A MOF-based Social Web Services Description Metamodel

Amel Benna[1,3], Zakaria Maamar[2] and Mohamed Ahmed Nacer[3]
[1]*DSISM, CERIST, Algiers, Algeria*
[2]*College of Technological Innovation, Zayed University, Dubai, U.A.E.*
[3]*LSI Laboratory, USTHB, Algiers, Algeria*

Keywords: MDA, MOF, Social Web Service, Metamodeling.

Abstract: To promote and support the development and use of social Web services by the IT community on the Web, both social Web service-based applications and their support platforms should evolve independently from each other while sharing a common model that represents the characteristics of these social Web services. To achieve this duality, this paper proposes a model-driven approach. First, the approach identifies a social Web service's properties. Then a Meta-Object-Facility (MOF)-based social Web services description metamodel is developed. Finally, a prototype illustrates how the MOF-based metamodel is used.

1 INTRODUCTION

Web services (WS)s offer a standardized way for deploying interoperable Web-based applications (Chung et al., 2003). However, several issues like discovery, composition, and monitoring continue to undermine their benefits. To address some of these issues, there is a research trend that looks into WSs from a social perspective (Xie et al., 2008; Chen et al., 2015).

A Social Web Service (SWS) is the result of blending social computing with service-oriented computing. Compared to regular WSs, SWSs *"establish and maintain networks of contacts; count on their (privileged) contacts when needed; form with other peers strong and long lasting collaborative groups; and, know with whom to partner so that ontology reconciliation is minimized"* (Maamar et al., 2011a). For instance, *prepareJob* SWS "likes" to collaborate with *postJob* SWS because of previous successful compositions and also "recommends", for similar reasons, *planJob* SWS as a replacement in the case of failure. *postJob* SWS and *planJob* SWS are part of *prepareJob* SWS's *"collaboration"* and *"substitution"* networks, respectively.

The existing solutions for discovering and composing WSs from a social perspective (Maamar et al., 2011b; Chen et al., 2015; Maaradji et al., 2011), rely on a specific SWS's networks (e.g., substitution network) and specific platform (e.g., Java EE platform, and .Net). Therefore, the evolution and update (e.g., new social networks and platform update) of these existing solutions are difficult. There is a serious lack of a common model that permits to describe SWSs' properties (e.g., who are a SWS's contacts) across different stakeholders (e.g., designers, developers, providers, and users) and regardless of the platform used. Moreover, Web Service Description Language (WSDL) does not capture the "social" dimension of a WS in terms of who are the collaborators and substitutes, for example. What if we think of ***S***-WSDL for Social-WSDL?

Our objective is to define ***S***-WSDL; a common representation of SWS to be made available for all stakeholders regardless of the implementation platforms. ***S***-WSDL enhances WSDL with a "social" dimension without altering WSDL's original content. This can be done using Model Driven Architecture (MDA). MDA represents everything from requirements to business modeling and to technology implementation (OMG, a). MDA requires models to be expressed in a Meta-Object-Facility (MOF)-based language (OMG, b). MOF is a standard for specifying, constructing, and managing technology neutral metamodels, i.e., models that describe other models. When a system model refers to a specific platform it is called Platform Specific Model (PSM). Contrarily, the model is called Platform Independent Model (PIM) (OMG, a). A model in PIM can be transformed into another PIM when it needs to be enhanced, filtered, or specialized or into one or more PSMs. PIM into PIM transformation represents model refinement.

Our approach refines WSDL metamodel, inde-

Benna, A., Maamar, Z. and Nacer, M.
A MOF-based Social Web Services Description Metamodel.
DOI: 10.5220/0005687302170224
In *Proceedings of the 4th International Conference on Model-Driven Engineering and Software Development (MODELSWARD 2016)*, pages 217-224
ISBN: 978-989-758-168-7

pendently of any platform. We inject some social elements into the description of this metamodel. The outcome called ***S***-WSDL metamodel is MOF based. ***S***-WSDL metamodel can serve different purposes such as specifying social elements of a WS (e.g., know with whom to partner and collaborate). Automatic transformation from WSDL PIM into ***S***-WSDL PIM can be carried out using model-to-model transformation languages such as Queries, Views, and Transformations (QVT) (OMG, b). ***S***-WSDL PIM generated can in turn be transformed into PSM.

