Review of User Interface Description Languages

Jürgen Engel, Christian Herdin, Christian Märtin

Augsburg University of Applied Sciences, Faculty of Computer Science Augsburg, Germany

{Juergen.Engel, Christian.Herdin, Christian.Maertin}@hs-augsburg.de

Abstract

In this paper we compare existing User Interface Description Languages (UIDL) in terms of their suitability, adequacy, practicability, and availability with regard to support model-driven as well as pattern-based software development. The results will be used to extend the potential of our PaMGIS framework for Pattern-based Modeling and Generation of Interactive Systems.

1 Introduction

In the scope of our research within the Automation in Usability Engineering group (AUE) at Augsburg University of Applied Sciences we develop an integrated approach for the design and semi-automated generation of user interfaces (UI) of interactive software applications. It combines both, model-based and pattern-based development techniques and methods. In this context we have reviewed existing User Interface Description Languages (UIDLs) in terms of their suitability, adequacy, practicability, and availability with regard to support model-driven as well as pattern-based software development. The results will be used to extend the potential of our framework for Pattern-based Modeling and Generation of Interactive Systems (PaMGIS) [8].

The further document is organized as follows: the review approach is described in section 2, brief descriptions of the considered UIDLs are provided in section 3, the review results are summarized in tabular format in section 4, and our lessons learnt and decisions regarding PaMGIS are depicted in section 5. Finally, a list of literature being consulted during the review process is provided.

2 Review Approach

Subject of the literature review have been existing XML-compliant UIDLs, including User Interface Markup Language (UIML), User Interface Extensible Markup Language (UsiXML), Dialog Modeling Language (DiaMODL), Interface Specification Meta-Language (ISML), Transformation Environment for Interactive Systems Representations (TERESA XML), Model-based Language for Interactive Applications (MARIA), Extensible Interface Markup Language (XIML), and XML User Interface Language (XUL). Initially the Unified Modeling Language for Interactive Applications (UMLi) and the Extensible Application Markup Language (XAML) were also planned to be reviewed, but were excluded during the review process (please refer to section 3.9).

Documentation of UIDL reviews and evaluations is already available [e.g. 30, 37]. But partially it is fairly old [30], lacks of evidence regarding the review results [30], or does not cover UIDLs of our interest [30, 37]. Nevertheless, these documents deliver valuable input notably for defining the UIDL characteristics to be investigated.

For each of the above mentioned UIDLs we captured (1) the name of the UIDL, (2) its originator, (3) date of first publication, (4) actuality respectively current version, (5) known tools available to support the usage of the UIDL, (6) availability in terms of whether the UIDL can freely be used or not, (7) the types of inherent models, i.e. the kinds of UI aspects that can be modeled, (8) the number of specified XML tags, (9) whether there is one generic UI specification valid for all different contexts of use or separate UI descriptions must be created for each context of use (analogous to Methodology in [30]), (10) the major concepts of the UIDL, (11) whether the UIDL was mainly designed for supporting multi-platform, multi-user, or multi-environment developments (analogous to Target in [30]), (12) the supported target programming languages, (13) the supported target computing platforms, and (14) the abstraction level of the UIDL in accordance with the CAMELEON Reference Framework (CRF), i.e. Model level, Abstract User Interface (AUI) level, Concrete User Interface (CUI) level, and Final User Interface (FUI) level [4] plus Meta-Model level.

Due to space limits it was not possible to present all details of the evaluated characteristics within this document. Therefore, we decided to provide as much information as possible within the textual UIDL descriptions (see section 3) and summarized in tabular form (see section 4).

3 Descriptions of Considered UIDL

3.1 UIML

The User Interface Markup Language (UIML) is a joint development of the Virginia Polytechnic Institute and State University and Harmonia Inc. [1]. The initial language design has been accomplished in 1997. The first version, i.e. UIML 1.0, has been released in 1998. The current version UIML 4.0 is available as Committee Draft of the Organization for the Advancement of Structured Information Standards (OASIS) and dates from 2008 [36].

