A Comparison of User Description Languages Concerning Adaptability Based on User Preferences

Christopher MAYER [a,1], Martin MORANDELL [a], Andreas KUNTNER and
Hilda TELLIOGLU [b]

[a] *AIT Austrian Institute of Technology GmbH, Health & Environment Department, Biomedical Systems, Austria*

[b] *Vienna University of Technology, Institute of Design and Assessment of Technology, Austria*

Abstract. The user interface is the most important feature of interaction between users and (AAL) services. Explicitly defined user interfaces are bound to a specific toolkit and programming language or markup language. Thus a separate user interface definition has to be created manually for different classes of I/O devices to be supported. Compared to manual user interface creation, the automatic or semi-automatic generation of user interfaces based on interaction descriptions considerably reduces the manual effort necessary for integrating a large number of devices and therefore automatically increases the number of supported devices. The main goal of this paper is to provide an overview of selected existing solutions for the definition of generic user interactions and the generation of user interfaces. The comparison shows that the aspect of adaptability is partly covered by the presented User Interaction Description Languages. Nevertheless it is important to analyze them with respect to additional criteria, like accessibility, context- and use-case awareness, to receive a meaningful overview of advantages and drawbacks of the different approaches leading to a good basis for choosing one of the presented approaches.

Keywords. Ambient Assisted Living, Human-Computer Interaction, User Description Languages, Comparison, Adaptability, User Preferences

1. Introduction

The user interface (UI) is the most important feature of interaction between users and (AAL) services. It can be critical to the success or failure of an ICT product or service[1]. In the course of aging the abilities and special needs of older adults change. Many services older people could benefit from lack of accessibility, adoptability, and usability of the user interface and interaction.

[1] Corresponding Author: Christopher Mayer, AIT Austrian Institute of Technology GmbH, Health & Environment Department, Biomedical Systems, Donau-City-Str. 1, 1220 Vienna, Austria; E-mail: christopher.mayer@ait.ac.at.

[1] http://www.etsi.org/website/Technologies/UserInterfaces.aspx [Accessed: 04/2013]

Explicitly defined user interfaces are bound to a specific toolkit and programming or markup language. Thus a separate UI definition has to be created manually for different classes of I/O devices to be supported. Compared to manual user interface creation, the automatic or semi-automatic generation of user interfaces based on interaction descriptions considerably reduces the manual effort necessary for integrating a large number of devices and therefore automatically increases the number of supported devices. The major challenge of using such an approach of automatic or semi-automatic user interface generation is the quality of the rendered UIs. Especially when developing services and UIs for older adults who might not be experienced in the use of electronic devices and are likely limited to certain input and output modalities, it is important to provide products that ensure maximum accessibility and usability.

The main goal of this paper is to provide an overview of selected existing solutions for the definition of generic user interactions and the generation of user interfaces – subsumed as User Interaction Description Languages (UIDL). It is inspired by the theoretical survey presented in [1]. The UIDLs are examined and compared to each other, whereas the focus is on adaptability based on the environmental settings. Other important criteria, such as level of abstraction, openness, organizational background, status, number of implementations, and number of supported target platforms, are of importance as well, but neglected in the presented work. The paper is structured in the following way: The introduction and objectives are followed by a description of the methodology in terms of a comparison approach and an overview of UIDLs. Thereafter the results of the comparison are presented, followed by some final conclusions.

2. Methodology: Comparison Approach

We have established a comparison study to analyze UIDLs based on various criteria, whereas adaptability is of uttermost importance to identify their applicability to Ambient Assisted Living environments. The criterion *adaptability* concerns the possibility to adapt user interfaces automatically based on different environmental settings. For AAL environments, the following three characteristics concerning adaptability are important:

- **Accessibility:** The UI should automatically adapt to user preferences based on the user's abilities and disabilities.
- **Use-case awareness:** In different use cases, different I/O devices are used. For example, for activating and deactivating a service, a mobile phone is mostly used because the user can carry it with him or her, while changing the basic setup of a service is carried out using a PC due to the more sophisticated input mechanisms. UIDLs should know about the capabilities of UI devices and automatically adapt UIs to provide different functionalities on different I/O devices as well as to present the UI in the way that is supported best by each type of device.
- **Context-awareness:** Finally, it is desirable to automatically adapt the presentation of a user interface based on environmental influences, for example physical conditions such as the intensity of light around the I/O device.