Our main contributions in this paper are as follows: identification of SWSs' properties; adoption of MDA to develop ***S***-WSDL metamodel (i.e., a way to abstract these properties), and demonstration of how ***S***-WSDL metamodel based on MOF and QVT Relation supports automatic transformation of models.

The rest of this paper is organized as follows. Section 2 provides an overview of SWSs approaches and WSs models. Section 3 defines SWSs' properties and then illustrates SWS modeling. Section 4 discusses how the proposed model is instantiated. Finally Section 5 draws some conclusions.

2 RELATED WORK

We provide first a brief overview of SWSs and then a discussion on MDA use for WSs development.

2.1 Social Web Service Overview

A significant amount of research looks into blending social computing with service-oriented computing.

To perform WSs discovery, Maamar et al. develop LinkedWS, a social network discovery model that captures interactions between WSs (Maamar et al., 2011b). This network identifies collaborators and substitutes of WS. For a better quality of WS discovery, Chen et al. (Chen et al., 2015) construct a global social network of WSs. They transform WSs' WSDL and OWL-S descriptions on the Web into RDF descriptions. In this global social network, social links between WSs that can work together during composition are defined. Chen et al.'s network allows users to identify a WS as an entry point in the network and then navigate along social links to discover the WSs deemed necessary for composition. El-Goarany et al. propose a WS recommendation system within a social network of WSs that makes use of generated ontologies to discover similar and complementary WSs (El-Goarany et al., 2008). Maaradji et al. use information collected from a user's social networks so that WS composition schemas are recommended (Maaradji et al., 2011). In (Maamar et al., 2012) Maamar et al. interleave social networks of users and social networks of WSs to support WSs composition, execution, and monitoring. To provide semantic WS composition, Xie et al. (Xie et al., 2008) define social networks based on trust relationship between users of WSs, providers of WSs, and WSs themselves. Xie et al's networks help select WSs or composition plans and make the search results meet users' requests. Last but not least, Bansal et al. (Bansal et al., 2010) present a framework for trust-based WS composition. This trust rating is used to filter composition results and then produce solutions that comprise WSs provided by trusted providers.

The aforementioned approaches use different representation of SWSs and depend on their platform support and technologies (e.g., Java, .Net, and Prolog) making it difficult to identify all WSs' social elements (e.g., trust and collaboration).

2.2 MDA for Web Services Development

MDA addresses specific issues related to WSs such as automating the generation of platform-specific implementations using WS models (Bézivin et al., 2004), extending WSDL for describing the qualities of WSs (D'Ambrogio, 2006), and promoting interoperability between WS_* protocols (Simon et al., 2013).

The adoption of MDA for describing WSs interactions is quite new and very few studies are available. Bouchakour et al. use MDA to develop SWSs (Bouchakour and Benslimane, 2013) in terms of building social networks from a log file that defines specific WSs interactions, assessing and modeling the social qualities of each WS, and then adding these qualities to a WSDL file using an XSLT template. However, social qualities update (e.g., new interactions between WSs) leads to rebuilding social networks, which is time consuming and error prone for SWSs developers, providers, and users. Zheng et al. (Zheng et al., 2012) extend OWL-S metamodel to support the description of WS providers social context.

It should be noted that (Bouchakour and Benslimane, 2013) and (Zheng et al., 2012) approaches do not embrace MDA principles (e.g., PIM and PSM) and standards (e.g., MOF and QVT). They do not consider the overall SWSs' properties, and do not allow incremental model transformation.

Contrarily, we propose to refine WSDL PIM in order to define a MOF-based ***S***-WSDL metamodel, a model abstracting a WS's social elements. The use of QVT Relation language provides an automatic transformation of WSDL model into ***S***-WSDL model, and support target incremental model transformation.

3 MODELING SOCIAL WEB SERVICE

This section identifies properties that should define WSDL description of SWS and describes the MDA approach for enriching WSDL with these properties.

