IX International PhD Workshop
OWD'2007, 20–23 October 2007

RAPID PROTOTYPING OF COMPLEX SERVICES IN SOA ARCHITECTURE

Piotr Grobelny, *Faculty of Electrical Engineering, Computer Science and Telecommunication, University of Zielona Gora*

Abstract

This paper addresses the problem of rapid prototyping and designing of systems composed of a set of Web Services. Its objective is to provide the proof of concept prototype for intelligent discovery and matchmaking atomic Web Services within integrated workflow called complex service. The proposed solution brings closer the usage of domain specific languages, domain ontologies and Expert Systems in the Services Oriented Architecture paradigm area. SOA is a collection of services that communicate with one another. The services are self-contained and do not depend on the context or state of the other services. They work within the distributed systems architecture.

The paper is focused on the computer system architecture which allows for searching particular Web Services from a vast number of them, registered as the enterprise software resources. The main role within this architecture is played by the knowledge-based Expert System and the knowledge representation, as well.

The described prototype is implemented as the Java framework because it is the flexible language for integrating Web Services, rules engine, application server and the domain system applications.

As the illustrative use case for described approach the author proposes the domain of Internet-based geographic information systems (GIS) which represent a new branch of information and communication technologies.

1. Introduction

The growing amount of Web Services maintained by the information and telecommunication companies also commonly available in Internet allows for discovery new functionalities based on previously developed software components. The background of the consideration is the Domain Engineering approach [1] which relies on developing software families from reusable components which are parts of common domain system. In the future, the software can be named service-ware, where all resources are services in a Service Oriented Architecture. The main idea of this approach is that business processes engineer operates on atomic services not on the software or hardware that implements the service [4].

2. Concept

2.1 Basic idea

Fig. 1 presents the main architecture components of developed proof of concept prototype.

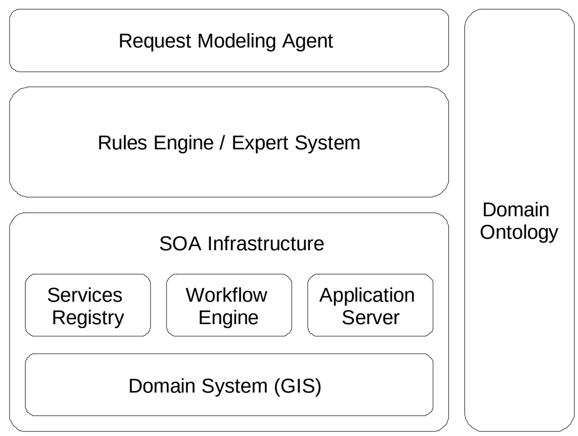

Fig. 1. Architecture overview.

The Domain Ontology is the component all remaining components depend on. The ontology describes the concepts and relations between them. It determines the common namespace for whole domain of software solutions (Domain System). The common meaning of the terms used for description of enterprise resources in particular domain is a very important issue because the Domain Ontology integrates all components of the architecture presented in Fig. 1.

Request Modeling Agent allows the user to model the request in terms of rules which will be used for knowledge inferring from Expert System. The requests can be modeled using Domain Specific Languages (DSL) [20] and next transformed to the format accepted by the shell of Expert System.

The Expert System works as the rule engine for modeled request. It represents the knowledge where the facts describe the atomic Web Services. The outcome of the reasoning process is the recipe for atomic services workflow for instance described as BPEL[1] [7] file.

[1] Business Process Execution Management

In this case the Expert System plays the role of complex service developer's advisory [17].

The SOA infrastructure is responsible for providing implementations of atomic Web Services through e.g. UDDI[2] [23] registrars and executing the workflow of atomic services (further called complex service) on the Application Server.

The author of this paper proposes to use the geographic information systems as the domain use case in SOA. Spatial information (geographic information) is a special kind of information – it is located in space. An increasing frequency of solving spatially oriented problems and then making decisions in both public and private sector of economy is one of the reasons why accessibility of spatial information is so important [13].

