A modular approach to user interface adaptation for people with disabilities in ubiquitous environments

Julio Abascal, Amaia Aizpurua, Idoia Cearreta, Borja Gamecho, Néstor Garay, Raúl Minón
{julio.abascal, amaia.aizpurua, idoia.cearreta, borja.gamecho, nestor.garay, raul.minon}@ehu.es

Abstract. This paper presents the EGOKI system designed to generate adapted mobile user interfaces in order to provide people with disabilities with access to ubiquitous services. The main goal of this system is to automatically create accessible user interfaces that allow access to services provided by local devices, such as ATMs, vending machines, information kiosks, smart home appliances, etc. In most ubiquitous computing environments the middleware provides an abstract description of the user interface after the “discovering” phase. From this description EGOKI creates an instance of the user interface tailored to the user characteristics and to the technical features of his or her mobile device. The adaptation is based on user and device models, managed by means of ontologies.

Keywords: Ubiquitous computing, Special needs, Assistive technologies, Mobile systems.

1 Introduction

The growing ubiquitous computing paradigm allows the provision of context-aware services to mobile users. In addition to the usual computing requirements, these environments entail wireless network infrastructures and special management software, usually called middleware. When a mobile computing device (smart phone, PDA, etc.) enters into such a space (frequently known as Ambient Intelligence environments), the middleware establishes the communication with the local network in a way that is transparent to the user. After the “discovering” and “presentation” phases, the available local services are offered to the user. In order to be operated some of them may require a specific user interface that is downloaded to the user’s mobile device.

This type of environment can be extremely helpful for people with disabilities who have mobile devices adapted to their characteristics. In this way, using their own device they can access several local services that can otherwise be inaccessible to them, such as ATMs, vending machines, information kiosks, smart home appliances, etc.
This is only possible if the downloaded user interface is rendered to the mobile
device in an accessible mode. The large variety of user characteristics and restrictions
(due to the broad range of disabilities) and the peculiarities of the devices used by
them makes it necessary to adapt the “basic” user interface supplied by the service
provider to the specific needs of the user and his or her device.

In this paper we present the EGOKI system, which aims to automatically generate
user interfaces adapted to the features and preferences of users with disabilities.
These interfaces are intended to provide access to ubiquitous services in intelligent
environments. In this context, the provider of the service has to supply a machine-
readable abstract description of the interface. From this description, EGOKI
dynamically creates an instance of the interface running on the user device. To adapt
the interface to the user characteristics, it is necessary to take into account what the
most suitable communication modalities are for each user, mapping them to the
appropriate media.

Ubiquitous systems handle a huge quantity of information that can be used to infer
knowledge about the user, the environment and the tasks. Modelling this knowledge
would contribute to enhancing the generation of adapted user interfaces. Ubiquitous
Computing itself frequently includes user modelling and personalization as a goal, in
order to take into account the human context [1]. This goal requires a component that
can manage the adaptation of the information resources and make the interaction
comfortable for each user of the ubiquitous environment. There are some interesting
precedents in this area. For instance, the UBI system [2] identifies resources for
universal access to mobile services. The UBI system allows adaptation of the user
interface to a particular device (Personal computer, PDA or mobile) or user interface
type (Web UI or GUI) in order to cater for changes in user needs. The INREDIS
project1 was intended to create a large infrastructure that would allow ubiquitous
interaction, service interoperation and adaptation to the user. Some of the ideas of the
EGOKI system were firstly tested in INREDIS.

The remainder of this paper is structured as follows: In section 2, we present the
rationale for the adopted adaptive methodology. The next two sections present the
middleware for the ubiquitous environment and the language selected for the abstract
user interface description. Section 5 describes the EGOKI adaptive system. Finally,
the validation, conclusions and future work are discussed.

2 Foundations of the EGOKI Adaptive System

User modelling techniques provide the opportunity to create personalized interfaces
that adapt the content and presentation to the needs of the user, greatly enhancing the
quality and accessibility of the interfaces. Nevertheless, adaptive interfaces for people
with disabilities have to consider several other factors included in the interaction
context (task, location, technology used, etc). A number of recent works in the
Ambient Intelligence area also consider device characteristics [3].

