



HAL
open science

Unifying Web Services Communities Modelling Through a UML Profile

Hela Limam, Jalel Akaichi

► **To cite this version:**

Hela Limam, Jalel Akaichi. Unifying Web Services Communities Modelling Through a UML Profile. 13th International Conference on Wired/Wireless Internet Communication (WWIC), May 2015, Malaga, Spain. pp.357-366, 10.1007/978-3-319-22572-2_26 . hal-01728823

HAL Id: hal-01728823

<https://inria.hal.science/hal-01728823v1>

Submitted on 12 Mar 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Unifying Web services communities modelling through a UML profile

Hela Limam¹, Jalel Akaichi²

¹Institut Supérieur d'Informatique
Université Tunis El Manar

hela.limam@isg.rnu.tn
²Institut Supérieur de Gestion

jalel.akaich@isg.rnu.tn

Abstract. Many works were interested in studying the ways of organizing Web services into communities which allowed the onset of a variety of definitions and models. The new paradigm, launched by the use of Web services communities models was adopted by users in various fields such as those related to marketing, commerce, health care, etc. However, the proposed conceptual models suffer from dispersed points of view that have to be unified in order to offer generic conceptual support for experts and clerical users. The purpose of this work is to propose a unified conceptual model able to unify different points of view through a generic UML description profile of Web services communities characteristics well adapted to new concepts imposed by the use of communities. Thanks to the proposed unified model, users will be able to build themselves their communities regardless the case of study involving Web services.

Keywords: Web services, communities, profile, modelling.

Introduction

The emergence of Web services in the information space raises many difficulties. Web services are developed by different entities and there is no consensus on the way to use services descriptions. In this context, the concept of community offers a solution for structuring and organizing a large set of unstructured Web services sharing the same area of interest. Indeed, the community infrastructure contributes to improving the availability of Web services and provides centralized access to distributed Web services, the organization into communities raises a modelling problem resulting from the lack of a generic tool for the specification and the modelling of Web services communities. We propose in this work a unified model for representing different aspects of Web services communities through a UML profile. We illustrate the useful-

ness of the proposed profile notation using a health care community example designed with the proposed modelling tool.

The paper proceeds as follows. In section 2 we argue our choice of the notion of the community and we situate our approach regarding the others proposed in the literature. The methodology for Web services communities modelling is described in section 3. The unified model is detailed in section 4. In section 5 we propose the Web services community model. In section 6 we illustrate our approach by a running example related to a health care community. Finally section 7 concludes our work and gives future directions of our research.

State of the Art

According to Wan et al., 2010, communities of Web services are defined as virtual spaces that can dynamically gather different Web services having complementary or related functionalities [1]. In this context, an approach is used to verify Web services communities modeled in UML activity diagram [2]. The first step consists on translating the activity diagram into an interpreted system model using predefined transformation rules. Then, specifications are expressed as formulae in a logic extending the Computation Tree Logic CTL with agent commitments needed for their communication. Finally, both the model and formulae are used as inputs for the multi-agent symbolic model checker. Yahyaoui et al., 2013 go beyond the definition of the concept of Web services communities and give some direct benefits that out of the adoption of communities of Web services [2]. Another approach for modeling Web services communities [3], is proposed by Sellami et al., 2013. It organizes Web service registries into communities according to the functionalities of the Web services in order to address the problem of the growing number of Web service registries and their poorly organized network.

Benharref et al., 2010 [4] proposed a managerial community of Web services for managing communities of Web services. This community is able to monitor, certify and assess the quality of Web services from other communities. A Web service will be allowed in the community if the quality of service is high or rejected if the quality of service is low. In this paper, the authors focused on the use of the managerial community in the selection of web services based on quality of service. In [5] Bouchaala et al., 2011 proposed a graph-based approach for managing

communities of Web services registries. This approach consists of a set of algorithms and operations management. Management operations are verified at each stage of the life cycle of the community (create, delete, merge, split). Authors in [5], use communities to group records of Web services. Indeed, a community of Web services registries is a set of records with Web services that provide similar functionality. The concept of Web services community highlights the importance of management and organization of Web services in a community.