At the highest level, a UIML document comprises of four major elements, i.e. *Head*, *Interface*, *Peers*, and *Template* [1]. The Head element contains metadata about the current UIML document and is neither considered as part of the user interface nor does it have effect on the rendering or operation of the UI. The Peers element facilitates extensibility and defines mappings from class, property, event, and call names used in a UIML document to entities specified outside of the document. The Template element features reuse by specifying UIML code that can be employed in other UIML sections [36]. Finally, the Interface element holds all information in terms of the representation of the user interface. It consists of four components, i.e. *Structure*, *Style*, *Content*, and *Behavior*. Within the Structure element the physical organization of the UI is specified. This also includes the relationships between the various UI elements. The Style element contains a list of properties and values which are used to render the user interface, e.g. background color, font type, and font size. The Content element represents the actual content associated with the various parts of the UI and facilitates a clear separation of the content and the structure of the user interface. Further, the Behavior element specifies the behavior of the UI. This is achieved by rules consisting of conditions and associated actions. UIML allows for four different types of action, i.e. (1) assign a value to a property of a part, (2) call an external function or method, (3) fire an event, and (4) restructuring the user interface [1].

UIML is designed to support the development of multi-platform user interfaces. However, the Structure element must be specified for each device or platform separately [1]. This means that the user interfaces for different platforms can be specified by one single language, but the actual design has still to be done repeatedly. Regarding the degree of abstraction UIML covers the Model level.

3.2 UsiXML

The User Interface Extensible Markup Language (UsiXML) has been developed and published by the Information Systems Research Unit (ISYS) of the Université Catholique de Louvain (UCL) [12] in the year 2003 [37]. The language stems from the CAMELEON FP5 Project [41] and therefore complies with the four levels of abstraction of the CAMELEON Reference Framework, i.e. *Tasks & Concepts*, *Abstract User Interface*, *Concrete UI*, and *Final UI* [12]. The prevailing edition is version 1.8 which dates from 2007 [39].

UsiXML comprehends multiple models involved in the design of user interfaces, including task, domain, presentation, dialog, and context of use models. The latter is decomposed into user, platform, and environment models [12]. Interrelationships between these models are consolidated and documented within mapping models. Moreover, UsiXML supports transformation models. Transformations are specified by means of transformation systems which, in turn, are based on the theory of graph grammars [13].

UsiXML allows designers to specify a user interface on multiple abstractions, i.e. Model, CUI, AUI, and FUI [13] and supports device-independent, platform-independent, modality-independent, and context-independent definition user interfaces [12]. There is extensive tool-support for UsiXML and a variety of target languages have been addressed (please refer to table 2). Furthermore, diverse platforms are supported, including mobile, pocket PC, interactive kiosk, wall screen, and PDA [37].

3.3 DiaMODL

The Dialog Modeling Language (DiaMODL) has been developed and introduced by the Norwegian University of Science and Technology (NTNU) [34]. The first publication dates from 2003 [33]. DiaMODL is a hybrid language combining *Pisa Interactor Abstraction* and *UML State Charts*.

An interactor plays the role of an information broker between a component and the interactive system by sending and receiving information through a set of *Gates* each equipped with a *Tip* and a *Base*. In principle, there are four different type of gates, i.e. (1) *Input/Send*: user input results in information sent out to the interactive system, (2) *Output/Receive*: system output is received and information is sent to the user, (3) *Input/Receive*: user input is received by the interactor for further processing or mediation, and (4) *Output/Send*: Output to the user is sent out by the interactor. A value received at the base of a gate is computed by means of a function and subsequently sent out via the tip. It is possible to build entire networks of interactors by connecting the gates. Like the gates, the connections can also be tied to functions which are defined in the domain modeling language, i.e. UML. Interactors are mainly used to describe the functionality of *Concrete Interaction Objects*.

The dynamic aspects of the user interface, such as triggering the information flow and activation respectively deactivation of interactors, are modeled by means of state charts. For this purpose the meta-model of UML has been extended.

DiaMODL is intended to support design and development of user interfaces for different interaction styles and platforms. In terms of the degree of abstraction it covers the Model level.

3.4 ISML

The Interface Specification Meta-Language (ISML) has been defined by the Bournemouth University (UK) in 2003. ISML is part of a model-based user interface framework making use of metaphor models. Metaphors are regarded as shared concepts between the user and the computer. The goal of ISML is to make metaphors explicit in design but it strictly decouples the metaphor models from any particular implementation. Basically, the framework combines task models and interactor definitions and envisages metaphor-based mappings between them [6].

