Adaptable UI for Web Service Composition: A Model-Driven Approach

Waldemar Ferreira Neto
Supervised by: Philippe Thiran

PReCISE Research Center, University of Namur, 5000, Belgium
{waldemar.neto,pthiran}@fundp.ac.be

Abstract. The main objective of this work is to provide User Interfaces (UI) for Web service compositions (WSC). We aim at investigating how user interfaces and their navigation can be derived from the WSC structures (data and control flows). We propose a model-driven engineering approach that provides models and transformational methods that allow deriving and adapting UI for any context of use.

Keywords: Web service composition, model-driven engineering, user interface, adaptation.

1 Introduction

Web services have gained attention due to the pressing need for integrating heterogeneous systems. A Web service is a software system designed to support interoperable machine-to-machine interactions over a network. It has an interface described in a machine-processable format. A main advantage of Web services is their ability of being composed. A Web service composition (WSC) consists in combining several Web services in a same process, in order to address complex user's needs that a single Web service could not satisfy [2].

There are several initiatives to provide languages that allow the description of a Web service composition. The current WSC languages are expressive enough to describe fully automated processes to build Web service compositions [2]. However, full-automated processes cannot represent all real-life scenarios specially those that need user interactions. In these scenarios, a user interaction may range from simple approvals to elaborate interactions where the user performs a complex data entry, for example, filling several forms.

Any computer system that involves users needs user interfaces (UI) to permit the interactions between the system and the user. The users of a WSC can interact with it through diverse devices (Desktop, Smart Phone, Tablet, among others) in diverse modalities (visual, aural, tactile, etc.). The adaptability of the UIs for a WSC has become necessary due to the variety of contexts of use.

In this work, we propose a model-driven engineering (MDE) approach for providing adaptable UIs from WSC. In particular, the approach relies on a modelization of user interactions within the WSC. Based on this modelization, the

G. Pallis et al. (Eds.): ICSOC 2011, LNCS 7221, pp. 177–182, 2012.

approach proposes a method to derive an abstract representation of the UI from a WSC. Interestingly, the derivation rules rely on the data/control flow of the WSC for specifying the navigation through the UIs. The obtained abstract representation can then be adapted to any specific context of use.

The remainder of this work is organized as follows. An overview of the works about user interactions and Web service composition is given in Section 2. Section 3 explores the research challenges associated with the generation of UI from WSC. Section 4 proposes an MDE approach to deal with challenges that were identified. Section 5 offers a preliminary plan for realizing our MDE approach and Section 6 concludes.

2 Related Work

There are several approaches that permit interactions between users and Web services. In some of these approaches, the information about the Web service (which can be WSDL or OWL-S) is used to infer a suitable user interface (e.g., [8]). To increase the usability of generated used interfaces, some approaches use additional information like UI annotation [9], platform-specific description [12], or user context [14]. In these approaches, the UI generation relies on type of the input and output described on the Web service description.

The development of Web interfaces for Web services has been addressed by the Web engineering community by the means of model-driven Web design approaches [15] and [4]. These approaches propose a model-based development method for process-based Web applications that use Web services. The former approach describes the Web service composition by BPMN and the UI navigation is described by a web-specific visual modeling language, WebML [4]. The latter relies on BPMN too, but the UI navigation is described on an object-oriented modeling language, OOWS [15]. Based on HTML templates, a set of UIs can be automatically generated form the WSC, and the navigation among these UIs is driven by the navigation model.

Another work that generates user interfaces for Web services is the Dynvoker [13]. This approach interprets a determined Web service and generates Web forms according to the Web service operation. Based on a BPEL-like language (GUI4CWS) this approach allows to handle complex service interaction scenarios. There are other approaches that allow a similar UI generation, but these approaches consider multiples actors [5] or/and context-aware UIs [11].