3.1 Properties of Social Web Service

To identify SWSs' properties, we looked at the steps that Beydoun et al. peformed in order to define the concepts of a generic multiagent metamodel (Beydoun et al., 2009). These steps are summarized in Table 1 and are looked at from both bottom-up and top-down perspectives. The bottom-up perspective (Steps 1-4) identifies common SWSs' properties using the existing literature on SWSs. The top-down perspective (Step 5) validates the identified SWSs' properties with regard to the existing SWSs literature.

In Step 1, we investigate SWSs literature using academic library databases (e.g., Web of Science) and academic search engines (e.g., Google Scholar). SWSs approaches published in journals with high impact factors and in conferences with high ranking[1] are the measures taken into account in selecting the top n (n=10) SWSs approaches.

In Step 2, we decide on the general concepts relevant for SWSs. These concepts are selected on the basis of tasks that are necessary for building a concept dictionary in Methontology[2] method (Gómez-Pérez et al., 2004). The tasks are as follows:

- Task 1. Build a glossary of terms that includes all relevant terms for SWS like concepts, instances of concepts, attributes of concepts, and relations between concepts, their synonyms, and descriptions. To build this glossary, we capture and collect terms related to SWSs' social networks and social element from approaches identified in Step 1.
- Task 2. Build concept taxonomies to structure concepts. We select terms that are of type concept from the glossary of terms. Concepts are examined based on middle-out strategy used in Uschold and King's method for building ontology (Gómez-Pérez et al., 2004). This strategy recommends identifying first, the core of basic domain terms and then specifying and generalising them as required. In our case, we start by identifying the main concepts that define a SWS's social network. As social relations mean connecting parties together, graphs capture them (King et al., 2009) using *node* and *edge* between nodes. We refer to *edge* as *relationship*. We have then generated top and bottom concepts for *node* and *relationship*. However, the existing SWS literature presents several terms that are similar and different, and sometimes overlap. To reach a consensus over SWS network community, we proceed as follows:
  - Two concepts with overlapping or disjoint meanings have the same name. As in ONIONS[3] method (Gómez-Pérez et al., 2004), we create a concept for each meaning. The new concepts are linked to others concepts through specialization or generalization (e.g., *trust* as rating of WS specializes *node degree* concept and is named *trust degree*, while *trust* between user and WS provider specializes *relationship type* concept and is named *trust relationship*).
  - Same concepts having different names. As in ONIONS method (Gómez-Pérez et al., 2004), we adopt a name based on what is widely agreed upon in the domain (i.e., in social networking community). For instance, we adopt *relationship* concept name for relationship between nodes which is referred as *explicit relationship* (Maaradji et al., 2011).
  - Some specific concepts are omitted, so they specialize more general concepts of the concept taxonomies. For instance, formulas for assessing *relationship weight* are omitted. These formulas can specialize *get relationship weight* concept which in turn specializes a *relationship weight* concept.
  - Other specific concepts can be obtained from the concept taxonomies by specializing more general concepts. For instance, *collaboration* and *substitution* between WSs in (Maamar et al., 2011b) can be obtained by specializing *relationship type* concept.
  - When a concept is closely related to a specific approach and cannot be obtained by specializing more concepts from concept taxonomies, we create this concept. For instance, *relationship dependency* concept is created to specify trust relationship between two WSs in (Xie et al., 2008).
  - Implicit concepts are included in the concept taxonomies. For instance, *relationship duration* concept is inferred from (Chen et al., 2015) and (Maamar et al., 2011b) approaches that refer to update or management of relationship.

[1] http://www.core.edu.au/

[2] Methontology enables building ontology from scratch

[3] ONIONS method define how to generate a unique ontology concepts from original ontologies concepts.

Table 1: Steps for identifying SWS properties.

Step 1	Identify SWSs approaches from a literature search
Step 2	Extract general concepts related to SWS from approaches identified in Step 1
Step 3	Short-list candidate definitions of selected SWS concepts in Step 2
Step 4	Refine list of SWS concepts, their corresponding definitions and relationships: output of this step is the initial SWS properties
Step 5	Validate SWS properties by their instantiation on SWS approaches selected in Step 1

Table 2: SWSs' properties defined and their use in the literature.

