Interactive Composition of WSMO-based Semantic Web Services in IRS-III

Denilson Sell[1,2], Farshad Hakimpour[2], John Domingue[2], Enrico Motta[2] and
Roberto C. S. Pacheco[1]

[1]Stela Group and INE, Universidade Federal de Santa Catarina, Brazil
{denilson, pacheco} @stela.ufsc.br
[2]Knowledge Media Institute, The Open University, Milton Keynes, UK
{d.sell, f.hakimpour, j.b.domingue, e.motta} @open.ac.uk

Abstract. The discovery and integration of services in a composition are challenging tasks due to the lack of semantic in the Web services' description. WSMO community is working on developing ontologies and infrastructures to support Semantic Web Services. In this paper, we present a tool that takes into account WSMO descriptions to support a user-guided, interactive composition approach whereby Web services are discovered and recommended to the users according to the composition context. The generated composition is orchestrated in IRS-III by our Java API for dataflow orchestration.

1 Introduction

Research on Web services composition is gaining a considerable attention motivated by the need to support business interoperation and re-use or extension of available services. Challenges related to the Web service composition include constant changes in the business rules, high diversity and heterogeneity of Web services and the ad-hoc character of each composition.

Semantic Web technology can support this complex task, whereby semantic descriptions associated with each Web service can be used to filter and match the services according to the users needs. In particular, IRS-III following the WSMO framework [6], provides at the semantic level a distinction between goals (i.e. abstract definition of tasks to be accomplished) and Web services (i.e. description of services that can achieve a goal) and as a result support capability-driven service matching and invocation [1]. Moreover, the clean distinction between goals and Web services in IRS-III enables the specification of flexible n:m mapping between problems and methods and a dynamic, knowledge-based service selection.

According to [2], the problem of composing Web service can be reduced to three fundamental problems: 1) to prepare a plan dividing a complex task in sub-tasks; 2) discover Web services that achieve the sub-tasks identified in the plan; and 3) monitor and manage the execution and the interaction with the discovered Web Services. The full automation of the Web service composition is still the objective of many ongoing research activities [3], but supporting the user in the definition of the compo-

sition process can achieve accomplishing this objective in a semi-automatic fashion, taking into account user's non-functional expectations on a service composition.

In this paper, we introduce a graphical tool developed in Java that supports users on the definition of dynamic compositions in IRS-III by recommending goals according to the context at each step of a composition. The generated composition is performed by our Java API for orchestration. Our approach is similar to those described in [3], [4] and [5] in the sense that human holds the control of the definition of the composition, but laborious work such as discovery of services according to the users needs is assumed by the machine. However, our approach introduces additional features such as dynamic invocation of Web services in the orchestration, control operator and mediation.

2   Defining a Composition

The Fig. 1 depicts the composition tool and some of its functionalities. The tool guides users in a step-by-step composition process by selecting goals, mediators and control flow operators. The composition starts with the selection of the first goal, when the user receives a list containing all the goals defined in the IRS-III Server. The user can select a goal scrolling the list or use the discovery functionality to search for goals by defining some search criteria, as illustrated in the Fig. 2.
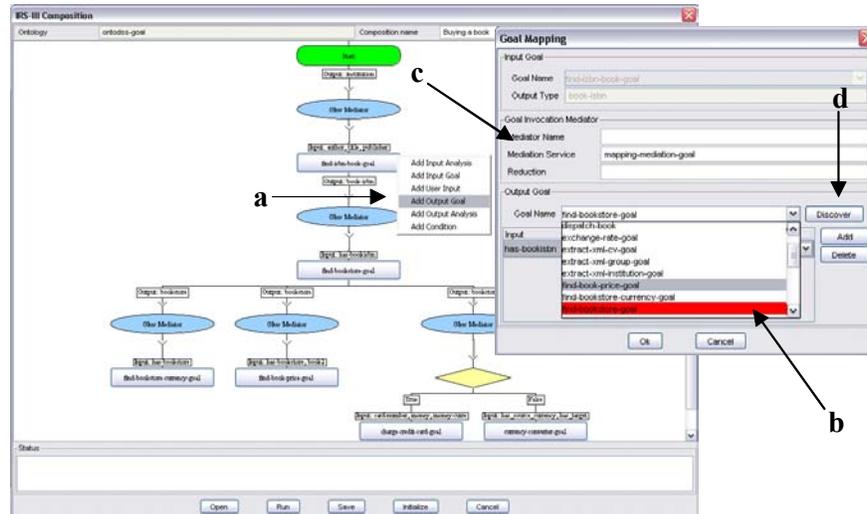
Fig. 1. An example of composition defined in our tool. Users right click over a component and select the desired action (a). In each step of the composition, users receive recommendations of services according to the automatic match of inputs and outputs of goals (b). Users also can define mediators (c) or call the discovery functionality (d).