Other approaches generate UI for Web services based on the annotated Web service descriptions and the UI defined from a task model [17]. The annotations are used to generate the UI for the Web services and the task model drives the navigation among the UIs and Web services. As such, these approaches do separate the data/control flows of the WSC and the UI navigation model.

Other works aim at extending WSC descriptions with user interactions. An example of such extensions is BPEL4People [1], which introduces user actors into a Web service composition by defining a new type of BPEL activity to specify user tasks. However, this extension focuses only on the user task and

does not deal with the design of a user interface for the Web service composition. Another example of BPEL extensions that addresses the user interaction is BPEL4UI (Business Process Execution Language for User Interface) [6]. BPEL4UI extends the Partner Link part of BPEL in order to allow defining a binding between BPEL activities and an existing UI. This user interface is developed separately from the composition instead to be generated. In another work, Lee et al. [10] extend BPEL by adding interactive activities that are embedded in the BPEL code. Unlike BPEL4UI, this work specifies the UI together with the WSC, however the UI is specified for a unique context of use.

3 Research Challenges

The main objective of this work is to derive adaptable UI from WSC. In the following, we present the research challenges that must be tackled to achieve this objective.

First, we need to investigate how user interactions can be integrated within WSC. Concretely, WSC must be extended with user interaction activities that express the different possible types of user interactions [16]: data input interaction, data output interaction, data selection, and interaction by user event.

Another challenge is the fact that the navigation and the composition of the UI can rely on the control/data flow structures of the WSC extended with user interaction activities. A simple example of generation is given in Figure 1 that presents a simple travel reservation management. This WSC comprises three user interaction actives. The UI generation can lead to a UI grouping of the two first user interactions activities (*initializing Service* and *transportation means selection*) as data provided by these user interactions are mutually independent. However, this UI could not comprise the third user interaction activity (*providing license number*), as the user interaction will only be enable if the transportation means is the private car.

The last challenge is to be able to generate a UI adapted to the user context (user preference, user environment, and user platform) and the usability criteria (e.g., the size of the device screen).

4 Proposed Approach

We propose a Model-driven Engineering (MDE) approach that provides models and transformations for deriving and adapting UI from WSC and the context of use. We identify 3 main models and 3 main methodological steps.

4.1 Models

Our MDE approach relies on 3 models:

- *UI-WSC*: an extension of WSC with user interaction activities. To be compliant with current standards, the model is to rely on existing standards: a standard for WSC (e.g. BPEL) and a standard for describing user interfaces (e.g. UsiXML).