The ISML framework is composed of five parts: Devices, Components, Meta-objects, Interactors, and Tasks. At this juncture, devices are simple abstractions of user input/output hardware and components are logical abstractions of user input and output objects. Objects feature attributes, states, constraints, and communication mechanisms. The meta-object layer is the foundation of the specification of the metaphor abstraction layer, its implementation (i.e. the interactors), and the task model [5]. Interactor definitions use meta-objects as a basis for a specific design solution using a mapping of components to interactor 'display parts'. The task layer combines meta-object definitions of objects and actions with a simple, hierarchical decomposition of tasks [6].

ISML targets to support the design and development of multi-platform user interfaces. In terms of the degree of abstraction it covers the Model level.

3.5 TERESA XML

TERESA XML is integral part of the Transformation Environment for Interactive Systems Representations (TERESA) [2] which has been developed and introduced by the Human-Computer Interaction (HCI) group of Institute of Information Science and Technologies (ISTI), which is an institute of the national Research Council of Italy (CNR) [31]. It targets to support the design and development of multi-device user interfaces. The initial development of TERESA dates from 2003 [14].

TERESA involves a method of model-based design starting with the preparation of a high-level task model and a domain model which covers all interaction objects required to perform the specified tasks. Subsequently, system task models are to be elaborated which can be regarded as platform-specific refinements and adaptations of the original task model. From a system task model an abstract description of the UI is obtained as a set of presentations and connections between them. Such an AUI is the basis for the generation of the final user interface of the platform of interest [3].

TERESA XML consists of two parts. On one hand the task model is represented in ConcurTaskTrees (CTT) notation and stored accordingly in an XML-compliant format. On the other hand it covers both, the abstract and concrete user interface descriptions. Here, the AUI incorporates the static structure of the UI (i.e. the presentations) as well as the dynamic behavior (i.e. the connections between presentations) [3, 17]. The connections can also be interpreted as a kind of dialog model.

In terms of the degree of abstraction TERESA XML covers the Model level as well as abstract and concrete UI level.

3.6 MARIA

Since the Model-based Language for Interactive Applications (MARIA) is an evolution of TERESA it has been developed by the HCI group of ISTI-CNR [14]. The first Paper dates from 2009.

On the basis of experiences with TERESA and state-of-the-art analyses additional requirements for a modern UIDL have been identified: (1) higher level of control regarding the produced UI for designers through an event model, (2) more flexible dialog and navigation model supporting parallel interactions, (3) flexible data model which allows to model the association of various types of data to the various interactors, (4) supporting recent dynamic technologies, e.g. AJAX scripts, and (5) streamlining AUI and CUI specifications in order to reduce their volume and improve the readability [26].

In MARIA, the data model is described using the XML Schema Definition (XSD) language. Regarding the event model two types of events have been defined: *Property Change Events* lead to a status change of certain user interface properties and *Activation Events* which allow for activating individual application functions through interactors. In order to achieve continuous updating of UI fields, interactors are equipped with a new Boolean attribute named *continuously-updated* which can be utilized, for instance, to employ AJAX asynchronous mechanisms. Moreover, MARIA supports to dynamically change parts of the user interface. This applies to the way how UI elements are arranged inside a presentation as well as to the navigation between presentations [26].

On the AUI level a user interface is composed of one data model and one or more presentations. The presentation, in turn, consists of a data model and a dialog model which contain information about the events that can be triggered by the presentation. The dynamic behavior of the events is specified by means of the temporal operators within the CTT task model. Finally, the concrete UI description is defined in a platform-dependent manner, but is still independent from the target programming language [26].

Considering the degree of abstraction MARIA covers the Model, abstract UI, and concrete UI level [3].

3.7 XIML

The Extensible Interface Markup Language (XIML) has been developed by a company named RedWhale Software [28]. XIML 1.0 has been published in 1999 [42]. A basic principle is the strict separation of the user interface definition and its actual appearance on a target platform [27]. XIML is designed to describe abstract UI aspects, such as the user interaction context, as well as concrete facets like particular widgets to be displayed on the screen [28].

