

Software Support for User Interface Description Language

Adrien Coyette¹, David Faure², Juan González-Calleros³, Jean Vanderdonckt¹

¹ Université catholique de Louvain, Louvain School of Management
Louvain Interaction Laboratory, Place des Doyens 1 - 1348 Louvain la Neuve, Belgium
{adrien.coyette,jean.vanderdonckt}@uclouvain.be

²Thales Research & Technology France - Campus de Polytechnique
F-91767 Palaiseau Cedex (France) - david.faure@thalesgroup.com

³University of Puebla, Faculty of Computer Science
Avenida San claudio y 14 sur, Puebla, Pue. (Mexico) – juan.gonzalez@cs.buap.mx

Abstract. A User Interface Description Language (UIDL) is a formal language used in Human-Computer Interaction (HCI) in order to describe a particular user interface independently of any implementation. Considerable research effort has been devoted to defining various meta-models in order to rigorously define the semantics of such a UIDL. These meta-models adhere to the principle of separation of concerns. Any aspect of concern should univocally fall into one of the following meta-models: context of use (user, platform, environment), task, domain, abstract user interface, concrete user interface, usability (including accessibility), workflow, organization, evolution, program, transformation, and mapping. Not all these meta-models should be used concurrently, but may be manipulated during different steps of a user interface development method. In order to support this kind of development method, software is required throughout the user interface development life cycle in order to create, edit, check models that are compliant with these meta-models and to produce user interfaces out of these methods. This workshop is aimed at reviewing the state of the art of software support for a UIDL in the context of any development method (e.g., formal method, model-based, model-driven). From this reviewing, a taxonomy of software support for UIDLs will emerge that will serve for describing, comparing, and exploring software support for UIDLs.

Keywords: Context of use, Model-driven architecture (MDA), Model-driven engineering (MDE), Service Oriented Architecture (SOA), situation engineering, user interface description language (UIDL).

Theme, Goals, and Relevance

A User Interface Description Language (UIDL) [2,4] is a formal language used in Human-Computer Interaction (HCI) in order to describe a particular User Interface (UI) independently of any implementation technology. As such, a UI may involve different interaction modalities (e.g., graphical, vocal, tactile, haptic, multimodal), interaction techniques (e.g., drag and drop) or interaction styles (e.g., direct manipulation, form filling, virtual reality). A common fundamental assumption of most UIDLs is that UIs are modeled as algebraic or model-theoretic structures that include a col-

lection of sets of interaction objects together with behaviours over those sets. Significant examples of UIDLs include: UIML (www.uiml.org) [5], useML (<http://www.uni-kl.de/pak/useML/>), MariaXML [7], UsiXML (www.usixml.org), and XIML (www.xml.org). Various UIDLs have been subject to discussion [6], understanding their common ground and their subsumed approach [9], a comparative analysis [3], and their consideration for standard [2]. Sometimes, alternative approaches have been considered and compared within a same UIDL such as UsiXML [4]. A UIDL can be therefore used during:

- *Requirements analysis*: in order to gather and elicit requirements.
- *Systems analysis*: in order to express specifications those address the aforementioned requirements.
- *System design*: in order to refine specifications depending on the context of use.
- *Run-time*: in order to realize a UI via a rendering engine.

The design process for a UIDL encompasses the definition of the following artefacts:

- *Semantics*. They express the context, meaning and intention of each abstraction captured by the underlying meta-models on which the UIDL is based on. Meta-Models are normally represented by means of UML Class Diagrams, OWL or other conceptual schemas. Semantics are usually conveyed using natural language.
- *Abstract Syntax*. It is a syntax that makes it possible to define UI models (in accordance with the UIDL semantics) independently of any representation formalism.
- *Concrete Syntax/es*. They are (one or more) concrete representation formalisms intended to express syntactically UI Models. Many UIDLs has an XML-based concrete syntax. In fact XML has been proven to be extremely useful in describing UIs according to the different levels of the Cameleon Reference Framework (CRF) [1] and for adapting UIs according to adaptation dimensions of the Similar Adaptation Space (SAS) [10].
- *Stylistics*. They are graphical and textual representations of the UIDL abstractions that maximize their representativity and meaningfulness in order to facilitate understanding and communication among different people. Stylistics are typically used by models editors and authoring tools.

Many UIDLs reveal themselves as a markup language that renders and describes graphical user interfaces and controls. But a UIDL is not necessarily a markup language (albeit most UIDLs are) and does not necessarily describe a graphical user interface (albeit most UIDLs abstract only graphical user interfaces). Figure 1 shows a general software architecture depicting typical software support for a UIDL. The workshop is aimed at defining a taxonomy for such a software support so that it can be used widely to refer to the same base. It is expected to review existing software support in the light of this taxonomy.

