ISSN No. 0976-5697

Volume 2, No. 2, Mar-Apr 2011

International Journal of Advanced Research in Computer Science

REVIEW ARITICAL

Available Online at www.ijarcs.info

WSC-UML: A UML Profile for Modeling Web services communities A Health Care Case Study

Hela Limam*
Dept of Computer Science. High institute of management
SOIE, university of Tunis
41, Rue de la Liberté, Cité Bouchoucha
Le Bardo 2000,Tunisia
hela.limam@isg.rnu.tn

Jalel Akaichi
Dept of Computer Science. High institute of management
SOIE, university of Tunis
41, Rue de la Liberté, Cité Bouchoucha
Le Bardo 2000,Tunisia
jalel.akaichi@isg.rnu.tn

Wided Oueslati
Dept of Computer Science. High school of management
University of Tunis
41, Rue de la Liberté, Cité Bouchoucha
Le Bardo 2000,Tunisia
Wided.oueslati @isg.rnu.tn

Abstract: In today's collaborative environments such as bioinformatics and healthcare, the access to an increasing number of distributed data sources is an important step towards enhancing decision support systems. In this context, we propose a mediation framework for integrating distributed Web services into communities and we show how the use of communities provides a scalable solution for gathering and managing functionally-equivalent Web services. Hence, in this paper we first propose a model of Web services communities. Secondly, as the specificities and design complexity of Web services communities call for a language and tool support to guide a Web services community design we propose a WSC-UML profile for the design of web services communities .The proposed tool extends UML to express specific concepts of web services communities. Finally, the paper illustrates the usefulness of the proposed profile notation using a health care community example designed with the WSC-UML Tool

Keywords: Web services, communities, UML extension, health care, model.

I. INTRODAUCTION

In today's medical environment, information technology was adopted for supporting digitized equipment, administrative tasks, and data management but less has been achieved in the use of computational techniques to exploit the medical information in research or practice. There is an emerging demand for the integration and exploitation of heterogeneous biomedical information for improved clinical practice, medical research and personalized healthcare [1].

In practice, biomedical information is spread over many independent data sources and there is no sharing of common knowledge among the different information sources. In this context, web services appear to be a privileged mean to assist the federation of distributed data sources into dedicated communities to share common knowledge between members of a health care community. In fact, medical information must be available and sharable among different members. To fulfill theses requirements, we propose a framework through which web services communities are built and interact with one another. This enabled a potentially large number of health information sources to act as a community to serve knowledge sharing.

The approach is based on Web services architecture. It uses the notion web services communities where services catering for similar customer needs are grouped together and form a single community. Information sources register themselves to a health care service community as members and the communities themselves interact with each others. The community can be viewed as a domain specific data integration mediator which holds metadata and registry information about its members

Our work, so far, has focused on designing Web services communities. In this paper, we draw the general design of the system.

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. Web services communities requirements are described in section 3. The associated meta- model is described in section 4. In section 5 we detail our proposed model. In section 6 we illustrate our approach by a running example related to a health care community. Section 7 describes our system implementation. Finally section 8 concludes our work and gives future directions of our research

II. STATE OF THE ART

A community has been defined as a group of people living together or united by shared social interactions, social ties, and a common 'space' [2]; as a social network of relationships that provide sociability support, information, and a sense of belonging [3], and as a set of relationships where people interact socially for mutual benefit [4]. The term community is not particular to Web services. In grid computing for example, solutions for sharing resources in a grid rely on communities [5].

When applied to Web services, communities help gathering Web services that provide a common functionality in order to simplify the access to Web services via a unique communication endpoint, which is the access point to the community. In [6], the concept of community gathers services from the same domain of interest and publishes the functionalities offered by Web services as generic operations. The authors provide a general template referred to as community ontology for describing semantic Web services and communities.