XIML can be regarded as an organized collection of user interface *Elements* which are in turn categorized in UI *Components*. Theoretically, the number and types of components is not limited, but XIML 1.0 provides five predefined basic interface components, i.e. *task*, *domain*, *user*, *dialog*, and *presentation* components. The task component determines the business process and user tasks supported by the UI while the domain component specifies classes and data objects which will be displayed to or manipulated by the user. The user component is intended to hold characteristics of individual users or user groups. The presentation component defines a hierarchy of interaction elements comprising the concrete objects of the UI while the dialog component determines possible interactions and the navigation [28].

Moreover, XIML facilitates *Attributes* that represent features or properties of elements to which *Values* can be assigned. Finally, it is possible to specify *Relations* which are *Definitions* or *Statements* that link any two or more XIML elements together. With regard to the degree of abstraction XIML covers the Model [28] as well as the abstract and concrete UI levels [27].

3.8 XUL

In the year 1998, the Netscape Communications Corporation published the source code of its Internet browser and founded the open source project *Mozilla* for further development. In the context of this change the XML User Interface Language (XUL) has been introduced in order to facilitate the development of the user interfaces of the Mozilla products [9].

With XUL it is possible to describe content with a particular behavior, a presentation of this content, and moreover, the user interface can be localized. When cross-platform web applications are developed with XUL further technologies can be applied, such as *Cascading Style Sheets* (CSS), *Extensible Binding Language* (XBL) which is a means to modify, replace or add XUL tags, *Overlays* which are sets of XUL files used for extra content and UI adaptions, *Cross Platform Component Object Model* (XPCom) / XPConnect for integrating new libraries, and *XPInstall* which is used to install XUL applications from the Internet or intranet [43].

While native XUL elements serve to create the layout of a user interface, the behavior is accomplished by the usage of script languages, usually JavaScript [9]. A basic concept of XUL layouts is the *Box* model: every XUL element is initially regarded as a rectangular box which might in turn contain further boxes. It is differentiated between horizontal boxes (*hbox*) in which the child boxes are arranged horizontally, and vertical boxes (*vbox*) with a vertical layout of child boxes. Finally, a user interface is represented by a set of structured UI elements, such as *Window*, *Menubar*, *Button*, *Checkbox* and the like [21].

Basically, XUL runs on all operating system platforms on which the Mozilla rendering engine *Gecko* is available, including BSD, Linux, OS X, Solaris, OS/2, AIX, OpenVMS, and Windows [20]. The degree of abstraction of XUL covers the concrete UI level.

3.9 Further Contributions

Initially we intended to include two further contributions to our review. On one hand this was the Unified Modeling Language (UML) respectively the related extension named Unified Modeling Language for Interactive Applications (UMLi). UMLi introduces a diagram notation for modeling user interface presentations. Furthermore, the UML Activity Diagram notation is extended in order to describe the collaboration between interaction and domain objects [29]. Owing to the semi-formal nature of UML and due to lack of space we decided to exclude UML(i) from the present document. On the other hand we omitted the Extensible Application Markup Language (XAML) [16] because it is more or less restricted to the Microsoft Windows platform.

4 Summary of Review Results

General UIDL characteristics are summarized in table 1.

Table 1: Review Results: General UIDL Characteristics

Language	Originator	Since	Latest Version	Tool Support	Availability
UIML	Harmonia Inc., Computer Science Department of Virginia Tech [1]	1998 (UIML 1.0) [36] start of deveteopment 1997 [1]	4.0 Committee Draft (OASIS) (23.2.2008) [36]	4 tools[1]	open specification without license agreement [36]
UsiXML	CAMELEON FP5 project [41]	2003 [37]	UsiXML V1.8 (2007) [39]	13 tools[2]	free to use [41]
Dia-MODL	Norwegian University of Science and Technology [34]	first publication 2003 [33]	not specified	Eclipse[3]	uses open source tools, UML, MOF/MXI [33]

→

1 Transformation-based Integrated Development Environment (TIDE), LiquidUI product suite (Harmonia) [1], VoiceXML Renderer, WML Renderer [37]

2 UsiGesture v1.0 (Editor), FlashiXML [38], SketchiXML, IdealXML, VisiXML, KnowiXML, TransformiXML [13], FlowiXML, QTKiXML, InterpiXML, Attributed Graph Grammars (AGG) tool [11], GrafiXML, ReversiXML [11, 13]