3. Overview User Interface Description Languages

This section presents 15 existing User Interaction Description Languages. For each of these, some high-level information are given based on available literature and tools as well as a short analysis of their adaptability.

The *Alternate Abstract Interface Markup Language (AAIML)* is part of the first Universal Remote Console (URC) specification [2]. It is a markup language used for defining UI descriptions in an abstract way, based on XML. In practice, an AAIML document contains an AAIML model that describes the UI of exactly one target service that can be controlled from a remote console device. As part of the abstract user interaction description language, classes may be defined that restrict certain interactions to special devices and therefore to certain use-cases.

Presentation Template (PreT) Markup Language is part of the ISO/IEC 24752 standard [3], which is the direct successor of the ANSI/INCITS 389 through 393 standards family. It defines the latest specification of Universal Remote Console (URC). The concrete UI is rendered individually for the UI device's platform at run-time, based on the capabilities of the available services. Thus it supports in theory an unlimited number of I/O devices. It lets the user choose the device as remote console, which is an important step towards accessibility.

The *Extensible Interface Markup Language (XIML)* is a framework for defining abstract UIs and transforming them to concrete ones. In addition, it offers several methods for the automatic adaptation of UIs, based on different criteria. Its main focus is the use case of a generic UI, defined once for an application, which can be executed on a variety of devices, using different platforms [4]. User profiles and preferences can be specified. Personalization is integrated using automatic exchange of UI widgets. The mechanism of automatically generating rules for mapping intermediate presentation elements to concrete UI widgets provides a flexible way to react to contextual settings, device capabilities, use-cases, user preferences, etc.

The *Extensible Interaction Scenario Language (XISL)* was designed with regard to changing Internet usage behavior. It is a language for describing online multi-modal interaction systems, using an XML based syntax. It describes interaction scenarios rather than concrete UIs, thus being applicable to many different interaction modalities. XISL is based on existing open standards such as VoiceXML [5] and SMIL [6], but advances their concepts in a modality-independent approach. It does not provide any framework for automatic adaptation, but extensions can be customized for rendering the UI and adapting it to users' needs or environmental settings.

The *Web Services Description Language (WSDL)* specifies an XML-based language to describe web services based on their functionality [7]. It is used to define the functionality a web service offers and the interactions to access a certain functionality. Although WSDL is often used in combination with SOAP in practice, the language is protocol-independent. It does neither define nor generate UI.

The *Web Service eXperience Language (WSXL)* was developed to reduce development effort while building web applications for different distribution channels by re-use. Service-based web applications specified using WSXL can easily be adapted to different presentation scenarios. This means that one web application can be accessed in different ways without the need of redevelopment. The adaptability description framework was designed for adaptation to different distribution channels inside the usage context of web services. No built-in support for the automatic adaptation is provided, but it could easily be added by external toolkits.

The *USer Interface eXtensible Markup Language (UsiXML)* is an XML-based markup language that can be used to define UIs in an abstract way, modality independent [8]. Its main goal is to reduce the effort necessary for generating UIs to be used in multiple contexts of use. It addresses mainly designers, in contrast to traditional UI toolkits that are often integrated into programming frameworks aiming at developers. So, full adaptability of UI is supported.

User Interface Markup Language (UIML) is a meta-language, not specifying concrete UI elements on its own, but providing a framework for the definition of custom vocabularies that can then be used to create generic UI descriptions. The development of UIML was inspired by the generic concept of HTML, which provided one markup language that could be rendered on several platforms [9]. Based on this, UIML's main goal is helping UI developers in creating UIs that are sufficiently generic to be used on different platforms, thus significantly reducing the effort in developing multi-platform UIs. UIs defined using UIML are either automatically transformed to different target languages, or interpreted and rendered on target devices in real-time.