Web services communities appear to be a solution towards reshaping online communication and collaboration between Web services. However, the organization into communities raises management issues: how to initiate, set up, and specify a community of Web services. Several works gather functionally-similar Web services into communities that are accessed via a common interface and propose solutions for tackling Web services communities' management issue described above.

Such a solution is proposed in SELF-SERV framework [7], which distinguishes 3 types of services: elementary services, composite services, and service communities. The fundamental element of a Web Service Community (WSC) according to [8] is subject-club which is something like a special service container in which some services localized in different place across the globe but with similar domain interest are clustered.

Subject-club ontology, one important part of the hierarchical community ontology, is divided into club local ontology and web service private ontology, respectively characterizing commonness and peculiarity of services with similar function, because practice has proved that the description on commonness and peculiarity of web service deserve same considerations.

Obviously Web service description in WSC combines explicit representation and implicit representation concordantly, taking both commonness and peculiarity of web service into account. This prominent feature helps the service discovery and selection process to improve efficiency and flexibility by narrowing search space into certain one or several subject-clubs.

In [9], authors propose an approach that supports the concepts, architecture, operation and deployment of Web service communities. The notion of community serves as an intermediary layer to bind to Web services. A community gathers several slave Web services that provide the same functionality. The community is accessed via a unique master Web service. Users bind to the master Web service that transparently calls a slave in the community. . This work details the management tasks a master Web service is responsible for. Such tasks include among other things registering new Web services into the community, tracking bad Web services, and removing ineffective Web services from the community. A master Web service represents the community and handles users' requests with slave Web services with the help of a specific protocol.

Built upon this work, authors propose in [10] context-based semantic mediation architecture for Web service communities. Indeed, the applicability of the mediation proposition goes beyond this domain. However, they specifically focus on its deployment with communities as defined in [9], where semantic mediation is performed between the community master and slave Web services.

A community-based architecture for semantic Web services is proposed in [11]. In this work, communities gather services from the same domain of interest and publish the functionalities offered by Web services as generic operations. Community ontology is used as a general template for describing semantic Web services and communities. A major advantage of this work is the peer-to-peer community management solution that addresses the problems of centralized approaches.

What sets us apart from the proposed approaches is that we aim at addressing the design of Web services communities in general. We neither rely on a globally fixed domain nor on ontology of permitted classes of data, both strong assumptions that are often not realistic.

To cope with the specific aspects of web services several design notations have been proposed (cf., Provost [12], Ortiz and Hernandez [13], Belouadha and Roudiés [14] [15], Skogan [16]). While most of the proposed notations agreed on a set of concepts specific to Web services, none of then addresses the issue of Web Services communities. This shortage motivated our work in proposing a Web services communities design language, called WSC-UML, which increases the expressiveness of UML for WS 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. Overall, three main motivations are behind our extensions of the original UML notation

III. WEB SERVICES COMMUNITIES REQUIREMENTS

Our system attempts to build a community for end users and service providers by providing the means of managing community memberships and member's contacts. The main requirements related to communities' management can be resumed by creating and updating Web Services Communities then building relationships between them.

Community Creation: The Community Manager creates a community by grouping Web Services related to the same domain then he defines its schema to provide a description of the field to which the community belongs without referring the Web Services providers.

Community Update: As communities evolve in the Web environment characterized by its dynamism, changes can frequently affect communities. Hence, communities should be permanently updated. The community update takes in general two forms, deletion or modification.

The Web Services providers' registration: In order to be accessible through a community, the Web Service provider must apply for registration. The registration of its Web Service is done by Web Service provider must apply for registration. The registration of its Web Service is done by the Member Manager. When a request for registration arrives, the Member Manager associates the web service provider with the correspondent community. By registration, the Web Service provider becomes a member of the community.

Converting Web Information Sources to Web Services: In case of Web information source does not take the form of Web Service, the Member Manager have to convert it to Web Service. Defining Communities Members: The Web Services provider defines communities members by feeding communities with Web Services related to its domain.