3 Eclipse-based editors, views and runtime [7]

Language	Originator	Since	Latest Version	Tool Support	Availability
ISML	Bournemouth University [6]	proposal 2000 [10] Publication 2003 [5]	initial version	ISML Framework [6]	not specified
TERESA XML	CNR-ISTI, HIIS Laboratory, Pisa [31]	2003 [14]	v 3.4 (2009) [31] predecessor version of MARIA [14]	TERESA [18, 3] CTTE [18, 3]	free download of TERESA (incl. TERESA XML) [31]
MARIA	CNR-ISTI, HIIS Laboratory, Pisa [15]	evolution of TERESA XML [14] 2009 [26]	MARIA v 1.4.12 (2011) [14] MARIAE v 1.5.4 (2013) [15]	CTTE [24] MARIAE v 1.5.4 [15]	free download of MARIAE (incl. MARIA) [15]
XIML	RedWhale Software [27, 28]	1999 [42] XIML 1.0 [28]	1.0 [28]	2 tools[4]	free XIML Research License Agreement [42]
XUL	open source project Mozilla [9]	1998 [9]	in context of Firefox 17 (2012) [19]	6 tools[5]	requires open source rendering engine Gecko

Further UIDL details are provided in table 2.

Table 2: Review Results: UIDL Details

Language	Models	Number of Tags	Target	Supported Languages	Supported Platforms	Abstr. Level
UIML	Presentation, Dialog, Domain [36]	44 [36]	multi-platform [36, 1]	Java, HTML, WML, VoiceXML [1] C++, CORBA, QT [37]	handheld, desktop, mobile phone, TV [37]	Model

→

4 MOBI-D Model-to-XIML Converter, HTML-to-XIML Converter [28]

5 Gecko, XPCom, XPConnect, XPInstall, XULRunner, XUL Explorer [43]

Language	Models	Number of Tags	Target	Supported Languages	Supported Platforms	Abstr. Level
UsiXML	Task, Domain, Presentation, Dialog, Context of Use (User, Platform, Environment) [12], Mapping, Translation[13], UI [40]	118 tags [37] vs. 345 tags [39]	multi-platform [37]	HTML, XHTML, VoiceXML, Java3D, VRML, X3D, XAML, Java, Flash, QTK, WML, X+V, C++ [37]	mobile, pocketPC, interactive kiosk, wall screen, PDA [37]	Model, CUI, AUI, FUI [13]
DiaMODL	Domain [33] Dialog [34, 33]	not specified	multi-platform [33]	Java Swing [34] Java, CORBA [33]	not specified	Model
ISML	Task, Metaphor [6], Presentation, Domain, Dialog [37], Device [5]	not specified	multi-platform [37]	Java [37]	desktop PC, 3D screen [37]	Model
TERESA XML	Task [17] [3], Domain, Dialog, Presentation, Device [3]	AUI: 38 CUI platform-dependent, e.g. desktop (86), DTV (67), voice (66), mobile (80) [32]	multi-platform [17, 3]	XHTML [3] C# [23]	desktop PC, mobile phone,voiceXML [17, 3], Digital TV, X+V [23]	Model, AUI, CUI [3]
MARIA	Task, Domain [14], Presentation, Event [26], Dialog [24]	not specified	multi-platform [25]	HTML 4/5, JSP, VoiceXML, X+V, SMIL [14]	destop PC, mobile [14], vocal [25]	Model, CUI [14, 24] AUI [24]
XIML	Task, Domain, User, Presentation, Dialog [27] [28]	32 [37]	multi-platform [27, 28] multi-user [28]	C++, HTML, WML [28] Java Swing [37]	desktop PC, PDA, cell phone [27]	Model [28] AUI, CUI [27]
XUL	Presentation, Dialog [30]	125 [21]	multi-platform [22]	XUL [30]	BSD, Linux, OS X, Solaris, OS/2, AIX, OpenVMS, Windows [20]	CUI

5 Conclusion

All UIDLs considered in our detailed literature review can be regarded as valuable contributions to the design and development of multi-platform user interfaces.