2.2 Problem statement and challenges

This paper focuses on the reference architecture and proof of concept prototype for rapid prototyping of complex services in Services Oriented Architecture. The author aims at using them to solve the challenges described within this section.

The problem of discovery complex services in SOA is known and considered by many research institutions and companies. Thus, it is only the background for further research. The author will work on the knowledge representation about services in Expert Systems. It consists of service models, which involve interface description and semantic specification as well as information about service quality (QoS) and non-functional properties. There are languages for building service models such as Flexible XML-based Language (FXL) [14] as well as Flora-2 [26] which basis on the semantic description, but the really challenge is to combine all the information in one model, which allows for registering Web Services as the facts in the deductive database. Additionally, the author in further research will focus on the formalization of this knowledge representation.

The next challenge is the modeling the requests to the Expert System because the description of service interfaces is not enough – two different Web Services may have the same input and output, thus the service semantic specification is needed. The author plans to provide example of Domain Specific Language (DSL) which can be helpful in semantic description approximated to the human natural language and simultaneously reasoned by Expert System.

2.3 Related work

The process-oriented enterprises need the seamless processes and rapid technology innovations also for sharing business processes with other companies. Existing IT resources, especially hardware and software, will be transformed into a common infrastructure such as telephone or power grids. Thus the new concepts like Software as Service appeared [11].

There are many approaches for discovery and composition of complex services [2][5][22] where the similar architecture is considered, but the knowledge representation in SOA differs for every solution. The paper brings closer the formalisms and common knowledge representation about service models for reasoning in Expert Systems.

Anyway, one element binds all the approaches. It is the ontology. Ontology aims at making the semantics in context of specified domain which is machine-readable as well as enables to infer logical conclusions from the presented facts. This functionality can be applied to the discovery of complex Web Services [24].

The authors of [4] propose to use an ontology-based approach instead of database solution in SOA. The main advantage of the ontology is the possibility of inferring new knowledge by utilizing a reasoning engine which considers existing facts and rules. The main part of presented architecture within this paper is the ontology. There are two leading technologies WSMO – Web Services Modeling Ontology [3] and OWL – Web Ontology Language [15] which allow for developing domain ontologies in SOA area.

Also the authors of [12] propose the system for Web Services discovery, which helps developers make a decision about which services to use by getting suggestions from the Internet community in situation where it is difficult to make the right decision due to the lack of knowledge. They use the theory of production rules "*if consequent then antecedent*" in accordance to the equation (3). The result of reasoning process is a set of services that have been used for similar requests in the past. This example shows the strength of rule-based knowledge representation in SOA.

The base for discovery, composition and enactment architecture presented in Fig. 1 was the reference architecture worked out by the European research project Adaptive Services Grid[3] described in [5]. In this approach the knowledge representation regards the service's semantic description in Flora-2 [26] language and Horn clause programs [10]. Basing on current state of the art the author of this paper will work on more complex as well as independent, and standarized knowledge representation is SOA. This paper presents the developed proof of

[2] Universal Description, Discovery, and Integration

[3] ASG is the research project supported by the Sixth Framework Programme of the European Commission under the Information Society Technology Objective Open Development Platforms for Software and Services. www.asg-platform.org

concept prototype which allows for evaluation of these research results.

The architecture proposed in Fig. 1 is useful for distributed systems based on services which may represent the business processes. They perform some calculations and usually generate some messages to other processes. The created workflow can be a base for parallel and distributed simulation, for instance presented in [19]. The example of such distributed applications which consists of services are geographic information systems.

The authors of [13] consider and present the wide range of GIS software packages, which are oriented on the web-based approach. Internet geographic information systems represent a new branch of information and communication technologies, because they allows for managing distributed data by crisis management, land management, urban planning and network maintenance [13].