To conclude, Web services communities modeling issues have been addressed in different ways according to several approaches. However, few studies have focused on proposing a comprehensive conceptual modeling of various aspects of functioning of communities of Web services.

Methodology for Web services communities modelling

In order to provide a formal description of the Web services community, we adopted the Model-driven architecture (MDA). In fact, MDA offers a conceptual framework that separates the organizational process from the technology platform decision [6]. A significant feature of the MDA is the use of automated tools and services to define the models (abstractions) and facilitates the transformations between different models. A UML profile for abstracting business information and business service specification was proposed [7], along with XMI's profile for selecting XML production options, exemplifying the MDA approach. Generators based on mappings of these languages to XML, WSDL, and other implementation technologies form the other crucial piece of an MDA framework for Web services. To cope with the specific aspects of Web services several design notations have been proposed ([8],[9],[10]; [11]). While most of the proposed notations agreed on a set of concepts specific to Web services, none of them addresses the issue of Web services communities. This shortage motivated our work in proposing a Web services communities modelling approach which increases the expressiveness of UML for Web services communities and guides their design. Stereotypes and graphical annotations have been added to UML diagrams in order to distinguish between the different aspects in a Web services community.

The Unified Model

The design process of profile consists on defining a set of stereotypes and icons appropriate to Web services communities, then attaching constraints to profile elements in order to enhance its expressiveness. Our conceptual model provides a high level of abstraction and is completely software or hardware independent.

4.1 The Unified Model Visual Elements

A UML profile is a package that belongs to the extension mechanism. This package is stereotyped <<profile>> and the extension mechanism is allowed thanks to UML characteristics. In fact, the latter serves as foundation for building domain specific languages by specifying stereotypes, which introduce new language primitives by subtyping UML core types and tagged values, which represent new properties of these primitives. Model elements are assigned to such types by labeling them with the corresponding stereotypes.

The main modeling element of our system is the class stereotyped <<community>> used to represent a Web services community as a collection of Web services related to the same area of interest. A community is identified by a stereotype <<ID-parameter>> representing an abstract definition of the data used to identify a community. Data used to identify the community is essentially its category which describes the area of interest of the community.

The stereotype <<Web service>> is an abstraction of services providers which become members of the community while accomplishing the registration process. The community is accessible via a set of operations.

The stereotype <<operation>> represents abstract operations that summarize the major functions needed by community members. Community providers define generic operations based on their expertise on the corresponding area of interest.

Quality of Service specifies a set of quality requirements on the behavior of a community. To model this aspect, we used a class stereotyped <<quality of services>> that offers run time, business and security attributes.

In addition to proposed stereotypes, we attached an icon to each of them in the behavioral, structural and interactional diagrams as de-

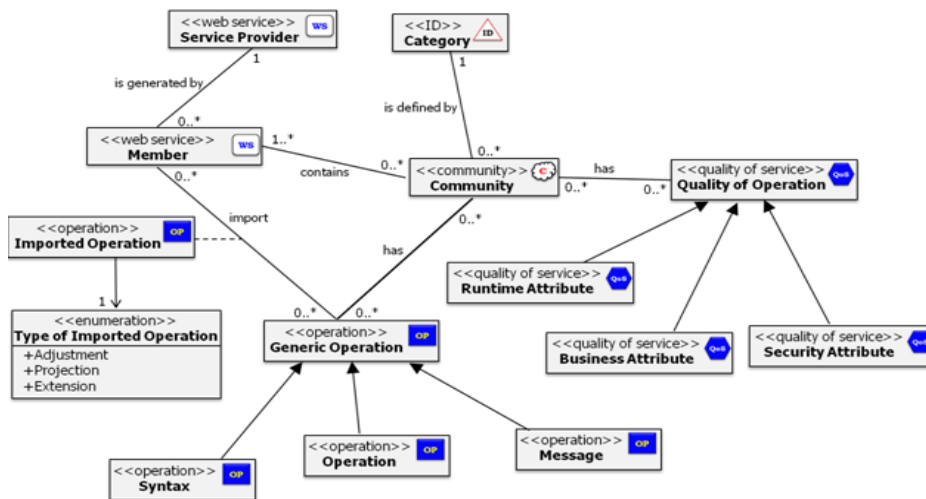
scribed in table 1. Stereotypes and icons associated to them constitute the core of our proposed design language.