In order to meet the system requirements described above, we aim to propose in these part a model of a community which is "container "that clusters Web services based on a specific area of interests. A community offers to its user's functions which allow them to share information about the community and a framework for selecting Web services registered in the community. We suppose here that providers interested by offering their services are in the origin of different Web services communities. All Web services belonging to a given community share the same area of interests' exp: Health care services. The community provides description of desired services without referring to any actual provider.

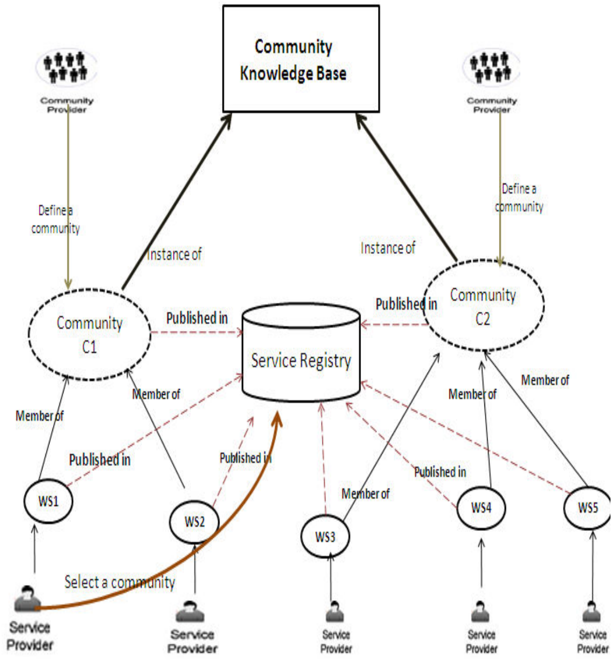


Figure 1. Web services communities model

We define a Community Knowledge Base (figure1) for creating a community of Web services. The Community Knowledge Base provides concepts that allow the description of the community. Communities are instances of the Community Knowledge Base. Community providers are generally groups that share a common domain of interest for example providers: medical centers, X-Ray storage center and Surgery Rooms would define a community which offers Health care services for its members. A community is itself a service that is created, advertised, discovered and invoked in the same way "regular" Web services are. The providers of a community register themselves in a Community Knowledge Base. Communities are published in a registry so that they can be discovered by service providers. Service providers (X-Ray Storage center) identify the community of interest and register their services with it . A Web service may belong to different communities.

A composite service may out-source operations that have different domain of interests. For example a patient historical data comes from Web services belonging to medical center and X-Ray storage center. Since "patient historical data "belong to two different communities. The composite Web service is registered in both of cited communities. End user "A doctor" selects a community of interest and invokes its operations. Each invocation of a community operation is translated to an invocation of community members operations .

IV. THE WSC-UML META-MODEL

The Unified Modeling Language (UML) is considered as the industry de-facto standard for modeling software systems and plays a central role in Model Driven Architecture (MDA) [17]. In UML, the structural aspects of software systems are defined as classes, each formalizing a set of objects with common services, properties, and behaviour. Services are described by methods. Properties are described by attributes and associations. Object Constraint Language (OCL) can be used to express additional constraints.

UML can also serve 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.

In addition, UML can also be used as metamodeling language, where UML diagrams are used to formalize the abstract syntax of another modeling language.

For the reasons presented above we adopted UML for the elaboration of our Metal-model and it's associated model

Our main objective in defining a UML-based meta-model using a design language called WSC-UML in order to cover perspectives related to Web services communities. Our design language is based on an extended UML class diagram. In addition, in defining each perspective, we made sure to provide for pertinent standard. The resulting meta-model illustrated in Figure 1.

The main modelling element 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 <<ID-parameter>> a stereotype representing an abstract definition of the data used to identifier a community. Data used to identify the community are essentially the identifier and the category which describe the area of interest of the community.