Starting with version 1.0 in 1998 UIML has been developed further to version 4.0 which is available as OASIS Committee Draft since 2008. Furthermore, extensive tool support is described within the literature but nowadays none of the mentioned tools can be retrieved from the Internet any longer. Therefore, we assume that any activities have been stopped.

Contrariwise, UsiXML which started in 2003 was subject of an EU-funded project[6] which ended in March 2013. However, the latest language specification available to us dates from 2007. UsiXML is designed according to the CRF and incorporates a sophisticated model conception. Furthermore, it is comprehensively supported by tools.

The hybrid modeling approach DiaMODL entails an extension of the UML meta-model. It has also been created in 2003; the latest available documentation dates from 2008 [35]. Obviously no further development of the UIDL takes place.

ISML is an interesting approach supporting metaphors for UI design. It has been developed in the context of a PhD thesis in 2003. More recent documentation could not be found.

Like UsiXML, TERESA XML and its successor MARIA are designed according to the CRF and have a long history starting in 2003. The current version of the MARIAE framework (version 1.5.4) has been released in August, 2013. It covers the Model, AUI, and CUI abstraction levels.

XIML is the only UIDL being reviewed that not only supports the design of multi-platform, but also multi-user interfaces. It covers Model, AUI, and CUI abstraction levels. Furthermore, the supported target languages are similar to PaMGIS. XIML started in 1999; no indications of later updates could be found.

Finally, XUL left the impression that it is easy and effective to use. It supports the CUI abstraction level and is available on every platform on which the rendering engine *Gecko* is running. It dates from 1998 and the last update took place in 2012.

With regard to the further development of our PaMGIS framework, we intend to inspect UsiXML, MARIA, and XIML in more detail and also to

6 Refer to http://usixml.eu.

gather practical experience with them. Our decision is based on the fact, that, like PaMGIS, the former two UIDL are designed in accordance with the CRF and the latter also supports the Model, CUI, and AUI abstraction levels. Furthermore, there are ongoing development activities at least regarding UsiXML and MARIA. In addition, the model conceptions of UsiXML and XIML are very close to the PaMGIS approach while the structure and model processing resembles MARIA. Currently we are not planning to adopt and replace the entire UIDL of PaMGIS by any other language, but exploit certain aspects and ideas thereof. Finally, we will integrate XUL as future target language of PaMGIS.

References

[1] Ali, M.; Pérez-Quiñones, M.; Abrams, M.; Shell, E. (2002): Building Multi-Platform User Interfaces with UIML. In: Kolski, C., Vanderdonckt, J. (eds.): *Computer-Aided Design of User Interfaces III* (15–17 May 2002, Valenciennes, France). Dordrecht u.a.: Springer Netherlands, pp. 255–266.

[2] Berti, S. et al. (2005): TERESA: An Environment for Designing Multi-Device Interactive Services. http://giove.isti.cnr.it/attachments/publications/2005-A2-80.pdf.

[3] Berti, S. et al. (2004): The TERESA XML Language for the Description of Interactive Systems at Multiple Abstraction Levels. In: *Proceedings Workshop on Developing User Interfaces with XML: Advances on User Interface Description Languages*. Online: http://giove.cnuce.cnr.it/projects/cameleon/pdf/workshop-avi.pdf.

[4] Calvary, G. et al. (2002): The CAMELEON Reference Framework. http://giove.isti.cnr.it/projects/cameleon/pdf/CAMELEON%20D1.1RefFramework.pdf.

[5] Crowle, S. (2003): *The Design and evaluation of the specification framework for user interface design* (PhD Thesis, Bournemouth University).

[6] Crowle, S.; Hole, L. (2003): ISML: An Interface Specification Meta-Language. In: Jorge, Joaquim A. (ed.): *Interactive Systems : Design, Specification, and Verification (DSV 2003)* (10th international workshop; revised papers; Funchal, Portugal, June 11–13, 2003). Berlin u.a.: Springer, pp. 362–376.

[7] DiaMODL project: http://sourcefourge.net/projects/diamodl/ <2013-09-09>

[8] Engel, J.; Märtin, C. (2009): PaMGIS: A Framework for Pattern-based Modeling and Generation of Interactive Systems, in: Jacko, J. A. (ed.): *Proceedings of HCI International '09* (San Diego, USA). Berlin u.a.: Springer, S. 826–835.

