Topological Functioning Model and Services Identification
An Approach for Services Identification from a Topological Functioning Model

Gundars Alksnis, Erika Asnina and Uldis Sukovskis

Department of Applied Computer Science, Institute of Applied Computer Systems,
Faculty of Computer Science and Information Technology, Riga Technical University, Meza iela 1 k-3, Riga, LV-1048, Latvia

Keywords: Service Oriented Architecture, Topological Functioning Model, Services Identification.

Abstract: To support the initial elicitation of software services, IT industry organizations have proposed various Service Oriented Architecture (SOA) specifications and frameworks. However most of them do not assume specific approach of SOA construction and what the service model should consist of. There are also approaches that suggest various practices for services identification, but the issue of proper services modeling still remains open. In this paper we propose an approach for services identification from a Topological Functioning Model, therefore enabling another viewpoint which fosters the quality of identified services.

1 INTRODUCTION

It is assumed that Service Oriented Architecture (SOA) analysis and modeling approaches mostly should focus on the business alignment with IT and the integration of existing assets than the development of SOA systems from the very beginning. Therefore the main focus is rather on the application assembly from available "bricks" with externalized interfaces (i.e., services), than the implementation of the service internals (which are just software assets implemented in particular technology platform, e.g. Java or .NET).

To support the initial elicitation and specification of software services, IT industry organizations have proposed various SOA specifications and frameworks (Kreger and Estefan, 2009), for example:

- The Open Group's SOA Reference Architecture, Ontology, Governance, etc.
- OASIS's SOA Reference Architecture, Reference Model, etc.
- OMG's SoaML, BPMN, etc.

However mostly these specifications talk about the definition of vocabulary and some common conceptual framework that can be applied to all SOA implementations – they do not assume or force specific process/method/approach of SOA construction and what the services model should consist of. Therefore the issue of proper selection/combination of services modeling approaches remain open.

The main scope of our research is to contribute to the services identification and modeling approaches in order to improve their quality.

In this paper we propose an approach in which the candidate services are obtained from Topological Functioning Model (TFM) of system's functioning. Our approach fosters the quality of identified (or candidate) services, because takes into account information obtained from the TFM. Our approach does not disregard other service identification approaches but rather can be used to complement them.

The paper is organized as follows. In Section 2 we review the main aspects for services identification and the Object Management Group's SoaML standard which is used in our approach. Section 3 reviews the basics of TFM and demonstrates it with the example case study.

Section 4 gives the main statement and discusses the propositions for candidate service identification from TFM. Finally in Sections 5 and 6 we review the related works and give conclusions.

2 SERVICES IDENTIFICATION AND MODELING

In SOA, the services constitute the main units of functionality for the integration of various legacy applications and information sources by using a shared and standards-based communication. According to the

Alksnis G., Asnina E. and Sukovskis U..
Topological Functioning Model and Services Identification - An Approach for Services Identification from a Topological Functioning Model.
DOI: 10.5220/0004090702410248
In *Proceedings of the 7th International Conference on Evaluation of Novel Approaches to Software Engineering* (MDA&MDSD-2012), pages 241-248
ISBN: 978-989-8565-13-6

best SOA practices, a service must exhibit particular architectural principles and patterns such as modularity, encapsulation, high cohesion, loose coupling, separation of concerns, reusability, composability and should have stand-alone implementation. These principles and patterns are required in order to support the following SOA goals:

- To enable the integration of the services across and within organizations.
- To be independent from the implementation technology and communication transport.
- To enable offering by service providers and used by service consumers.
- To have support for intermediaries (e.g., Enterprise Service Bus) which provide various supporting services like services discovery, dynamic selection, monitoring, QoS levels, etc.

In order to support the achievement of these goals it is a common practice to perform system's business modeling in order to obtain various viewpoint models, formalized processes and other documents which thereafter are used as an input for service identification approaches. Services identification is the process in which candidate services are identified, catalogued and specified so that they are aligned and support business. Afterwards services specifications are used as a basis for business process composition, integration and implementation.

