Web accessibility, W4A 2008*, 128-129. ACM Press.
7. Miñón, R., Moreno, L., Abascal, J. A Graphical Tool to Create User Interface Models for Ubiquitous Interaction Satisfying Accessibility Requirements. In *Universal Access in the Information Society*, 2012, Springer, 1-13. http://link.springer.com/article/10.1007/s10209-012-0284-x#
8. National Institute on Aging and National Library of Medicine. Making Your Web Site Senior Friendly: A Checklist. , NIH & NLM, 2002, http://www.nlm.nih.gov/pubs/checklist.pdf
9. Paternò, F., Santoro, C., Spano, L.D. MARIA: A Universal Language for Service-Oriented Applications in Ubiquitous Environment. In *ACM Transactions on Computer-Human Interaction*, 2009, Vol.16, N.4, 19:1-19:30.
10. Peissner, M., Häbe, D., Janssen, D., Sellner, T. MyUI: generating accessible user interfaces from multimodal design patterns. In *Procs. of the 4th ACM SIGCHI symposium on Engineering interactive computing systems*, 2012. ACM, 81-90.
11. Richards, J.T., Hanson, V.L. Web Accessibility: A Broader View. In *Procs. of the 13th Int. conference on World Wide Web, WWW 2004*, ACM Press , 72-79.
12. Serenoa Project, AAL-DL: Semantics, Syntaxes and Stylistics, http://www.serenoa-fp7.eu/wp-content/uploads/2012/07/SERENOA_D3.3.1.pdf
13. Vanderheiden, G.C., Treviranus, J. Creating a Global Public Inclusive Infrastructure. In *Procs. of HCI Internationanl, 2011* (5), LNCS 6765, Vol. 5, 517-526.
14. W3C, WAI, Web Content Accessibility Guidelines (WCAG) Overview, 2012, http://www.w3.org/WAI/intro/wcag.php