[9] Düsterloh, K. (2005): XUL-Tutorial I: Basiselemente der Browsererweiterung: Schachtelsatz. In: *iX – Magazin für professionelle Informationstechnik*

No. 7/2005. Online: http://www.heise.de/ix/artikel/Schachtelsatz-506516.html <2013-09-09>.

[10] Crowle, S. (2005): hompage: http://web.archive.org/web/20050119132150/dec.bournemouth.ac.uk/staff/scrowle/ISML/Presentations/index.html; last website call via web.archive.org (25.5.2005) on Sept. 9, 2013.

[11] Limbourg, Q. et al. (2004): UsiXML: A Language supporting Multi-Path Development of user Interfaces. In: Bastide, R.; Palanque, P.; Roth, J. (eds.): *Proceedings of Engineering Human Computer Interaction and Interactive Systems, Joint Working Conferences EHCI-DSVIS 2004* (Hamburg, Germany, July 11–13, 2004). Berlin/Heidelberg/New York: Springer, pp. 200–220.

[12] Limbourg, Q. et al. (2004): UsiXML: A User Interface Description Language for Context-Sensitive User Interfaces. In: Costabile, M. F. (ed.): *Proceedings of the ACM AVI'04 Workshop Developing User Interfaces* (Gallipoli, Italy, May 25–28, 2004). New York, NY: ACM, pp. 55–62.

[13] Limbourg, Q.; Vanderdonckt, J. (2004): UsiXML: A User Interface Description Language Supporting Multiple Levels of Independence. In: Matera, M.; Comai, S. (eds.): *Proceedings of Workshops in Connection wtih 4th International Conference on Web Engineering (ICWE 2004)* (Munich, Germany, July 28–30, 2004). Princeton, NJ: Rinton Press, pp. 325–338.

[14] MARIA fact sheet: http://giove.isti.cnr.it/tools/MARIA/MARIA%20Fact%20Sheet.pdf <2013-09-09>.

[15] MARIAE homepage: http://giove.isti.cnr.it/tools/MARIAE/home <2013-09-09>.

[16] Microsoft Developer Network: http://msdn.microsoft.com/de-de/library/vstudio/ms788723.aspx <2013-09-09>.

[17] Mori, G., Paternò, F., Santoro, C. (2004): Design and Development of Multidevice User Interfaces through Multiple Logical Descriptions. In: *Journal IEEE Transactions on Software Engineering* 30 (8), 507–520.

[18] Mori, G., Paternò, F., Santoro, C. (2003): Tool support for designing nomadic applications. In: Johnson, W. L. (ed.): *IUI '03 – Proceedings of the 8th International Conference on Intelligent User Interfaces* (Miami, USA, January 12–15, 2003). New York, NY: ACM, pp. 141–148.

[19] Mozilla Developer Network: Firefox: https://developer.mozilla.org/en-US/docs/Mozilla/Firefox/Releases <2013-09-09>.

[20] Mozilla Developer Network: Gecko: https://developer.mozilla.org/en-US/docs/Mozilla/Gecko <2013-09-09>.

[21] Mozilla Developer Network: XUL Reference: https://developer.mozilla.org/de/docs/XUL_Reference <2013-09-09>.

[22] Mozilla Developer Network: XUL Tutorial: https://developer.mozilla.org/en-US/docs/XUL/Tutorial <2013-09-09>.

[23] Paternò, F. et al. (2008): Authoring Pervasive Multimodal user Interfaces. In: *International Journal of Web Engineering and Technology* 4 (2), 235–261.

[24] Paternò, F.; Santoro, C.; Spano L. D. (2011): A Design Space for User Interface Composition. In: Hussmann, H.; Meixner, G.; Zuehlke, D. (eds.): *Model-Driven Development of Advanced User Interfaces.* Berlin/Heidelberg: Springer, pp. 43–65.

[25] Paternò, F.; Sisti, C. (2010): Deriving Vocal Interfaces from Logical Descriptions in Multi-device Authoring Environments. In: Benatallah, B. et al. (eds.): *Proceedings of the 10th International Conference on Web Engineering (ICWE'10)* (Vienna, Austria, July 5–9, 2010). Berlin/Heidelberg: Springer, pp. 204–217.