Icon	Stereotype	Concept
	<<community>>	Community of Web services
	<<ID-parameter>>	Identification parameter of the community
	<<web service>>	Web service
	<<operation>>	Operations of the community
	<<quality of service>>	Quality of services related to the community

Table 1. Web services community notation

In order to cover specific perspectives related to Web services communities, we defined a UML based metamodel using the design notation introduced in the profile elements. This includes attaching a stereotype for each needed element inside a package named “profile”. Only extended elements shall be represented as stereotypes. The proposed metamodel is given in (figure 1).

Fig. 1. Web services community metamodel



We have extended the class diagram to represent the structure of a community of Web service. A community is associated with a specific domain and is identified by a category and an identifier. It dynamically collects different Web services with similar functionality. The Web service providers, who register their Web services in a community, represent the communities' members.

A community is accessible via a set of generic operations. Community services providers define these operations based on their expertise in the field of community. A generic process is defined by a set of functional and non-functional attributes. Functional attributes describe the syntactic functions, which represent the structure of a generic operation, and semantic features, which refer to the significance of the operation or its messages. The semantic attributes are static or dynamic. The quality of operation is based on a set of qualitative attributes which are transverse to the overall operations such as cost and the response time.

4.2 The Profile Constraints

For instance, constraints should be added to the aforementioned stereotypes to govern the metamodel structure.

In fact, constraints can be associated to stereotypes imposing restrictions on the corresponding metamodel elements. In this way, a designer can define the properties of a "well formed" model. In our work we adopted OCL (Object Constraint Language) for expressing constraints since it is a declarative language that describes constraints on UML metamodels [11].

As a constraint language, OCL allows to specify among other invariants on classes and types data in a UML model to express the pre and post conditions of operations and methods, defines invariants on stereotypes of a UML profile.

In our work, we mainly use the OCL invariant type to express constraints on meta-level UML stereotypes profile. An OCL invariant (keyword *inv*) is defined in the context of a class or a stereotype and specifies the conditions that must be met by its instances.

The context of this constraint definition is the class (keyword *context*). The keyword *self* is used in OCL to reference the current object of the specified class in the context; *Implies* the operator corresponds to the logical implication.

Using stereotypes, icons and constraints, we define a profile that helps us in the design of the Web services community.

In the following, we used the OCL syntax to describe some constraints associated to the presented metamodel in order to enhance its expressiveness while modeling a Web services community.

We express in the following constraints attached to the stereotype <<community>>

- A community has exactly one category: `context Community
inv: self.ownedMember->select(m|m.ocllsTypeOf
(Category))>size()=1`
- A community has exactly one Identifier: `context Community
inv: self.ownedMember->select(m|m.ocllsTypeOf
(Identifier))->size()=1`

Constraints attached to the stereotype <<web service>>

- The class << web service >> is only associated to the class << community >> or <<web service>> or <<generic operation>>:
`self.allOppositeAssociationEnds->forall(participant.ocllsTypeOf or ocllsTypeOf(web service) or ocllsTypeOf(generic operation))`

Constraints attached to the stereotype identifier << ID >>

- The class stereotyped << ID >> is only associated to the class << community >>: `self.allOppositeAssociationEnds->forall(participant.ocllsTypeOf (community))`

Constraints attached to the stereotype <<generic operation>>

- Class <<generic operation>> is only connected to <<community>> or <<web service >>: `self.allOppositeAssociationEnds->forall(participant.ocllsTypeOf (Community) or ocllsTypeOf(web service))`
- An element imported operation is an association between a departure element member and a receiver element generic operation: `context Imported Operation
inv : (self.client.ocllsKindOf (member)) and
(self.supplier.ocllsKindOf (generic operation))`

