Intelligent Design of Multi-Device Service Front-Ends with the Support of Task Models

by Fabio Paternò, Carmen Santoro and Lucio Davide Spano

We present a novel environment that supports the design of multi-device service-based interactive applications by using several user interface abstraction levels and utilizing the composition of annotated services.

Web services are increasingly used to support remote and distributed access to application functionalities, thus enabling a large gamut of interactive applications. However, when the user support (the service front end) is developed after delivering the Web services, it is often based on ad-hoc solutions that lack generality and do not consider the user viewpoint, thus providing front-ends with low usability.

BPMN (Business Process Modelling Notation) and BPEL (Business Process Execution Language) -based approaches are often used in Service–Oriented Architecture (SOA) efforts, but these mainly focus on direct service composition in which the output of one service acts as input for another, creating more complex services. Thus, they provide little support to the development of applications that interact with the users and perform service composition by accessing various services, processing their information and presenting the results to the end users.

One of the main advantages of logical User Interface (UI) descriptions is that they allow developers to avoid dealing with a plethora of low-level details associated with UI implementation languages. Our objective is to provide designers, who want to exploit the potential of Web services in their interactive applications, with a systematic approach that effectively addresses the specific issues raised by interactive applications when they are accessed through multiple devices. The method is based on the use of logical user interface descriptions and is also accompanied by an automatic tool (MARIAE, Model-based lAnguage foR Interactive Applications Environment), which supports all the phases. The approach uses different UI abstraction levels (task, abstract interface, concrete interface), which can also use additional UI information associated with web services (so-called annotations).

Task models describe the various activities that are supported by an interactive application, together with their relationships, through a number of relevant concepts. We describe tasks by exploiting the ConcurTaskTrees (CTT) notation, which is widely used thanks to the availability of the associated public tool (CTT Environment). Using MARIAE, it is possible to establish an association between some elementary tasks in the task model (namely, the tasks that cannot be further decomposed) and the corresponding operations specified in the Web services. This association, which involves only system tasks (namely, the tasks that are carried out by the application) is facilitated in CTT because different allocations of tasks are represented differently and therefore system tasks are easy to identify.

Provided that a suitable granularity is used to specify the task model, this association has the advantage of making the task
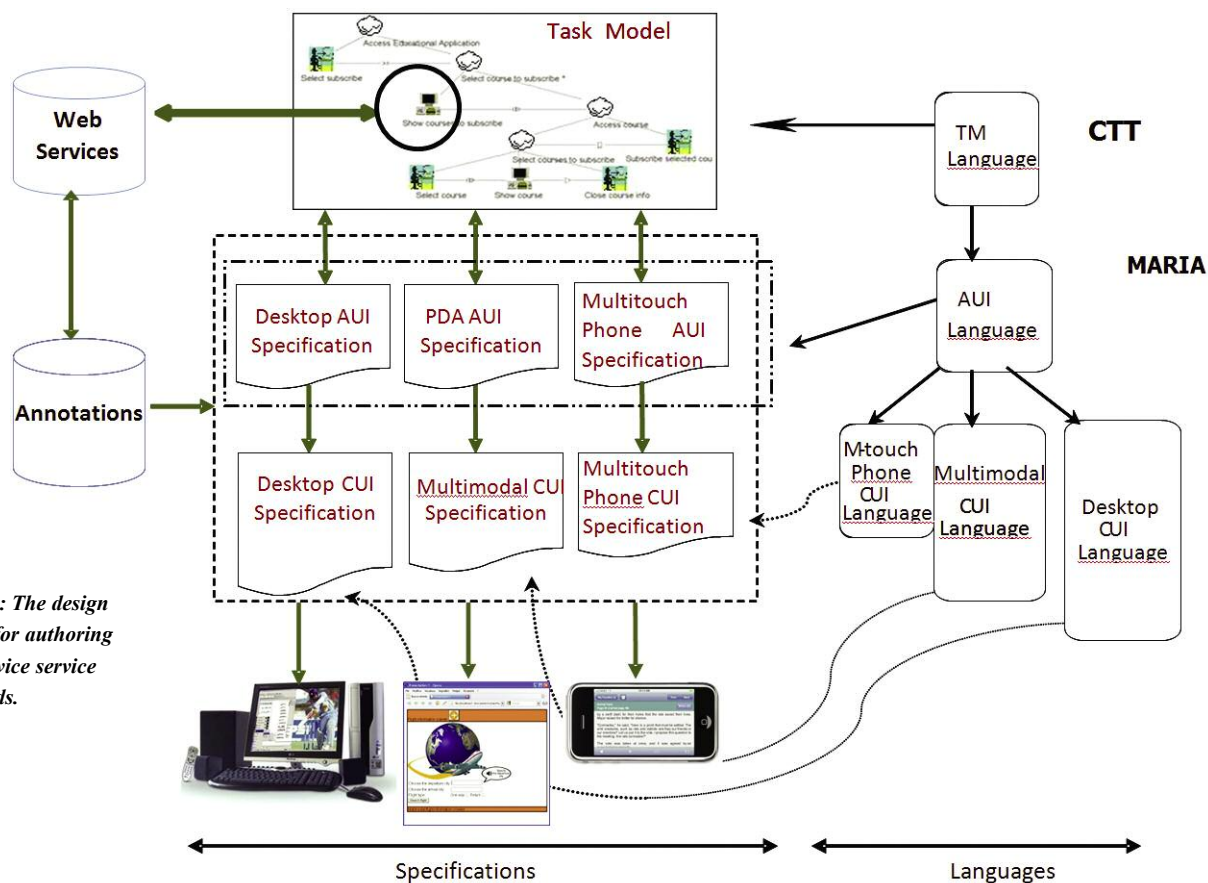