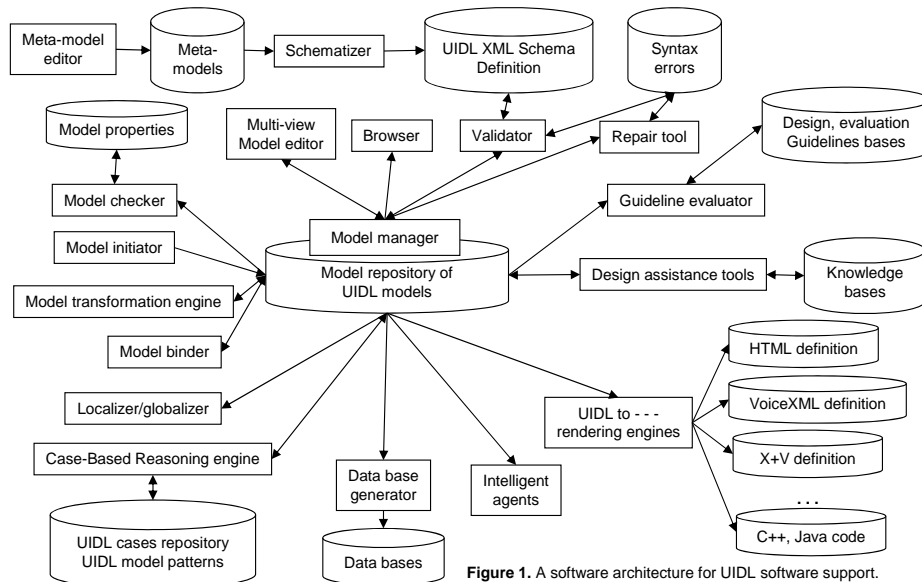


Figure 1. A software architecture for UIDL software support.

References

1. Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouil-lon, L., and Vanderdonckt, J. A Unifying Reference Framework for Multi-Target User Interfaces. *Interacting with Computers* 15, 3 (2003) pp. 289–308.
2. J.M. Cantera Fonseca (Ed.), *Model-based User Interface XG Final Report*, W3C Incubator Group Report, W3C, 4 May 2010. Accessible at: <http://www.w3.org/2005/Incubator/model-based-~ui/XGR-mbui-20100504/>
3. Faure, D., Vanderdonckt, J., (Eds.), *Proc. of 1st Int. Workshop on User Interface Extensible Markup Language UsiXML'2010* (Berlin, 20 June 2010). Thales Research and Technology France, Paris, 2010. ISBN 978-2-9536757-0-2.
4. Guerrero-García, J., González-Calleros, J.M., Vanderdonckt, J., and Muñoz-Arteaga, J. A Theoretical Survey of User Interface Description Languages: Preliminary Results. In *Proc. of LA-Web/~CLIHIC'2009* (Merida, November 9-11, 2009), IEEE Computer Society Press, Los Alamitos, 2009, pp. 36-43.
5. Helms, J., Schaefer, R., Luyten, K., Vermeulen, J., Abrams, M., Coyette, A., and Vanderdonckt, J. *Human-Centered Engineering with the User Interface Markup Language*. In: Sef-fah, A., Vanderdonckt, J., Desmarais, M. (eds.), *Human-Centered Software Engineering*, Springer, 2009, pp. 141-173.
6. Luyten, K., Abrams, M., Vanderdonckt, J. and Limbourg, Q. *Developing User Interfaces with XML: Advances on User Interface Description Languages*, Galipoli, 2004.
7. Paternò, F., Santoro, C., and Spano, L.D. MARIA: A universal, declarative, multiple abstraction-level language for service-oriented applications in ubiquitous environments. *ACM Trans. Computer-Hum. Interaction* 16, 4, (2009), Article #19.
8. Puerta, A.R. A Model-Based Interface Development Environment. *IEEE Software* 14, 4 (1997), pp. 40-47.
9. Shaer, O., Green, M., Jacob, R.J.K., and Luyten, K., *User Interface Description Languages for Next Generation User Interfaces*. In *Proc. of Extended Abstracts of CHI'08*, ACM Press, New York (2008), pp. 3949-3952.

10. Vanderdonckt, J., Coutaz, J., Calvary, G., and Stanciulescu, A. Multimodality for Plastic User Interfaces: Models, Methods, and Principles. In: D. Tzovaras (ed.), *Multimodal user interfaces: signals and communication technology*, Lecture Notes in Electrical Engineering, Springer, 2007, pp. 61-84.
11. Vanderdonckt, J. Model-Driven Engineering of User Interfaces: Promises, Successes, and Failures. In: *Proc. of 5th Annual Romanian Conference on Human-Computer Interaction ROCHI'2008* (Iasi, September 18-19, 2008). S. Buraga, I. Juvina (Eds.). Matrix ROM, Bucharest (2008), pp. 1–10.

Participants

Participants are welcome from any background or discipline that is concerned by UIDL ranging from psychology, social sciences to discipline of computer science, like software engineering, software development and testing. The expected amount of participants will be around 15, but could be extended. We will also accept attendees who have not submitted any work to the workshop provided that they have already worked in this domain (e.g., based on publications or implemented products or services).