SWS properties	Approaches								
	Bansal et al. (Bansal and Bansal, 2011)	Bansal et al. (Bansal et al., 2010)	Maamar et al. (Maamar et al., 2011a), (Maamar et al., 2011b)	Maaradji et al. (Maaradji et al., 2011)	Maamar et al. (Maamar et al., 2012)	Xie et al. (Xie et al., 2008)	Chen et al. (Chen et al., 2015)	El-Goarany et al. (El-Goarany et al., 2008)	Li et al. (Li and Chen, 2010)
Node	WS Provider	WS Provider	WS	User and WS	WS's user and WS	User, WS, and Provider	WS	WS	WS, User and Provider
Relationship Type	Trust between WSs's providers	Trust between WSs's providers	Collaboration, substitution, and competition between WSs	Recommendation confidence of WS to a user	Recommendation of WS to a user, competition, collaboration, and substitution between WSs	Trust	Pattern of social link (e.g., Sequential, and choice Incoming/ Outgoing social link)	Similar and complementary WS	WSs similar functionalities or object relationships in the real world
Node degree	Reputation per WS	Trust rating per WS	×	×	×	Trust degree of WS, of WS user and of WS Provider	degree of node	×	Reputation of WS provider
Relationship weight	×	×	√	√	√	√	√	√	√
Relationship duration	×	×	√	√	×	×	√	×	×
Relationship transitivity	√	√	√	√	√	√	×	×	×
Relationship dependency	×	×	×	√	×	√	×	×	√

legend: √ means available property in the approach and × means the opposite.

- Task 3. Build a concept dictionary which will include all concepts, their corresponding definitions, their attributes, and relations.

In Step 3, we short-list the candidate definitions of the selected concepts in Step 2. This happens by identifying what knowledge the definition specifies and if the definition defines explicitly the word. For knowledge that is required in SWS social networks domain but not explicit, it is inferred from other definitions.

Differences between definitions were reconciled in order to ensure consistency across concept names. We adopt a definition that is most coherent to maintain generality when contradictory use of concepts between two or more literatures occurred. For instance, *node* refers to WS in (Chen et al., 2015) while it refers to WS and user in (Maamar et al., 2012) and to WS provider in (Bansal et al., 2010). Therefore, we define *node* as an entity that could be WS provider, WS consumer, or WS. The final output of Step 3 is a refinement of the list of concepts obtained in Step 2 with their corresponding definitions.

In Step 4, we refine the list of SWS concepts obtained in Step 3 and identify dependencies between SWS concepts. The final output of this step is the following defined key properties of SWS.

- **Node** refers to an entity (physical or moral) that could be WS provider *(e.g., organization)*, WS consumer *(e.g., user)*, or WS as well. In the rest of this paper, we use entities and nodes interchangeably.
- **Relationship** establishes a link (or association) between pairs of nodes. More than one relationship can exist between nodes (e.g., collaboration and substitution), whether the nodes are of the same kind or not. In the rest of this paper, we use relationship and association interchangeably.
- **Relationship Type** specifies the label of a relationship. For instance, *competition, collaboration, and recommendation* relationships between WSs, *trust* relationship between WSs customers, WSs providers, and WSs themselves, and *friendship* or *business partnership* relationships between WSs providers and WSs consumers.
- **Relationship Weight** evaluates the relationship between two nodes in terms of how collaborative they are, how cooperative they are, etc. This evaluation may depend on the number of interactions that involve nodes.
- **Relationship Duration** establishes how long a relationship between nodes will last. This could be *permanent* like similarity between WSs' functionalities or *temporary* like upon-request partnership.
- **Relationship Transitivity** determines relationship type that supports transitivity *like similarity between WSs*. Transitivity cycles may be limited by a threshold that is defined by WS provider.
- **Relationship Dependency** captures the reliance of a relationship on another. For instance, trust relationship between two WSs in (Xie et al., 2008) relies on existing relationships between their WS users and WS providers.
- **Node Degree** reflects the social qualities of a node as it interacts with other nodes. A social quality of node is a mapping between a set of social parameters (e.g., trustworthy, and cooperative) onto a set of their values. For instance, a node social quality can be used to build its reputation degree and help selection in the case of competition.

