Software Support for User Interface Description Language

Adrien Coyette, David Faure, Juan González-Calleros, Jean Vanderdonckt

HAL Id: hal-01597021

https://hal.inria.fr/hal-01597021

Submitted on 28 Sep 2017

Software Support for
User Interface Description Language

Adrien Coyette[1], David Faure[2], Juan González-Calleros[3], Jean Vanderdonckt[1]

[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
[2]Thales Research & Technology France - Campus de Polytechnique
F-91767 Palaiseau Cedex (France) - david.faure@thalesgroup.com
[3]University 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 define rigorously the semantics of 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 review, 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) 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 collection of sets

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

References

1. Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouillon, L., and Vanderdonckt, J. A Unifying Reference Framework for Multi-Target User Interfaces. Interacting with Computers 15, 3 (2003) pp. 289–308.
2. 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.

Workshop web site at: http://itea.defimedia.be/node/130