Choreography of Web Services based on Natural Language Storybooks

Kurt Englmeier[1], Javier Pereira[2] , Josiane Mothe[3].
[1] LemonLabs GmbH, Germany
KurtEnglmeier@computer.org
[2]Escuela de Informática, Universidad Diego Portales,Santiago, Chile
Javier.pereira@udp.cl
[3]Université Paul Sabatier, IRIT Lab., France,
mothe@irit.fr

Abstract

Business processes usually span beyond the boundaries of single operations and many a process spans even beyond the boundaries of organizations. Web Service orchestration or choreography languages address the middle layer where atomic services (or operations) are integrated for more complex applications. They endow tech people to compose application logic into the required logic of a business process. The key actors, the experts of these processes, however, are not IT experts, and thus not the main designers. Nevertheless they have to be involved in the design of business processes. This paper presents the WS-Talk Service Designer which enables business process experts to write storybooks in their own language which are transformed automatically into semantics that can be handled by applications. The WS-Talk Service Designer currently supports organizations in managing their own and individual information.

1. Introduction

Web Service technology provides universal interchangeability and thus universal availability of application logic. This availability raises the potential to develop more applications, or more facets of applications, for broader quantity and variety of business processes (business logic). However, this availability does not mean an automatic pathway to new horizons in designing IT-based business processes.

What organizations expect from Web Services is first of all a reduction of "integration headaches" [1]. A survey – recently published on WebServices.Org – shows that a majority of companies take up Web Service technology in order "to integrate disparate systems". A further motivation for take-up addresses "tangible benefits in terms of reuse, developer productivity, and cost savings" [1]. Web Services, once conceived to facilitate more seamless e-commerce transactions beyond the firewall, get a role that is clearly focused on internal integration. Web Services standards and Web Service orchestration languages enable an essential move towards a new middleware-layer for integration [2].

Using Web Services, an application is rather a coalition of standardized and almost ubiquitous software modules than the typical monolithic block as we know it since decades. This coalition can be composed as well as adapted on-the-fly. WSDL (Web Service Definition Language), SOAP (Simple Object Access Protocol) and UDDI (Universal Description, Discovery and Integration) are proven standards to define coalitions of highly interoperable components and to propagate them in a distributed environment [3]. To form dynamic coalitions, they need to be orchestrated or choreographed. [4]

In this paper we investigate how natural language can enhance Semantic Web standards in order to bring the human expert of business processes and the IT expert closer together in the design of IT-based business processes. Natural language (NL) can provide us with semantics to write storybooks in our language, i.e. to compose and choreograph software modules in the way humans think about their everyday work. This rationale is the focus point of the WS-Talk project. The objective of this project is to develop an instrument that enables the experts in business processes rather than the IT expert to define business processes. WS-Talk's focus on natural language processing extends towards information retrieval applications or features, by nature. The vision of WS-Talk is to provide companies with natural language interfaces for their enterprise search systems.

This paper thus focuses primarily on coexistence of natural language and semantic web standards in a layered architecture for applications based on Web Services: in section 2 we present the framework of the layered architecture. The bottom layer comprises the Web Service stack. In the middle resides the choreography stack and on top of that the NL storybook serving the role of the NL modeling layer. Section 3 outlines and illustrates the rationale and advantages of extending service semantics by natural language. It demonstrates how WS-Talk can support enterprise search applications. Section 4 concludes the paper.

2. Three-layer Integration Architecture

Making services available for business purposes (with or without resorting to web service technology) is extremely intertwined with Business Process Management [5, 6]. The WS-Talk approach is thus inclined to Business Process Management where definition and management of business processes rests on two shoulders: the ones of the domain expert and of the IT expert.

This rationale has to be reflected by the architecture for business integration that follows a three-layered approach (see figure 1): the NL storybook at its top layer resides on the choreography layer which in turn resides on a Web Service stack. Our approach is inclined to the three-layer stack developed by W3C [7]. It considers the top level as the one that still needs to be developed. Standards are available for the middle and bottom layer. Web Service orchestration and Web Service choreography are concepts addressing the middle layer. In WS-Talk we investigate to what extend natural language processing capabilities can enrich the choreography (or orchestration) layer to make its semantics human-understandable. The WS-Talk storybook process engine, the WS-Talk Service Designer, operates primarily on the third layer, but can include the logic of the following layers as well without resorting to orchestration engines.