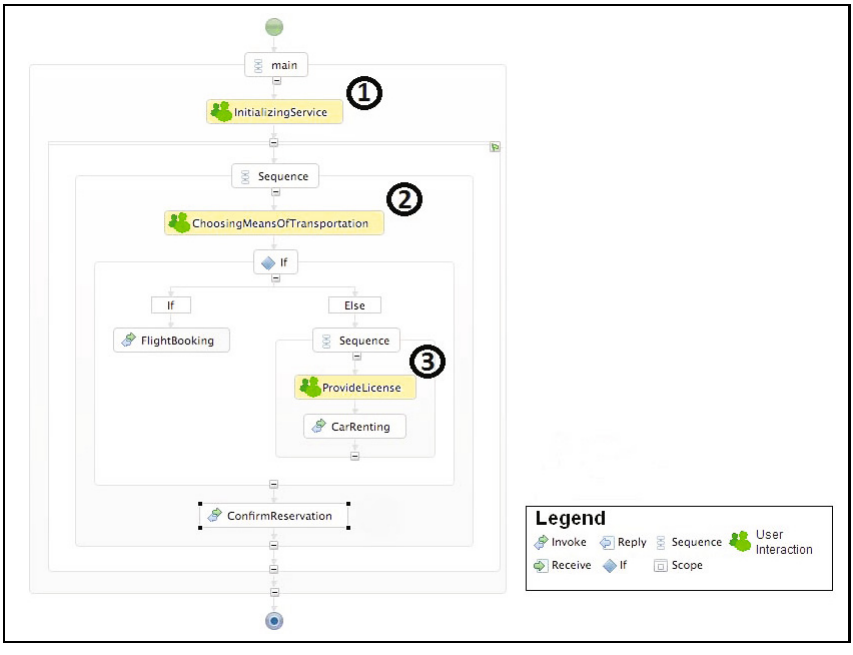

Fig. 1. Web service composition to manage travel reservations

- *Abstract user interface (AUI)*: this model describes the UI independently to any interaction modality (e.g. graphical modal, vocal modal) and computing platform (e.g. PC, smart phone). This model only specifies the UI components, their elements, and the navigation among the components.
- *Concrete user interface (CUI)*: this model is an adaptation of an AUI to a specific context of use (user preference, environment, and platform). For example, for visually handicapped person, an output abstract component could be transformed to a (concrete) vocal output.

4.2 Method

Our MDE method consists in 3 main steps:

- *Modeling*: where the WSCs are modelized within its user interactions by a designer using the UI-WSC.
- *Transformation*: where the AUI is derived by applying transformations to the UI-WSC model.
- *Adaptation*: where the CUI is derived from the AUI and the context of use. Additionally, the user can interact with the CUI through an interpreter, while a runtime component arbitrates the communication between the CUI and the WSC.

5 Research Methodology

The first part of our research consists in the definition of the different models of our MDE approach. In particular, we investigate and modelize how the user interaction can be specified within WSCs. Our goal here is to propose a extension to WSC meta-model (UI-WSC meta-model) with the user interaction activities representing the different possible types of user interactions. For the AUI and CUI meta-models, we refer to existing works in UI meta-modeling.

Next, we define the transformation rules for deriving an AUI description from a UI-WSC model. We plan to define these rules in an incremental way: starting with simple UI-WSC patterns (e.g., input/output sequence, choice) to continue with more complex ones (e.g., loop or interruptible area).

AUI adaptation is the next step. As there are existing approaches, we plan to investigate and evaluate these approaches so that we can to adopt the more suitable to our approach.

As a proof of concept, we develop a tool that not only supports the three main steps of your MDE method (design) but also orchestrate the WSC execution and the user interactions with the user (runtime).

Finally, we evaluate our approach. We first aim at evaluating our approach against other approaches (e. g. [6,15] and [4]). As comparison criteria, we adopt the usability criteria proposed by the ISO 9241 [7]: satisfaction, effectiveness, and efficiency. We also aim at evaluating our approach in real scenarios with real users.

6 Conclusion

In this work, we propose an MDE approach for providing adaptable UI from WSC. This approach aims at specifying all types of user interactions within WSC process, as well as the derivation of an abstract representation of the UI. The derivation rules rely on the data/control flow of the WSC for specifying the navigation through these abstract representations. Finally, the obtained representation can then be materialized to any specific context of use in order to provide an adapted UI.

So far, we have reviewed the literature about the users interactions and Web services. We have already proposed a BPEL extension able to modelize all types of user interactions within WSC processes, named UI-BPEL meta-model [3]. We have also implemented a design tool that is dedicated to edit a WSC conform to our UI-BPEL meta-model. The tool is an Eclipse plug-in based on the Eclipse BPEL Designer[1]. As future work, we plan to work on the transformation rules for deriving AUI from UI-BPEL and integrate these rules into our modeling tool.