According to the IBM Rational SOMA Practices (Arsanjani, 2008) there are three main service identification approaches that can be used:

- Top-down or Domain Decomposition – the refinement of business processes and business use cases. These approaches focus on the details regarding the tasks and roles that are involved in the realization of the business processes.
- Bottom-up or Existing Asset Analysis – such existing assets as systems, legacy applications models and components are examined for a candidate services.
- Middle-out or Goal-service Modeling – focuses on the allocation of candidate services such as to fulfill and serve particular business goals. This approach looks both at the business processes from the business domain, and at the existing services and IT assets.

Usually the use of combination of those approaches is essential for the proper determination of candidate services. Thus the weaknesses of each approach can be mitigated and the overall quality of identified services can be increased.

Even though there are proliferation of various notations for service modeling and specification, we believe that only the movement to standardization will substantially influence SOA adoption across and within organizations, especially in today's globalization. We can draw analogy with the Unified Modeling Language (UML) – because of the common notation, nowadays modeling adoption and tool support is integral. We also support this and to demonstrate our approach we will use standardized notation – SoaML.

SoaML (SOA Modeling Language) (SoaML, 2012) is OMG's specification to provide a standard ways to architect and model SOA solutions using the UML profile.

SoaML supports modeling of both business and IT perspectives, that collaboratively support the organization's mission. This is achieved by creating the services architectures therefore "separating the concerns of what needs to get done from how it gets done, where it gets done or who or what does it." (SoaML, 2012, p. 7)

"SoaML can leverage Model Driven Architecture (MDA) to help map business and systems architectures, the design of the enterprise, to the technologies that support business automation, like web services and CORBA." (SoaML, 2012, p. 7)

Our approach will use the SoaML vocabulary, therefore we will summarize the main notation used.

In order to understand the main principles of SoaML the following terminology needs to be reviewed (SoaML, 2012, p. 8):

- Participant – represents specific entity (people, organizations, information systems, etc.) that provides and/or consumes services via explicit interaction points called ports.
- Capability – the feature that participant provides and what is exploited to provide a service, or something that an organization needs that can be used to identify candidate services.

Service specification in SoaML can be take one of the following forms (SoaML, 2012, p. 8):

- Simple Interface – are used to model simple services which need not to know anything about the service consumers.
- ServiceInterface based – are used to model services which impose specific requirements regarding the service consumers, e.g. to model callbacks of asynchronous service calls.
- ServiceContract based – are used to model services collaboration which usually involves more than two participants that play in specific sequence (contract) in order to achieve the goal or add value.

SoaML ServicesArchitecture is UML collaboration diagram that “[..] puts a set of services in context and shows how participants work together to support the goals [..]” (SoaML, 2012, p. 15)

ServiceContract is UML collaboration diagram that “[..] defines the terms, conditions, interfaces and choreography that interacting participants must agree to (directly or indirectly) for the service to be enacted.” (SoaML, 2012, p. 18)

”ServiceArchitectures and ServiceContracts provide a formal way of identifying the roles played by parties or Participants, their responsibilities, and how they are intended to interact in order to meet some objective using services.” (SoaML, 2012, p. 29)

However in some cases Participants may not yet be known, but it is important to be able to specify the behavior of a service or capability that will realize a ServiceInterface. In these situations it is useful to express a ServicesArchitecture in terms of the logical Capabilities of the services.

“SoaML Capabilities represent an abstraction of the ability to affect change. [..] [They] identify or specify a cohesive set of functions or resources that a service provided by one or more Participants might offer. [..] [N]etworks of capabilities are used to identify needed services, and to organize them into catalogues in order to communicate the needs and capabilities of a service area, whether that be business or technology focused, prior to allocating those services to particular Participants.” (SoaML, 2012, p. 29)

Having reviewed the main SoaML aspects, we will continue by reviewing the Topological Functioning Model which will be used in our approach as a source and will show how mentioned SoaML aspects can be supported.

3 TOPOLOGICAL FUNCTIONING MODEL

In this section we give the review of Topological Functioning Model (TFM) (Osis and Asnina (2), 2011) and then continue with the demonstration of TFM construction (Asnina and Osis, 2011) by using example case study.

3.1 The Basics of Topological Funtioning Model

Theoretical foundations for TFM (also called a topological model of system functioning) was initially developed in 1969 by J. Osis. Since then it has found numerous applications and extensions for various problem domains. But since the introduction of OMG’s Model Driven Architecture, TFM also has been successfully applied in the context of MDA, see, for example (Osis, Asnina and Grave, 2008), (Osis, Asnina and Grave, 2007) and (Osis, Donins, 2010).