2.4 The knowledge representation in Decision Tables

As described previously, the creation process of services workflow will be done by the rules engine of the Expert System. All service instances available in particular domain are treated as the knowledge representation system and can be explained as the Decision Table which contains production rules. Decision Tables specify what decisions should be made when some conditions are fulfilled [16]. This document considers the knowledge reasoning problem employing Decision Tables formalism

$$K = (U, A) \tag{1}$$

K (1) is the knowledge representation system, where:

U is a nonempty, finite set called universe
A is a nonempty, set of primitive attributes

The knowledge representation system which distinguishes the condition and decision attributes can be called Decision Table T:

$$T = (U, A, C, D) \tag{2}$$

Where $C, D \subset A$ are two subsets of attributes called condition and decision attributes.

Any implication

$$\Phi \rightarrow \Psi \tag{3}$$

is considered as the decision rule and Φ, Ψ are called predecessor and successor respectively. If $\Phi \rightarrow \Psi$ is decision rule and P contains all attributes occurring in Φ (condition attributes) and Q contains all attributes occurring in Ψ (decision attributes) then this decision rule can be called PQ-rule.

Tab. 1.

The knowledge representation of services landscape.

Operation	Input Parameters	Output Parameter	Service Name
P1	P2	P3	Q1
getMap	{Coordinates}	Map	GisMap
drawPoint	{Coordinates, Map}	Map	DrawPoint
drawSegment	{Coordinates, Coordinates, Map}	Map	DrawSegment
computeDistance	{Coordinates, Coordinates}	Distance	ComputeSegmentDistance

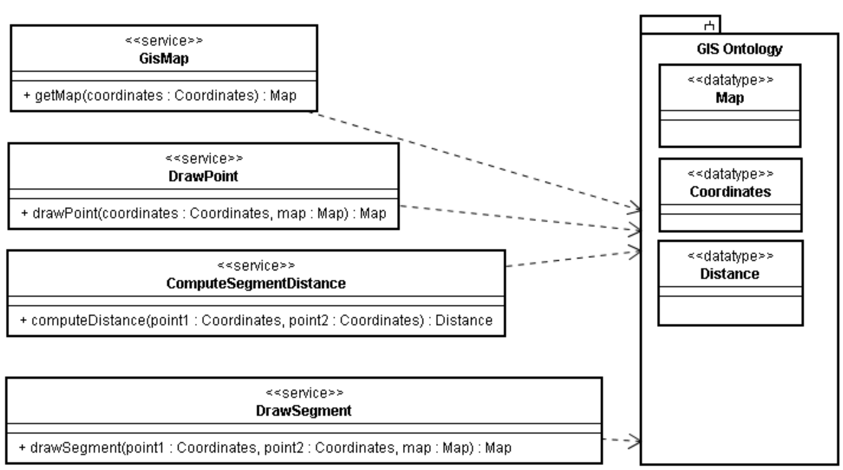

Fig. 2. The example of atomic services landscape in GIS domain.

Decision algorithm is a finite set of decision rules. If all decision rules in decision algorithm are PQ-rules then the algorithm is called PQ-decision algorithm or shorter PQ-algorithm [16].

Let's consider the real Decision Table (see Tab. 1). which represents the knowledge system from geographic information systems domain (see Fig. 2) in Services Oriented Architecture.

3. Implementation

3.1 Use case scenario

The objective of the use case scenario is to present the applicable results of research. The author selected the geographic information systems domain because it combines the spatial data with geographic artifacts like maps. Fig. 2 shows the example services landscape from GIS domain. They are the atomic Web Services which can be parts of discovered complex service.

The Decision Table which describes knowledge about services landscape looks like in Tab. 1.

The columns P1-P3 represent the condition attributes and column Q1 represents the decision attribute of the PQ-rule. These PQ-rules are stored as the facts in Expert System database.