In Step 5, we validate the SWS properties by their instantiation on SWS approaches selected in Step 1. Table 2 summarizes how the defined SWS properties are mapped onto concepts reported in the SWS literature.

3.2 MDA for Social Web Service Description

In this section, we propose a MOF-based ***S***-WSDL metamodel for abstracting SWS's properties. ***S***-WSDL metamodel consists of refining WSDL metamodel at the PIM level. In MDA terms, WSDL model is an instance of WSDL metamodel. Likewise, ***S***-WSDL model (i.e., WSDL model with social dimension) is an instance of ***S***-WSDL metamodel. These metamodels are also instances of the MOF model (Fig. 1). A ***S***-WSDL metamodel's concepts are identified and grouped according to *Interaction* and *service* views.

3.2.1 Interaction View

Interaction View models the aforementioned proposed SWS properties. The basic view of the *Interaction* metamodel is illustrated in Fig. 2. A metaclass defines the behavior of certain classes and their instances. 1 or 1..* cardinalities denote the required association, while 0..1 or 0..* cardinalities denote optional associations. 'SN" for "Social Network" is added as a prefix to name the new *Interaction* metamodel classes. The following outlines the role of the key metaclasses, their metaattributes, and metafunctions.

- *SNNode* refers to *node*. It is characterized by *name*, *state* and a set of *properties* represented by *SNProperty* metaclass.

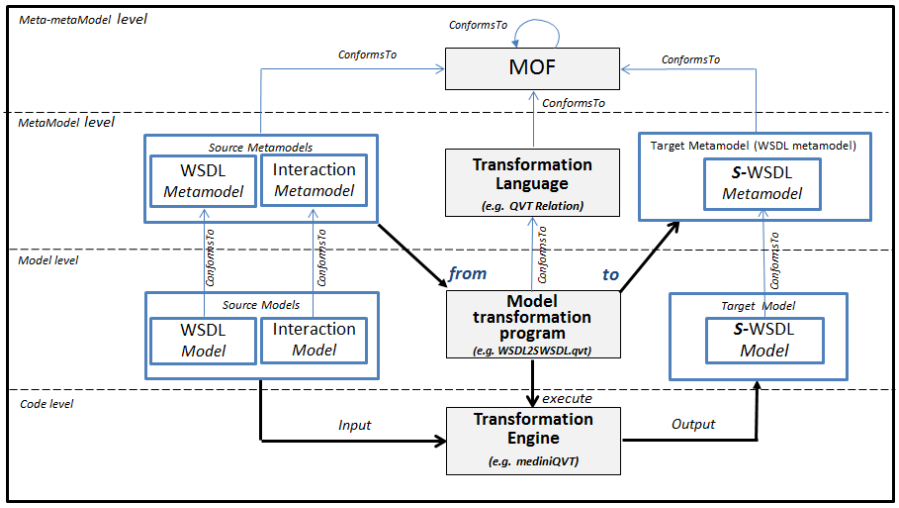

Figure 1: Overview in using MDA for SWS description and transformation.