In general, TFM allows to represent the formal functionality of a complex system (organization) in a form of topological space which consists of finite set of functional features (system’s properties) and a topology between them to indicate the existence of cause-and-effect relations.

The main advantages that TFM provides is the possibility of analysis of both topological (connectedness, closure, neighborhood, continuous mapping) and functional (cause-and-effect relational, cyclic structure, inputs, outputs) properties of the system being modeled.

In the context to the MDA, TFM is mainly used to represent the computationally independent (CIM) viewpoint of the system, i.e. it depicts the business domain model (Asnina and Osis, 2011).

In order to construct TFM, first functional features must be obtained ”through the acquisition of the experts knowledge about the complex system, verbal description, and other documents concerning the structure and functioning (in documental, analytical, statistical, etc. form)” (Asnina and Osis, 2011, p. 46). Functional features can be represented in the form of a tuple (1).

$$< Id, A, R, O, PreCond, PostCond, Pr, Ex, S > \quad (1)$$

Where:

- Id – an identifier (e.g., an integer) that used to identify topological node.
- A – an object’s action (description of functional feature).
- R – the result of object’s action.
- O – an object which gets the result of the action and/or object which participates in an action.
- PreCond – the precondition(s) (if any) which must be satisfied for an action A to be executed.
- PostCont – the postcondition(s) (if any) which must be satisfied after an action A has completed.
- Pr – an actor(s) that provides or suggests an action.
- Ex – an actor(s) that enacts a concrete action.
- S – denotes whether the functional feature is inner or external to the system being modeled.

For more elaborated explanation TFM’s background and approach in general, we refer to (Osis and Asnina (1), 2011), (Asnina and Osis, 2011) and (Osis and Asnina, 2008).

3.2 TFM with Example Case Study

Let us now introduce the example case study for which we demonstrate TFM construction and later perform services analysis. We will start with the informal problem domain description.

Consider the fictional company "MuSt" which operates online business by providing licensed audio content to consumers by using the Web. The informal description of this company follows (*objects* and *actors* are marked in italic, **verbs** are marked in bold).

> "In order to **provide** licensed *audio content* to the consumers "MuSt" has *signed agreements* with various music *publishers*. "MuSt" also has *agreements* with online *payment provider* and *shipping* companies. In order to **advertise** company's services it has various *advertisement agreements* with music related *Web portals*. In order for a *consumer* to **buy** a *single*, an *album* or a *set of singles*, (s)he must **register**. (S)he does it by **filling out** and **submitting** online *registration form* via the "MuSt" *Web site application*, which contains such information items as *account name* and *password*, *personal information/address*, *various interests* related to music genres, and also *payment related account information*. Registration *form data* **is validated** against fraudulent actions, and *confirmation e-mail* **is sent** to the *consumer*. From this point on the *consumer* can **log on** on the "MuSt" *Web site* by using his/her *account*, **browse** for *an audio content* and **place** *an order* for both *electronic only* and/or *Compact Disc (CD) versions* of *an audio content*. If *electronic only version* **is ordered**, a *payment request* **is sent** to the online *payment provider* and if *successful acknowledgment* **is received**, the *consumer* **is notified** and **is able to download** MP3 type file(s) of the *purchased audio content*. If (also) a *CD version* **is purchased**, then if it is not available in the *stock*, it **is ordered** from the *publisher*. Then a new *shipment order* **is created** with the *shipping company* which **ships** the *ordered CD* to the *consumer's address*. Upon successful or unsuccessful shipment, the *shipping company* **notifies** "MuSt" and therefore the *purchase* **is closed**. Required *hardware* **is located** on the ISP's *data center* premises to whom *monthly payments* **are paid**. *Monthly payments* also **go to** the *shipping company*, the online *payment provider* and the *advertisers*."

By analyzing the verbs and associated nouns of this description, we have obtained the list of functional features (see Table 1).The analysis of the verbs are required, because as stated in (Osis and Asnina (2), 2011, p. 20):

> [..] functional feature is a characteristic of the system (in its general sense) that is *designed* and *necessary* to achieve some system's goal. Functional features must be defined in accordance with the verbs (actions) defined in the description of the system.

