Towards a choreography for IRS-III

Conference or Workshop Item

Towards a Choreography for IRS-III

S. Galizia[1,2] and J. Domingue[2]

[1]Department of Mathematics, Università della Calabria, Rende (CS), Italy
galizia@mat.unical.it
[2]Knowledge Media Institute, The Open University, Milton Keynes, UK
J.B.Domingue@open.ac.uk

Abstract. In this paper we describe our ongoing work in developing a choreography for IRS-III. IRS-III is a framework and platform for developing WSMO based semantic web services. Our choreography framework is based on the KADS system-user co-operation model and distinguishes between the direction of messages within a conversation and which actor has the initiative. The implementation of the framework is based on message pattern handlers which are triggered whenever an incoming message satisfies pre-defined constraints. Our framework is explained through an extensive example.

1 Introduction

Web services promise to turn the web of static documents into a vast library of interoperable running computer programs and as such have attracted considerable interest, both from industry and academia. For example, IDC [IDC, 2002] predicts that the Web Services market, valued at $416 million in 2002, will be worth $21 billion by 2007. Current web service technologies are, however, relatively inflexible and ongoing research is investigating how semantic web technology can alleviate this.

Existing web service technologies are based on a manual approach to their creation, maintenance and management. At the centre of the conceptual architecture is a registry which stores descriptions of published web services. Clients query the registry to obtain relevant details and then interact directly with the deployed service. The descriptions, represented in XML based description languages such as WSDL [WSDL, 2001] and UDDI [UDDI, 2003], mostly focus on the specification of the input and output data types and the access details. These specifications are obviously not powerful enough to support automatic discovery, mediation and composition of web services. A software agent cannot find out what a web service actually does, by reasoning about a WSDL specification. Analogously the same agent cannot locate the appropriate service in a UDDI registry, given a specification of a target functionality. As a result, existing web service infrastructures by and large support a manual approach to web service management: only manual discovery can be supported and only 'static', manually configured web applications are possible.

The above issues are being addressed by ongoing work in the area of semantic web services [OWL-S, 2002; Fensel and Bussler, 2002]. The overall approach is that by augmenting web services with rich formal descriptions of their competence many aspects of their management will become automatic. Specifically, web service location,

composition and mediation can become dynamic, with software agents able to reason about the functionalities provided by different web services, to locate the best ones for solving a particular problem and to automatically compose the relevant web services to build applications dynamically.

Recently a number of initiatives based on the WSMF framework [Fensel and Bussler, 2002] have started. WSMF is an extension of the UPML framework [Fensel and Motta, 2001] revised to integrate fully with web services and to support e-commerce. WSMF is centered on two complementary principles: a strong de-coupling of the various components that realize an ecommerce application; and a strong mediation service enabling Web services to communicate in a scalable manner. Mediation is applied at several levels: mediation of data structures; mediation of business logics; mediation of message exchange protocols; and mediation of dynamic service invocation. Three related initiatives associated with WSMF have recently begun. These are WSMO [WSMO, 2004] which will develop an ontology for WSMF, WSML [WSML, 2004] which will develop a formal language for representing the WSMO based descriptions and WSMX [WSMX, 2004] which will develop a reference implementation.

In this paper we describe our ongoing work on choreography in IRS-III (Internet Reasoning Service) [Domingue et al., 2004] a framework and implemented infrastructure which supports the creation of semantic web services according to the WSMO ontology.

The paper is organized as follows: in the following section we give a brief overview of choreography followed by an overview of IRS-III. We then describe our approach to choreography IRS-III in detail. The final section of the paper contains a summary.

2 Choreography

Web service choreography deals with all the interactions between web services and their users [W3C a, 2004], [Dijkman and Dumas, 2004] where the users in some cases can be automated services. Additionally a choreography globally describes the behaviour observable from an external point of view.

W3C presents a notion of choreography at several levels of abstraction [W3C b, 2004], [W3C c, 2004], [W3C d, 2004]. An abstract choreography definition will contain descriptions of the data types used and the conditions under which a given message is sent. The advantage of an abstract choreography is that it is relatively easy to reuse. However, more detail is often required. For this reason W3C defines other two types of choreography: portable and concrete, that extend the abstract choreography definition.

A portable choreography includes descriptions of the physical structure of the information exchanged and of the technologies used. A concrete choreography extends a portable description including destination URLs, and specific rules, such as information about digital certificates to be used for securing messages.

