Towards Ontology-driven Composition of Personalized Mobile Services by End-users

Rune Sætre[1], Mohammad Ullah Khan[2], Erlend Stav[3], Alfredo Perez Fernandez[1], Peter Herrmann[2], and Jon Atle Gulla[1]

[1] Computer and Information Science (IDI) NTNU, satre@idi.ntnu.no
[2] Telematics (ITEM) NTNU, 7491 Trondheim, Norway
[3] Information and Communication Technology (ICT) SINTEF, 7031 Trondheim, Norway

Abstract. With mobile devices being an integral part of the daily life of millions of users having little or no ICT knowledge, mobile services are being developed to save them from difficult or tedious tasks without compromising their needs. Starting with a number of real life scenarios we have been working towards supporting end-users in managing their services in an efficient and user-friendly manner. We observe that these scenarios consist of sub-tasks that can be solved with collaborative service units. Therefore, a composition of such service units will serve the needs of the end-user for the complete scenario. We envisage that a visual formalism and tools can be developed to support these end-users in creating such service compositions. Moreover, methodologies and middleware can significantly reduce the complexity of developing composite services. This paper focuses on the role of ontologies within that context. Ontologies can assist the users in selecting appropriate services and setting composition parameters within a composition tool. For our prototype demonstration system we target the open source Android cellphone architecture supporting a number of different runtime platforms.

1 Introduction

The world of cell-phones and mobile WiFi-devices is changing rapidly. Modern mobile phones now have the same computational power as a desktop computer had only a few years ago. Many small programs and services are being made to harness this power in order to save the users from difficult or boring tasks. For example, the address book application on the phone remembers all the phone numbers for the user, so he or she can call someone just by typing (or saying) their name, or perhaps just by clicking on the correct portrait picture. We have studied several usage scenarios like "incoming call handling", "doctor's appointment" and "city-guiding". These scenarios consist of several sub-tasks that can be solved with various existing applications. We are developing a *composition tool* to allow non-experts (without programming experience) to compose their own services. The hypothesis is that a visual formalism and tools can be developed to support end-users in composing (their own) service collaborations. We

want to show that service composition can be supported by generic solutions in the form of methods and middleware that significantly reduce the complexity of developing these composite services. We have started implementing service- and domain-ontologies that allow reasoning about the user's intentions, and that can provide useful feedback in the compositional phase.

After a paper-prototyping phase to settle on an intuitive end-user composition notation and approach, and a comprehensive state-of-the-art survey, we are now moving into the runnable prototyping phase. In this paper we briefly provide the overall picture, while focusing on the usage of ontologies within the context of end-user service composition.

2 The City-guide Scenario and the Ontology

We describe the City-guide scenario, which is one of the scenarios [4] that we are using to test our approach. We will first investigate the need for ontologies and then very briefly mention the concepts provided in the developed ontology to serve that purpose.

In this scenario, the composer creates an Android-based service which should assist a person (scientist or business man) in making a short guided tour in a not yet visited city within an afternoon. The available time is between the end of the conference or business meeting (which may be delayed), and the check-in time at the airport. The starting point is the meeting venue, and the final target is the airport. Means of transportations are walking, bus, train, taxi etc. The composer, still at home, wants to specify these constraints, together with a list of user profiles, stating what kind of sights are interesting to the user. For example, the *cultural* profile might include one visit to a church, like the *Nidaros Cathedral*, and one to a museum, e.g. the *Ringve Museum* in Trondheim.

To achieve that, the composer discovers the following service *building-blocks*[5]

1. Guided tours at the Nidaros Cathedral.
2. Guided tours at the Ringve Museum.
3. Bus time-tables and information about the necessary time to walk to and from the bus stops that are closest to the starting and the end point.
4. A call service for a taxi company.
5. A guiding map for walking from the present position to a certain point.
6. A scheduler for events, which provide alerts some time before the events.

Our scenarios are quite complex for an end-user composer, as it takes lot of effort to reflect the plan in the service composition. Also, we notice that the successful organization of events at different places seems to have a recurring pattern. It would be nice if we could relieve the composer from repetitive tasks such as transportation planning between composed events.

[4] More details on the other scenarios like "incoming call" or "doctor's appointment" are available in [2].

[5] A building block is a system abstraction used by a service composer. It can represent a component, a service or a system part in terms of a sequence of actions.

Ontologies can be of help here, since they define suitable terms which may guide the system to understand what it should do with the different building blocks. For example, they can be used to connect places of events with nearby places of transportation.

