ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 1, Issue 7, September 2012

Evaluating QOSMOS Using Markov Decision Process Model

D. MEENAKSHI, R. RAJA SEKHAR M.Tech,(Ph.D.)

Abstract— **Service discovery has been recognized as an important aspect in the development of service centric systems, i.e., software systems which deploy web services. To develop such systems, it is necessary to identify services that can be combined in order to fulfill the functionality and achieve quality criteria of the system being developed. In this paper, markov decision process model has structural and Behavioral Analysis for Service Quality (SBA) - that is the exploit of services in QOS based ranking order, which can provide functionalities, shows whether vulnerabilities are present and satisfy properties and constraints of systems as specified during the software design phase. The framework is composed of a query extractor, which derives queries from design specification of service centric systems, and a query execution engine that executes these queries against service registries. The paper describes a prototype tool that is developed to demonstrate and evaluate our framework and the results of a set of preliminary experiments are conducted to evaluate it.**

Index Terms— **Adaptive systems, markov decision processes, QoS management, QoS optimization, Service-oriented software engineering**

I. Introduction

Service-Based Systems (SBSs) are playing an increasingly significant role in application domains ranging from research and health care to defense and aerospace. Built through the dynamic composition of loosely coupled services offered by autonomous providers[1], SBSs are operating in environments characterized by frequent changes to requirements, state of component services, and system usage profiles. Achieving and maintain well-defined Quality of Service (QoS) properties in a changing environment represents a key challenge for self-adapting architectures [2]. For example, a highly dynamic system where a set of services are fitted together may change over time, either due to service provider publishing (or withdrawing) service descriptions or due to the availability of certain services which vary according to the users location or the network connectivity. In these settings, a more reliable or efficient service might become available, and thus self-adaptation will allow to improve the overall QoS. To deal with the QoS management of SBSs, QoS MOS (QoS Management and Optimization of Service based systems) a generic architecture for SBS is defined in [3]. QoS MOS is a tool-supported framework for the QoS management of self-adaptive, service-based systems

With the motivation of QOS MOS this paper proposes, a markov decision process model for the Structural and Behavioral Analysis (SBA) of a Service. This proposed model considers the integration structure of the QOSMOS and includes the structural and behavioral analysis to validate the QOS of the selected services.

SBA adds self-adaptation (e.g., self-configuration and self-optimization) capabilities to service based systems through continuous verification of quantitative properties at runtime derived from high-level, user-specified system goals encoded with multi objective utility functions. The self-adaptation capabilities include service selection [4], runtime reconfiguration [5], and resource assignment. Consequently, SBA subsumes most of the existing approaches. The rest of the paper is organized as follows- Section 2 describes the related work. Section 3 describes the SBA architecture. Validation of the proposed framework is presented in Section 4. Section 5 concludes the paper.

II. RELATED WORK

There are many techniques which are used for the self-adapting architectures [2] and for evaluating the performance of the service based systems by using the different markov models [6] like discrete time markov chains and continuous time markov chains. These models check the services in terms of reliability and performance. This paper consists of

1) Query specification
2) Query execution
3) Checking of spoofing attacks
4)Checking the vulnerabilities in the services

The proposed work detects the vulnerabilities present in the services by specifying the functional requirements and the query execution based on the requirements. This shows the spoofing attacks for the services and also intruder detection [7] using the markov decision process model [8].

.