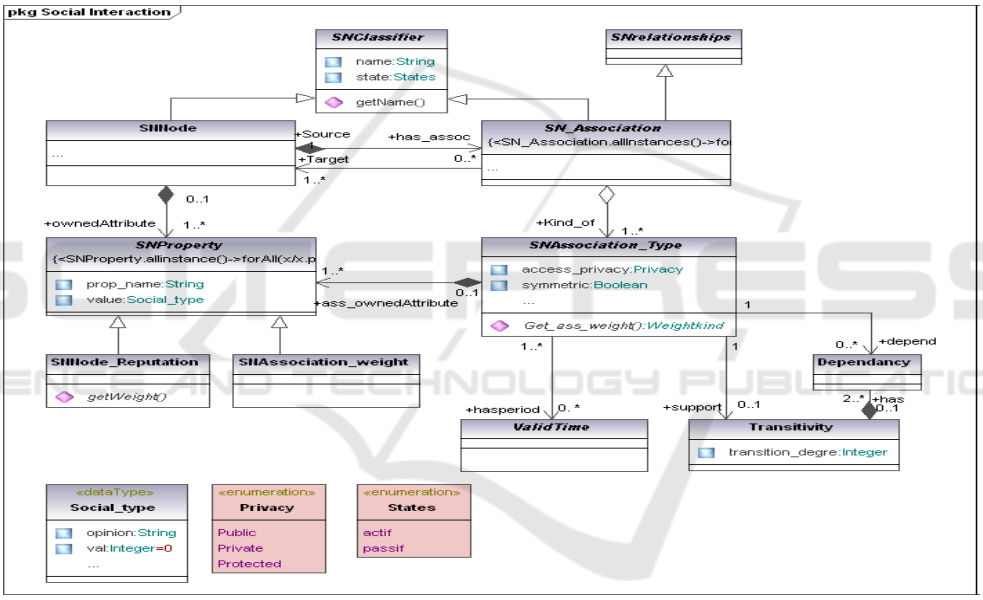

Figure 2: Interaction core metamodel View.

- *SNAssociation_Type* refers to *relationship type* and aims to identify the nature of the social relationships *(e.g., collaboration and recommendation relationship as value instance of the SNAssociation_Type metaclass)* and the representation of information regarding symmetry, transitivity, dependency, and temporal aspects.
- *SNAssociation_Weight* specializes the metaclass *SNproperty*. In this case, *prop_name* attribute refers to the label of *association_weight (e.g., collaboration weight as value instance of association_weight metaclass)* and *Value* attribute to its weight. This value depends on the relationship type and is calculated using the metafunction *Get_ass_weight*.

3.2.2 Service View

Service View extends WSDL metamodel with *Interaction* to create ***S***-WSDL metamodel. WSDL metamodel is generated from WSDL 2.0 XML schema description. WSDL 2.0 XML schema allows elements representing a specific technology, called *extensibilityelements*, under various elements defined by WSDL. ***S***-WSDL metamodel gives rise to social characteristics as an optional entry for WSDL metamodel. Thus, any service *(e.g., Prepare_Job)* can refer to services *(e.g., Post_Job, and Plan_Job)* with whom it has a relationship *(e.g., collaboration)* using *SNNode* and *SN_Association* metaclasses.

4 IMPLEMENTATION AND VALIDATION

In order to validate our proposed MOF-based ***S***-WSDL metamodel, we need to prove its completeness. However, incompleteness is a fundamental problem in an open environment such as the Social Web. In fact, we cannot prove the completeness of a proposed MOF-based ***S***-WSDL metamodel nor the completeness of its definitions, but we can prove the incompleteness of an individual SWS property or its definition, and therefore we can deduce the incompleteness of the proposed metamodel if at least one SWS property definition is missing in the established ***S***-WSDL metamodel. So, ***S***-WSDL metamodel is complete if and only if all that is supposed to be in the SWS is explicitly stated in it, or can be inferred for various social WS scenarios.

We validate our ***S***-WSDL metamodel by showing its applicability and relationships to existing SWS approaches such as (Maamar et al., 2011b) and (Xie et al., 2008) described in Table 2.

Upon MOF compliant metamodels for sources *(e.g., WSDL, and Interaction metamodels)* and target *(e.g., **S**-WSDL metamodel)* introduced, *Model Transformation* can be carried via transformation rules (not included here for space reasons) between MOF compliant metamodels, and implemented via a *transformation engine* (Fig. 1). Transformation rules are described in QVT Relation model transformation language. The use of QVT Relation language provides an automatic transformation of WSDL model into ***S***-WSDL model, and support target incremental model transformation. Thus, the formal semantics of MOF and QVT Relation language reflects the conformance relation between models and metamodels, and the satisfaction of transformation rules between pairs of models (Calegari and Szasz, 2013).

We have implemented a prototype in Eclipse Modeling Framework (EMF)[4] and integrates *mediniQVT*[5] plugin. This later supports transformations expressed in QVT Relation. The WSs models used are built on real-world WSs extracted from WS-DREAM[6] a Web Service QoS Datasets (Zhang et al., 2010) and converted to XMI standard representation. XMI representation of Interaction models is created from Interaction Ecore diagram (e.g., using Eclipse Rich Client Platform). We executed our transformations on JVM version 1.8, running on MicrosoftWindows 7 Professional, in computer system of Intel(R) Core (TM) i5-920CPU, running at 2.67 GHz, and 4 GB of RAM.