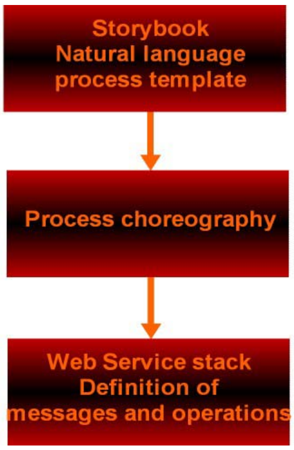

Figure 1: Three layers for a service-oriented architecture including the coexistence of natural language and semantic web standards.

At the second layer we concentrate on the observable behavior of Web Services in the context of message exchange between them. In WS-Talk we do not apply web service orchestration (or choreography) standards, but developed a simple choreography layer which inclines to those principles. Storybook statements are processed by the WS-Talk Process Engine. This includes also error handling and compensation actions as well as the conditional execution of storybook statements. Again, the main focus of the WS-Talk project is on the use of natural language on the third layer of this architecture.

The Process Engine's Service Stack defines a set of Web Services as atomic entities. It does not define a choreography language or any other language that helps to coordinate atomic operations. It contains their protocols and message characteristics. The messages themselves may be wrapped in SOAP envelopes.

3. Business Process Storybook

The WS-Talk process engine associates each statement of the storybook with instructions and executes the corresponding services and handles communication and data transfer between them. The following example shows a storybook used for a helpdesk application. It refers to a WS-Talk pilot application – an enterprise search system for the Chilean health insurance company "Cruz del Sur". The search system uses product descriptions as database and retrieves appropriate documents and extracts text passages from these documents in order to produce tailored retrieval results. The domain expert uses a storybook to describe the application logic – composition of and transactions over software components (or web services) – for a specific retrieval situation (retrieving information from product descriptions, for instance). In this case the helpdesk manager, for instance, describes how incoming user queries have to be handled: First, the user is prompted by the system to enter his query. The query is then processed, i.e. the type of the query is determined and stopwords are eliminated. Text analysis as applied in retrieval depends on the type of the query (see figure 2). And finally the system prints the result which should be a small number of text passages.

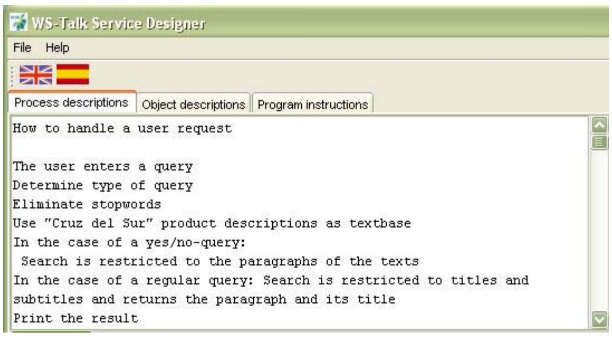

Figure 2: Example of a storybook.

The process engine is equipped with text interpretation capabilities. Storybooks are analyzed by the engine using a controlled vocabulary reflecting the language of the respective business domain. We use this vocabulary to describe operations, business objects, collaborations, etc.

While analyzing a storybook the WS-Talk process engine treats each natural language statement as a single instruction and considers the whole set of instructions as necessary to complete a particular process (handling a user request, in this case). The process may have different facets (alternative instructions are selected according to the results produced in course of the execution of its instructions). However, for each facet the ACID[1] rationale known from database technology is applied.

Before continuing we outline briefly the role of the controlled vocabulary. It is used to support information retrieval features that comprise primarily the functionality of WS-Talk's first application context. We improve the accuracy of text analysis by classification and taxonomy features by resorting to this vocabulary. In WS-Talk, classification is used to set up taxonomies which provide a way to see information around thematic categories [8]. Enterprise search in WS-Talk uses taxonomies *and* inverted term lists to annotate a text document, to determine document or document passage descriptors). The same process can be used for identifying suitable services for a given storybook statement.

Semantic co-ordinates – i.e. controlled vocabularies derived from taxonomies and structured according to concept hierarchies – are elements of orientation that can be communicated. Semantic co-ordinates enable us to develop a context map for the respective domain where the services are used. In addition, these hierarchies can be mapped into different natural languages in parallel [9].

3.1. The role of the WS-Talk Service Designer

The task of the WS-Talk Service Designer is

1) *to associate a task to be completed (in natural language correctly with the corresponding storybook,*

2) to identify Semantic Web representations of services and business objects corresponding to this storybook, and

3) to execute storybook statements using business object representations.

In the context of WS-Talk, "description" refers to NL representations of processes and objects (storybook statements) that can be processed by text analyzing features, but are human-understandable by nature. "Instructions" shall be machine-processable elements, not necessarily understandable to humans. Instructions are generally expressed by semantics of a programming language or semantic web standards like XML. The transformation of descriptions into instructions is achieved through an incremental process where

- a storybook statement is linked to one or more instructions and
- its components necessary to perform the instruction.