Previous Workshop

This workshop is a generalization of [3]. While the first edition was focusing on meta-models useful for UIDLs, whatever their scope are, and their associated development methods, this second edition is focusing on software support and on discussing what are the appropriate and expected families of software for supporting the management and the usage of a UIDL. It is expected that the taxonomy of these services will result into a clearer identification of implementation efforts to be devoted to software support

Organization

Adrien Coyette is a scientific collaborator at Université catholique de Louvain and is an IT consultant for major inter-national companies. Jean Vanderdonckt is Professor of Computer Science at Université catholique de Louvain where he leads the Louvain Interaction Laboratory (LILab). David Faure is senior researcher at Thales Research & Technology France and the project coordinator of the ITEA2-Call3 project UsiXML (User Interface eXtensible Markup Language).

Format

The workshop will be a full day workshop that will result into discussion: the software architecture for a UIDL support (Fig. 1) and a taxonomy of such software according to a common definition format of services inspired by SOA. The workshop will start with an invited keynote by Prof. S. Kanai (Hokkaido University, Japan) based on his paper entitled “3D digital prototyping and usability enhancement of information appliances based on UsiXML”. A video projector and a flip chart will be used.

Tentative Program Committee (Pending)

- C. Aguéro, Thales Research & Technology, The Netherlands
- D. Akoumianakis, Technological Education Institution of Crete, Greece
- Malte Philipp Armbruster, Technical Univ. of Berlin, Germany
- B. Bomsdorf, University of Fulda, Germany
- P. Bottoni, University of Rome “La Sapienza”, Italy
- G. Calvary, Laboratoire d'Informatique de Grenoble, France
- D. Ciocarelli, Hewlett Packard, Italy
- A. Coyette, Université catholique de Louvain, Belgium
- S. Dupuy-Chessa, Laboratoire d'Informatique de Grenoble, France
- D. Faure, Thales Research & Technology, France
- E. Furtado, University of Fortaleza, Brazil
- J. M. González Calleros, Univ. of Puebla, Mexico

G. Gronier, Centre de Recherche Public Henri Tudor, Luxembourg
 J. Guerrero, Univ. of Puebla, Mexico
 S. Kanai, Hokkaido University, Japan
 M. Kizito, Univ. of Kaiserslautern, Germany
 K. Luyten, Univ. of Hasselt, Belgium
 V. M. Lopez Jaquero, University of Castilla-La Mancha, Spain
 G. Meixner, German Research Center for Artificial Intelligence (DFKI), Germany
 J.P. Molina, University of Castilla-La Mancha, Spain
 F. Montero, University of Castilla-La Mancha, Spain
 Ph. Palanque, IRIT Toulouse, France
 J. Plomp, VTT, Finland
 S. Praud, Thales Research & Technology, France
 J.C. Preciado, Univ. of Extremadura, Spain
 C. Pribeanu, ICI, Romania
 D. Raneburger, Univ. of Vienna, Austria
 D. Roscher, Technische Universität Berlin DAI-Labor, Germany
 A. Serra San Valero, ITACA, Spain
 D. Segers, Barco, Belgium
 R. Tesoriero, University of Castilla-La Mancha, Spain
 J. Vanderdonckt, Université catholique de Louvain, Belgium
 M. Winckler, IRIT Toulouse, France
 ...

Preliminary Call for Papers

Prospective participants should submit a position paper (ranging from 4 pages to 10 pages in ACM SIGCHI format) describing their interests and work in the topic of the workshop. First comments on (one or more of) the following topics we intend to discuss at the workshop are also welcome, but not mandatory (additional questions are also welcome):

- What are the major challenges (e.g., conceptual, methodological, technical, organizational) for software support for a UIDL?
- For which kinds of systems or applications are UIDLs appropriate, efficient, and desirable?
- When and how could we measure the effectiveness, the efficiency of UIDLs and their software support?
- How could we measure the quality of the user interface resulting from a development method based on a UIDL and supported by some software?
- In which ways will UIDL affect HCI practice in the future and how will they evolve?
- What kinds of software support are particularly desirable for a UIDL and what kind of implementation reveal themselves to be the most appropriate, efficient for this purpose?

We will accept participants based on the paper quality and the diversity of their backgrounds, aiming at an interdisciplinary group. The authors of accepted papers will be asked to provide a refined version of their paper one month before the workshop based on a preliminary version of workshop material. These position papers will be circulated in advance for participants to get an understanding of the mutual views and to provide a starting point for the discussion.

Paper Publication

Accepted submissions will be published in the following book (with ISBN): Coyette, A., Faure, D., Gonzalez-Calleros, J., Vanderdonckt, J. (Eds.), Proc. of Int. Workshop on User Interface Description Language UIDL'2011 (Lisbon, 6 September 2011), Thales Research and Technology France, Paris, 2011 (ISBN 978-2-9536757-1-9). The papers should be submitted via EasyChair at <https://www.easychair.org/conferences/?conf=uidl2011>.

Important dates:

- July 10th, 2011: submission deadline
- August 5th, 2011: notification of acceptance
- August 15th, 2011: final version of papers accepted for the workshop.
- September 6th, 2011: Workshop at Interact 2011 with book available, Room Q4.7 in the Conference center.