SOPHIE - A CONCEPTUAL MODEL FOR A SEMANTIC CHOREOGRAPHY FRAMEWORK

Sinuhe Arroyo
Digital Enterprise Research Institute (DERI). University of Innsbruck, Austria.

Alistair Duke
BT, Next Generation Web Reasearch Group

Abstract:

Choreography models the external visible behavior of services as a set of message exchanges that realize Message Exchange Patterns (MEP). A number of language exist that try to tackle the problem. However they constitute incomplete solutions, either because they lack of support of semantics, have a tight relation to the underlying communication framework, or mix structural, behavioral and operational aspects resulting in confusing specifications. This position paper defines a conceptual model for **SOPHIE**, a choreography engine for Semantic Web Services following a Semantic Service-Oriented Architecture. The conceptual model of **SOPHIE** is especially suitable for supporting the fine grained interaction among services with different structural, behavioral or operational models. A case study in the Telecommunication sector is presented.

Key words: Choreography, Semantic Services, MEPs.

1. INTRODUCTION

This paper presents the conceptual framework of a choreography engine named **SOPHIE**. A number of approaches exist, such as (BPEL4WS [Andrews et al., 2003], WS-CDL [Kavantzas et al., 2004], WSCI [Arkin et al., 2002] or WSMO - Choreography [Roman et al., 2005]), that try to model the external visible behavior of services, so interoperation among heterogeneous interaction patterns can occur. Still none of these approaches represents a complete solution to the problem due to: lack of proper support for semantics (BPEL4WS, WSCI, WS-CDL), lack of technological independence and tight relation to the underlying communication framework (BPEL4WS, WS-CDL), lack of a clear model that separates structural, behavioral and operational aspects (BPEL4WS, WS-CDL, WSCI or WSMO-Choreography), lack of use of MEP as a central unit to solve the mismatches among message exchanges (BPEL4WS, WS-CDL, WSCI or WSMO-Choreography)

Thus, new initiatives are needed that overcome these limitations. **SOPHIE** (Semantic web services chOreograPHi servIcE) proposes a framework, which in a first cut is divided into syntactical and semantic models. The semantic model details the semantic support. The syntactical model details the syntax of the engine. The syntactic model further depicts three different complementary models, namely: structural, behavioral and operational. The structural model deals with the provision of a reusable collection of entities following different levels of abstraction that provides the basis for the description of a conceptual model. The behavioral model cares for the description of the dynamic interaction among the entities defined in the structural model. Finally, the operational model facilitates the means to allow the interoperation among different behavioral models as shown in Figure 1.

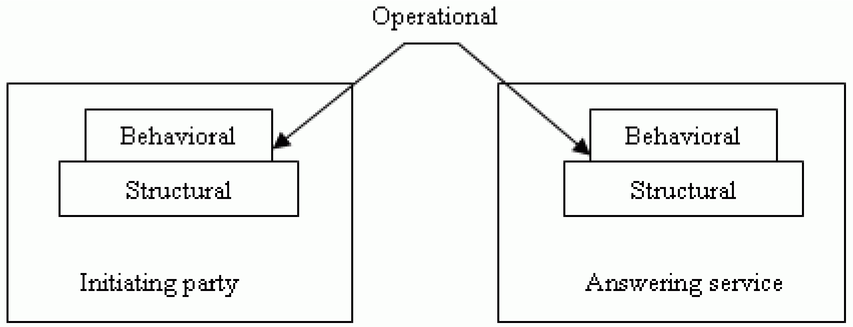

Figure 2:Relation among models

A clear separation of aspects facilitates the addition, replacement and modification of the underlying paradigms without imposing the need to redesign the overall conceptual model and affecting the remaining aspects. For example Petri nets, temporal logic or transaction logic can be used instead of Abstract State Machines (ASMs) for the behavioral model, or OWL instead of WSML for the semantic one. The work presented here defines the behavioral model as ASMs, and thus the terms used, in particular the terms choreography, state and guarded transition, follow the framework proposed in [Roman et al., 2005] and [Booch et al., 1999]. Further the ontologies of semantic model are WSMO ontologies [Roman et al., 2004]. Still, any other suitable paradigm can be easily modeled.