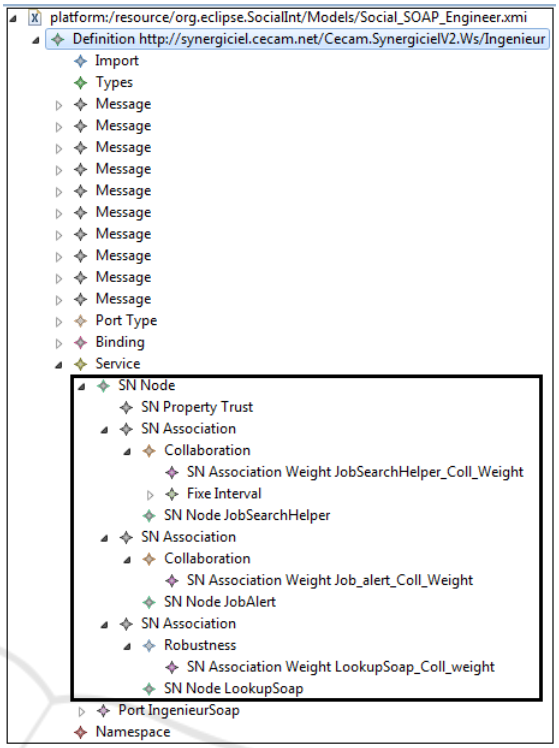

Figure 3: An excerpt of Social_SOAP_engineer ***S***-WSDL model.

Transformation takes as inputs *SOAP_engineer* WSDL model and its Interaction model and returns *Social_SOAP_engineer* ***S***-WSDL model (Fig. 3). QVT evaluation created 1 new element, set 19 features and takes 250 ms. *Social_SOAP_engineer* ***S***-WSDL model is validated using EMF validation, as instances of WSDL metamodel.

The social dimension of *Social_SOAP_engineer* ***S***-WSDL is described, as optional elements in WSDL, inside dash line shape in Fig. 3. It includes *Trust* property as in (Xie et al., 2008) approach and *Collaboration* and *Robustness* relationship type as in (Maamar et al., 2011b) social networks approach:

- Collaboration with *JobAlert* and *JobSearchHelper* SWSs. The collaboration weight value between *SOAP_engineer* and *JobAlert* is labeled *Job_alert_Coll_Weight*, while collaboration weight value between *SOAP_engineer* and *JobSearchHelper* is labeled *JobSearchHelper_Coll_Weight*.
- Robustness with *LookupSoap* SWS. The latter has similar functional and non-functional properties with *SOAP_engineer* and can replace it in case of failure. The robustness relationship weight value is labeled LookupSoap_Rob_weight.

[4]http://www.eclipse.org/modeling/emf/
[5]http://projects.ikv.de/qvt
[6]http://www.wsdream.net/

5 CONCLUSION

In this paper, we identified SWS properties and proposed a MOF-based ***S***-WSDL metamodel for the description of SWS. ***S***-WSDL metamodel is defined to ensure consistency between different SWS application models and extensibility in terms of new interactions, regardless of the implementation platform. MOF-based ***S***-WSDL metamodel and QVT Relation are used to support automatic transformation from WSDL model into ***S***-WSDL model (WSDL with social dimension) without altering the original content of WSDL model. We implemented a prototype to test our approach. The prototype illustrates how the MOF-based ***S***-WSDL metamodel is defined and how to automate the transformation of a WSDL model into a S-WSDL model, using EMF and QVT Relation tools. Furthermore, the proposed ***S***-WSDL can be applied to serve different purposes such as adding social dimension when querying WSs registries and mapping from WSs model (e.g., existing UML models of BPELs process) to WSs with social dimension. As future work, we aim to validate our metamodel in real-world use-cases.

REFERENCES

Object Management Group, MDA Guide Revision 2.0.

Object Management Group, OMG Specifications.

Bansal, S. K. and Bansal, A. (2011). Reputation-Based Web Service Selection for Composition. In *World Congress on Services*. IEEE.

