

Multidimensional Context-Aware Adaptation of Service Front-Ends

Javier Caminero¹, Mari Carmen Rodríguez-Gancedo¹, Jean Vanderdonckt², Fabio Paternò³,
Joerg Rett⁴, Dave Raggett⁵, Jean-Loup Comeliau⁶, Ignacio Marín⁷

¹ Telefónica R&D, Spain; ² Université catholique de Louvain, Belgium; ³ Consiglio Nazionale delle Ricerche, Italy;
⁴ SAP AG, Germany; ⁵ GEIE ERCIM, France; ⁶ W4 S.A., France; ⁷ Fundación CTIC, Spain

Email: fjc@tid.es, mcrg@tid.es, jean.vanderdonckt@uclouvain.be, fabio.paterno@isti.cnr.it, joerg.rett@sap.com,
dsr@w3.org, jean-loup.comeliau@w4global.com, ignacio.marin@fundacionctic.org

ABSTRACT

The EU-funded Serenoa Project (www.serenoa-fp7.eu) is creating a novel open platform for developing context-aware application user interfaces. Such user interfaces are aware of the changes in the context and can react to them in a continuous way. This includes adaptation to the user's devices, tasks, preferences and abilities, thereby improving the user's satisfaction and performance compared to traditional approaches to user interface design. An additional benefit is a reduction in the cost and development time for creation and maintenance of user interfaces.

DESCRIPTION OF THE PROBLEM

Even though software development environments, standards and engineering methodologies have significantly evolved and matured over the last decade, designing and implementing context-aware application user interfaces is still complex, and therefore expensive.

Businesses are seeing increasing diversity in the kinds of devices being used (i.e. workstations, desktops, laptops, smart phones, tablets, embedded devices, etc.), but this is putting up the cost of developing and maintaining user interface designs. Aligning business and Information Technology (IT) is a constant challenge, not only because of the constantly evolving standards and underlying technologies, but also because of the accelerating pace at which functional requirements change throughout the lifetime of applications. Business applications need to adjust rapidly to fulfill changing requirements, and this trend is accelerating as users require more and more mobility.

Moreover, implementing Human-Computer Interaction (HCI) systems requires forecasting user-driven interaction combinations that are not always easy to anticipate. Adding context management to the HCI domain increases complexity further because it requires taking into account multimodality (i.e. keyboard & mouse, touch screens, voice, gesture, etc.), geographical location, social context, accessibility and user preferences.

Finally, the underlying technologies involved in the development of user interfaces are multiple, keep changing and emerging at a rapid pace. Development therefore often requires technical experts, who are hard to find and retain. As a matter of fact, most of the effort in a traditional project is spent on infrastructure and technical tasks, and not on business issues. This often contributes to frustration of business stakeholders and end-users. There is also a steep learning curve for each new platform that has to be mastered: Android with Java, iOS with ObjectiveC, Windows with C#, Linux with C/C++, the Web with HTML and JavaScript, and so forth. Translating user interface designs into each of these platforms is time-consuming and expensive.

MODEL-BASED DESIGN

A variety of approaches have evolved in response to the above challenges. These include the Eclipse Integrated Development Environment, the Java 2 Platform Enterprise Edition (J2EE), W3C's HTML5, JavaScript libraries (e.g. jQuery), and OMG's Model-Driven Architecture (MDA). These have been supported by work on software development methodologies, such as Model-Driven Engineering (MDE) or Agile project methodologies, such as Scrum or xP, or with UI design approaches such as User-Centred Design (UCD). When it comes to designing user interfaces, Model-Based approaches bring many benefits by filling the gaps where traditional approaches are inadequate for agile development, and struggle to cope with the reality that requirements aren't fully understood at the outset of a project, but rather emerge as designs are tried out and found useful, relevant and necessary.

Model-based user interface design is about separating out design concerns at different levels of abstraction using both declarative and procedural knowledge. The declarative part is expressed as task models, abstract user interface (AUI) models and concrete user interface (CUI) models, whilst the procedural part is expressed with rule languages for adaptation at both design-time and runtime.