---

1 INREDIS Project [http://www.inredis.es/Default.aspx]
Modelling the different participants in the interaction (users, devices and their contexts of use) is essential in order to be able to perform adaptations. User characteristics are taken into account in order to automatically adapt the interface to his or her characteristics, abilities, needs and preferences. In addition, the device used to access services has a determinant influence on the user’s experience. Device characteristics are classified into hardware and software features in order to model relevant issues such as display size, supported browsers and mark-up languages, available peripherals, etc. Once the information is obtained, through either explicit or implicit user intervention, it is possible to automatically adapt the functionalities of the interface of a service as is done, for instance, in the Unified User Interface [4].

Apart from modelling, another essential foundation of the system is the adaptation process. A wide variety of adaptive techniques and methods exist and they can be used for specific purposes. In the web context, three main adaptation areas are distinguished: content, presentation and navigation adaptation [5]. However, not all systems require the application of all three types of adaptations. The use of multimodal interfaces can be included in the adaptation method: by providing various resources oriented to different user characteristics and transmitted through different communication modalities, people with different capabilities will be able to access the service.

3 Technological support for the EGOKI ubiquitous environment

As previously mentioned, a ubiquitous environment is supported by wireless networks managed by means of a middleware level. The role of the middleware is essential because it ensures the discovering, communication and interoperation between the diverse computers involved. Although this mechanism must be transparent to the user, it is necessary to summarily describe the underlying middleware level because it determines key issues for the interface generation performed by EGOKI: how the abstract description of the service is created and provided, and what is the method of communication between the user device (where the user interface runs) and the target device (where the service is provided), as can be seen in Figure 1.

For this research work, we selected the Universal Control Hub (UCH)² [6], an interoperability framework that provides the abstract user interface to the devices that have been connected to the ubiquitous system. It is used to handle interoperability issues and to facilitate communication between the user interfaces and the ubiquitous environment, by providing abstract descriptions of the user interfaces. This framework is mainly designed to support Ambient Assisted Living (AAL) environments. It has been used in a number of European projects [7], such as 12HOME (2006-2009), VITAL (2006-2010), Brainable (2010-2012) and MonAMI (2006-2010). However, to our best knowledge, the automatic generation of interfaces based on user modelling has not been addressed in this context.

² UCH is the implementation of the URC ISO Standard (ISO/IEC 24752-1 to 5)
### 3.1 Ubiquitous middleware infrastructure supporting EGOKI

The URC Standard describes three different kinds of files: User Interface Server (UIS), Resource Sheet (RSHEET) and Target Description (TD) [6]. The Standard also includes its own User Interface Description Language called Presentation Template [8], although this feature is not yet implemented in the UCH. To ensure future compatibility with middleware and ubiquitous systems other than UCH, EGOKI does not make use of Presentation Template, a tool specific to the URC standard. Instead, we use User Interface Mark-up Language UIML [9] to describe the user interface with resources and the behaviour. As we will explain later, a UCH service provider must include a UIML file with the information from the UIS, TD and RSHEET xml files in order to allow access to EGOKI.

Currently, UCH has a JavaScript Library called WebClient which is used to build simple user web interfaces automatically (without multimedia resources) and provide complex user web interfaces manually (with full design and multimedia resources developed by a human designer). When a user wants to control a Target, a JavaScript-enabled web browser is used to render the interface. In our case, EGOKI uses the WebClient library functions. This work has also contributed to addressing one of the open UCH issues [7]: how to automatically select or generate suitable user interfaces using all types of resources (text, video, audio, images…).

The general steps of the interaction between the EGOKI and the UCH are described below (as it can be seen in Figure 1):

1. Through its mobile device the user requests an adapted interface to the EGOKI System in order to use a specific service.
2. The EGOKI system generates an adapted user interface in XHTML.
3. The user interacts with the service, producing changes in the state of the ubiquitous service.
4. The JavaScript embedded in the XHTML web document updates the state and informs the UCH.
5. The UCH updates the state of the ubiquitous service and sends the changes to all the users of that service (if it is shared by more than one person).