When creating a choreography the level of abstraction chosen would depend on the current context (e.g. the type of organization it was designed for) and the level of re-

usability and extendibility required. Actually the abstraction levels just illustrated represent a logical division, it is possible to define a choreographies that contain a description of the message physical structure, as required in a portable choreography, but without information about the technologies used. We define a choreography with a high level of concreteness, as we specify the message exchange pattern that includes the destination URLs, the model used for the information flow representation, but it does not include still information about digital certificates. Our choreography description may be considered as portable/concrete level. Furthermore, W3C choreographies are described from the viewpoint of a group of web services [W3C c, 2004], specifying a global peer-to-peer model of cross-enterprise interactions and their semantics. Within IRS-III (and WSMO) our viewpoint, instead, is at a local single web service level. That is we describe how one web service talks to one other.

3 Overview of IRS-III

The IRS project has the overall aim of supporting the automated or semi-automated construction of semantically enhanced systems over the internet. IRS-I supported the creation of knowledge intensive systems structured according to the UPML framework and IRS-II [Motta et al., 2003] integrated the UPML framework with web service technologies. Within IRS-III we have now incorporated and extended the WSMO ontology.

IRS-III has four main classes of features which distinguish it from other work on semantic web services. Firstly, it supports *one-click publishing* of 'standard' program code. In other words, it automatically transforms programming code (currently we support Java and Lisp environments) into a web service, by automatically creating an appropriate wrapper. Hence, it is very easy to make existing standalone software available on the net, as web services.

Secondly, by extending the WSMO goal and web service concepts users of IRS-III directly invoke web services via goals that is IRS-III supports *capability-driven* service invocation.

Thirdly, IRS-III is programmable. IRS-III users can substitute their own semantic web services for some of the main IRS-III components.

Finally, IRS-III services are web service compatible – standard web services can be trivially published through the IRS-III and any IRS-III service automatically appears as a standard web service to other web service infrastructures.

4 Choreography in IRS-III

4.1 Message Exchange Events

According to the WSMO standard model [WSMO, 2004], a web service *interface* description is composed of *choreography* and *orchestration*.

Choreography describes how one web service interacts with another web service and orchestration specifies how a complex web service calls subordinate web services. The detailed description of interactions are based on Message Exchange Patterns (MEPs) [W3C a, 2004].

The IRS-III choreography framework classifies messages according to two dimensions following the system-user cooperation model proposed in KADS [Greef and Breuker, 1992], namely,

- The initiative in the communication, and
- The direction of the communication.

The initiative expresses which actor, either a web service or a user, is responsible for starting the communication, while the direction represents the communication route, which can be from the system to the user or vice-versa.

In a message exchange pattern we can specify two main parameters: a sender and a receiver. A message exchange *event* is a kind of transfer task, an elementary action executed by an actor during a conversation.

From the IRS-III perspective we consider four kinds of events: *receive, provide, obtain* and *present*. The overall framework shown in the figure below.

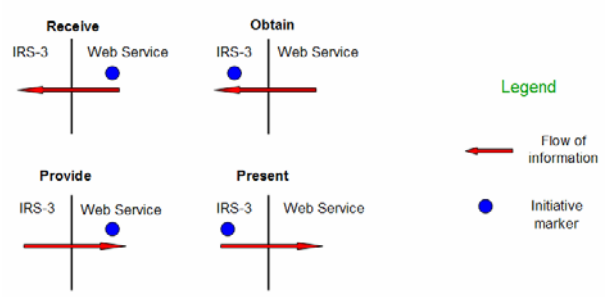

Fig. 1. Message exchange events

The message transfer events depicted above are obtained by permuting the possible values for direction and initiative in the communication; the arrows (coloured red) show the communication direction and the balls (blue) represent which actor, involved in conversation, has the initiative.

Initiative is associated with actors who in some sense have control of the conversation. For example, only actors with initiative are allowed to start a conversation. Also actors with initiative are allowed to *negotiate* a conversation, where "negotiate" means to alter the topic of a communication or to update data sent within previous messages. Within IRS-III initiative state (as part of conversation state) is held by an instance of a choreography class.

Receive is the first exchange message event; when this event occurs the external web service has the initiative and the IRS-III server receives a message sent by the web service. The second type of event, *Obtain*, represents information transferred from the web service to the server; IRS-III obtains a message and holds the initiative. *Provide* is the third type of event; in this case the web service has the initiative and it

receives a message provided by IRS-III. The last event type shows the situation in which IRS-III has the initiative and sends a message to the web service.