Other Model-Based approach benefits include simplification through automation and code generation, a

clean separation of concerns, e.g., IT and business issues, and the fact that the model becomes a common language for all project stakeholders and helps IT align with business. Designers vary in how they approach the challenges of user interface design. We would not take advantage of the true potential of Model-Based user interface design if we only supported a top-down methodology. A significant number of designers find it easier to start at the concrete level and refine the abstract level later. Subsequent revisions to the abstract level should be possible without throwing away the design work done at the concrete level. However, traditional Model-based design also suffers from a lack of attention to adaptation to the context of use, and this is exacerbated by a lack of standards for expressing adaptation rules.

DESCRIPTION OF THE SOLUTION

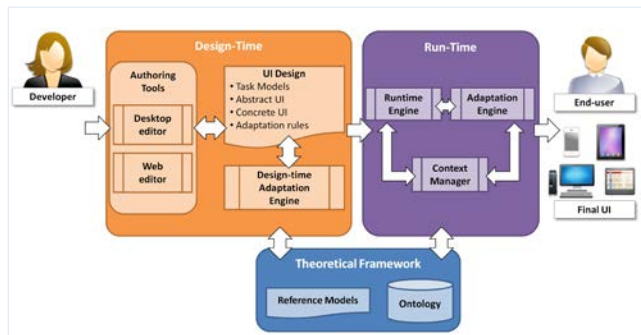


Figure 1. Serenoa's architecture

Serenoa's proposed architecture for supporting user interface adaptation is depicted in Figure 1.

Firstly, at design-time, 'Authoring tools' are used to support the design of model-based user interfaces. The authoring tools support Serenoa's abstract user interface description language, as well as the Serenoa language for expressing adaptation rules (see 'Languages'). For those developers who find it easier to start the definition of the interface at the concrete level and then refine the abstract level as they go, there is a design-time adaptation engine which makes it possible to work at both levels.

Secondly, the 'Theoretical framework' consists of the reference models and the ontology. These reference models are aimed at guiding developers and designers during the complete software life-cycle, listing alternative possibilities for implementing context-aware adaptation, and permitting the analysis and comparison of adaptive and adaptable applications. The ontology, based on the reference models, is intended to gather all the knowledge involved in advanced adaptation logic for user interfaces. It is used to inform developers as well as to support adaptation processes in the runtime.

Finally, the runtime phase will transform the description and associated rules of user interfaces into a final user interface implementation. The adaptation engine determines the optimal adaptation for the current context of

use, based upon the context models and adaptation rules. To achieve this goal, the context manager provides information related to all the possible contextual dimensions (i.e. the user preferences, the environment, social relationships, etc.) The run-time engine generates the final interactive application according to the context. Currently, this module is composed of a set of sub-modules which cover several modalities (i.e. mobile web applications, vocal interfaces, avatar-based interaction or desktop business applications). In fact, 'Applications prototypes' showing the functionality of these sub-modules and the whole Serenoa framework are under development.

AUTHORING TOOLS

Authoring tools help designers, engineers and web authors to easily create context-sensitive user interfaces for different platforms and different interaction modalities, e.g. visual, aural and tactile. The authoring tools will be usable by non-expert programmers, and fulfill some additional key success factors, including the usability of the graphical interface, the availability on multiple platforms and the support for concurrent work by multiple users (business experts, user interface designers and programmers working as a team) appear as very important requirements. The authoring tools will support editing of model-based descriptions at both abstract and concrete levels, together with context-dependent transformation rules. Two types of authoring tools are being developed as proof of concept for the Serenoa architecture:

- *Eclipse-based Plug-in*: this type is a 'plug-in' for Eclipse, one of the most widely used Integrated Development Environments (IDEs) in the research and industrial communities for software development. Since most of the open source and research based authoring environments/tools, i.e. MARIA and LEONARDI and other related libraries are developed in Java, we have also chosen the Java programming language for Serenoa's authoring environment and tools.
- *HTML5-based Browser Application*: the second type being developed is an HTML5-based web browser application that operates on the models that are held on the web server side. This approach allows live concurrent editing by multiple users, so that they could see and discuss the changes which a remote user is making in real time. A server-based design-time adaptation engine allows designers to work at different levels of abstraction, synchronising changes in a cooperative workflow.

Thus, Model-Based languages, i.e. MARIA, UsiXML, and IDEAL2 have been used as basis for not only developing the abstract UI and adaptation logic languages, but also the generation of Abstract/Concrete UIs.