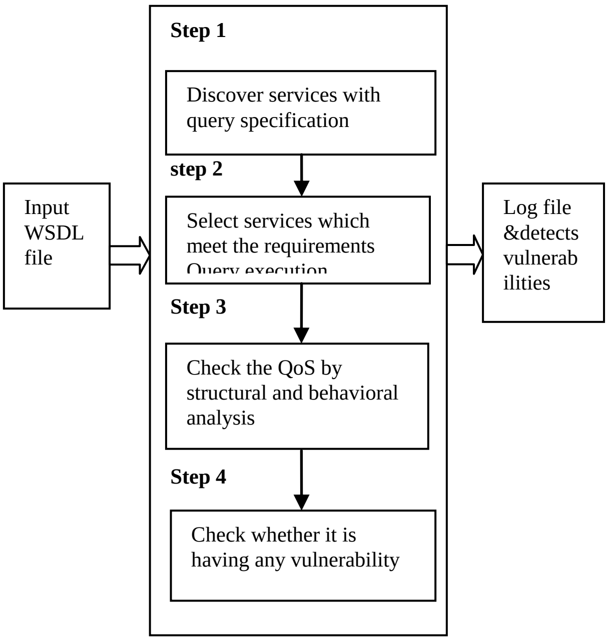


Fig1: Architecture of the proposed model

III. PROPOSED SYSTEM APPROACH

A. *Query specification*

Query specification defines stereotype properties, which are used to specify parameters and constraints for the elements to which the stereotypes containing these properties are applied. Both <<query package>> and <<query message>> stereotypes can specify query parameters. Some of these parameters are inherited from the abstract stereotype <<query element>>. The query parameters specified for query package are global (i.e., applied to the whole query). The query parameters specified for query message are local (i.e., applied to specific messages of the query interaction). The global parameters are considered as default values in the query and can be overridden by local parameters.

The constraints include (a) the type of the constraint (hard or soft), (b) the body of the constraint as an OCL expression, and (c) an optional weight of the constraint if the constraint is soft (real value between 0.0 and 1.0). Hard constraints must be satisfied by all the discovered services and operations. Soft constraints influence the identification of the best services/operations but may not be satisfied by all the services/operations that are discovered. The use of OCL to specify constraints is motivated by the fact that OCL is the standard formal language for specifying constraints for UML models and therefore SBA queries which are based on them.

B. *Query execution*

The SBA query package is submitted to the query execution engine to be processed. The execution of queries is performed in a two-stage process. In first stage, filtering, the query execution engine searches service registries in order to identify services with operations that satisfy the hard constraints of a query and retrieve the specifications of such services. In the second stage, referred to as best operation matching, the query execution engine searches through the services identified in the filtering phase, to find the operations that have the best match with the soft constraints of the query.

Selection of best operation matching: Detection of the best possible matching between the operations required by an SBA query and the candidate service operations identified in the filtering stage is formulated as an instance of the assignment problem as proposed in [9]. More specifically, given the set of operations required by an SBAF query Q, Oper(Q) and the set of service operations identified in the filtering stage, OperS(Q), an operation matching graph is constructed with two disjoint sets of vertices VQ and VS defined as

$$V^Q \equiv \mathrm{Oper}(Q) \cup DV_k \text{ and } V^S \equiv \mathrm{Opers}(Q)$$

Where DV_k is a set of k special vertices representing dummy operations (k = |Oper(Q)| − |OperS(Q)|). This formulation assumes, without loss of generality, that |Oper(Q)| > |OperS(Q)|. If this is not the case, k dummy vertices are added to VQ where k = |OperS(Q)| − |Oper(Q)|.

The set of edges of the graph, $E(V^Q,V^S)$, includes all the possible edges between the required operations in V^Q and the retrieved service operations in V^S. These edges are weighted by a measure $D(v^Q_i,v^S_j)$ indicating the overall distance between v^Q_i and v^S_j where $v^Q_i \in V^Q$ and $v^S_j \in V^S$. This measure is computed as the weighted sum of a set of partial distance measures $df(v^Q_i,v^S_j)$ quantifying the semantic differences between v^Q_i and v^S_j with respect to each facet f in the descriptions of v^Q_i and v^S_j according to the following distance function (the weights are assumed to be normalized):

$$v_j^s \epsilon \mathrm{Opers}(Q) \;\; D\left(F, v_i^Q, v_j^S\right) = 1$$
$$\text{if } v_i \epsilon DV_k$$
$$D\left(F, v_i^Q, v_j^S\right) = 1$$

if vi should not be mapped onto v^S_j where F is the set of facets in the descriptions of operations.