References

1. Agrawal, A., Amend, M., Das, M., Ford, M., Keller, C., Kloppmann, M., König, D., Leymann, F., Müller, R., Pfau, G., et al.: Ws-bpel extension for people, bpel4people (2007)

[1] http://webapps.fundp.ac.be/wse/wiki/pmwiki.php?n=Projects.UIBPEL

2. ter Beek, M.H., Bucchiarone, A., Gnesi, S.: Web service composition approaches: From industrial standards to formal methods. In: ICIW, p. 15. IEEE Computer Society (2007)
3. Boukhebouze, M., Neto, W.P.F., Erbin, L.: Yet Another BPEL Extension for User Interactions. In: De Troyer, O., Bauzer Medeiros, C., Billen, R., Hallot, P., Simitsis, A., Van Mingroot, H. (eds.) ER Workshops 2011. LNCS, vol. 6999, pp. 24–33. Springer, Heidelberg (2011)
4. Brambilla, M., Dosmi, M., Fraternali, P.: Model-driven engineering of service orchestrations. In: Proceedings of the 7th Congress on Services, pp. 562–569. IEEE Computer Society, Washington, DC (2009)
5. Daniel, F., Casati, F., Benatallah, B., Shan, M.-C.: Hosted Universal Composition: Models, Languages and Infrastructure in mashArt. In: Laender, A.H.F., Castano, S., Dayal, U., Casati, F., de Oliveira, J.P.M. (eds.) ER 2009. LNCS, vol. 5829, pp. 428–443. Springer, Heidelberg (2009)
6. Daniel, F., Soi, S., Tranquillini, S., Casati, F., Heng, C., Yan, L.: From People to Services to UI: Distributed Orchestration of User Interfaces. In: Hull, R., Mendling, J., Tai, S. (eds.) BPM 2010. LNCS, vol. 6336, pp. 310–326. Springer, Heidelberg (2010)
7. ISO (ed.): ISO 9241-11: Ergonomic requirements for office work with visual display terminals (VDTs) – Part 9: Requirements for non-keyboard input devices (2000)
8. Kassoff, M., Kato, D., Mohsin, W.: Creating GUIs for web services. IEEE Internet Computing 7(5), 66–73 (2003)
9. Khushraj, D., Lassila, O.: Ontological Approach to Generating Personalized User Interfaces for Web Services. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 916–927. Springer, Heidelberg (2005)
10. Lee, J., Lin, Y.Y., Ma, S.P., Lee, S.J.: BPEL extensions to user-interactive service delivery. J. Inf. Sci. Eng. 25(5), 1427–1445 (2009)
11. Pietschmann, S., Voigt, M., Rümpel, A., Meißner, K.: CRUISe: Composition of Rich User Interface Services. In: Gaedke, M., Grossniklaus, M., Díaz, O. (eds.) ICWE 2009. LNCS, vol. 5648, pp. 473–476. Springer, Heidelberg (2009)
12. Song, K., Lee, K.H.: Generating multimodal user interfaces for web services. Interacting with Computers 20(4-5), 480–490 (2008)
13. Spillner, J., Feldmann, M., Braun, I., Springer, T., Schill, A.: Ad-Hoc Usage of Web Services with Dynvoker. In: Mähönen, P., Pohl, K., Priol, T. (eds.) ServiceWave 2008. LNCS, vol. 5377, pp. 208–219. Springer, Heidelberg (2008)
14. Steele, R., Khankan, K., Dillon, T.S.: Mobile web services discovery and invocation through auto-generation of abstract multimodal interface. In: ITCC (2), pp. 35–41. IEEE Computer Society (2005)
15. Torres, V., Pelechano, V.: Building Business Process Driven Web Applications. In: Dustdar, S., Fiadeiro, J.L., Sheth, A.P. (eds.) BPM 2006. LNCS, vol. 4102, pp. 322–337. Springer, Heidelberg (2006)
16. Trewin, S., Zimmermann, G., Vanderheiden, G.C.: Abstract representations as a basis for usable user interfaces. Interacting with Computers 16(3), 477–506 (2004)
17. Vermeulen, J., Vandriessche, Y., Clerckx, T., Luyten, K., Coninx, K.: Service-Interaction Descriptions: Augmenting Services with User Interface Models. In: Gulliksen, J., Harning, M.B., van der Veer, G.C., Wesson, J. (eds.) EIS 2007. LNCS, vol. 4940, pp. 447–464. Springer, Heidelberg (2008)