The *Dialog and Interface Specification Language (DISL)* [10] is an extended subset of the UIML language specification. It provides a modeling language for specifying dialog models in an abstract way that can be used to generate UIs for multiple modalities and platforms. The language is designed to support switching between end devices on the fly. Adaptability is a built-in core component.

Model-based lAnguage foR Interactive Applications XML (MARIA XML) [11] mainly focuses on the definition of UIs used to access web service functionalities. The language follows a semi-automatic approach for generation of UIs: Basic final UIs are generated automatically from abstract UI descriptions, but developers are given the possibility to refine these concrete interfaces. This concept allows human intervention, but reduces manual effort. The mechanism of migratory UI implements an automatic adaptation based on contexts of use.

The *eXtensible Application Markup Language (XAML)*[2] is a declarative markup language based on XML developed by Microsoft. While XAML was created as general markup language for the initialization of structured objects, its main use is the specification of UIs as part of the Windows Presentation Foundation (WPF) and the definition of UIs for web applications that build upon the Microsoft Silverlight framework. It does not provide mechanisms to store users' preferences and to react to environmental factors.

The *XML User interface Language (XUL)* is based on XML and not compiled but interpreted at run-time by a special rendering engine. Its main use is the definition of UIs for applications developed by the Mozilla community, although also some additional projects make use of it. Adaptation could be integrated using the flexible CSS styling system. XUL UI automatically adapt to different hardware capabilities such as screen resolution, but limited to simple dynamic resizing and positioning of widgets.

The *Macromedia eXtensible Markup Language (MXML)* is a declarative markup language based on XML originally developed by Macromedia, now used by Adobe as part of the Flash product series. The main use of MXML is the specification of UIs of Adobe Flex applications. The compiled applications (SWF) can be run as web applets

[2] http://msdn.microsoft.com/en-us/library/ms747122.aspx [Accessed: 04/2013]

embedded in any web page, or as standalone applications. Through the use of layout managers, MXML UI automatically adapt to different screen resolutions by re-positioning UI elements. The display of different UI on different devices however is not supported.

VoiceXML is an XML-based markup language used to specify user interaction with speech-based systems. It was designed with a similar goal as HTML: HTML is used to specify visual web pages, while VoiceXML allows the specification of audible web content, featuring both one-way presentations and interactions. No support for adaptability is provided since the language focuses on speech-based interfaces that are presented in the course of one telephone call using exactly one device.

The *HyperText Markup Language (HTML)* is the markup language used for the specification of web pages. Version 4 [12] is the first version that separates semantics and appearance. The analysis also applies to the *eXtensible Hypertext Markup Language (XHTML)* [13]. HTML is based on the Standard Generalized Markup Language (SGML), while XHTML is based on XML. XML actually is a subset of SGML that adds additional restrictions to the base language [14]. No automatic adaptation support is provided, whereas CSS supports multiple media types by enabling styling the control elements differently for different devices or hiding of some elements for certain devices.

	Accessibility	Context-awareness	Use-case awareness
AAIML	unknown	unknown	yes
PreT	medium	no	no
XIML	yes	yes	yes
XISL	no	no	no
WSDL	N/A	N/A	N/A
WSXL	no	no	no
UsiXML	yes	yes	yes
UIML	prepared	prepared	prepared
DISL	no	yes	yes
MARIA XML	prepared	through migrational UI approach	prepared
XAML	no	no	N/A
XUL	no	partly	no
MXML	no	no	no
VoiceXML	no	no	N/A
HTML/XHTML	through browser	no	through CSS

Table 1. Comparison of all analyzed User Interaction Description Languages regarding adaptability.