Instructions and components may be represented in ontology-like Semantic Web representations. In the end, concepts like "customer profile" or "inform users", for instance, have to be mapped to ontologies suitable for these business objects and processes.

3.2. Associating storybook statements with program instructions.

A particular service may address a business process reflecting also the process' inner logic. After a service is described in the way as explained above it is available under its title (like "How to handle a user request"). It can be selected and executed by the user (or by other services) like any other application. The process engine operates like a program interpreter. It takes a statement and looks for the corresponding semantic web representation. This "translation" process starts with a simple analysis of the statement's structure. In general, the schema subject-predicate-object is applied to identify what actor (user or other instruction) uses a function (represented by the predicate) to operate on a certain object. The engine applies the simple grammatical relationship

[subject] predicate [object]

in its first step of the analysis of a statement. While a predicate is always required, subject and object are optional. Each statement may contain more than just one predicate as well as predicate-object pairs. For each predicate there may be restrictions that apply like "Search is restricted to the paragraphs of the texts". These restrictions can be considered as parameters that are passed to and processed to the function represented by the predicate. The application of a sentence may depend of a condition extending the grammatical principles to

[condition] [subject] predicate [object].

The grammatical structure of a statement can be represented by (an asterisk indicates that the element may occur iteratively)

[condition] [[subject] [[predicate] [restriction] [object]]*]

[1] The ACID model is one of the oldest and most important concepts of database theory. It sets forward four goals that every database management system must strive to achieve: atomicity, consistency, isolation and durability.

The restriction may also reflect an attribute that should be related to an object. The condition itself can be represented by a simple relationship of "[object] predicate [restriction]" where "predicate" simply has the quality such as "is_a" used in semantic networks. Subjects refer to system components that reside within a certain application, not necessarily the Service Designer where the required appliance is composed and tested. The statement "The email system sends the results to the user" indicates that the remote mailing system is in charge with passing the retrieved information to the user. Or "The user enters a request" addresses the subject "user" which in fact refers to the user interface of the system. A statement thus indicates a process represented by a predicate owned by a subject and producing an object, a query, for instance, as in our example. This object is automatically made available (by a message bus) to the subsequent predicates (within the same statement or the following statements). From a different point of view it is an implicit restriction or attribute that is used by the subsequent process instructions. "Determine type of query" takes the query as input and produces a further object which reflects the nature of the query as stated by the user. The object produced by this statement is used further down to resolve the conditions "in case of a yes/no-query" and "in case of a regular query".

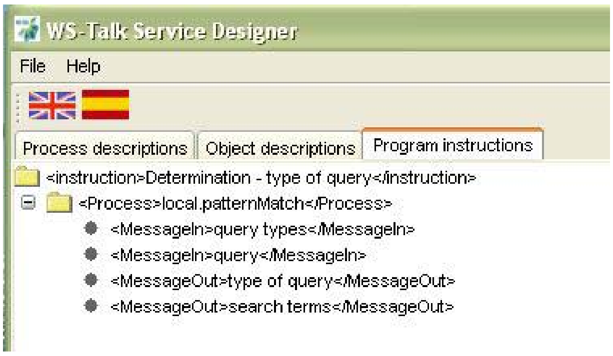

Figure 3. Representation of an instruction that corresponds to a statement's predicate.

While processing each statement the engine looks first for an instruction matching the statement's predicate (eventually including the object's name). If a corresponding description exists, the engine looks for objects that are required for this instruction (indicated as incoming messages). These ("MessageIn")-parameters are passed to the instruction. The results of the instruction are sent to the bus as outgoing message ("MessageOut").

The example of figure 3 shows an instruction that corresponds to the predicate "determine type of query". It looks for objects on the message bus that are labeled "query types" and "query". If the required information is available the engine sends the parameters to the process "local.patternMatch" that returns to the bus the two parameters "type of query" and "search terms". These parameters are later used by the subsequent instructions such as search, for instance.

Business objects are represented in a similar way. There is a key word ("databases" annotated to a XML structure that contains a number of access points of data collections. Each access point has again a descriptor in natural language.

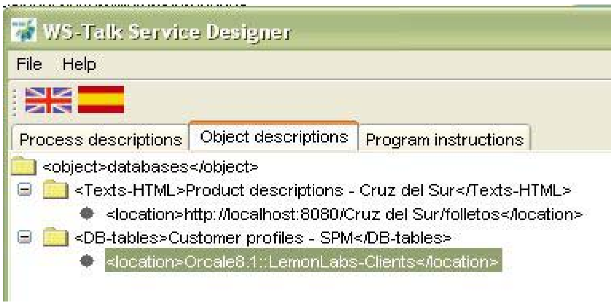

Figure 4. Representation of the access points to different data collections.