Bansal, S. K., Bansal, A., and Blake, M. B. (2010). Trust-based Dynamic Web Service Composition using Social Network Analysis. In *Workshop on: Business Applications of Social Network Analysis*. IEEE.

Beydoun, G., Low, G. C., Henderson-Sellers, B., Mouratidis, H., Gómez-Sanz, J. J., Pavón, J., and Gonzalez-Perez, C. (2009). Faml: A generic metamodel for mas development. *IEEE Trans. Software Eng.*, 35(6).

Bézivin, J., Hammoudi, S., Lopes, D., and Jouault, F. (2004). Applying MDA Approach for Web Service Platform. In *8th International Enterprise Distributed Object Computing Conference*. IEEE.

Bouchakour, E. H. and Benslimane, S. M. (2013). Social Web Services Development Based on MDA:Extending WSDL to Inject Social-QoS. In *14th Arab Conference on Information Technology, Proceedings*.

Calegari, D. and Szasz, N. (2013). Institution-Based Semantics for MOF and QVT-Relations. In *Formal Methods: Foundations and Applications*. Springer.

Chen, W., Paik, I., and Hung, P. C. K. (2015). Constructing a Global Social Service Network for Better Quality of Web Service Discovery. *IEEE Transactions on Services Computing*, 8(2).

Chung, J.-Y., Lin, K.-J., and Mathieu, R. G. (2003). Guest Editors'Introduction : Web Services Computing-Advancing Software Interoperability. *IEEE Computer*, 36(10).

D'Ambrogio, A. (2006). A Model-driven WSDL Extension for Describing the QoS of Web Services. In *International Conference on Web Services*. IEEE.

El-Goarany, K., Saleh, I., and Kulczycki, G. (2008). The Social Service Network - Web 2.0 Can Make Semantic Web Services Happen. In *10th Conference on E-Commerce Technology & Enterprise Computing, E-Commerce and E-Services*. IEEE.

Gómez-Pérez, A., Fernández-López, M., and Corcho, Ó. (2004). *Ontological Engineering: With Examples from the Areas of Knowledge Management, e-Commerce and the Semantic Web*. Springer.

King, I., Li, J., and Chan, K. T. (2009). A brief survey of computational approaches in social computing. In *International Joint Conference on Neural Networks*. IEEE.

Li, S. and Chen, Z. (2010). Social Services Computing: Concepts, Research Challenges, and Directions. In *Int'l Conference on Green Computing and Communications,& Int'l Conference on Cyber, Physical and Social Computing*. IEEE.

Maamar, Z., Faci, N., Sheng, Q. Z., and Yao, L. (2012). Towards a User-Centric Social Approach to Web Services Composition, Execution, and Monitoring. In *Web Information Systems Engineering*. Springer.

Maamar, Z., Hacid, H., and Huhns, M. N. (2011a). Why Web Services Need Social Networks. *IEEE Internet Computing*, 15(2).

Maamar, Z., Wives, L. K., Badr, Y., Elnaffar, S., Boukadi, K., and Faci, N. (2011b). LinkedWS: A Novel Web Services Discovery Model Based on the Metaphor of "Social Networks". *Simulation Modelling Practice and Theory*, 19(1).

Maaradji, A., Hacid, H., Skraba, R., Lateef, A., Daigremont, J., and Crespi, N. (2011). Social-Based Web Services Discovery and Composition for Step-by-Step Mashup Completion. In *International Conference on Web Services*. IEEE.

Simon, B., Goldschmidt, B., and Kondorosi, K. (2013). A Metamodel for the Web Services Standards. *J. Grid Computing*, 11(4).

Xie, X., Du, B., and Zhang, Z. (2008). Semantic Service Composition based on Social Network. In *Proceedings of the 17th International World Wide Web Conference*. Springer.

Zhang, Y., Zheng, Z., and Lyu, M. R. (2010). WSExpress: A QoS-aware Search Engine for Web Services. In *International Conference on Web Services*. IEEE.

Zheng, X., Wu, Q., Ke, D., Li, H., and Shi, Y. (2012). Social Context Enabled Description Model for Web Services. In *Information Computing and Applications conference*. Springer.