4. Results

In Table 1 the results of the comparison regarding adaptability of all analyzed UIDLs are presented. As shown the characteristics are in a wide range and just some UIDLs are covering these important aspects (XIML, UsiXML). DISL lacks on accessibility support, but provided context- and use case-awareness. The solutions studied differ strongly concerning their field of application. To form a complete user interface generation system for AAL environments, several of these solutions need to be combined since all operate on a different degree of abstraction.

5. Conclusion

The comparison and the results presented in Section 4 show that the aspect of adaptability is partly covered by the UIDLs. Nevertheless it is important to analyze them with respect to additional criteria, like accessibility, context- and use-case awareness, to receive a meaningful overview of advantages and drawbacks of the different approaches. These characteristics are chosen due to their importance and relevance in the design of user interfaces and interaction for users with special needs to access (AAL) services. The project AALuis[3] uses for example Maria XML.

References

[1] Guerrero-Garcia, J., Gonzalez-Calleros, J. M., Vanderdonckt, J., Muoz-Arteaga, J. (2009). A theoretical survey of user interface description languages: Preliminary results. In Web Congress, 2009. LA-WEB'09. Latin American (pp.36–43).

[2] Zimmermann, G., Vanderheiden, G., Gilman, A. (2003). Universal remote console – prototyping for the alternate interface access standard. In: Proceedings of the User interfaces for all 7th international conference on Universal access: theoretical perspectives, practice, and experience, ERCIM'02. Springer-Verlag, (pp. 524–531).

[3] Information technology - user interfaces - universal remote console. International Organization for Standardization, ISO/IEC 24752, 2008.

[4] Puerta, A., Eisenstein, J. (2001). XIML: A universal language for user interfaces. Technical report, RedWhale Software.

[5] Oshry, M., Auburn, R.J., Baggia, P., Bodell, M., Burke, D., Burnett, D. C., Candell, E., Carter, J., McGlashan, S., Lee, A., Porter, B., Rehor, K. (2007). Voice extensible markup language (VoiceXML) version 2.1. W3C recommendation, W3C.

[6] Bulterman, D. (2008). Synchronized multimedia integration language (SMIL 3.0). W3C recommendation, W3C.

[7] Chinnici, R., Moreau, J.-J., Ryman, A., Weerawarana, S (2007). Web services description language (WSDL) version 2.0 part 1: Core language. W3C recommendation, W3C.

[8] Limbourg, Q., Vanderdonckt, J., Michotte, B., Bouillon, L., López-Jaquero, V. (2004). USIXML: A language supporting multi-path development of user interfaces. In Proceedings of the 9th IFIP Working Conference on Engineering for Human-Computer Interaction jointly with the 11th Int. Workshop on Design, Specification, and Verification of Interactive Systems, volume 3425 of EHCI-DSVIS'2004, Springer Verlag (pp.200–220).

[9] The relationship of the UIML 3.0 spec. to other standards/working groups. Technical report, Organization for the Advancement of Structured Information Standards (OASIS), 2003.

[10] Schaefer, R., Bleul, S., Mueller, W. (2007). Dialog modeling for multiple devices and multiple interaction modalities. In Proceedings of the 5th international conference on task models and diagrams for users interface design, TAMODIA'06, Springer Verlag (pp. 39–53).

[11] Paternó, F., Santoro, C., Spano, L.D. (200)). MARIA: A universal, declarative, multiple abstraction-level language for service-oriented applications in ubiquitous environments. ACM Transactions on Computer-Human Interaction, Vol. 16, No. 4, Article 19.

[12] Raggett, D., Le Hors, A., Jacobs, I (1999). HTML 4.01 specification. W3C recommendation, W3C.

[13] Pemberton, S. (2002). XHTML$^{\text{TM}}$1.0 the extensible hypertext markup language (second edition). W3C recommendation, W3C.

[14] Clark, J. (1997). Comparison of SGML and XML. W3C note, W3C.

[3] www.aaluis.eu. The project AALuis is co-funded by the AAL Joint Programme (AAL-2010-3-070) and the following National Authorities and R&D programs in Austria, Germany and The Netherlands: bmvit, program benefit, FFG (AT), BMBF (DE) and ZonMw (NL).