Constraints attached to the stereotype << quality of service >>:

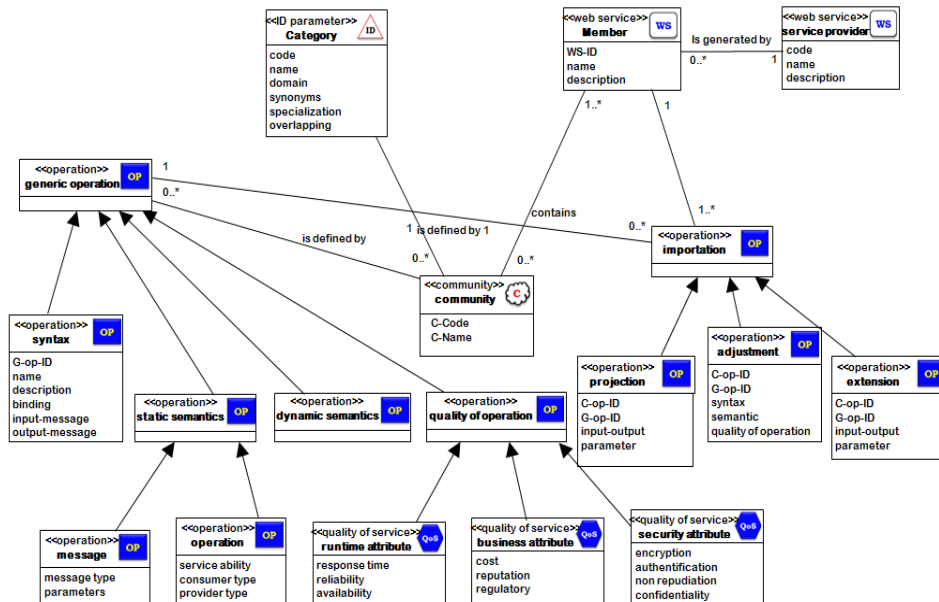
- The class `<< quality of service >>` can be only associated to the class `<< community >>`: `self.allOppositeAssociationEnds->forall(participant.oclIsTypeOf(Community))`

After the definition of the metamodel with its OCL constraints, we propose in the next step to instantiate the metamodel in order to generate a model of a Web services community.

The Web services community model

The Web services model is built upon the Web services community classes. To represent the Web services community classes, the current prototype allows the user to draw class diagrams according to the language notation presented in table 1. We begin with loading the UML meta-model. This later offers an editor for viewing and editing UML models. The edited model for the representation of the Web services community classes and description of relationships is given in the Web services community class diagram (figure 2).

Fig. 2. Extended UML Web services community class diagram



The community is defined by an identifier, a category and a set of generic operations. All Web services that belong to community class have the same category. The category of a community is formally defined by a set of attributes (Domain, Synonyms, Specialization, Overlapping). Domain gives the area of interest of the community (e.g., “healthcare”). It takes its value from taxonomy for domain names. For flexibility purposes, different communities may adopt different taxonomies to specify their category. Specialization is a set of characteristics of the community domain. For example, “Patient Administration Services” and “Patient Care Services” are specialization of “healthcare”. Communities are generally not independent. They are linked to each other via association relationships. These relationships are specified in the Overlapping attribute.

The community is accessible via set of operations called generic operations. Community providers define generic operations based on their expertise on the corresponding area of interest. A generic operation is defined by a set of functional and non-functional attributes. Functional attributes describe syntactic and semantic features of generic operations. Syntactic attributes represent the structure of a generic operation. An example of syntactic attribute is the list of input and output parameters that define the operation’s messages. Semantic attributes refer to the meaning of the operation or its messages. We consider two types of semantic attributes: static and dynamic semantic attributes. Static semantic attributes (or simply static attributes) describe non-computational features of generic operations. Those are semantic attributes that are generally independent of the execution of the operation. An example of static attribute is the operation’s category. Dynamic semantic attributes (or simply dynamic attributes) describe computational features of generic operations. They generally refer to the way and constraints under which the operation is executed. An example of dynamic attribute is the business logic of the operation i.e., the results returned by the operation given certain parameters and conditions.