Function D is defined to have a value in the range [0,1] for all the pairs of operations drawn from Oper(Q) and OperS(Q). In the case of comparisons between an existing operation and a dummy operation D's value is defined to be 1. This favors the possibility of mapping an existing operation onto a requested operation rather than leaving without a counterpart. Finally, D is defined to take an infinitum value (∞) in the case of operations which − by virtue of the constraints defined in Q − should not be mapped onto each other. This precludes the matching of such operations when the optimal matching between V^Q and V^S is selected. Following the computation of the D distances for all the edges of the graph, the matching between the operations in V^Q and V^S is detected in two steps. In the first step, a subset $O(V^Q,V^S)$ of $E(V^Q,V^S)$ that is a total morphs between V^Q and V^S (or onto morphs if |Oper(Q)| < |OperS(Q)|) and minimizes the function $\Sigma(v^Q_i,v^S_j) \in O(V^Q,V^S)\, D(F,v^Q_i, v^S_j)$ is selected. $O(V^Q,V^S)$ is selected using standard algorithms for the assignment problem [9]. In the second step, $O(V^Q,V^S)$ is restricted to include only the edges whose distance $D(F,v^Q_i, v^S_j)$ does not exceed a threshold value Dt. Partial distance

ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 1, Issue 7, September 2012

functions. For the facets corresponding to soft constraints defined as Boolean tests, the partial distances d_f are defined as 1 if the test returns false or the facet is not available for a specific service, and 0, otherwise. The partial distance that is used to compare the facets specifying the signatures of two operations is defined as

$$d_{f=signature}(v^Q, v^S) = w_N^* d_L\left(name(v^Q), name(v^S)\right) + w_{IN}^* d_{PS}\left(in(v^Q), in(v^S)\right) + w_{OUT}^* d_{PS}\left(out(v^Q), out(v^S)\right)$$

In this formula, d_L is a linguistic distance built on top of WordNet lexicon [11] and d_{PS} is a function computing the distance between the sets of input or output parameters of two operations. For two sets of parameters P1 and P2, d_{PS} is computed by finding the best possible morphism pm between the elements of these sets, defined as:

$$d_{PS}(P1, P2) = \min_{pm}\left(\Sigma_{(x,y)\in pm}\, d_p(x, y)\right)$$

Where $d_P(x,y)$ is a function that computes the distance between two specific parameters. This distance is computed by finding the best possible matching between the structures of the types of the given operation parameters using a variant of the class distance measures defined in [10].

Our operation matching framework has been designed to support modifications to the set of facets F for service specifications. More specifically, when new facets are added to F, our framework could be extended to support them by incorporating partial distance functions enabling operation comparisons with respect to these facets.

C. *Approach*

As shown in the fig.1, it takes the WSDL file of service provider which provides the QoS initially at the time of service registration. It lists the operations and the subservices related to it. A test request will be sent which shows the html requests sent to the server. It checks for the vulnerabilities present in the services. Then Service consumers rate the services after its usage. The web-service ranking will be the aggregate of initial QoS information provided by web service provider stored in XML format and the feedback ratings by the service consumers. The initial QoS information and feedback ratings can be averaged to derive the ranking for the web service to be published. Using this QoS ranking selection of appropriate web service in business-to-business interactions can greatly benefit.

D. *QoS Information*

Quality of Service, or QoS, is “a combination of several qualities or properties of a service”. It is a set of nonfunctional attributes that may influence the quality of the service provided by a Web service.

The process of detecting web services based on the framework discussed in related work will be initiated, if services are found to match both the functional and QoS requirements and ratings requirements have also been specified, then the Web Service Broker ranks the services based on consumer’s QoS and ratings requirements. Service consumer then selects the web service with the highest rank.