4.2 An Example

Let us give an example of message exchanges between a web service and IRS-III focusing on the different events happening.

We want to describe the behaviour (visible from the outside) of IRS-III during a conversation with a Travel Agency System that deals with travel reservations.

This travel agent service has the following capabilities:

- Replying to the requests of the traveller,
- Proposing the best itinerary according to some criteria (e.g. cheapest, quickest),
- Reserving tickets,
- Cancelling reservations, and
- Sending statements.

Imagine that a user is planning a trip from Milton Keynes (UK) to southern Italy for a holiday. The traveller does not have a specific preference regarding the plane company or the itinerary to follow, but she is only concerned about finding the cheapest ticket. As there are no airports in Milton Keynes it is necessary to consult coach and/or train line services to plan the whole itinerary.

The user sends her request to IRS-III which using predefined WSMO goal and web service descriptions invokes airline, train and coach travel related web services.

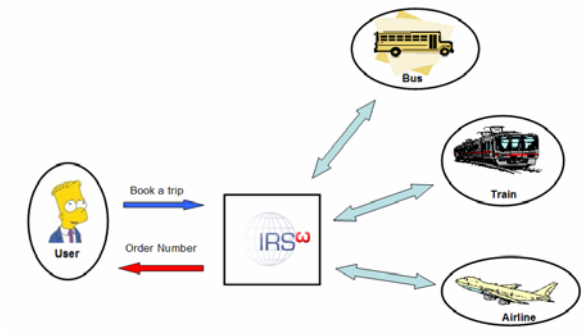

Fig. 2. The interactions involved in booking a trip via bus, train and airline booking services with IRS-III.

The final solution that IRS-III presents to the user will be composed of the different contributions from the given pool of heterogeneous web services. In this example, however, we analyze only the conversation between IRS-III and the airline web service.

We will now describe the conversation shown in figure 3. IRS-III asks the airline service to book the cheapest trip from Milton Keynes to a city in southern Italy. Note that IRS-III starts the conversation with the airline service, thus IRS-III has the initiative. The airline service returns the details of a two legged flight Luton to Amsterdam

followed by Amsterdam to Naples. The travel ontology used as a basis for the WSMO descriptions contain flight evaluation criteria. Running the criteria results in a low convenience rating for the proposed flight. As the IRS-III has the initiative it is able to start a negotiation process. IRS-III sends a new request to the airline service, changing the journey parameters, in particular, giving a higher priority to the criteria related to the number of legs in the journey. The web service then returns a direct flight from Luton to Reggio in Calabria.

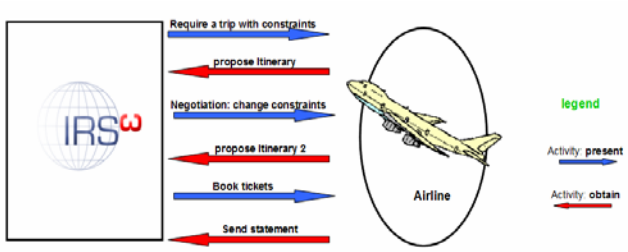

Fig. 3. An example of an IRS-III/web service conversation

We can see in the above example that only messages of type *obtain* and *present* were used - in fact these are the only possible types when the IRS-III has the initiative. Another aspect of the example we should emphasise is that only actors who currently hold the initiative can negotiate the information. In our example IRS-III was able to change the journey constraints when the proposed trip failed to satisfy evaluation criteria.

4.3 Movement of the initiative

Within web service conversations there will be occasions when it is necessary to move the initiative. Our framework supports the transfer of initiative through the use of an initiative marker.

Fig. 4. Initiative movement events

Figure 4 displays the only two possible events regarding the movement of initiative from the IRS-III point of view. The initiative flow is indicated by an arrow (red), the ball (blue) indicates which actor has the initiative. The two types of events are:

- *Present-initiative* - IRS-III sends the initiative to the web service. IRS-III has the initiative while the send-event is performing, if the action does not generate errors, the initiative is transferred to the web service.
- *Obtain-initiative* – the initiative is transferred from a web service to IRS-III.

The initiative can also be transferred within the four other 'standard' message types.