![Figure 1](image_url). Access to the service though EGOKI using UCH.
6. Each XHTML web document receives the changes in the state of the ubiquitous service made by each user.
7. The JavaScript code changes the value of the fields updating the user interface.

4 Abstract User Interface Description

As previously stated, the information exchange between the adaptive module and the Interoperability module is specified using UIML, which is a User Interface Description Language (UIDL). For our purposes, the UIDL must fulfill certain requirements in order to be able to generate suitable user interfaces:

- To provide an interface abstraction level to describe the interface elements as Abstract Interaction Objects (AIO). Subsequently, the adaptive system transforms the AIO into Concrete Interface Objects and Final Interface Objects.
- To add and describe user interface resources requires building a complete and usable interface for a specific user.
- To define mappings to add support to UCH or other interoperability middleware in order to generate functional user interfaces.

Several approaches have been proposed in recent years [10] to define description languages. Of these, UIML is used since it offers adequate mechanisms to describe a user interface in a robust manner; as a result all necessary resources types can be properly specified. Although UIML provides ways of rendering the final user interface through specific vocabularies and renderers, we define our own rendering process since there are no available renderers for XHTML. In EGOKI, UIML is used as this is recommended by the standard. Labels and structures are compliant with UIML 4.0 cs1. Nevertheless, due to the specific characteristics of UCH some modifications were needed.

The main elements of the UIML scheme used by EGOKI are described below:

- The “structure” element contains the specific user interaction elements grouped in three “part” levels.
- The “content” element describes the available resources for a target.
- The “rules” element describes the events that will have to address the user interface.
- The “peers” element describes the mapping between the “rules” section and the WebClient JavaScript library.

Based on this UIML, EGOKI can provide a user interface with the appropriate resources for the user’s capabilities. This interface interacts with the UCH and updates the value of the variables described using the WebClient library. The interface is automatically generated by the system and does not need to be modified once it has been generated.
5 The EGOKI adaptive system

The main objective of the UI adaptation component is to automatically generate accessible user interfaces. Since the UI adaptation component is a part of the general architecture (see Figure 1), it has to meet the following requirements:

- User identification: the user must be logged in to use any target device provided through the proposed architecture. In this way, the Information about the user that has previously been collected in the model can be used.
- Service description: each service offered through the system must provide an associated UIML document, where information on the functionalities and resources provided by the service are described. Every UIML document, for a target service, must at least offer textual resources.
- User device: The mobile user device must run a JavaScript-enabled browser (in order to be able to interact with the user interface generated).

5.1 Architecture of the system

The implemented architecture is sufficiently modular to accommodate future enhancements (see Figure 4). The system is currently composed of two main parts: the first is the component responsible for modelling information on the user and is called the Knowledge Base (KB). The second is the adaptation component that automatically generates user interfaces tailored to the user needs specified through the modelling. Each of these two parts is composed of diverse modules, as described in the following paragraphs.

5.1.1 Knowledge Base (KB)

Several modelling approaches have been studied and eventually an ontological model was chosen, as in [3], since ontologies offer automated reasoning, dynamic classification and consistency checking, allowing the extraction of other relevant features without impacting on the user [11]. Therefore, an ontology was built with the aim of modelling users, devices, context and other adaptation-related concepts.

Several user features are represented in the designed ontology, such as the level of capability for each modality of communication (visual, auditory, motor, speech, etc.). In addition, other data, such as the preferences of the user, are modelled (for instance: preferred colour contrast, font size, font type, etc.).

Moreover, device features are also modelled, such as device-supported fonts and formats, available communication channels and other characteristics (for example, availability of voice recognition or video display).

In addition, the ontology contains a user location attribute for the context-aware services that require user localization.