Figure 5. Representation of query types in Spanish. The first set comprises questions like "Is there …?" or "Does … contain …?" where the answer is only yes or no. The second group addresses more general questions like "Who is …?" or "What is …?" or "How and where can …?" etc..

3.3 A scenario

Business process experts describe not only services (or orchestrated services) in natural language, but also requests for services and process or object descriptions. From a different angle, a request may be nothing else, but a query to be matched with a service description, which may be the title of a storybook or an additional and more elaborate description. In many a case it will be inevitable to find a number of necessary process and/or object descriptions before a full transformation of the description into machine-processable instructions is possible.

Let us come back to our scenario which demonstrates the appliance as it results from the storybook presented. The user describes the requested service using natural language. This request, in turn, may also be considered as the description of a particular service that needs to be developed. In WS-Talk, controlled vocabularies are available in different languages in parallel. Even if a particular storybook is written in English, for instance, the retrieval feature as triggered by the storybook mentioned above can handle queries and texts in Spanish.

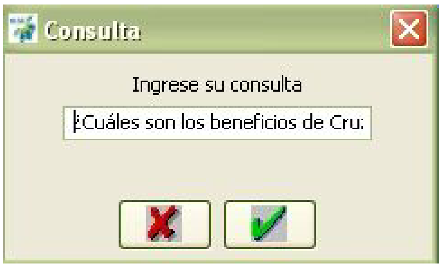

Figure 6. The first instruction of the storybook: querying the system. In this case the user asks for benefits of a certain house insurance.

Prompting the user to state a query is the first statement of the storybook as mentioned above. The query here is in Spanish and asks for the benefits of an insurance product called "Cruz Hogar" (Figure 6). From all available product information the system retrieves the most appropriate paragraphs from a product description concerning the household insurance "Cruz Hogar". The following monitor windows show the first the original text after its text structure has been identified (Figure 7). Structure information of a text is taken into account in the retrieval process. The annotation process using concept hierarchies is used to support retrieval of the most suitable document within a document retrieval is restricted to inverted term lists including structural information, which takes into account where matching terms occur (in the title, an enumeration, the body of a paragraph, etc.) [10]. The two statements preceding the final print statement in our storybook refer to this matching process From the whole product description (about two pages) only two paragraphs are retrieved that refer to the benefits of the insurance (Figure 8). The second paragraph was selected because of the term "advantages" which was considered equivalent to "benefits". The results as shown here are only available if the whole "retrieval" storybook could be executed successfully; otherwise the user gets information about the step where the execution of the application failed.

The inner logics of a business process (like the one presented in our storybook) are covered by Web Service Orchestration or Choreography languages whereas the operations within a process (database access, for instance) are performed by web services. In WS-Talk, we investigate to what extend process templates (in NL) can be used for orchestration purposes. In this context it is of minor importance if we consider Web Service Orchestration or Web Service Choreography.

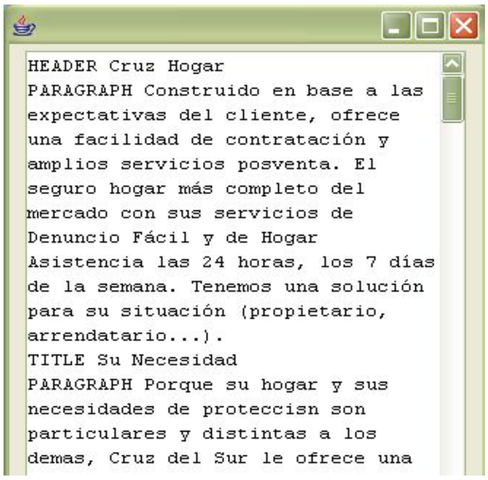

Figure 7. Extract of a structured resource description. Texts are marked with text structure information such as paragraphs, titles etc.