[26] Paternò, F.; Santoro, C.; Spano L. D. (2009): Model-based Design of Multi-device Interactive Applications Based on Web Services. In: Gross, T. et al. (eds.): *Proceedings of the 12th IFIP TC 13 International Conference on Human-Computer Interaction – INTERACT 2009: Part I* (Uppsala, Sweden, August 24–28, 2009). Berlin/Heidelberg/New York: Springer, pp. 892–905.

[27] Puerta, A.; Eisenstein, J. (2002): XIML: A Common Representation for Interaction Data. In: Gil, Y.; Leake, D. B. (eds.): *Proceedings of the 7th International Conference on Intelligent User Interfaces (IUI '02)* (San Francisco, USA, January 13–16, 2002). New York, NY: ACM, pp. 214–215. Online: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.101.5152&rep=rep1&type=pdf <2013-12-10>.

[28] Puerta, A.; Eisenstein, J. (2001): XIML: A Universal Language for User Interfaces. http://www.ximl.org/documents/XimlWhitePaper.pdf <2013-09-09>.

[29] da Silva, P. P.; Paton, N. (2000): UMLi: The Unified Modeling Language for Interactive Applications. In: Evans, A.; Kent, S. J. H. (eds.): *UML 2000 – The Unified Modeling Language: Advancing the Standard (3rd International Conference on the Unified Modeling Language)* (York, UK, October 2–6, 2000). Berlin u.a.: Springer, pp. 117–132.

[30] Souchon, N.; Vanderdonckt, J. (2003): A Review of XML-compliant User Interface Description Languages. In: Goos, G. et al. (eds.): *Interactive Systems. Design, Specification, and Verification*. Berlin/Heidelberg: Springer, pp. 377–391.

[31] TERESA homepage: http://giove.isti.cnr.it/teresa.html <2013-09-09>.

[32] TERESA tool; downloadable from http://giove.isti.cnr.it/teresa.html <2013-09-09>.

[33] Trætteberg, H. (2003): Dialog Modelling with Interactors and UML Statecharts – A Hybrid Approach. In: Goos, G. et al. (eds.): *Interactive Systems.*

Design, Specification, and Verification. Berlin/Heidelberg: Springer, pp. 346–361. Online: http://www.idi.ntnu.no/emner/tdt12/dsvis-2003.pdf <2013-12-09>.

[34] Trætteberg, H. (2004): Integrating dialog modelling and application development. In: *Proceedings of the First International Workshop on Making model-based UI design practical: usable and open methods and tools* (Funchal, Portugal, January 13, 2004). Paper no. 11. Online: http://ceur-ws.org/Vol-103/traetteberg.pdf <2013-12-09>.

[35] Trætteberg, H. (2008): Integrating Dialog Modeling and Domain Modeling: The Case of DiaMODL and the Eclipse Modeling Framework. In: *Journal of Universal Computer Science* 14 (19), 3265–3278.

[36] UIML Version 4.0 Committee Draft: https://www.oasis-open.org/committees/download.php/28457/uiml-4.0-cd01.pdf <2013-09-09>.

[37] UsiXML Consortium: State of the Art of User Interface Description Languages (2013): http://www.itea2.org/project/workpackage-document/download?document=468&file=08026_UsiXML_WP1_D1_1_v2_State_of_the_Art_of_UIDL (1).doc <2013-12-09>.

[38] UsiXML homepage: http://usixml.org/ <2013-09-09>.

[39] UsiXML: Reference Manual (2007)

[40] Vanderdonckt, J. et al. (2004): UsiXML: A User Interface Description Language for Specifying Multimodal User Interfaces. In: *W3C Workshop on Multimodal Interaction* (July 19–20, 2004, Sophia Antipolis, France), pp. 19–20. Online: http://www.w3.org/2004/02/mmi-workshop/vanderdonckt-louvain.pdf <2013-12-09>.

[41] W3C Member Submission: User Interface extensible Markup Language (UsiXML): http://www.w3.org/wiki/images/5/5d/UsiXML_submission_to_W3C.pdf <2013-09-09>.

[42] XIML homepage: http://www.ximl.org/ <2013-09-09>.

[43] XUL tutorial with examples: http://www.xul.fr/tutorial/ <2013-09-09>.