Resources and adaptations for each user have been also modelled in the ontology by means of the “adaptation modelling”. These resources are alternative media provided

---

3 Developed by means of the Protégé 3.4.4 ontology editor [http://protege.stanford.edu/] and the KAON2 reasoner [http://kaon2.semanticweb.org/], which supports the OWL ontological language [http://www.w3.org/TR/owl-features/]
by the service for each functionality. With respect to the adaptations, currently only presentation-related techniques are used in this system. These resources and adaptations are categorized into the different communication media available: audio, video, text and image. In this way, it is possible to know whether a user is able to access the information provided by each medium or conversely which is the best alternative to an inaccessible medium.

5.1.2 Generation/Adaptation Component
This component is in charge of generating and adapting the user interface for a particular service, taking into account the user’s needs and preferences. Based on the information contained in the Knowledge Base, this component generates and adapts the user interface by applying an XSL stylesheet to the UIML using an XSLT processor. It has three main modules: the Manager, the Alternative Resource Selector (ARS) and the Adapted Interface Constructor (AIC) (see Figure 4).
- The objective of the Manager is to orchestrate the processes related to this component.
- The ARS module identifies the most appropriate resources and adaptations for the user and stores this information in a document called functionalities.xml (see Figure 2). This document is automatically generated from the KB during the resource selection phase. It contains references to the most appropriate resource and the transformation type required to represent that resource in the user interface for each functionality specified in the UIML. The transformation type is related to the rules.xml file and indicates how each element will be rendered. In addition, this document contains data on the adaptations required for the user.

```
<?xml version="1.0" encoding="UTF-8"?>
<functionalities target="UCH">
   <adaptations>
      <adaptation id="enlarge_text"/>
      <adaptation id="high_contrast"/>
   </adaptations>
   <functionality id="powerMenu">
      <resourceID id="ref_text:powerMenu"/>
      <transformation id="containerText"/>
   </functionality>
</functionalities>
```

**Figure 2. An example of the Functionalities.xml file.**

- The AIC applies the XSL to the UIML document in order to generate the adapted user interface. In this process, the XSL stylesheet requires additional information from the functionalities.xml and rules.xml documents. The rules.xml file is a static file integrated in the architecture (see Figure 3). Its main goal is to relate resources to real XHTML elements. It guides the XSL on how to associate functionalities with required resources and transform them into XHTML elements to be rendered in the user interface. Each UIML element for each type of media has its corresponding rule element. In addition, appendix elements describe how to include resources in XHTML elements.

---

4 http://www.w3.org/TR/xslt
5.2 Interaction process

Figure 3. An example of the rules.xml file.

Figure 4. The architecture of the EGOKI system
The above-mentioned components and the information flow between them are illustrated in Figure 4. The process of the automatic generation of a user interface tailored to a particular user’s needs occurs as follows:

1. When a user logs into the system using his mobile device, the system queries the Knowledge Base in order to identify the user and to determine his or her needs and preferences (Figure 4: 1, 2, and 3). As a result, the List of Alternative Resources (LAR) and the List of Alternative Adaptations (LAA) are returned (Figure 4: 4a). The LAR includes the most appropriate types of resources for each media type (audio, video, text and image). For example, a LAR for a user with hearing impairments could be: Text → simple_text; Audio → transcription; Image → image; Video → video subtitles. The LAA specifies the adaptations required for the user. For instance, for a person with low vision the adaptations would be: Text → enlarge_text; Background → high_contrast.

2. Once the user selects a service (that is provided by a service), the system retrieves the corresponding UIML document for the service (Figure 4: 4b). The UIML is parsed (in ARS), and for each functionality of the service the system checks the existence of the appropriate resources for the user (resources matching the LAR, in the KB). The available resources listed in the UIML have a priority assigned to indicate which is the best way to represent each functionality. Therefore, the system will try to match the information in the LAR with the resources described in the UIML that have the highest priority. Information on adaptations in the LAA is also taken into account. The outcome of this step is an automatically generated XML document called functionalities.xml (Figure 4: 5). As previously stated, this document contains a list of all functionalities of the service along with the specific resource and adaptation to be used, as well as a reference to another XML document, rules.xml, in order to indicate how the resource should be rendered in the user interface.