LANGUAGES

The languages developed in Serenoa will cover the specification of adaptive SFEs at different abstraction

layers, and of the context-dependent transformation rules to be applied on the user interfaces. With the Serenoa solution, the exploitation of both these languages will be supported not only at design time but also at runtime. At design time, the authoring tools will help the designers, engineers and web authors to easily create and edit context-sensitive SFEs for different platforms (at both abstract and concrete levels) and relevant context-dependent transformations rules. At runtime, the logical descriptions of the SFEs and of the adaptation rules will be transformed in a final, adapted user interface implementation.

- The *Advanced Service Front-End Description Language* (ASFE-DL) is aimed at enabling the development and authoring of context-aware SFEs. The user interfaces modeled through this language will be adapted to the context by exploiting the rules defined through the *Advanced Adaptation Logic Description Language* (AAL-DL). By leveraging on past expertise on user interface languages that Serenoa members have already authored or co-authored, and on previous experiences they gathered by working in relevant industrial case studies to support requirements of most modern service-based user interfaces, the Serenoa consortium plan to build a more complete language that will allow ASFE-DL to meet the Serenoa requirements and to go beyond the state-of-the-art in this field. The ASFE-DL has been already specified at the Abstract user interface level: it describes the UI through a number of abstract interaction units and associated connections in a modality-independent manner. In the next months the Serenoa project plans to cover also the Concrete user interface level. The ASFE-DL is currently being submitted as an input to standardization work at W3C.

- The *Advanced Adaptation Logic Description Language* (AAL-DL) is a high-level language intended to express advanced adaptation logic in a declarative manner. The basic idea is that the user interfaces modeled through ASFE-DL will be adapted to the context by exploiting the rules defined through the AAL-DL. The AAL-DL rules have been expressed through an Event-Condition-Action (ECA)-based format where: i) events are changes that can occur in the context state or in the UI state; ii) conditions are Boolean predicates referring to context state or UI state; iii) actions are changes affecting the interactive application. In the current specification of the AAL-DL we have considered the definition of first-order adaptation rules (simple adaptation rules like e.g., adapt this service front-end for this platform) and second-order adaptation rules (those that govern the application of adaptation rules by e.g. selecting first-order rules: the action part of a rule can be in turn another rule). In the next version of this language we will consider third-order adaptation rules (strategies that privilege some adaptation approach for usability, performance, reliability or rules that promote or demote sets of second-order rules).

THEORETICAL FRAMEWORK

The Theoretical ground of the Serenoa Project is structured in three main components: a *Context-aware Reference Framework* (CARF), a *Context-aware Design Space* (CADS) and a *Context-aware Reference Ontology* (CARFO). The CARF provides to stakeholders the core concepts for defining and implementing adaptive and adaptable systems. The CADS provides means to analyze, evaluate and compare multiple applications regarding their coverage level of adaptation, especially concerning certain specific dimensions (such as: modality types). The CARFO not only formalizes the core concepts defined by CARF and their relationships, but also enables the request and retrieval of relevant information for defining and executing the adaptation process. Both industrial and scientific domains can benefit of these theoretical models, once they provide support for the whole development life-cycle of adaptive and adaptable applications, i.e. design, specification, implementation and evaluation.

APPLICATION PROTOTYPES

The *Warehouse-Management Scenario* for the intelligent picking prototype is aimed at providing a seamless context (environment and task) adaptation experience to users in one of the partner's Living Lab facilities in the field of Future Retail Concepts (FRC). This scenario motivates how proactive applications can provide unobtrusive and adequate help (e.g. missing parts, location of necessary parts, etc.) when the user needs it. Thereby, the service time can be reduced while increasing the quality of service. At the FRC, the supply chain continues to Retail Management with five demos, e.g. on price strategy and smart vending and concludes in Retail with eight demos, e.g. on Mobile Payment and RFID Shelves. This opens up the possibility, while following the modular design of the Serenoa components, to create further prototypes, moving from Logistics to Sales scenarios.

With the *E-commerce Scenario*, we aim at illustrating how different online end-users can take advantage of adaptive SFEs while connecting to both a front-end application and a back-end application. Typical user roles involved in the scenario include online shoppers and employees, acting either as supervisors or customer representatives in charge of following-up with online orders. Based on their roles, such users can access different features, but their UIs are capable of adapting based on different factors such as language, colour-blindness or type of device (either a home computers or a mobile device).

