

NEXOF-RA

NESSI Open Framework – Reference Architecture

IST- FP7-216446



**Open Architecture Specification Process
Open Construction Cycle #1**

**Declarative UI Authoring Languages
Position Paper**

Contact: Paternò Fabio, ISTI-CNR, HIIS Lab

Date of publication: October 9, 2008

This work is licensed under the Creative Commons Attribution 3.0 License.

To view a copy of this license, visit <http://creativecommons.org/licenses/by/3.0/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

1 DETAILS ABOUT THE CONTRIBUTORS

Main Contact / Lead

- Name: Fabio Paternò
- Affiliation: ISTI-CNR, HIIS Lab
- Email: fabio.paterno@isti.cnr.it
- Phone: +39 050 3153066
- Mobile Phone: +39 348 3962624

Projects Represented

- Project: OPEN
- Programme: FP7 – ICT

- Project: SevFace
- Programme: FP7 - ICT

2 TOPIC OF POSITION PAPER

2.1 Towards a Universal Declarative User Interface Definition Language

3 SUMMARY

The main characteristic of the technological evolution in recent years has been the increasing availability of a wide variety of interactive devices in the mass market. New offerings range from small mobile devices to stationary ones with large screens. Model-based approaches for user interfaces [1][4] have been considered in order to address such increasing complexity. They tend to hide a plethora of low-level details and allow designers to concentrate on the semantics of the user interface elements and their compositions, which determines the purpose for which such elements are used to support the user activities. The W3C Consortium has recently started a group [6] to investigate the adoption of such an approach.

There is general agreement that there are four relevant possible abstraction levels:

- The task and object level, which describes the activities to be performed and the objects manipulated during their performance;
- The abstract description level, where the user interface elements and their compositions are described in a modality-independent manner;
- The concrete level, in which the description is modality-dependent but independent of the implementation language;
- The implementation level, which refers to specific implementation languages.

It is important to point out that the adoption of this multi-layer approach does not impose any particular method. It is possible to exploit the description levels through both reverse and forward engineering transformations, which can support any kind of method: top-down, bottom-up, or middle-out.

Regarding the first level, the ConcurTaskTrees notation [1] has become a de facto standard for task modelling, and has been widely used at the international level in various universities and companies. It also has an associated environment (CTTE) [2] for editing and analysing the content of such models, which has received almost 7000 downloads. The notation provides hierarchical descriptions of the tasks to accomplish to reach the user goals with the possibility of specifying flexible temporal relations among such tasks (concurrent, disabling, order independence, ...) and a number of task attributes, including the associated objects and the platforms suitable for their performance. Business processing modelling notations are suitable to describe various pieces of application logic and how to orchestrate them. However, they provide poor support to derive the corresponding user interfaces: task models can be representations able to fill this gap.

Regarding the description of the abstract and concrete levels, a number of XML-based languages have already been put forward. However, they still suffer from various limitations. For instance, XForms [7] represents an example of how the research in model-based approaches has been incorporated into an industrial standard. In the same notation both abstract and concrete descriptions are included. There is a control vocabulary and constructs in abstract terms, and then presentation attributes and data types are described in concrete terms. However, (as its name indicates!) only the form-based interaction style is considered and in the end only the possibility of form-based interfaces on desktop and mobile devices are supported through the appearance attribute. This means that the notation is unsuitable to address other interaction modalities (such as vocal or gestural interaction).

A different approach has been addressed in TERESA XML [3], in which the abstract and concrete levels are described through different languages; the concrete languages are refinements of the abstract one depending on the specific interaction modality (form-based graphical, direct manipulation, vocal, vocal and graphical, ...). A tool supporting authoring multi-device user interfaces with TERESA XML [5] is publicly available at <http://giowe.isti.cnr.it/teresa.html>

Based on the experience with TERESA XML, at the HIIS Lab of ISTI-CNR we are developing a new modelling language, MARIA, which aims to provide a richer set of features, including abstract data types, support for Ajax-based applications, dynamic management of interface elements,... This work is carried out within the framework of the ServFace EU project (<http://www.servface.eu/>). This project aims to create a model-driven service engineering methodology for the design of user interfaces for applications based on web services; and the composition and integration of user interfaces for applications based on web-services. The resulting language will also be applied, tested and critically analysed in the OPEN project (<http://www.ict-open.eu/>) in such a way to provide useful information for further improving it. OPEN aims to deliver seamless and transparent support to users in carrying out their tasks when changing services and/or devices. The goal is to support migratory user interfaces for ubiquitous environments, in which users can change device and still continue from the point they left off interacting with user interfaces that adapt to the varying interaction resources available. The dynamically generated user interfaces for various platforms are obtained through the application of logical user interface descriptions and associated reverse and forward engineering transformations.

The new language will be also supported by a new authoring tool in which the supported transformations will be specified externally in order to allow their modification without requiring the modification of the tool implementation. The tool will also provide automatic support for the generation of user interfaces for Web services starting with their WSDL description and will provide effective interactive representations of the specifications under development to ease its use by designers and developers.

REFERENCES

- [1] F.Paternò, Model-based Design and Evaluation of Interactive Applications, Springer Verlag, ISBN 1-85233-155-0, 1999.
- [2] G. Mori, F. Paternò, C. Santoro, "CTTE: Support for Developing and Analysing Task Models for Interactive System Design", IEEE Transactions on Software Engineering, pp.797-813, August 2002 (Vol. 28, No. 8), IEEE Press.
- [3] G. Mori, F. Paternò, C. Santoro, Design and Development of Multi-Device User Interfaces through Multiple Logical Descriptions, IEEE Transactions on Software Engineering, August 2004, Vol.30, N.8, pp.507-520, IEEE Press.
- [4] F.Paternò, Model-based Tools for Pervasive Usability, Interacting with Computers, Elsevier, May 2005, Vol.17, Issue 3, pp. 291-315.
- [5] F. Paternò, C. Santoro, J. Mantyjarvi, G. Mori, S. Sansone, Authoring Pervasive MultiModal User Interfaces, International Journal of Web Engineering and Technology, Inderscience Publishers, 4(2) pp.235-261, 2008.
- [6] W3C Model-based Group, <http://www.w3.org/2005/Incubator/model-based-ui/charter/>
- [7] XForms, <http://www.w3.org/MarkUp/Forms/>