Figure 1: The design process for authoring multi-device service front-ends.

model temporal relationships automatically and consistently inherited by the associated Web service operations. This is important because task model relationships should be derived taking into account the user's perspective. Once the task model and Web services are connected, a first draft of an abstract UI description can be generated and then further edited by designers; when a satisfactory customization is reached, it can be refined into a concrete, platform-dependent one.

Designers can also include annotations associated with Web services, which provide (also partial) hints about the possibly related UIs. At the abstract UI level, the annotations can specify grouping definitions, input validation rules, mandatory/optional elements, data relations (conversions, units, enumerations), and languages. At the concrete UI level, the annotations can provide labels for input fields, content for help, error or warning messages, and indications for appearance rules (formats, design templates etc.).

When designing a specific multi-device application, the models created with device-independent languages should also take into account the features of the target interaction modality. Thus, for example, in the task model we can have tasks that depend on a certain modality (eg selecting a location in a graphical map or showing a video), and are neglected if they cannot be supported (eg in a platform having only the vocal modality).

For each target platform it is possible to focus just on relevant tasks and then derive the corresponding abstract description, which is in turn the input for deriving a more refined description in the specific concrete language available for each target platform. Since the concrete languages share a common core abstract vocabulary, this work is easier than working on a number of implementation modality-dependent languages. Our tool MARIAE is currently able to support the design and implementation of interactive service-based applications for various platforms: graphical desktop, graphical mobile, vocal, multimodal (vocal + graphical) and can be freely downloaded.

The tool has been developed in the ServFace project, whose goal was to create a model-driven service engineering methodology to build interactive service-based applications using UI annotations and service composition. The project began in 2008 and finished in 2010. The institutions involved were ISTI-CNR, SAP AG (Consortium Leader), Technische Universität Dresden, University of Manchester, W4.

Links:
ConcurTaskTrees Environment:
http://giove.isti.cnr.it/tools/CTTE/home
MARIAE Tool: http://giove.isti.cnr.it/tools/MARIAE/home
HIIS Laboratory: http://giove.isti.cnr.it/
ServFace EU Project: http://www.servface.eu/

Please contact:
Fabio Paternò
ISTI-CNR, Italy
Tel: +39 050 315 3066
E-mail: fabio.paterno@isti.cnr.it

Training Crisis Managers in Strategic Decision-Making

by Amedeo Cesta, Gabriella Cortellessa, Riccardo De Benedictis, and Keith Strickland

The goal of PANDORA is to apply state-of-the-art ICT technologies to build a learning environment for strategic crisis managers. We are currently refining a first version where training sessions are animated by reproducing realistic crisis events and fostering creative decision-making. Central to PANDORA is an original use of the timeline-based planning strategies used to diversify crisis scenarios by creating alternative training paths and to model trainees behavioral patterns for personalized training.

Crisis management in emergency situations helps significantly to avoid major losses and may prevent emergencies from becoming disasters. The success of crisis management depends heavily on the effectiveness of high-level strategic choices and the reasoning abilities of decision makers.

Three different levels of decision makers exist in current approaches to crisis management:
• operational level or bronze commanders, directly operating on crisis scenarios, whose actions and results are monitored and communicated to higher levels
• tactical level or silver commanders, responsible for translating high-level strategic decisions into actions and related resources allocations
• strategic level or gold commanders, who identify key issues and decide strategies to resolve the crisis.

The PANDORA project aims at creating an advanced training environment for crisis decision makers who operate in highly stressful situations. They must react effectively and coordinate interventions with different authorities to limit the dangerous effects of crises and to enable quick recoveries. Contrary to almost all the state-of-the-art training systems, aimed at the operational or tactical level, PANDORA targets decision-making at the strategic level, which presents interesting open challenges.

There are two main approaches to training: (a) tabletop exercises (group discussions guided by a simulated disaster); (b) real world simulation exercises (field-tests replicating emergency situations). Tabletop exercises are low cost and easy to organise, but they cannot recreate the real atmosphere, in terms of stress and pressure. On the other hand, real world simulations can be very effective but are extremely expensive and difficult to organize.

PANDORA aims at replicating the benefits of both training approaches by developing a system capable of guaranteeing the realism of the real world simulation and the practicality of tabletop exercises. A user-centered approach is followed and we have worked in close collaboration with the UK Emergency Planning College, which has identified the main