4.4 Structure of Messages

The table below shows the structure of choreography messages. We elected not adopt all of the Fipa ACL specification [Fipa a, 2001; Fipa b, 2002] because it did not completely conform to our requirements. In our conversations one of the actors is a WSMO web service description sitting within an IRS-III server. To identify the actor we therefore require an identifier for the server (a URL) and an identifier for the web service description, namely, an ontology and a web service class name.

Obviously, each web service will have its own message structure. Internally, within a web service choreography IRS-III will transform, as far as possible, the messages received from web services into the IRS-III message structure.

Within IRS-III deployed web services are attached to WSMO web service descriptions through a publishing process (see section 3). During the publishing process an identifier is generated for the web service which is used as the value for *id-sender* and *id-receiver*. The value for *sender-WSMO-obj* and *receiver-WSMO-obj* is simply the name of the WSMO web service class. Correspondingly the value for *sender-WSMO-ontology* and *receiver-WSMO-ontology* is the name of the home ontology for the WSMO web service class.

Parameters	Description	Type
Id-message	Message identifier	String
Sender-WSMO-ontology	Home ontology of sender WSMO object	WSMO ontology
Receiver-WSMO-ontology	Home ontology of receiver WSMO object	WSMO ontology
Sender-WSMO-Obj	Sender identifier of WSMO object	WSMO web service concept
Receiver -WSMO-Obj	Receiver identifier of WSMO object	WSMO web service concept
Id-sender	Identifier of the sender	URL
Id-receiver	Identifier of the receiver	URL
Event-type	Kind of event (e.g. Receive, obtain)	Event type
Initiative	Who has the initiative	WS/IRS-III (actor)
Content	Denotes the content of message	String
Time	Records the time when the action is executed	Date/time

Tab. 1. Message Structure

The other parameters of a message are relatively straightforward. Every message is recognized by a unique identifier *id-message.* The parameter *event-type* indicates the type of message sent (receive, provide, obtain, present, present-initiative or obtain-initiative). The *initiative* parameter specifies who has the initiative and the *content* parameter contains the actual data exchanged. For initiative transfer messages (present-initiative and obtain-initiative) the initiative marker is passed within the content parameter. The final parameter *time* contains a timestamp for the event.

4.5 Representation in IRS-III

When a web service communicates with IRS-III, it instantiates a message exchange pattern as described in the choreography. In this subsection we show how an example pattern handler is represented in IRS-III. Following from the previous example, we represent in OCML the pattern handler for managing communication between IRS-III and an Easy-Jet flight booking web service.

```
(def-wsmo-pattern-handler easy-jet-airline-pattern-handler
  (message ?id ?easy-jet-ontology
      easy-jet-web-service-ontology
      easy-jet-airline easy-jet-flight-booking-service
      ?id-message-sender irs-iii obtain-message-event
      ?current-initiative-holder
      ?content ?time)
  (flight-passenger ?content ?person)
  (flight-passenger ?content ?departure-location)
  (flight-arrival-location ?content ?arrival-location)
  (flight-departure-time ?content ?departure-time)
  (has-travel-plan ?person ?travel-plan)
  (matches-travel-plan ?travel-plan ?departure-time
       ?departure-location ?arrival-location)
  (has-travel-plan-message ?travel-plan
       ?travel-plan-message)
   then
  (send-message
    (create-message-id)
    easy-jet-web-service-ontology
    ?easy-jet-ontology easy-jet-flight-booking-service
    easy-jet-airline
    irs-iii ?id-message-sender present-message-event
    ?current-initiative-holder
    ?travel-plan-message (current-time)))
```

Fig. 5. An example of an IRS III choreography pattern handler

The above pattern handler is invoked when flight details for a passenger are received which match the passenger's stored travel plan. When invoked a message associated with the travel plan is retrieved and sent. The definition of the pattern follows the message structure defined in the previous section. In fact the arguments for the `message` and `send-message` relation are the same as table 1. In this specific scenario, the slot *id-receiver* would have the URL of the IRS-III server and *initiative* would have as its value an instance of the web service `easy-jet-flight-booking-service`'s associated choreography class. The value for *event-type* would be `obtain-message-event`.

A collection of pattern handlers are stored within a choreography description. When a deployed web service is published against a WSMO web service description the name of the appropriate choreography concept would be given.

Summary

IRS-III is a framework and implemented infrastructure which allows WSMO based semantic web services to be created and used. One of the main aims in designing IRS-III was to make the process of semantic web service based application development as easy as possible.