By analyzing the cause-and-effect relations between the obtained functional features, we have constructed corresponding TFM (see Fig. 1).Again, these relations were obtained in accordance with (Osis and Asnina (2), 2011, p. 21):

> [..] a cause-and-effect relation between two functional features of the system exists if the appearance of one feature is caused by the appearance of the other feature without participation of any third (intermediary) feature.

Currently the identification of such cause-and-effect relations is rather intuitive task – the proper identification relies upon the analyst's understanding of the nature of such relations.

It is important to note that the causal implication is characterized by the nature or business rules not by logic rules and that cause-and-effect relations also have a *time dimension*, since a cause chronologically *precedes* and *generates* an effect. For more detailed explanation we refer to (Osis and Asnina (2), 2011).

According to the TFM requirements, each TFM model must have one main functional cycle which represents the sequence of crucial functional features. It is assumed that functional feature is crucial if the failure of it will lead to the whole system's inability to function. Usually they represent functional features from the main line of business. In our case, the main functional cycle consists of the functional feature sequence "9-10-11-18-20-9," i.e. audio content provisioning and order processing (in the figure it is marked with the bold cause-and-effect relations).

There also can be present other functional cycles which usually extends the main functional cycle by including connected subsequences of functional features. However the malfunction within such subsequence must not interrupt the whole systems' functionality. For example, the subsequence "..-11-13-19-18-.." represents the CD delivery.

4 SERVICES IDENTIFICATION

In this section we review various sources for services identification with the specific emphasis in this pro-

Table 1: “MuSt” business domain functional features.

Id	Functional Feature (A)	Precondition (PreCond)	Actor (Ex)	S
1	Signing an agreement with a publisher		Owner	I
2	Signing an agreement with an online payment provider		Owner	I
3	Signing an agreement with a shipping company		Owner	I
4	Signing an agreement with an advertiser		Owner	I
5	Registering a new consumer	Not yet registered	Consumer	E
6	Validating a registration form		Employee	I
7	Log in to a web site		Consumer	E
8	Browsing an audio content	Logged in	Consumer	E
9	Placing an audio content order	Logged in	Consumer	E
10	Opening an order		Employee	I
11	Notifying a consumer	Payment acknowledged	Employee	I
12	Downloading an audio content		Consumer	E
13	Checking availability of a CD	CD version purchased	Employee	I
14	Ordering CD to a publisher	Purchased CD is not in the stock	Employee	I
15	Receiving CD from a publisher		Employee	I
16	Shipping CD to a customer address	CD is in the stock	Shipping Company	E
17	Notifying about a shipment		Shipping Company	E
18	Closing an order		Employee	I
19	Placing CD shipment order		Employee	I
20	Provisioning an audio content	Agreement with publisher is signed	Web Application	I
21	Performing a payment		Payment Provider	E
22	Acknowledging a payment		Payment Provider	E