2. SYNTACTIC MODEL

2.1 Structural model

A *conversation* represents the logical entity that permits a set of related message exchanges among parties to be grouped together. Conversations are composed of a set of building blocks. *Elements* represent units of data that build up *documents*. Documents are complete, self-contained groups of elements that are transmitted over the wire within *messages*. Messages characterize the primitive piece of data that can be exchanged among parties. As messages are exchanged, a variety of recurrent scenarios can be played out. *Message Exchange Patterns (MEP)*, identify placeholders for messages, that allow sequence and cardinality to be modeled, defining the order in which parties send and receive messages. A set of messages sent and received among parties optionally following a MEP that account for a well defined part of a conversation, are referred to as a *message exchange*. A conversation can be thus defined as a set of message exchanges among parties with the aim of fulfilling some goal. Every conversation is carried out over a communication facility, referred to as a communication network by parties. The specification differentiates among two type of parties, *initiating parties* and *answering services*. Both parties produce and consume messages, and additionally initiating parties take care of starting the message exchange.

2.2 Behavioral model

A *choreography* describes the behavior of the answering service from the initiating party's point of view [Roman et al., 2005]. It governs the message exchanges among parties in a conversation. ASMs allow the sequences of states the choreography goes through during its lifetime to be s pecified, together with its responses to events. In this sense the main building blocks of abstract state machines are, to some extent, redrawn. A *state* is a situation during the lifetime of a choreography which satisfies some condition or waits for some *event*. Events are occurrences of a stimulus that can trigger a state transition. An action represents the atomic task of sending a message to a party. Transitions among states are defined by *guarded transitions*. Guarded transitions allow the relationship between two states, by means of *guard conditions*, to be modeled. Guard conditions are *Boolean expressions* that are evaluated when the transition is triggered. *Parts* permit the relationship between guarded transitions and message exchanges to be expressed, thus defining the message exchange in terms of state transitions. Finally, a *choreography* comprises a set of parts that define a conversation.

2.3 Operational model

The atomic building blocks that permit a number of mismatches among interacting parties to be resolved are *logic boxes*. A logic box facilitates the reorganization of the content of documents, its mapping to messages, and the order and cardinality of messages, thus enabling the interoperation among parties following different message exchange patterns. Additionally, and depending on the type of box, the differences in the vocabulary used to describe the application domain can be overcome. Currently the specification defines five different types of logic boxes, namely: *refiner box*, *merge box*, *split box*, *select box*, *add box*. Logic boxes are grouped into *logic diagrams*. Logic diagrams permit the relationship between the message exchange pattern followed by the initiating party, and the one used by the answering service to be modeled. A number of logic diagrams defining a conversation are referred as a *logic group*.

2.4 Semantic Model

Ontologies define the semantics of the engine. They provide a vocabulary that can be mediated for the understanding of interacting parties. *Domain ontologies* facilitate the general vocabulary to describe the application domain of the answering service and the initiating party. The *Choreography ontology* provides the conceptual framework and vocabulary required to describe the behavioral model. In doing so it defines and allows instantiation of the entities of the different models. They are defined as concepts of the ontology, taking part in a choreography. Thus reasoning and the generation of the operational model based on the different behaviors and structures is possible.

3. USE CASE

SOPHIE is currently being trialed as part of the DIP project B2B in Telecommunications case study, hosted by BT. **SOPHIE** has been applied to BT Wholesale's B2B Gateway which allows BT's ISP partners to integrate their Operation Support Systems with those of BT and, for example, carry out tests on BT's network as part of their broadband assurance activities. The B2B Gateway currently uses the Business Process Specifications of ebXML to represent the required choreography.

The example applies the operational model of **SOPHIE** to the broadband test interface in order to illustrate how a partner's differing choreography could be integrated. Figure 1 shows a simplified message exchange pattern for the test interface. The service provider sends a *testRequest* message to BT. The requested test is examined and then accepted or rejected (it may be rejected if, for example, a test is requested on a line not owned by BT) resulting in a *testAccept* or *testReject* message being sent to the service provider. Following an acceptance notification, the test is carried out and the *testCompleted* message is sent which contains the result of the test.