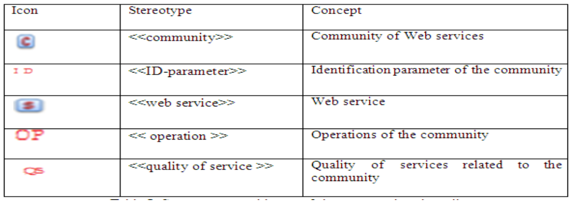
The stereotype <<web service>> is an abstraction of services providers which become members of the community while accomplishing the registration process. The community is accessible via a set of operations. The stereotype << operation >> represents "abstract" operations that summarize the major functions 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 behaviour of a community. To model this aspect, we used class stereotyped <<quality of services >> that offers RunTime, business and security attributes

In addition to proposed stereotype we attached an icon to each of them in the class diagram as described in the following.

Table I. Stereotypes and icons of the community class diagram



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

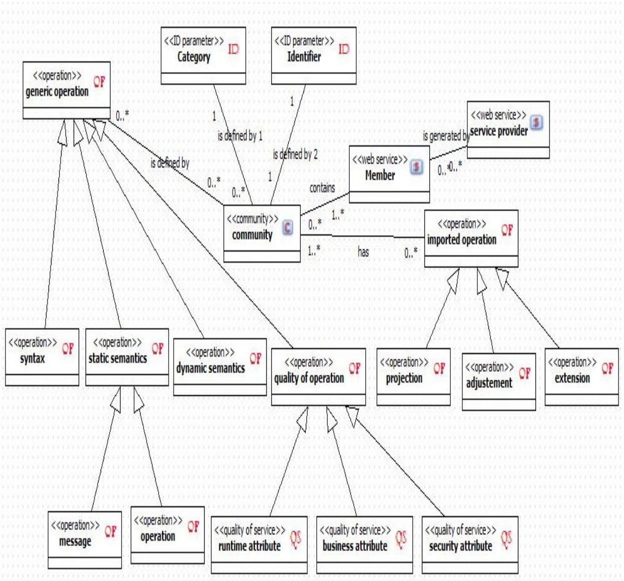

<<ID parameter>> Category ID
<<ID parameter>> Identifier ID
<<operation>> generic operation OF
<<web service>> service provider
<<web service>> Member
<<community>> community
<<operation>> imported operation OF
is defined by 1
is defined by 2
is defined by
is generated by
contains
has
0..*
1..*
1
<<operation>> syntax OF
<<operation>> static semantics OF
<<operation>> dynamic semantics OF
<<operation>> quality of operation OF
<<operation>> projection OF
<<operation>> adjustement OF
<<operation>> extension OF
<<operation>> message OF
<<operation>> operation OF
<<quality of service>> runtime attribute QS
<<quality of service>> business attribute QS
<<quality of service>> security attribute QS

Figure 2 . Meta-model of the Web services community

V. DESIGN OF THE WEB SERVICES COMMUNITY

After the definition of the WS-UML meta-model, our next step was to complement the WS-UML meta-model with a mod l (figure 3). The current WSC-UML Tool prototype allows the user to draw class diagrams according to the WS-UML language notation presented in table 1. To implement the UML extensions added by WSC-UML, we begin with loading the UML meta-model. This later offers an editor for viewing and editing UML models

Specifications of our system refer and are inspired from Web services communities structure presented in [19].

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, "insurance" and "children" are specialization of "healthcare". Communities are generally not independent. They are linked to each other via inter-ontology 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. It is hence 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.