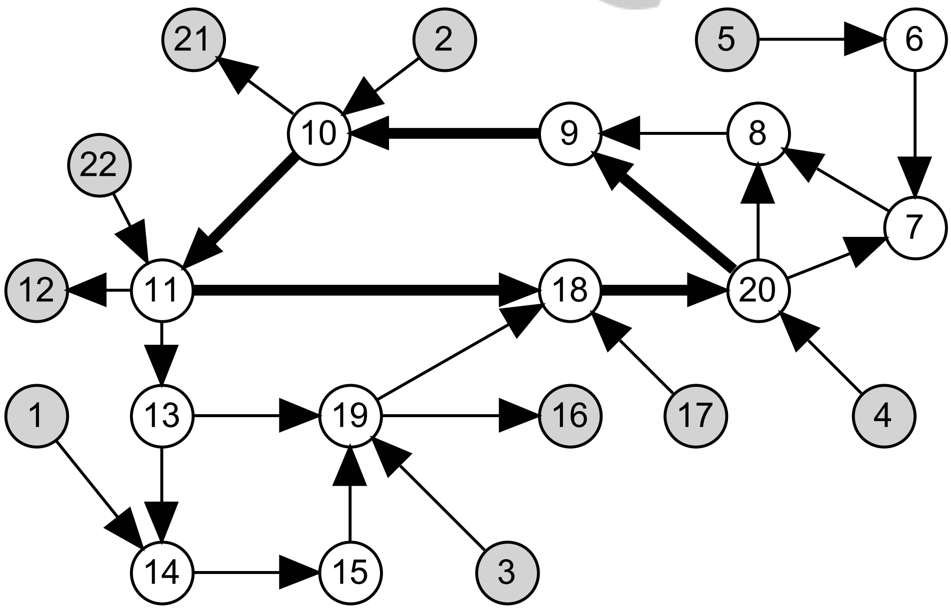

Figure 1: “MuSt” business domain Topological Functioning Model. White nodes represent inner functional features, while the gray nodes represent system’s inputs or outputs. Bold arrows links the main functional cycle of the system.

cess on the role of business goals.

In order to identify (candidate) services, we can use any available domain analysis method to obtain various business domain information from such artifacts as:

- Business process model;
- Business goals;
- Information (data) model;
- Existing non-SOA systems and applications; and others.

Therefore we propose that among such artifacts also can be used TFM. In the remainder of the section we argue regarding the following statement:

Statement 1. *It is possible to use the problem domain's TFM to identify SoaML Capabilities and/or Participants for ServicesArchitectures and ServicesContracts.*

But before that we want to review what role in the service identification plays the system's business goals.

4.1 The Role of Business Goals

According to the goal-service approach (Ang, 2008), before the elicitation of services, it is important to state what business goals the system (organization) must achieve. Namely, all eventual services should directly support at least one stated business goal in order for them to be business aligned.

In order to show that our approach also follows this principle, we must state some business goals. Let assume that for our example case study the following business goal and subgoals have been stated:

1. Increase Revenue by 10% each quarter
   (a) Increase the active consumer base by 3% each month
   (b) Increase the amount of provisioned content
      i. Seek for agreement with additional publishers
      ii. Question the consumers about missing content

As we later show, most of the identified candidate services will directly support specific business goal or subgoals.

4.2 Services Identification from Topological Functioning Model

In this subsection we give two propositions and detailed explanations regarding candidate services identification from Topological Functioning Model.

Proposition 1. *Topological Functioning Model's inner functional feature can be transformed to SoaML Capability or Participant classifiers' operation.*

This statement is true because functional feature, SoaML Capability and Participant all have an ability to affect cause.

To make the Proposition 1 clear, we must remind that the inner functional features (shown as white nodes in the Fig. 1) are the properties that are not inputs or outputs of the system, whereas the inputs and the outputs (shown as gray nodes) identify what functionality from and to the external environment the system receives or generates, respectively.

By referring to the example case study, we have elicited the following SoaML Capabilities:

- Purchasing – capability which deals with consumer activities regarding purchase of audio content (with the operations as functional features No. 9, 10, 11 and 18).
- Provisioning – capability which deals with the presentation of available licensed audio content, both with online and CD versions and the latter ordering from the publisher (with the operations as functional features No. 6, 7, 8 and 20).
- StockManagement – capability to deal with CD stock management and ordering from publishers (with the operations as functional features No. 13, 14, 15 and 19).
- SiteManagement – for the Web site application support (with the operations as functional features No. 6 and 7)

This list of capabilities were obtained by analyzing functional features and grouping them under one coherent name.

If there are already exists Capabilities that, for example, were obtained from other business domain artifacts, then it might be the case that the existing Capabilities are just amended with the new operations from the functional features.

Proposition 2. *Topological Functioning Model's input or output functional feature can be transformed to SoaML Capability or Participant classifiers' operation if it directly supports at least one business goal.*

This proposition covers the aspects of the problem domain which are related to system's environment or can lead to the need for an access of external services.

The following Capabilities can be selected from the example case study:

- OnlinePaymentProvisioning – for interaction with the online payment provider (with the operations as functional features No. 2, 21 and 22).
- Shipping – for interaction with the shipping company for the CD delivery (with the operations as functional features No. 3, 16 and 17).
- Publishing – for interaction with the publishers for obtaining licenses for both the online (MP3) and the CD versions of singles and albums (with the operations as functional features No. 1, 12 and 4).

Table 2: The goal-service model for the "MuSt" example case study.

