# Semantic web service composition in IRS-III: The structured approach

## Conference or Workshop Item

# Semantic Web Service Composition in IRS-III: The Structured Approach

Farshad Hakimpour[1], Denilson Sell[1,2], Liliana Cabral[1], John Domingue[1] and Enrico Motta[1]
[1]Knowledge Media Institute, The Open University, Milton Keynes, UK
[2]Stela Group, Universidade Federal de Santa Catarina, Brazil
{f.hakimpour, j.b.domingue, e.motta}@open.ac.uk, denilson@stela.ufsc.br

## Abstract

*Semantic Web Services facilitate activities including automatic discovery and composition of Web Services. Research initiatives such as WSMO have been developing specifications for this technology. This paper describes a model for composition of Web services. The proposed model complements the WSMO orchestration in IRS-III, a framework for Semantic Web Services based on WSMO specification. We present a tool based on the above model that supports a user-guided interactive composition approach, by recommending component Web services according to the composition context.*

## 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. 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 (Internet Reasoning System [2]) following the WSMO framework [7], 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.

OWL-S [1] and WSMO [7] are two prominent initiatives that address the service composition in the domain of Semantic Web services. This paper describes a model for Web service composition developed in the context of the IRS-III [2]. We present also a tool for semi-automatic composition of Web services developed on top of this model.

The Web Service Modeling Ontology (WSMO) [7] is a formal ontology for describing the various aspects related to Semantic Web Services. The main components of WSMO are Goals, Web services, Ontologies and Mediators. Goals represent the types of objectives that users would like to achieve, describing the state of the desired information space and of the world after the execution of a given Web service. WSMO Web services specifications describe the functional behavior of an actual Web service, including its capabilities and interfaces. The interface description contains two closely related notions of choreography and orchestration. Choreography describes information required to interact with a Web service. Orchestration, which is the focus of this work, can contain information describing a composite Web service. Orchestration provides the necessary details for the execution of all the component services and may be a proprietary item for the provider –i.e. not accessible to others. Ontologies support the semantic interoperability among the three other components. Mediators specify interoperability mechanisms and link the three components described above.

IRS-III [2] is a framework and implemented infrastructure for Semantic Web Services that implements the WSMO descriptions. IRS-III automatically transforms programming code into a Web service and supports capability-driven service discovery and invocation. In addition, any service published on IRS-III automatically appears as a standard Web service to other Web service infrastructures.

## 2. Composition Modeling

In this section, we show a simplified scenario for a service composition. We describe how we model service composition in a knowledge representation language (in OCML [5]) and how one can compose Web services in IRS-III by this model.

Figure 1 shows a composition tree for composition of five services in a virtual bookshop. FindISBN receives the book information and returns an ISBN for the book. FindBookstore finds a bookstore that has the book in stock. It produces the bookstore name and the book price. Finally, ChargCard charges the card and DispatchBook arranges sending the book from the appropriate bookshop to the clients address. Both of


1530-1354/05 $20.00 © 2005 IEEE

the later components send an acknowledgement message to their output.

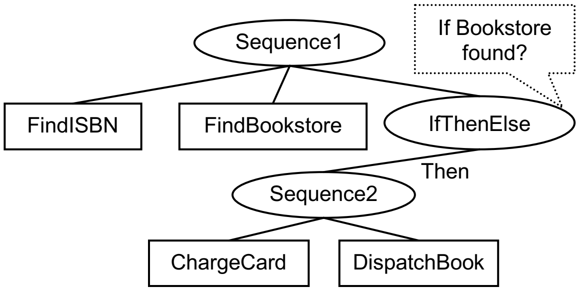


**Figure 1.** An example of a composition tree for buying a book in a virtual bookstore.

The structured composition in IRS-III is defined as an extension of WSMO *Orchestration* –i.e. it is a subclass of WSMO *Orchestration* class in [7]. The composition model is made of two types of components, namely control components and service components. Figure 2 shows the hierarchy of the composition component types. The control components provide the capability to define the control flow of the composition and they are of different types: sequence, concurrent, while and if-then-else. Figure 1 illustrates control components by ovals and service components by boxes. Listing 1 shows the definition of the root component for the example in Figure 1.

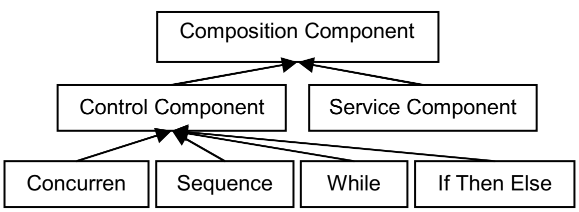


**Figure 2.** Hierarchy of composition component types in the model.

A service component is actually a wrapper that keeps the necessary information about the data bindings for an invocable (i.e. Goal or Web service). The example in Listing 3 shows how a service component is defined. Service components wrap exactly one invocable description. All data bindings are defined by means of mediators. That is, not only we can map data between component services, but also, perform other processes such as conversions and calculations on data between them, by means of mediators for that purposes. In the service component in Listing 2, we can see three internal bindings to other service components and one binding to the output of the composition. One of the three internal bindings is to the IfThenElse control component. Necessary data values are bound to the conditional control components, as they need the appropriate input for evaluating their conditions.