The equation (4) formalizes a possible representation of PQ-rule from Tab. 1. in accordance to the equation (3).

$$P1=getMap \wedge P2=\{Coordinates\} \wedge P3=Map \rightarrow Q1=GisMap \quad (4)$$

3.2 Prototype

Prepared prototype refers to the architecture presented in Fig. 1. The base for implementation is the Java language and the Eclipse framework. Java became a powerful development platform for Service Oriented Architecture [9]. Java 2 EE 5 significantly enhanced the power and usability of the Web Services capabilities on the Application Server [9]. Also the Eclipse framework, which has been written in Java is the flexible platform for developing many kinds of plug-ins.

Request Modeling Agent where user designs the desired complex service by means of rules (as shown in Fig. 3) uses the shell of Jboss DROOLS Expert System [18] for modeling the request to the rules engine.

The outcome of the knowledge reasoning is the set of services and descriptions which are the parts of complex service. In the future it can be the BPEL [7] document. The author used the JBoss DROOLS [18] rule engine based on the RETE algorithm [6] provided as the Eclipse plug-in.

The request rules are stored in the Production Memory and the facts that the Inference Engine matches in the Working Memory as Fig. 4 shows.

```
package soarules.rulesengine

import soarules.facts.AtomicService;

rule "serviceProposition1"
when
#conditions
as : AtomicService( operation == "getMap",
inputParameters contains
"soarules.ontology.Coordinates", outputParameter
== "soarules.ontology.Map", serviceName :
serviceName, serviceDescription :
serviceDescription )
then
#actions
System.out.println( "Proposed service1: " +
serviceName + " - " + serviceDescription);
End
```

Fig. 3. The request rule definition.

Facts are asserted into the Working Memory where they may be modified or retracted. A system with a large number of rules and facts may result in many rules being true for the same fact assertion, and these rules are said to be in conflict. The Agenda manages the execution order of these conflicting rules using a Conflict Resolution strategy [18].

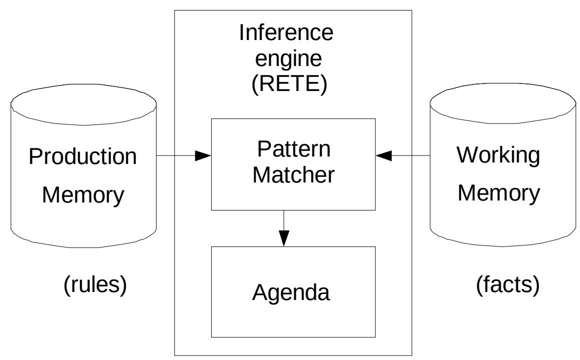

Fig. 4. The architecture of the used Expert System [17].

The author prepared the facts database in terms of production rules regarding equation (4) and Tab. 1 as the Java class which is loaded into the Working Memory of Expert System. Fig. 5 presents the example of the defined fact.

So, beginning from the request modeling by end-user through defining knowledge by domain engineer (facts) and ending on interacting with domain system (GIS) the domain ontology has been used. The author developed the needed ontology using WSMO [3] technology and WSML language [21]. The defined concepts have been converted to the corresponding Java classes: *Coordinates.class, Map.class, Distance.class* etc. using WSMO4J API [25], therefore the integration of the whole architecture (Fig. 1) was enabled on the Java platform.

```
public class FactsDatabase {
WorkingMemory rulesEngineMemory;
public FactsDatabase(WorkingMemory
rulesEngineMemory) {
this.rulesEngineMemory = rulesEngineMemory; }
public void activateFacts() {
AtomicService as;
Collection inputParameters;
// PQ rule
// P-parameters
as = new AtomicService();
as.setOperation("getMap");
inputParameters = new ArrayList();
inputParameters.add(new
Coordinates().getClass().getName());
as.setInputParameters(inputParameters);
as.setOutputParameter(new
Map().getClass().getName());
//Q-parameter
as.setServiceName("GisMap");
as.setServiceDescription("Service creates a map
according to provided longitude and latitude.");
rulesEngineMemory.insert(as);
… }
```

Fig. 5. The example of fact implemented in Java class.