Service providers can, at any time, select a community of interest based on categories and register their services with it. The registration process requires giving an identifier (WS-ID), name, and description for the Web service. Multiple Web services that belong to the same community may import the same generic operation. Hence, it is important to define a set attributes that help select the “best” Web service supporting a given functionality. For this purpose, we define a Quality of Service

(QoS) model based on a set of qualitative attributes that are transversal to all operations such as the cost and response time.

UML profile realization

A UML profile is a package belonging to the extension mechanism. This package is stereotyped "Profile" and is written in XML. In StarUML platform we added an approach called 'CSW Approach' and a UML profile for Web services community composed of three models or diagrams are: CSWClassDiagram and CSWUeCaseDiagram and CSWSequenceDiagram. Indeed, we created two XML files, the first one is used to define the approach and the second one is used to create the profile. When we start StarUML, the 'New Project by approach' appears to select the desired approach. Our proposed approach called 'CSW Approach' is integrated in the platform. Consequently, it can be selected as all the approaches of the platform (figure 3).

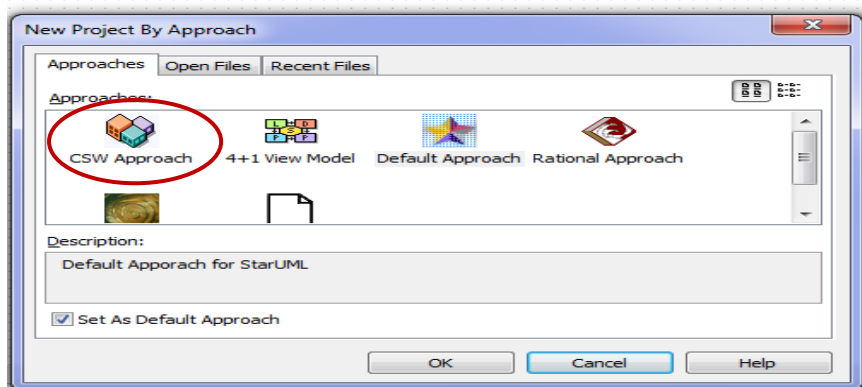


Fig. 3. CSW approach

After selecting the approach, we find in the "Model Explorer" of StarUML models called behavioral CSW model, structural model CSW and interactional CSW Model representing the functional, structural and interactional aspects of a Web services community.

In order to exemplify how the previously described model can be applied to a specific field we chose a running example related to a health care community. In our work, we followed the UML profiling mechanism

Figure 4 describes the general class diagram for the health care community called 'CSWClassDiagram HealthCareCommunity'. The main

domain classes are community health care, community members and health care provider health services.