The *E-Health Scenario* is intended to be shown through the improvement of two existing developments:

- The SARA project, intended to provide a user interface for chronic disease patients self-monitoring in the form of a (Windows based) tablet PC. The project wants to evolve to provide multi-device support (Android tablet devices, smartphones, etc.) and an expressive virtual assistant in order to engage patients in the usage of the

application. This project is now in a pre-market phase, after successful field tests using real patients from the Andalusian health system.

- The HealthDrive project, aiming to leverage on consumer devices such as computers, tablet PCs and phones to provide its users access to their personal file on the Andalusian health system. In order to do so, all medical information is digitized and shared by the institutions, with a publicly accessible interface for each user in which she can interact with doctors and see their health records. The inclusion of ECAs (Embodied Conversational Agents) technology for guiding the navigation through all this personal information is also being considered.

ADDED VALUE AND IMPACT

In a general way, the Serenoa project may contribute to the following aspects:

- *Faster time-to-market:* given the support provided by the Serenoa technologies (i.e. authoring tools, adaptation languages and models) the implementation of adaptive and adaptable applications will be easier and more efficient.
- *Reduction of development effort:* instead of spending a lot of resources looking for knowledge about context-aware adaptation, stakeholders can rely in a centralized information source and a dedicated development platform, provided respectively by the theoretical framework, and by the technological framework (i.e. tools, languages, etc.).
- *Promote re-use:* by means of standards and a consistent terminology, the applications can be implemented in a more flexible manner, i.e. with the integration of components and with the extension and updates of existing applications.
- *Efficient solutions:* Serenoa is filling a gap in context-aware adaptation systems by allowing non-expert developers to develop efficient solutions. This is achieved, in particular, by adopting a model-based approach.
- *Increased agility:* Because context is often dynamic by nature, adaptation needs to be tuned little by little (and sometimes continuously), based on the feedback provided by the different stakeholders, starting with the end-user. Therefore, the capability for systems to change its behaviour rapidly and to take into account new adaptation rules, thus fitting new contexts, is essential. Serenoa's adaptation engine adds substantial agility to the development process of context-aware systems, which can provide great benefits.
- *Joining efforts:* By analyzing the technological landscape and the current status of the market, Serenoa identifies actual users' needs and orient its efforts in order to progress simultaneously in both scientific and industrial domains. Besides this, Serenoa tries to establish a formal link between the research community and the industry (for instance by means of standardization and dissemination actions).

As a part of the individual exploitation strategies the consortium members demonstrate the Serenoa 'Applications prototypes' and Development Tools to product groups, customers and partners. This aims at encouraging these industrial stakeholders for an adoption of the Serenoa technologies to enhance their existing products or to create new ones.

From the point of view of the academic community, several branches of knowledge are involved with research of context-aware adaptation. As the closest related ones, we can highlight: Human-Computer Interaction, Software Engineering and Architecture, Distributed Systems and Ubiquitous Computing. In the context of Serenoa project, the scientific field may benefit from: concrete requirements for designing and implementing adaptive systems, authoring tools and languages that support the creation of adaptive applications, theoretical frameworks that provide a catalogue of information to support the research in the field, evaluation criteria and possible architectural approaches.

Furthermore, releasing some Serenoa components as open source facilitates the adoption of the results by other members of the community, even any other projects, communities, organizations or anyone interested in this focused area. This issue implies that any of them could take the results and evolve them into something more complete, allowing to increase the impact of the project outside the community of the project. Specifically, Serenoa is expected to produce three types of results: theoretical frameworks, languages, and application prototypes.

- *Theoretical frameworks:* the reference models (CARF and CADs) will be made available for the public by means of written documents (as deliverables and scientific papers). They will also include a detailed description about their methodology, creation, and application available online by means of a web page. The Ontology will be developed based on this theoretical framework.
- *Languages:* the XML schemas describing the languages developed in Serenoa will be made public by putting them available on the project web site. In addition, such results will be submitted to W3C for standardization. The W3C working group on Model-Based User Interfaces (MBUI) has started its work and various Serenoa partners are involved in this group.
- *Application prototypes:* the applications, which will illustrate the Serenoa framework, lay on various software modules: context manager, adaptation engine, runtime engine and the rest of components. In general, the software prototypes will be made publicly available. We envision various policies for making them public: open source, public executable code, videos, etc. The way how they will be made public depends on the organization that is developing them and the type of prototype.