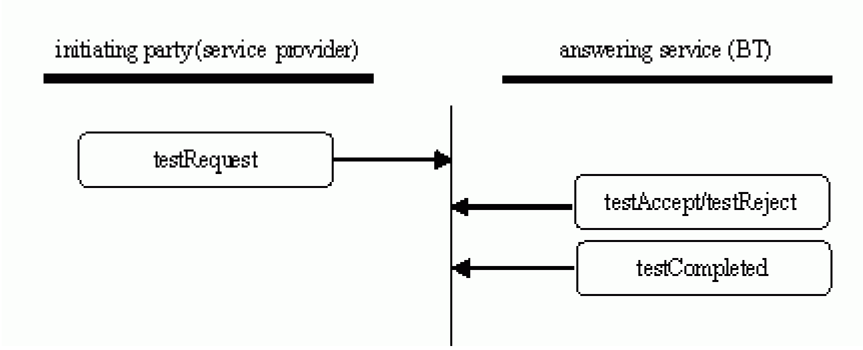

Figure 2:In-Multi-Out Message Exchange Pattern

If the service provider were to use a different MEP where it expects BT to provide a single response message (testReply) indicating whether the test was accepted or rejected and if accepted the result of the test, then the operational model of SOPHIE can be applied. As stated above, logic boxes constitute the key elements of the operational model. A logic box is represented as a tuple of the form:

logicBox = [name, URI, type, sequenceNumber&sup?, inputMessages&sup, inputMep&sup?, outputMep&sup?, ontologyMapping&sup?]*

Table 1: Tuple of a logic box

Where, type specifies the type of logic box, *sequenceNumber* labels the sequential order of execution of the logic box, *inputMessages* define the messages sent by the initiating party, and *outputMep* stand for the message exchanges patterns followed by the initiating party and the answering service respectively, and *ontologyMapping* specifies a mapping among the domain ontologies of the interacting parties.

In this example, a merge box is required. The merge box, maps the input messages sent by one party following a particular message exchange pattern into a single output message, received by the another party, following a different message exchange pattern. In this case BT's response messages to the testRequest message are merged and send to the service provider." Additionally, a merge box permits the types and names of elements, documents, messages and any other entities (as specified in the structural model) according to the ontologies of the semantic model to be matched. In this case it would allow the documents contained in the messages sent by BT to be refined to reflect those expected by the service provider.

Figure 3 shows the merge box required in this case.

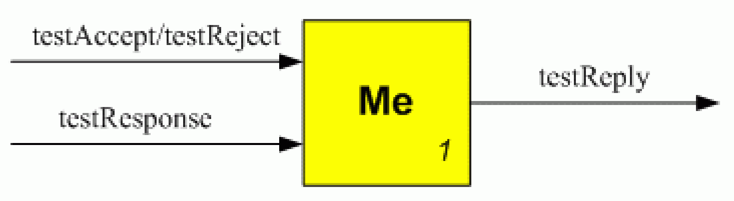

Figure 3: Merge Box for textReply message

Listing 2 shows the instance of the tuple for the merge box is:

name: *testReplyMerge*
URI: *http://www.bt.com/assuranceIntegration/logicBox/ testReplyMerge/*
type: *mergeBox*
sequenceNumber: *001*

inputMesages: *http://www.bt.com/message/testAccept/,*
http://www.bt.com/message/testReject/,
http://www.bt.com/message/testResponse/
inputMep: *http://www.bt.com/mep/outOnly/*
outputMep: *http://www.bt.com/messageExchange/inOnly/*
ontologyMapping: *http://www.bt.com/ontologyMapping/assuranceMapping/Listing/*

Table 2:Instance of tuple for a logic box

The URIs for the messages and MEPs refer to entities that have already been defined according to the structural model. Logic boxes can be combined to form logic diagrams allowing more complex transformations to take place over conversations.

From the point of view of the case study, SOPHIE provides a comprehensive way to semantically represent the choreography of the B2B Gateway's interfaces. With appropriate tool support, it can ease the integration problem and allow dynamic integration between partners to be realized. Tool support is required to automate the expression of ebXML Business Process Specifications into the ontological format of SOPHIE and to support to user in performing mappings between the choreographies of the different parties.