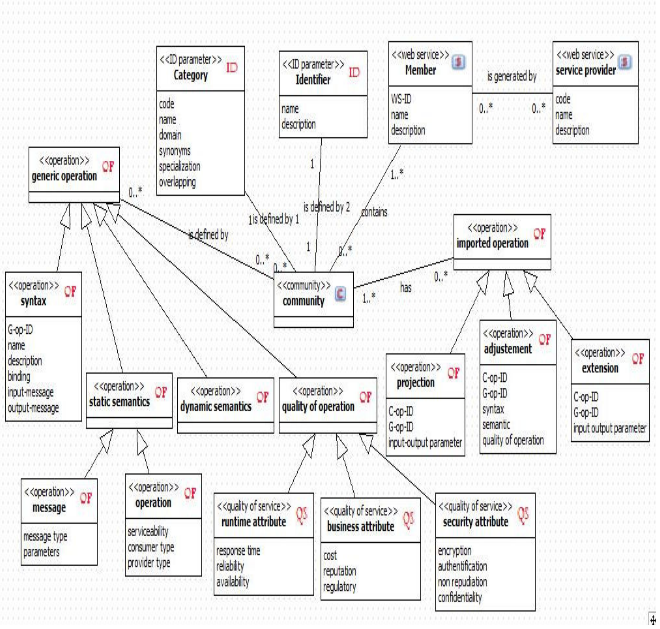


Figure 3 . Design of the Web services community;

VI . RUNNING EXAMPLE

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

The main diagram in the model is a UML class diagram describing the system information entities, its properties and relation. Figure 4 describes the analysis model for the helth care community system as a UML class diagram. The main classes in the domain are represented: Health care community, Health care community members and the Health care Service Provider.

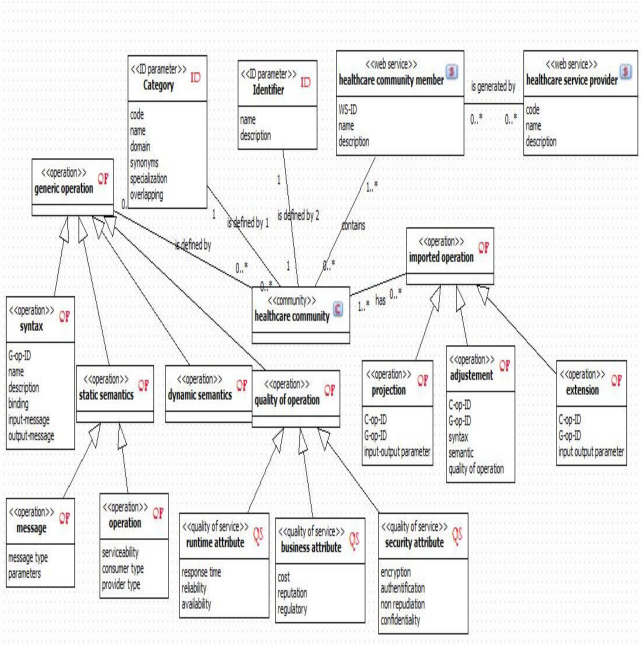


Figure 4 . Design of health care community

VII. IMPLEMENTATION

To ensure the deployment of the tool, and to provide for its potential extensions, we adopted tan open source platform called StarUml. In fact, StarUML [18] is a software modelling platform which supports UML (Unified Modeling Language). It is based on UML notations and eleven different types of diagram. It actively supports the MDA (Model Driven Architecture) approach by supporting the UML profile concept. StarUML allows adding new functions which are adaptable to our domain of interest. For these reasons we adopted StarUML for generating our profile. The adoption was facilitated thanks to our UML-based meta-model defining WSC-UML generated by adding stereotypes and icons specific to Web services communities.

Thus, after the definition of the WSC-UML meta-model, we completed the WSC-UML meta-model with a model for the tool's GUI and afterwards apply the necessary transformations to generate the code. To conduct these last two steps, we used the Eclipse Ganymede and the Eclipse Graphical Modeling The current WSC-UML Tool prototype allows the user to draw class diagrams according to the WSC-UML language notation. Due to space limitations, in this paper we focus only on the GUI for specifying the class diagram.