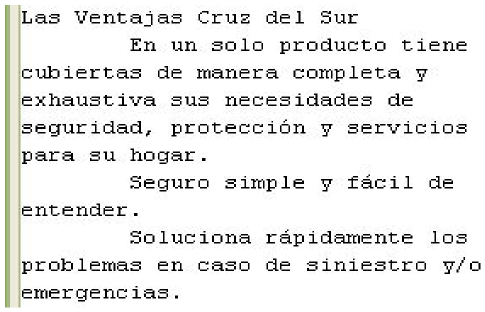

Figure 8. Retrieval results: second selected paragraph. From all available texts on house insurance product information the retrieval features select two paragraphs that matches with the user's query (see figure 6). The key expression of the paragraph shown here is "advantage" that determined the assignment to the user's query. "Advantage" (Spanish: ventaja) is used here as a synonym to "benefit"

The test environment of WS-Talk as presented here is an "Interactive Assistance Manual", an automatic information system based on an assistance manual for the support centre. Requests are questions stated in natural language. For the time being, we consider the simple case that one single question is answered by one single answer. Usually an answer is composed by observing a number of task steps. This composition can be steered by a storybook. And, in turn, it may resort to object representations that help to specify concepts appearing in the answers. Each storybook represents a complete transaction that can be performed only as a whole and must be rolled back completely if an error occurs. In that case the engine triggers in addition a compensation process that consists at least in a message to the users informing them why the transaction could not be completed and if it will be completed in the future.

The engine creates for each template an execution stub. It lists the sequence of operations to be executed. Each entry in the stub refers not only to its operation, but also to the required parameters and return values produced. Parameters are passed to the operation by incoming messages whereas results are passed to the operation's environment by outgoing messages. There are a number of object templates that help to "translate" concepts from the process template into concepts of the object templates.

4. Conclusions

Being less integrative than they should be can be a crucial downside of current Web Service technologies while approaching integrated and interoperable architectures. They still fail to integrate humans as the important resource both as an interactive part of the business processes and as a competent designer of integrated business solutions. Developing models for complex and dynamic solutions is a difficult activity for most system analysts because it demands both modeling experience and domain knowledge. To make the domain expert the principal designer will be crucial in the development of future Web Service technology [11].

In WS-Talk we opt for a different approach that proposes a co-existence of natural language and Web Service technology. Semantic Web representations of objects as well as processes are extended by natural language descriptions. They let users directly interact with web services, business logic representations, or other such objects that are rendered by or operating on Semantic Web standards. The best way for humans to develop a network of integrated business processes is to use their own language to describe the processes.

Acknowledgement

Research outlined in this paper is part of the project WS-Talk that is supported by the European Commission under the Sixth Framework Programme (COOP-006026). However views expressed herein are ours and do not necessarily correspond to the WS-Talk consortium.

5. References

[1] WebServices.Org, "From Web Services to SOA and Everything in Between: The Journey Begins". Retrieved June 14, 2005, from *WebServices.Org* website: http://www.webservices.org/index.php/ws/content/view/full/63404

[2] Q.H. Mahmoud, "Service-Oriented Architecture (SOA) and Web Services: The Road to Enterprise Application Integration (EAI)." Retrieved November 16, 2005, from *Sun Microsystems* website: http:// java.sun.com/ developer/technical/Articles/WebServices/soa/

[3] C. Peltz, "Web Services Orchestration and Choreography", *IEEE Computer*, (10) 36, 2003, p. 46.

[4] M.N. Huhns and M.P. Singh, "Service-Oriented Computing: Key Concepts and Principles", *IEEE Internet Computing* 9 (1), 2005, pp. 75-81.

[5] F. Casati,, E. Shan, U. Dayal, and M.-C. Shan, "Service-oriented computing: Business-oriented management of Web services", *Communications of the ACM* 46 (10), 2004, pp. 55-60.

[6] N. Milanovic and M. Malek, "Current Solutions for Web Service Composition", *IEEE Internet Computing* 8 (6), 2004, pp. 51-59.

[7] W3C, "Web Service Choreography Interface 1.0". August 2002, Retrieved November 4, 2005, from *W3 Consortium* website: http://www.w3c.org/ TR/wsci

[8] C. Moore, "Taxonomy tools organize searching. InfoWorld". Retrieved June 27, 2004 from http:// *www.infoworld.com*/article/04/06/21/25NNsearch_1.html

[9] K. Englmeier and J. Mothe, "Natural language meets semantic web". Retrieved July 16, 2003 from *ktweb.org* website: http://www.ktweb.org/doc/ Englmeier-NLP-SW.pdf.

[10] Karen Sauvagnat, Gilles Hubert, Mohand Boughanem, Josiane Mothe, IRIT at INEX 2003 INitiative for the Evaluation of XML Retrieval (INEX 2003), pp 142-148, 2003.

[11] V. Ovchininnikov, "Architecture of Heterogeneous Concept Space Managed by non-IT people". *Journal of Conceptual Modeling*, September 2004. Retrieved October 18, 2004 from www.inconcept.com/JCM/ September 2004/Ovchinnikov.html