3. The XSL stylesheet is applied to the UIML (Figure 4: 6b) using data from the functionalities.xml (Figure 4.6a) and rules.xml (Figure 4: 6c) files. As a result, the system (Figure 4: 7) automatically generates a fully functional user interface in XHTML (Figure 4: 9) by applying an XSL stylesheet and linking the corresponding media resources (Figure 4: 10c), CSS (Figure 4: 10b) styles and JavaScript code (Figure 4: 10a). Eventually, the user will be able to use the service from the adapted user interface.

6 Validation

The scenario for the validation of the system consists of a ubiquitous service, provided by the UCH middleware, and three users with different characteristics using their smart phones. The service, called Satellite Box5, plays the role of a TV set

5 http://myurc.org/tools/SatelliteBox/index.php
control. This service offers a number of functionalities, such as switching on/off, changing the channel and to turn the volume down/up.

The selected users are able to interact with the service through different communication modalities using their own smartphones, which is adapted to their own characteristics. The users are: a person with visual impairments, a blind person and a person with motor impairments. Based on the functionalities of the service, these users were provided with resources tailored to their communication needs. In addition, several adaptations were automatically performed for each user; for example, large font size and adequate colour contrast for the visually impaired user.

The first test was intended to verify that all the generated user interfaces offer the full functionality of the service, in order to avoid user interfaces that filter or fade out specific functions or commands. The current prototype of EGOKI demonstrated that it is able to generate fully functional and well-adapted user interfaces.

The second test verified that each user was able to perform all the possible tasks using the user interface adapted to his or her own features. The EGOKI prototype showed that the generated prototypes can be used by the modelled users.

These tests are, of course, preliminary. Further formal accessibility and usability evaluations are needed in order to validate the quality of the system and the user interfaces generated. Nevertheless, at this stage of the research the goal of the validation was to demonstrate that the EGOKI prototype is able to automatically generate interfaces that fulfill the specified requirements.

7 Conclusions and future work

The EGOKI system presented in this paper has been demonstrated to be able to automatically generate user interfaces tailored to the features of users with different types of disabilities. EGOKI has been developed for ubiquitous environments where the user interfaces are created from scratch, starting from a formal specification of the functionalities provided by the service. This procedure differs from web adaptation, which takes previously designed Web pages as a starting point. The adaptation has been enhanced to represent the information about each user as a combination of parameters related to his or her most appropriate modes of communication, with each mapped to a set of communication media.

It is a system requirement that the designer of ubiquitous services describes beforehand all the functionalities and controls, as well as to provide the necessary resources (such as icons), as the system cannot infer the semantic components of the interface. In EGOKI, this is done by means of the User Interface Description Language UIDL. In fact, one the main contributions of this work is the definition of the UIML extension that allows the formal specification of all the elements, procedures and the behaviour necessary to build instances of the user interface that are fully functional. In addition, we are currently developing an application to provide service designers with a wizard, to aid the complete formal specification of services.

Since the creation of the user interface does not involve human intervention, the accessibility of the resulting interface is guaranteed by the EGOKI system. Nevertheless, further user evaluations are required in order to comprehensively assess
the usability of the automatically generated user interfaces. In relation to usability, although the adapted user interfaces render adequately on mobile devices and are fully functional, we are working to enhance the presentation of the automatically generated interfaces in order to make them more attractive and improve the user experience.

This work has also contributed to addressing one of the open issues in UCH: how to automatically select and generate a suitable user interface using different types of resources (text, video, audio or images). Nevertheless, UCH technology presents a limitation due to the low number of devices compliant with UCH. For this reason, we are working on the extension of the EGOKI architecture in order to accommodate web services in addition to UCH targets.

Acknowledgments.
This research work has been partly funded by the Department of Education, Universities and Research of the Basque Government. We would like to thank the people involved in the INREDIS project, which has been the foundation for this work.

References
6. Universal Control Hub 1.0 (DRAFT). http://myurc.org/TR/uch/