In order to accomplish these results and to arouse interest in the scientific and industrial community, an action planning has been envisaged. Firstly, on August-September 2012, a delivery of the final version of the Serenoa's architecture and the second release of the modules will be available for the evaluation. From then to the end of the project (September 2013) the development efforts will be mainly focused on the final release of the Serenoa's components and tools, the exploitation of them in the Serenoa use cases and the final evaluation.

SERENOA BENEFICIARIES

Serenoa is a framework that simplifies and accelerates the development of context-sensitive applications. This implies the reduction of the time-effort of the development process and its corresponding maintenance costs. The value of the framework is based on a solid model-based architecture, a good methodology and advanced tools to automatically generate applications. Different user roles are susceptible to take advantage of the aforementioned features:

- *Developer*: both final applications developers and those who are interested in adaptive technologies.
- *Researcher*: scientific community interested in context-aware adaptation and related topics.
- *End-User*: indirect beneficiary who gets faster results because of the time-to-market reduction.
- *Executive Director*: professional who may be interested in the creation of new business opportunities.

CONCLUSIONS

The Serenoa project is intended to fill a gap in the market for developing context-aware Service Front Ends. Providing simple tools and methodologies to address user-centric systems for a better and more enjoyable user experience potentially benefits a multitude of stakeholders, from researchers and developers to end-users and CIOs.

Starting from a theoretical model consisting of a reference framework and a design space, Serenoa addresses this challenge by adopting a Model-Based approach and by offering a set of tools that can be used both at design time and at runtime for implementing context-aware systems.

At design time, Serenoa's main components include authoring tools that are able to describe the user interfaces and their adaptation rules using two languages to design systems at different abstraction levels. At runtime, Serenoa's main components include the Context Manager to capture the current context related information, an Adaptation Engine to dynamically compute the optimal adaptation strategy and a Runtime Engine to generate the final user interface for each target platform.

The project provides various scenarios as proofs of concept, including a warehouse management application, an example of e-commerce application and an avatar-based e-health project. The project outcomes are expected to

accelerate time-to-market of context-aware systems, in order to increase agility and to promote reuse. Dissemination and exploitation efforts are expected to impact the tools used in the industry in the ICT sector and to influence academic actors.

ACKNOWLEDGMENTS

This work received funding from the European Commission's Seventh Framework Program under grant agreement number 258030 (FP7-ICT-2009-5).

REFERENCES

1. Ali, S., Rett, J., Lewandowski, A. (2011). A SOA based Context-Aware Order Picking System for Warehouses using Laser Range Finder and Wearable Computer. Lucca, Italy: IEEE WoWMoM, 2011, pp. 1 - 8.
2. Calvary, Gaëlle, et al. (2003). A Unifying Reference Framework for multi-target user interfaces. *Interacting with Computers*, vol. 15, pp. 289-308. *Computer-Aided Design of User Interfaces*.
3. Cavazza, Marc, et al. (2010). How was your day? An Affective Companion ECA Prototype. *Association for Computational Linguistics (ACL)*, 2010, pp. 277-280.
4. Limbourg, Q., et al. (2005). USIXML: A language supporting multi-path development of user interfaces. Springer, 2005, *Engineering Human Computer Interaction and Interactive Systems*, pp. 134-135.
5. Losada, B. et al. (2011). Agile Development of Interactive Software by means of User Objectives. Barcelona, Spain: ICSEA 2011 - The Sixth International Conference on Software Engineering Advances, 2011, pp. 539-545.
6. MyMobileWeb (2008-2011). "Open source software platform for the development of top-quality mobile web applications and portals, providing an advanced content & application adaptation environment". <http://mymobileweb.morfeo-project.org>, 2008-2011.
7. Paternò, F., Santoro, C. and Spano, L.D. (1999). MARIA: A universal, declarative, multiple abstraction-level language for service-oriented applications in ubiquitous environments. *ACM 2009, ACM Trans. Computer-Human Interaction*, vol. 16, pp. 19:1-19:30.
8. W3C (2011). Model-Based UI Working Group Charter, <http://www.w3.org/2011/01/mbui-wg-charter>
9. The SERENOA Project: Multidimensional Context-Aware Adaptation of Service Front-Ends J. Caminero, M.C. Rodríguez, J. Vanderdonck, F. Paternò, J. Rett, D. Raggett, J. Comelieu, I. Marín; *Proc. Of the 8th LREC International Conference*, Istanbul, Turkey, May 2012