The enterprise's ontology should describe all the services divided on specialized groups within geographic information systems.

The GIS are usually based on the n-tier client-server architecture and consist of the following groups of services:

- Data layer services – data management system which stores spatial and non-spatial data.
- Application layer services – a part responsible for processing business logic.
- Presentation layer services – user interface [13].

Fig. 6 presents the reasoned complex service workflow. Each atomic service represents the business logic as well as the workflow represents the business process of application layer. The prototyped GIS application will be used by the end-user through presentation layer.

```
Proposed service1: GisMap -
Service creates a map according to provided longitude
and latitude.

Proposed service2: DrawPoint -
Service draws point's marker on the map.

...
```

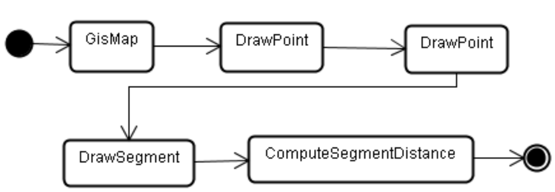

Fig. 6. Proposed atomic services workflow.

Paper [13] points the examples of atomic services for GIS platform (e.g. Google Maps [8]) such as overview map, scale tools, zoom, adding and removing objects on the map, measurement of distance, finding given object etc. The example of complex service can be biking paths, infrastructure networks, multi-layer maps.

The atomic services as realized according to the Web Service standards are deployed on the Application Server. Also the service implementations proposed by the Expert System are registered in the UDDI registrars, thus, the prototype knows where the particular Web Service has been deployed.

3.2 Limitations

The developed prototype provides the architecture for discovery complex services (workflow of atomic services) through Expert System. So, presented proof of concept example should be treated as the grounding for further considerations. The major limitation is the insufficient knowledge representation. In the next work the author will focus on issues of knowledge representation in SOA. The atomic service models as facts in Expert Systems should be extended with semantic description as well as information about quality (QoS) and non-functional properties. Thus, the modification of the formal knowledge representation and knowledge reasoning approach will be pointed out.

4. Conclusion

The paper describes the architecture for rapid prototyping of complex Web Services using the Expert System. The focus was put on the prototype implementation which proofs the feasibility of proposed architecture. The author demonstrated that the goal was reached using Java programming language which is the leading technology in Services Oriented Architecture area.

The author discussed the presented architecture and the relevant important issues such as usability of domain ontologies, the knowledge representation in SOA, knowledge reasoning through Expert Systems and the use case scenario based on the GIS domain.

The defined limitations point out the necessity of further research in knowledge representation in SOA extended by semantic techniques.

Bibliography

[1] Czarnecki Krzysztof, Eisenecker Ulrich: *Generative Programming – Methods, Tools and Applications,* Addison Wesley, Boston, MA, 2000

[2] *DIP Integrated Project, Data, Information, Process Integration with Semantic Web Services,* http://dip.semanticweb.org, accessed August 2007

[3] Dumitru Roman et al.: *Web Service Modeling Ontology,* Applied Ontology, 1(1), pp. 77-106, 2005

[4] Ekelhart Andreas et al.: *Security Issues for the Use of Semantic Web in E-Commerce,* Business Information Systems, Witold Abramowicz (Ed.) 10th International Conference BIS 2007 proceedings, LNCS 4439 pp.1-13, Springer, 2007

[5] Fahringer Thomas et al: *ASG Whitepaper - ASG technology advantages and disadvantages, exploitation possibilities and its business impact,* www.asg-platform.org, accessed August 2007

[6] Forgy Charles: *Rete: A fast algorithm for the many pattern/many object pattern match problem,* Artificial Intelligence, Volume 19, Issue 1, pp.17-37, September 1982

[7] Gaur Harish et al.: *BPEL Cookbook: Best Practices for SOA-based integration and composite applications development,* Packt Publishing, 2006