4. SUMMARY

Based on the work presented, the rationale is summarized as follows:

Separation of models

The definition of a conceptual framework requires to clearly differentiating the models that build it. As a result a modular framework should be designed where the underlying different paradigms of models can be readily added, extended or replaced in order to place the most suitable one depending on the target application domain.

Semantic-driven

Semantic descriptions enable the dynamic interoperability among heterogeneous parties. As new ontological languages based on different logical formalisms are developed, independence from existing and emerging formalisms should be obliged in any conceptual design. In doing so, the conceptual model is driven by the semantic description of its constituent entities, not making any constraint or assumption on the ontological language used to model such descriptions.

Mediation

Mediation allows heterogeneous parties to communicate. It originates due to the natural heterogeneity of the open environment where parties reside. Mediation by means of logic boxes facilitates an intermediate layer among parties that provides a generalized solution to resolve communication mismatches.

Technological independence

The design of a fully extensible conceptual framework should not make any assumptions about underlying technologies. In particular, the details regarding transport and communication frameworks, should be left aside, as do not help to neither characterize nor solve the problem addressed in this thesis, and are subject to change. In doing so, a conceptual framework should rely on such underlying technologies, defining a clear border, which allows keeping separated the particular communication details from the conceptual model.

Global view vs. decentralized approach

Traditional models call for centralized approaches where the interaction among parties is control from a unique point. In contrast, the nature of services is decentralized. While a decentralized approach is preferred due to its flexibility, eventually a global point of view is chosen to control the message exchange.

Separation of internal and external behavior

Choreographies deal with the external visible behavior of parties. Thus, the internal details should be clearly separated from the external ones, allowing its independent definition. Deeply in this direction, the description of collaborative process is out of the scope of this work, same as orchestration in general.

SOA-based

As a realization of the all this basic principles, especially regarding semantic-driveness and technological independence, the conceptual framework should be realized as a SOA with support for semantics.

5. REFERENCES

[Andrews et al., 2003] Andrews, T., Curbera, F., Dholakia, H., Goland, Y., Klein, J., Leymann, F., Liu, K., Roller, D., Smith, D., Thatte, S., Trickovic, I., Weerawarana, S.: Business Process Execution Language for Web Services, Available from "href="ftp://www6.software.ibm.com/software/developer/library/ws-bpel.pdf">ftp://www6.software.ibm.com/software/developer/library/ws-bpel.pdf.

[Arkin et al., 2002] Arkin, A., Askary, S., Fordin, S., Jekeli, W., Kawaguchi, K., Orchard, D., Pogliani, S., Riemer, K., Struble, S., Takacsi-Nagy, P., Trickovic, I., Zimek, S.: Web Service Choreography Interface (WSCI) 1.0, Available from "href=" http://www.w3.org/TR/wsci/"> http://www.w3.org/TR/wsci/.

[Booch et al., 1999] Booch, G., Rumbaugh, J., and Jacobs, I. The Unified Modelling Language User Guide. Addison-Wesley, 1999.

[Kavantzas et al., 2004] Kavantzas, N., Burdett, D., Ritzinger, G.,: Web Services Choreography Description Language Version 1.0, Available from "http://www.w3.org/TR/2004/WD-ws-cdl-10-20040427/">http://www.w3.org/TR/2004/WD-ws-cdl-10-20040427/.

[Roman et al., 2005] Roman, D., Scicluna, J., Feier, C., (eds.) Stollberg, M and Fensel, D.: D14v0.1. Ontology-based Choreography and Orchstration of WSMO Services. , Available from "http://www.wsmo.org/TR/d14/v0.1/"> http://www.wsmo.org/TR/d14/v0.1/.

[Roman et al., 2004] Roman, D., Lausen, H. and Keller, U. (eds): Web Service Modeling Ontology. WSMO Working Draft v0.3. , Available from "http://www.wsmo.org/2004/d2/v1.0/"> http://www.wsmo.org/2004/d2/v1.0/.

6. ACKNOWLEDGMENTS

The work is funded by the European Commission under the projects DIP, Knowledge Web, InfraWebs, SEKT and ASG; by Science Foundation Ireland under the DERI-Lion project; by the FIT-IT (Forschung, Innovation, Technologie - Informationstechnologie) under the projects RWÂ² and TSC.