In this paper we have described our ongoing work on choreography within IRS-III. Our choreography framework is based on the KADS system-user co-operation model and distinguishes between the direction of messages and which conversation actor currently has the initiative. In implementing the framework we have devised a message structure and a representation for handling conversations based on pattern handlers.

Successfully managing the choreography of heterogeneous web services, each with its own message organization, process and underlying ontology, is a non-trivial problem. We believe that our framework is step towards addressing this problem within the WSMO framework.

Acknowledgements

This work is supported by the DIP (Data, Information and Process Integration with Semantic web services) and AKT (Advanced Knowledge Technologies) projects. DIP (FP6 - 507483) is an Integrated Project funded under the European Union's IST programme. The AKT Interdisciplinary Research Collaboration (IRC), is sponsored by the UK Engineering and Physical Sciences Research Council under grant number GR/N15764/01. The AKT IRC comprises the Universities of Aberdeen, Edinburgh, Sheffield, Southampton and the Open University.

References

J. Domingue, L. Cabral, F Hakimpour, D. Sell, and E. Motta (2004) IRS-III: A Platform and Insfrastructure for Creating WSMO-based Semantic Web Services. 1st WSMO Implementation Workshop (WIW), Frankfurt, 29th and 30th September 2004, CEUR Workshop Proceeding.

D. Fensel and C. Bussler. (2002) The web service modeling framework WSMF. Electronic Commerce Research and Applications, 1(2):113–137, 2002.

D. Fensel and E. Motta. (2001) Structured Development of Problem Solving Methods, IEEE Transactions on Knowledge and Data Engineering, 13(6):9131-932.

Fipa [a]. (2001) Fipa ontology service specification. Document number XC00086D, Available from http://www.fipa.org/specs/fipa00086/XC00086D.pdf

Fipa [b]. (2002) Fipa acl message structure specification. Document number SC00061G, Available from http://www.fipa.org/specs/fipa00061/SC00061G.pdf/.

H. P. Greef and J. A. Breuker. (1992) Analysing system-user cooperation in KADS. Knowledge Acquisition, 4:89–108, 1992.

IDC. (2002) U.S. Web Services Market Analysis, 2002 IDC (http://www.internetnews.com/xSP/article.php/1579171)

E. Motta. (1999) Reusable Components for Knowledge Modelling. IOS Press, Amsterdam, The Netherlands, 1999.

E. Motta, J. Domingue, L. Cabral, and M. Gaspari. (2003) IRS-II: A framework and infrastructure for semantic web services. In 2nd International Semantic Web Conference (ISWC2003), 20-23 October 2003, pages 406–417. Sundial Resort, Sanibel Island, Florida, USA, 2003.

OWL-S. (2002) Web Service Description for the Semantic Web. In the Proceedings of The First Int'l. Semantic Web Conf. (ISWC), Sardinia (Italy) (2002).

R. Dijkman and M. Dumas. (2004) Service-oriented Design: A Multi-viewpoint Approach. CTIT Technical Report Series No. 04-09, Centre for Telematics and Information Technology, University of Twente, The Netherlands, February 2004.

UDDI. (2003) UDDI Spec Technical Committee Specification v. 3.0, http://uddi.org/pubs/uddi-v3.0.1-20031014.htm

W3C [a]. (2004) Web services architecture. W3C Working Group Note 11 February 2004, Available from http://www.w3.org/TR/2004/NOTE-ws-arch-20040211.

W3C [b]. (2004) Web services choreography requirements. W3C Working Draft 11 March 2004, Available from http://www.w3.org/TR/2004/WD-ws-chor-reqs-20040311.

W3C [c]. (2004) Web services choreography model overview. W3C Working Draft 24 March 2004, Available from http://www.w3.org/TR/2004/WD-ws-chor-model-20040324.

W3C [d]. (2004) Web services choreography description language version 1.0. W3C Working Draft 27 April 2004, Available from http://www.w3.org/TR/2004/WD-ws-cdl-10-20040427.

WSDL. (2001) Web Services Description Language (WSDL) 1.1, http://www.w3.org/TR/2001/NOTE-wsdl-20010315

WSML. (2004) Languages for WSMO, http://www.wsmo.org/2004/d16/

WSMO. (2004) Web Service Modeling Ontology – Standard, http://www.wsmo.org/2004/d2/

WSMX. (2004) Overview and Scope of WSMX, http://www.wsmo.org/2004/d13/.