VIII. CONCLUSION

This paper first overviewed the UML-based design language for web services WSC-UML. It then presented a CASE tool for the WSC-UML language that provides for the graphical representation of Web services communities applications. The tool offers the design Web services communities' elements and helps the developer to distinguish between these elements. The tool usage was illustrated through the design of Web services communities for health care information.

In our work, the proposed profile tackles only the community structure which represents the static part in the community modeled using the class diagram. We are currently investigating how to improve the tool by integrating the dynamic part including community activity modeling using sequence diagram.

VIII. REFERENCES

[1] The Information Societies Technology Project: Health-e-Child EU Contract IST, 2004.

[2] Lee, S. Hyun, and K. M. Na, "This is my paper, " ABC Transactions on ECE, Vol. 10, No. 5, pp. 120-1, 2008.

[3] Gizem. Aksahya and A. Ozcan, " Coomunications & Networks, " Network Books, ABC Publishers,2009.

[4] Kozinets, R. V, "The Strategic Implications of Virtual Communities of Consumption," European Management Journal, Vol. 17, No.3, pp.252-264, 1999.

[5] Smith, M, "Tools for Navigating Large Social Cyberspaces, " Comm. of the ACM 45(4), pp. 51-55, 2002.

[6] Xuejuan ,H . Xinmeng ,C and L. Jinshuo, "Dynamic Selection of Best Service in Semantic Web Service Community, " Proc. International Conference on Computer Science and Software Engineering, Vol. 03 , pp 407-410,2008.

[7] H. Paik, B. Benatallah. And F. Toumani, "WS-CatalogNet: Building Peer-to-Peer e-Catalog, " Proc. of 6th International Conference on Flexible Query Answering Systems,2004.

[8] I. Foster, T. Freeman, K. Keahey, D. Scheftner, B. Sotomayer, and X. Zhang "Virtual Clusters for Grid Communities, " Sixth IEEE International Symposium on Cluster Computing and the Grid, Singapore, IEEE Computer Society, pp. 513–520, 2008.

[9] Benatallah,B Dumas,M , and Sheng, Q .Z, "Facilitating the Rapid Development and Scalable Orchestration of Composite Web Services," Distributed and Parallel Databases, Vol , 17. pp. 5-37, 2005.

[10] Sheng, Q.Z., Benatallah, B., Dumas and M. Mak, E.O-Y "SELF-SERV: a platform for rapid composition of web services in a peer-to-peer environment, " Proceedings of the 28th VLDB Conference, Hong Kong, China, 2002.

[11] S. Sattanathan, P. Thiran, Z. Maamar and D. Benslimane " Engineering communities of web services," iiWAS Austrian Computer Society, Vol.229, pp.57-66, 2007.

[12] W. Provost, "UML for Web services", XML.com, August 5, 2003.

[13] G. Ortiz and J. Hernandez, "Toward UML Profiles for web services and their Extra-functional Properties", IEEE International Conference on Web services (ICWS'06), 2006.

[14] F. Belouadha and O. Roudiés, "Un profil UML de spécification de services web composites sémantiques," CARI ,ROC, pp. 537-544, 2008.

[15] F. Belouadha and O. Roudiés, " Vers un modèle de spécification pour la composition de services web," Proc of SIIE'08, Tunisia,2008.

[16] D. Skogan, R. Gronmo and I. Solheim, "Web service

Composition in UML", Proc of the 8th IEEE Intl enterprise Distributed Object Computing Conf, EDOC 2004.

[17] D. S. Frankel. Model Driven ArchitectureTM: Applying MD ATM to Enterprise Computing. John Wiley & Sons, 2003.

[18] http://staruml.sourceforge.net/docs/developerguide(en)/ch01.html

[19] B. Medjahed and A, Bouguettaya "A Dynamic Foundational Architecture for Semantic Web Services, "Distributed and Parallel Databases, pp. 179–206, 2005