In the subsequent steps, users can define whether they want to add goals that will receive the result or feed input to the previously selected goals. Each goal can have

more than one feeding source, for instance, a goal that have three inputs can have one input fed by the main flow of the composition and the remaining inputs fed by other goals. Users can also define the values for the inputs of the selected goals in design or orchestration time. Finally, users can add If-Then-Else control operators to the composition. This interactive process is supported by the tool, which in each step recommends goals by matching the inputs and outputs of the goals that were previously selected considering also the subsumption of the input and output types

One important characteristic of our approach is that the tool enables users to select mediators to map and perform transformations between goals. Those mediators (i.e. WSMO Mediators [6]) can solve mismatches between different parties in the data, protocol and process levels. In addition, users can select a Goal invocation mediator (GInv Mediator) that can bind and handle any other transformation required between the inputs and outputs of goals. The GInv mediator is not part of the WSMO specification but specific added to IRS-III to support flexible mappings in our composition model (see [1] for a complete description of the extensions implemented in IRS-III).

The adoption of mediators gives more flexibility to users, since it is inevitable to select services defined and implemented by different parties while building a composition (in fact, this is a basic requirement to support business interoperation). Therefore, we do not restrict the list of goals that the user can select in each step of the composition, allowing users to define mediators that could perform required transformations between goals.

Fig. 2. The Goal Discovery functionality. Users can define search criteria using a logical operator and identifying properties and correspondent values. The search criteria is translated to an OCML expression, processed against the IRS-III Server and presented to the user.

3   Orchestration of Composition

Once a composite service has been defined, the composition tool instantiate the workflow using our Java API for orchestration. In this process, the tool instantiates the service components and control operators defined in the composition according to their data dependencies using the constructors defined in our API for orchestration. The API offers necessary features to build, validate and write a composite service to

IRS Server, as well as, loading a composition from the server and editing it. The saved descriptions can be executed by the orchestration engine included in the API.

The API offers three categories of components to support compositions, namely service components, control components and mediators. A service component is actually a wrapper that keeps the necessary information about the goal to be achieved and its binding mediators. The control components provide the capability to define the control flow through the If-Then-Else operator. The mediator components bind the service components and point to WSMO mediators described in IRS-III Server for any data transformation required between service components.

The order of the execution will depends on the data provided to a service component at the execution time and it will not be defined at the design time. A service starts to execute when the necessary data is provided for its inputs. For a stateless service, that means, if all inputs to the service are provided it will be executed. The necessary means to define mediators are provided just as described above.

During the orchestration, the user is requested to enter values to feed input to the goals where the inputs were not specified in design time and that are not fed by other goals. The orchestration API relies on the IRS-III Server to achieve each goal defined in the composition, which in turn, dynamically discovers the most appropriated Web services that should be invoked according to their applicability conditions. Users can monitor the status of the orchestration by examining the status bar provided in the composition tool.

Acknowledgement

References

1. Domingue, J., Cabral, L., Hakimpour, F., Sell, D. and Motta, E. IRS-III: A Platform and Infrastructure for Creating WSMO-based Semantic Web Services. In: Proceedings of the Workshop on WSMO Implementations (WIW 2004) Frankfurt, Germany, September 29-30 (2004).
2. Sycara, K., Paolucci, M., Ankolekar, A.; Srinivasan, N.: Automated Discovery, Interaction and Composition of Semantic Web Services. In: Journal of Web Semantics, Vol. 1, Issue 1 (2003).
3. Sirin, E., Parsia, B., Hendler, J. Filtering and selecting semantic Web services with interactive composition techniques. In: IEEE Intelligent Systems, Vol. 19, Issue 4 (2004) 42-49.
4. Kim, J., Spraragen, M., Gil, Y. An Intelligent Assistant for Interactive Workflow Composition. In: Proceedings of the International Conference on Intelligent User Interfaces (IUI-2004) Madeira, Portugal, 2004.
5. Ponnekanti, S. R., Fox, A. SWORD: A developer toolkit for web service composition. In: The Eleventh World Wide Web Conference, Honolulu, HI, USA, (2002).
6. WSMO Web Service Modeling Ontology – Standard, http://www.wsmo.org/2004/d2/