Business Goal	KPI	Candidate Service
1. Increase Revenue	10% each quarter	OnlinePayment-Provisioning
1 a. Increase the active consumer base	3% each month	Purchasing Shipping
1 b. Increase the amount of provisioned content	-	Provisioning
1 b i. Seek for agreement with additional publishers	-	Publishing

- SiteManagement – for the Web site application support (with the operation as functional feature No. 5)

To support this selection, we matched them with the appropriate business goals (see Table 2). As we can see, each candidate service (SoaML Capability) has direct relation to the business goal. Thus we can assure that eventual services also will be business aligned. However, there can also be business goals for which there are no corresponding Capability. Such business goals then will be out of the SOA implementation domain.

The next logical step is to specify all obtained Capabilities by specifying then in SoaML ServiceContracts, ServicesArchitectures, Participants and ServicesInterfaces.

For our example case study we could identify such Participants as OnlinePaymentProvider, Shipper, Consumer, Provisioner and Publisher. There could be one ServiceContract based specification in which Provisioner, Consumer and Shipper are involved. There also could be two ServiceInterface based specifications. The first in which Provisioner and Publisher and the second in which Provisioner and OnlinePaymentProvider are involved.

We do not give the process of their acquirement in this paper because basically it is the same as, for example, described in (Arsanjani, 2008) or (Amsden, 2012). We just show the possible ServicesArchitecture which can be obtained from these ServiceContract specifications (Fig. 2).

Depending on the tool availability, such specification process can be performed in the model driven way thus contributing to the automated SOA implementation.

5 RELATED WORKS

Since the coining of the term SOA in 1996 by Gartner Research and refreshing it in 2003 (Natis, 2003), many approaches to service identification and design were proposed. However mostly all of them follow either bottom-up or top-down or both approaches.

As were mentioned, top-down approaches assume the refinement from business processes and use cases, whereas bottom-up approaches seek a ways to adapt existing IT assets for the service oriented communication. It is important to note, that the combination of both ways are important to use in order to successful determine proper services architecture.

IBM was among those who first announced SOA related methodology – Service-Oriented Modeling and Architecture (SOMA) (Arsanjani, 2008) and continues to extend it by incorporating newest trends, adopting to the emerging standards like OMG's SoaML and providing tool support for them.

There are also other attempts to standardize SOA service specification notations. For example, Unified Service Description Language (USDL) (USDL, 2011). However we see that SoaML mainly aims to standardize analysis and modeling aspects, whereas USDL has different goals, namely to cover the specification not only of IT and business but also to serve for a reference and exchange purposes of implemented services.

Our approach can be integrated with IBM's SOMA top-down approach by using TFM as an additional source for candidate services identification. In fact our approach can be adapted to any top-down approach of services identification and modeling. However because we assume the usage of SoaML, it is the best suited especially for approaches which support standardized model driven frameworks.

6 CONCLUSIONS

In this paper we have proposed an approach to the services identification from TFM in order to elicit business aligned candidate services (Capabilities in terms of SoaML) so that they fulfill the typical properties of the services. Our approach can be integrated and used in conjunction with other top-down services identification approaches, however it is the best suited with approaches which incorporate the SoaML standard.

The future plans are regarding more propositions in order to provide transformation rules from TFM to SoaML classifiers. Also we want to investigate how TFM functional feature's cause-and-effect relations and properties can be used to support auto-