**Listing 1.** An example of a structured orchestration and its root component from Figure 1.

```
book-selling-orchestration
                  (structured-orchestration)
  has-root-component :value sequence1

sequence1 (sequence)
  has-components
     :value (find-isbn-SC
             find-bookstore-SC
             if-bookstore-found)
```

Listing 3 shows one of the bindings and its mediator description for the service described in Listing 2. The binding specifies that the mediator `mapping-mediation01` should be applied to the output of the service component and produce required input for `service-component-find-bookstore`.

**Listing 2.** Description of a service component in the "sequence" described in Listing 1.

```
find-bookstore-SC (service-component)
  has-invocable-description
     :value find-isbn-goal
  has-internal-bindings
     :value (binding-to-dispatch-book
             binding-to-charge-card
             binding-to-if-bookstore-found)
  has-binding-to-composition-output
     :value output-binding2
```

Use of mediators is an important feature of IRS-III. IRS-III supports several types of mediators. The only type of mediators, that is specific to compositions, is *Goal Invocation Mediator* (GInv). GInv Mediator inherits source, target and mediation-service roles from super class Mediator described in IRS-III based on WSMO description -see [7]. The mediator in Listing 3 shows that the output parameters taken from FindBookstore (i.e. book-price) are passed to the mediation service and then provided as input for ChargeCard (i.e. cost).

**Listing 3.** An example of a binding and its mediator for the service component in Listing 2.

```
binding-2-charge-card (internal-binding)
  uses-mediator :value mapping-mediation01
  to-component-service :value charge-card-SC

mapping-mediation01 (GInvMediator)
  has-source :value find-bookstore-goal
  has-target :value charge-card-goal
  has-mediation-service
     :value mapping-mediation-goal
```

## 3. Semi-Automatic Composition Tool

In this section, we introduce a graphical tool based on our composition model. This tool supports users on the definition of dynamic compositions in IRS-III by recommending Goals according to the context at each step of designing the composition. The full automation of the composition process is still the objective of ongoing research activities. This objective can be achieved in a semi-automatic fashion by supporting the user during the process of designing the composition [6]. We chose an approach similar to those described in [4] and [6] in the sense that human holds the control of the definition of the composition, but laborious work such as discovery and invocation of services according to the abstract representation of users requirements is assumed by the machine. However, our approach introduces features such as dynamic invocation of Web services, control operator and mediation.

The Figure 3 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, using a logical operator and identifying properties and correspondent values. The search criteria is translated to an OCML expression and processed against the IRS-III Server.

Using our tool, users can add Goals. Each Goal can receive input from or provide output to other existing component Goals. Goals can receive input from more than one source. For instance, a Goal that have three inputs can have one input entered from the main input to the composition and the remaining inputs from other component Goals. Users can define the values for the inputs of the selected Goals in either design or orchestration time. Finally, users can add if-then-else control operators to the composition. The 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 tool is that it enables users to select mediators to handle heterogeneities between Goals. Mediators can solve mismatches between different parties in the data, protocol and process levels. The GInv mediator presented in the last sections is an extension of WSMO mediator specifically added to IRS-III to support flexible mappings in our composition model.

We consider mediators a basic requirement to support business interoperations. 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. Once a composite service is defined, the composition tool instantiates the workflow using orchestration engine. During the orchestration, the user should enter required values for inputs to Goals. Inputs are required if their values are not specified in design time or they are not provided by other Goals. The orchestration reports to the users the transformations performed, conditions satisfied and values that should be entered to complete the orchestration. Finally, the result of the composition is presented to the user in a separated dialog.

## 4. Conclusion and Related Work

Web service composition is a new but essential issue to enhance B2B e-commerce over the Internet. This paper discusses a model for Web service composition in IRS-III. We present an approach to describe a composition by extending the existing WSMO ontology and describe different aspects of it. As WSMO is developing its *orchestration* model [8], it may adopt any approaches. Furthermore, we built a composition tool based on the model. This tool facilitates building compositions and it is a step towards an automatic composition tool.

In our approach, the composition designer breaks a Goal to sub-goals that matches to one or more Web services. An advantage of our composition approach is the use of Goals, while other initiatives such as BPEL4WS [3] compose only Web services. This feature provides a certain level of dynamism for compositions, that is, suitable Web services are discovered at the execution time. Another advantage of our tool is the use of mediators. All data bindings in our tool are using mediators which makes the visualization of the transformations in a composition more comprehendible, as compared to the approach used by OWL-S [1] and BPEL4WS [3], where no distinction between mediators and Web services is made. While the full automation of the composition is still subject of ongoing research, we provide the necessary means for users to define their expectations on a service composition through our composition tool. The composition tool suggests Goals according to a composition context and supports definition of mediators and control operators in an interactive composition process. Our composition model and tool are compliant with WSMO specifications.