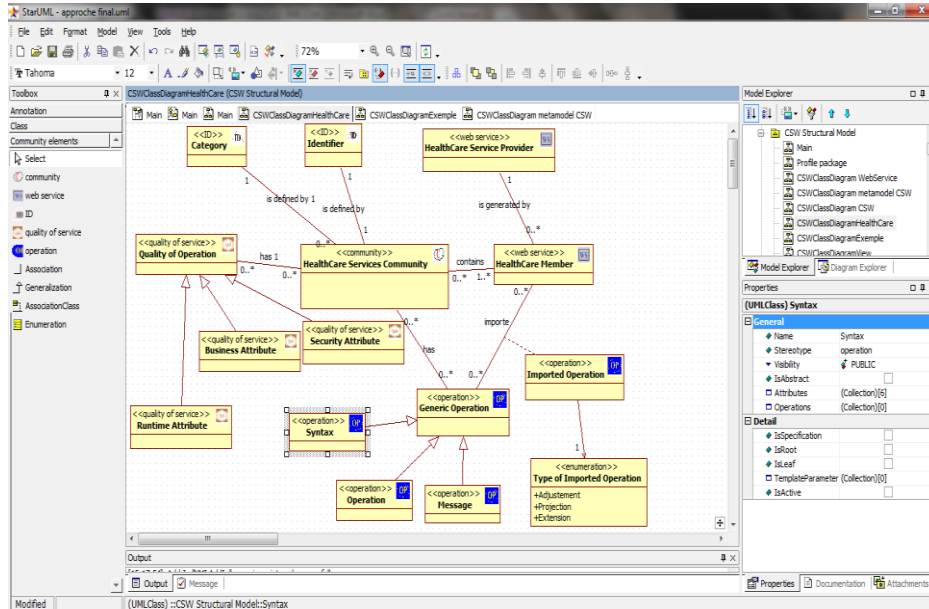


Fig. 4. CSWClassDiagram HealthCareCommunity

A Health care community is designed to combine data from the large ancillary services, such as pharmacy, laboratory, and radiology, with various clinical care services. It has an identifier and is described by a set of attributes. It is composed of a set of Health care Web services: Insurance Service, Care Service, Patient Referral Service, Physician Referral Service and scheduling Service. Web services inside a community are associated to each other's with peer relationships.

Each Web service is modelled as a class in the extended class diagram and is described by a set of attributes as shown in figure 4. The number of integrated Web services involved in the Health care community is dependent upon the data structures and has to provide an interface that allows clinicians to access the systems through a portal.

The Health care community portal must be able to schedule patient visit with physicians. The system also records drugs prescribed during the visit. Additionally patient referral services should be permanently updated and the information is used to track the visit.

7 Conclusion

In this work, we proposed a generic model of Web services communities. It consists on a UML profile destined to describe Web services communities using specific and well-adapted stereotypes and notations. Based on the general structure of the proposed model, we described a community of health care Web services necessary to ensure the community main requirements. The modelling of Web services communities was performed using a UML profile based on enriched diagrams; however enhancing the model with other behavioural diagrams and context-aware components, such as activity diagrams extended with mobility and context-aware features, is necessary for proposing a full design methodology starting from the requirement specifications to code generation.

References

1. Wan, W., Bentahar, J., Ben Hamza, A.: Modeling and Verifying Agent-based Communities of Web Services. Proceedings of the 23rd International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems . Cordoba, Spain(2010).
2. Wang, Y., Zhang, J., Vassileva, J.: Effective Web Service Selection via Communities Formed by Super-Agents. IEEE / WIC / ACM International Conferences on Web intelligence(2010) 549-556. Toronto. doi: 10.1109/WI-IAT.2010.221
3. Yahyaoui, H., Maamar, Z., Erbin, L., Thiran, P.: Towards a community-based, social network-driven framework for Web services management. Future Generation Computer Systems (2013) 29(6), 1363-1377.
4. Benharref, A., Serhani, M.-A., Bouktif, S., Bentahar, J.: A Managerial Community of Web Services for Management of Communities of Web Services. 10th international conference on New Technologies of Distributed Systems(2010) 97-104.
5. Bouchaala, O , Sellami, M. , Gaaloul, W., Tata, S., Jmaiel, M.: Modeling and Managing Communities of Web Service Registries, WEBIST (2011)88-102.
6. Frankel, D., Parodi, J.:Using Model-Driven Architecture to Develop Web Services. The John Wiley & Sons OMG Series (2002).
7. Provost, W.:UML for Web services. Retrieved from <http://www.xml.com/pub/a/ws/2003/08/05/uml.html> (2003).
8. Ortiz, G., Hernandez, J.:Toward UML Profiles for Web services and their Extra-functional Properties. IEEE International Conference on Web services Chicago. doi:10.1109/ICWS.2006.130 (2006) 889-892.
9. Belouadha, F., Roudiès, O. : Un profil UML de spécification de services Web composites sémantiques. 1ère Conférence Internationale Systèmes d'Information et Intelligence Economique. Tunisia (2008) 14-16.
10. Skogan, D., Gronmo, R., Solheim, I.:Web service Composition in UML. Proceedings of the Enterprise Distributed Object Computing Conference. Washington (2004) 47-57.