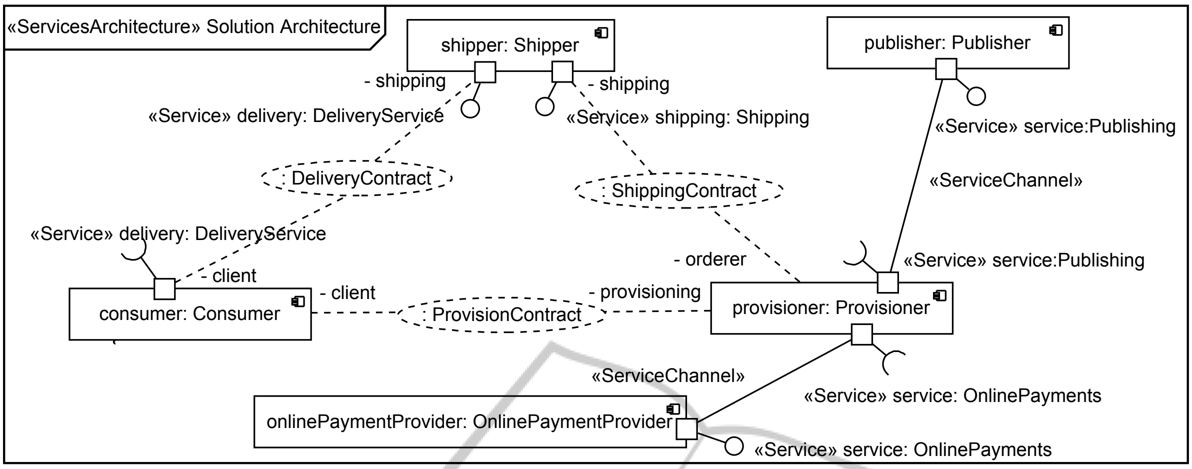

Figure 2: SoaML services solution architecture for "MuSt" example case study.

mated specification of SoaML ServiceContract and ServicesArhitecture.

REFERENCES

Amsden, J. (2012). *Using SoaML services architecture*. IBM developerWorks. Retrieved May 7, 2012, from http://www.ibm.com/developerworks/rational/library/soaml-services-architecture/

Ang, J. S. H., et al. (2008). *Goal-service modeling. Patent No. 20080027784*. United States. Retrieved May 7, 2012, form http://www.freepatentsonline.com/y2008/0027784.html

Asnina, E. and Osis, J. (2011). Topological Functioning Model as a CIM-Business Model. In Osis, J. and Asnina, E. (Eds.), *Model-Driven Domain Analysis and Software Development: Functions and Architectures*. Hershey - New York: IGI Global. 40-64.

Arsanjani, A., et al. (2008). SOMA: A method for developing service-oriented solutions. *IBM Systems Journal*, 47(3), 377-396. doi: 10.1147/sj.473.0377

Kreger, H. and Estefan, J. (2009). *Navigating the SOA Open Standards Landscape Around Architecture. The Open Group.* Retrieved May 7, 2012, from http://www.opengroup.org/onlinepubs/7699909399/toc.pdf

Natis, Y. V. (2003). *Service-Oriented Architecture Scenario*. Gartner Research. Retrieved May 7, 2012, form http://www.gartner.com/resources/114300/114358/114358.pdf

Object Management Group. (2012). *Service Oriented Architecture Modeling Language (SoaML), Version 1.0*. Retrieved May 7, 2012, form http://www.omg.org/spec/SoaML/1.0/PDF

Osis, J. and Asnina, E. (1) (2011). Is Modeling a Treatment for the Weakness of Software Engineering?. In Osis, J. and Asnina, E. (Eds.), *Model-Driven Domain Analysis and Software Development: Architectures and Functions*. Hershey - New York: IGI Global. 1-14.

Osis, J. and Asnina, E. (2) (2011). Topological Modeling for Model-Driven Domain Analysis and Software Development: Functions and Architectures. In Osis, J. and Asnina, E. (Eds.), *Model-Driven Domain Analysis and Software Development: Functions and Architectures*. Hershey - New York: IGI Global. 15-39.

Osis, J. and Asnina, E. (2008). A Business Model to Make Software Development Less Intuitive. *Proceedings of the 2008 International Conference on Innovation in Software Engineering*. Los Alamitos: IEEE Computer Society CPS, 1240-1246.

Osis, J., Asnina, E. and Grave, A. (2008). Formal Problem Domain Modeling within MDA. *Communications in Computer and Information Science. Vol. 22, Software and Data Technologies*. Berlin Heidelberg: Springer-Verlag, 387-398.

Osis, J., Asnina, E. and Grave, A. (2007). Computation Independent Modeling within the MDA. *IEEE International Conference on Software-Science, Technology Engineering (SwSTE'07)*. 22-34. doi: 10.1109/SwSTE.2007.20.

Osis, J. and Donins, U. (2010). Formalization of the UML Class Diagrams. *Evaluation of Novel Approaches to Software Engineering*. Berlin Heidelberg: Springer-Verlag. 180-192. doi: 10.1007/978-3-642-14819-4_13

Unified Service Description Language. (2011). Retrieved May 7, 2012, from http://www.internet-of-services.com/index.php?id=264&L=0