As the first step towards simplifying end-users' task in composing services with the intended support we have developed an ontology in OWL based on the presented scenario. This ontology will be integrated with the service composition tool (see section 3) so that the service composer (an end-user) can benefit from it during the service composition. We also foresee that the run time platform will also use the ontology in making automatic decisions; e.g., in the case of selecting alternative services.

The concepts contained in the ontology consists of Classes representing different entities. It contains entities like *Nidarosdomen* which is of subclass *Cathedral* in the class of *Church*. Other classes are *Museum*, *Station* and *Vehicle*. These concepts can be populated (semi-) automatically from a database of sights for the city/area, or from textual descriptions, using existing research, e.g., [1,3].

3 Methodology and Tools

We are developing a methodology that provides a step-by-step procedure both for the end-users creating service compositions and for the service developers creating individual services. This involves the usage of a composition tool that helps end-users to define the composition, using an end-user-friendly and intuitive notation Here we briefly introduce the contribution of the ontology within the methodology.

At design time, the composer creates application compositions by using the service building blocks. The composition is created in a scenario-like fashion; e.g., a particular block executes or a particular sequence of actions are performed only when a certain condition holds. Such conditions are dynamic in nature so that the evaluated result may depend on some information obtained at run time. Ontologies may be used to discover certain building blocks that help to solve a certain problem, in particular, to glue building blocks that should be executed in a specific sequence.

The composed scenarios are selected based on the conditions evaluated at run time. The run time platform should take care of the fact that the user's needs may change; e.g. the user may re-compose the application at run time. Ontologies can be used in identifying and selecting from alternative scenarios and services that serves the same composition and the same building block, respectively.

The methodology is supported by a meta-model representing the concepts, a set of end user-friendly notation and a composition tool that creates the composition taking into account. The tool is created using the Google Web Toolkit (GWT[6]), which is a development toolkit for building and optimizing complex

[6] `http://code.google.com/webtoolkit/`

browser-based applications. The composition tool works both on a PC and a mobile phone. We target end-users with limited or no ICT knowledge, but with the ability to use a PC or a mobile phone. Therefore, the composition tool also needs to be easy-to-use and easy-to-install. That triggered the choice of a composition tool that runs on a web browser. The tool development also takes advantage of tools like WindowBuilder[7] to work with GWT in addition to the automatic generation of editors from EMF meta-models.

4 Conclusion and Future Work

In this paper, we advocated the usage of *ontologies* within a methodology that supports end-user service compositions. End-users with minimum ICT knowledge needs intuitive, intelligent and easy-to-use notation and tools in managing their tasks. We argued that the use of ontologies can be of great help towards that direction. By analyzing different scenarios, we developed initial ontologies supporting different application domains, which are in the process of being integrated with an end-user service composition tool.

The usage of good ontologies can substantially make end-users' lives easy. Ontologies can help predicting what the user will type while composing a service. For example, there is only one *Church (or Cathedral)* with its name starting with the letter *N* in a particular city. These ontologies can be created either at compose-time for given cities, or for general-purpose (multi-city) services as soon as the composed service is started on the phone, and the GPS etc. is used to determine which city the user is in.

We are currently implementing the concepts presented in this paper. For our prototype demonstration system we target the open source Android cellphone architecture supporting a number of different runtime platforms. For example, we are currently developing end-user composition support for self-adaptive mobile applications running on the MUSIC platform. The implementation follows the incremental approach of updating the concepts and the implementation based on the feedback from user-testing on several different platforms/scenarios.

References

1. Gulla, J.A., Brasethvik, T.: A Hybrid Approach to Ontology Relationship Learning. In: Kapetanios, E., Sugumaran, V., Spiliopoulou, M. (eds.) Natural Language and Information Systems (NLDB), Lecture Notes in Computer Science, vol. 5039, chap. 9, pp. 79–90. Springer Berlin / Heidelberg, London, UK (2008)
2. Shiaa, M.M., Vaskinn, J.E., Sanders, R.T.: Tool Chain for End-Users Service Composition. In: Proceedings of the 2010 VERDIKT conference. p. 56 (November 2010)
3. Witte, R., Khamis, N., Rilling, J.: Flexible Ontology Population from Text: The OwlExporter. In: The Seventh International Conference on Language Resources and Evaluation (LREC 2010). pp. 3845–3850. ELRA, Valletta, Malta (May 2010)

[7] `http://code.google.com/javadevtools/wbpro/`