[8] *Google Maps API documentation,* http://www.google.com/apis/maps/, accessed August 2007

[9] Hansen Mark: *SOA Using Java™ Web Services,* Person Education Inc., Prentice Hall, 2007

[10] Horn Alfred: *On sentences which are true of direct unions of algebras,* Journal Symb. Logic, 16 pp. 14-21, 1951

[11] Hoyer Volker, Christ Oliver: *Collaborative E-Business Process Modelling: A Holistic Anaysis Framework Focused on Small and Medium-Sized Enterprises,* Business Information Systems, Witold Abramowicz (Ed.) 10th International Conference BIS 2007 proceedings, LNCS 4439 pp.41-53, Springer, 2007

[12] Kokash Natalia et al.: *Web Service Discovery Based on Past User Experience,* Business Information Systems, Witold Abramowicz (Ed.) 10th International Conference BIS 2007 proceedings, LNCS 4439 pp.97-107, Springer, 2007

[13] Komarkowa Jitka et al.: *Usability of GeoWeb Sites: Case Study of Czech Regional Authorities Web Sites,* Business Information Systems, Witold Abramowicz (Ed.) 10th International Conference BIS 2007 proceedings, LNCS 4439 pp.411-423, Springer, 2007

[14] Kossmann Donald, Reichel Christian, *SLL: Running My Web Services on Your WS Platforms,* Proceedings of WWW 2005, May 10-14, 2005, Chiba, Japan

[15] *OWL Web Ontology Language,* http://www.w3.org/TR/owl-features/, accessed August 2007

[16] Pawlak Zdzisław: *ROUGH SETS Theoretical Aspects of Reasoning about Data,* Kluwer Academic Publishers, 1991

[17] Pieczyński Andrzej, *Knowledge Representation in the Expert Diagnostic System,* Lubuskie Towarzystwo Naukowe w Zielonej Górze, 2003

[18] Proctor Mark et al.: *Drools Documentation,* http://labs.jboss.com/drools/, accessed August 2007

[19] Sikora Andrzej, Niewiadomska-Szynkiewicz Ewa: *A Federated Approach To Parallel and Distributed Simulation of Complex Systems,* International Journal of Applied Mathematics and Computer Science Vol.17, No. 1, 2007

[20] Spinellis Diomidis, *Notable design patterns for domain-specific languages*, The Journal of Systems and Software 56 (2001) pp. 91-99, Elsevier 2001

[21] *The Web Service Modeling Language WSML,* www.wsmo.org/wsml, accessed August 2007

[22] *SUPER, Semantic Business Process Management,* http://super.semanticweb.org, accessed August 2007

[23] *UDDI Specifications,* http://www.oasis-open.org/committees/uddi-spec/doc/tcspecs.htm, accessed August 2007

[24] Weber Ingo et al.: *A Conceptual Framework for Composition in Business Process Management,* Business Information Systems, Witold Abramowicz (Ed.) 10th International Conference BIS 2007 proceedings, LNCS 4439 pp.54-66, Springer, 2007

[25] *WSMO4j API and a reference implementation for building Semantic Web Services and Semantic Business Process applications based on the Web Service Modeling Ontology and the Business Process Modeling Ontology,* http://wsmo4j.sourceforge.net/, accessed August 2007

[26] Yang Guizhen et al.: *FLORA-2: A Rule-Based Knowledge Representation and Inference Infrastructure for the Semantic Web,* Second International Conference on Ontologies, Databases and Applications of Semantics (ODBASE), Catania, Sicily, Italy, November 2003

Author:

Mgr inż. Piotr Grobelny
Faculty of Electrical Engineering, Computer Science and Telecommunication
University of Zielona Gora
ul. Podgorna 50,
65-246 Zielona Gora,
POLAND
Tel. +48605066517

email: P.Grobelny@weit.uz.zgora.pl