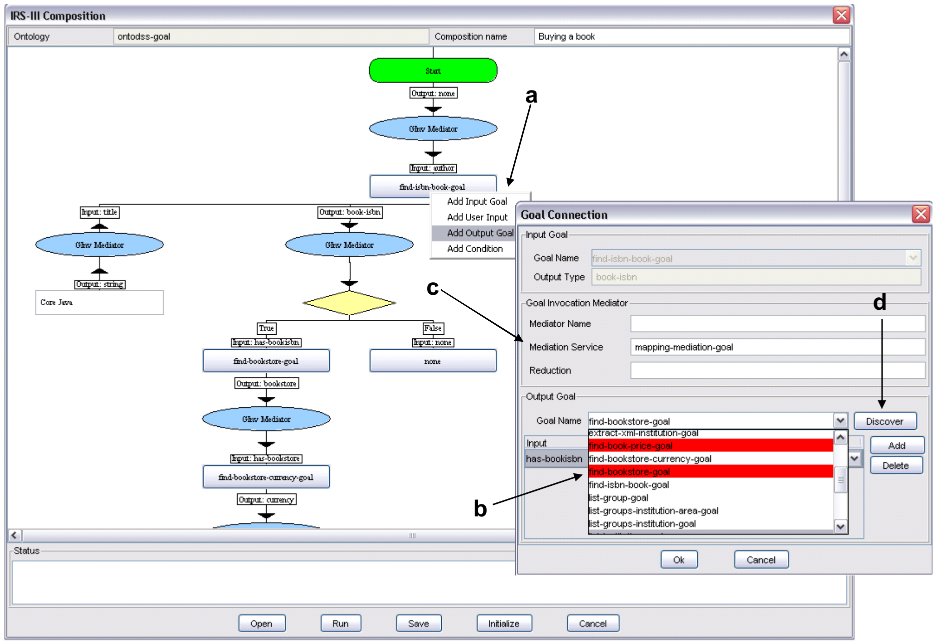


**Figure 3.** Illustration of the definition of the buying book scenario in the composition tool. Users interactively define a composition (a) receiving recommendations according to the automatic match of inputs and outputs of Goals (b). Users also can define mediators (c) or call the discovery feature (d).

Our composition tool is a step towards an automatic composition tool. The composition tool suggests Goals according to a composition context in an interactive process. CAT [4] uses a similar approach and integrates planning techniques to track relations among the composition components. However, CAT does not support the use of mediators as well as control components (similar to [6]). The OWL-S composer [6] supports users in the composition by narrowing the list of Web services based on the match of their inputs and outputs. We adopted a more flexible process, by recommending Goals through the match of their properties but allowing users to define mediators between Goals that do not match.

As for future work, we are considering to enhance our composition model by error handling and compensation. We are also investigating semantic aspects related to the automatic composition following an approach based on parametric design and incorporating planning techniques to our composition tool to support partial automation of the composition process.

## Acknowledgement

This work is supported by the AKT (Advanced Knowledge Technologies) project, the DIP (Data, Information and Process Integration) project and the CNPq, Brazil.

## References

[1] DAML Services Coalition, “Web Service Description for the Semantic Web”, Proc. of the First Int’l. Semantic Web Conf. (ISWC), www.daml.org/services/owl-s/ISWC2002-DAMLS.pdf, 2002.
[2] J. Domingue, , “IRS-III: A Platform and Infrastructure for Creating WSMO-based Semantic Web Services”, Proc. of the Workshop on WSMO Implementations, Germany, 2004.
[3] IBM Corporation, Business Process Execution Language for Web Services, www-128.ibm.com/developerworks/library/ws-bpel/, 2003.
[4] J. Kim, M. Spraragen, Y. Gil, “An Intelligent Assistant for Interactive Workflow Composition”, Proc. of the Int’l Conf. on Intelligent User Interfaces, Portugal, 2004.
[5] E. Motta, “An Overview of the OCML Modelling Language”, the 8th Workshop on Knowledge Engineering Methods and Languages, 1998.
[6] E. Sirin, B. Parsia and J. Hendler, “Filtering and selecting semantic Web services with interactive composition techniques”. IEEE Intelligent Systems, 19(4), 2004, pp. 42-49.
[7] WSMO, “Web Service Modeling Ontology–Standard”, http://www.wsmo.org/2004/d2/, 2004.
[8] WSMO, “Orchestration in WSMO”, http://www.wsmo.org/2004/d15/, 2004.