IV. SBA RESULTS ANALYSIS

The results of an SBA query identified by the query execution engine (i.e. Best candidate services with smallest distances) are specified by using the SBA profile. Figure 2 presents the part of the SBA profile for SBA results. The SBA results are represented as a UML package stereotyped as <<results_package>.

The results package contains a refinement of the query interaction used by the designer to create the query together with the structural model for the elements in the interaction, and a number of UML packages stereotyped as <<service package>>, one for each candidate service identified by the query execution engine. Each service package contains elements representing a concrete discovered service together with the class diagram of all data types and their relationships used in the XSD schemas reverse engineered from the WSDL specification of this service. The structural model in the results package contains copies of all data types from the query package together with the mapping to the data types in the service packages. The data mapping based on the data distances computed for each bound operation in the service against the query message associated to the service

The operations in an service_package may be stereotyped as (i) bound operations <<bound_operation>> that denote the service operations with the best match to a query message or the one that the designer selects as the best candidate; (ii) candidate operations <<candidate_operation>> that reflect another possible result for the query message, but not necessarily the best match; and (iii) service operations <<service_operation>> that are all the remaining operations in the WSDL specification of the service. The above operations are grouped together in a UML component (contained in the service_package) stereotyped as either <<bound_service>> or <<candidate_services>>, depending on the existence of any bound operations.

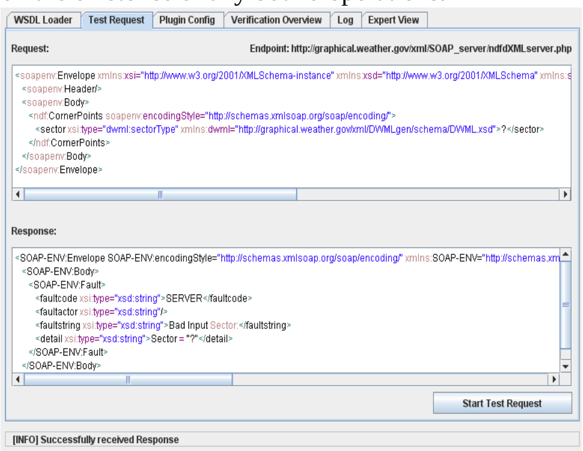
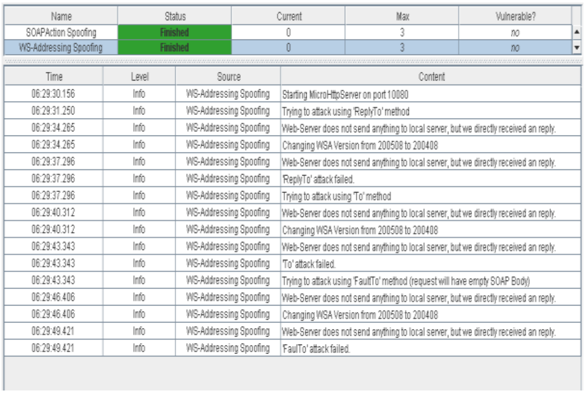


Fig. 2: SBA Results: Structural Information

ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 1, Issue 7, September 2012



Name	Status	Current	Max	Vulnerable?
SOAPAction Spoofing	Finished	0	3	no
WS-Addressing Spoofing	Finished	0	3	no

Time	Level	Source	Content
06:29:30.156	Info	WS-Addressing Spoofing	Starting MicroHttpServer on port 10080
06:29:31.250	Info	WS-Addressing Spoofing	Trying to attack using 'ReplyTo' method
06:29:34.265	Info	WS-Addressing Spoofing	Web-Server does not send anything to local server, but we directly received an reply.
06:29:34.265	Info	WS-Addressing Spoofing	Changing WSA Version from 200508 to 200408
06:29:37.296	Info	WS-Addressing Spoofing	Web-Server does not send anything to local server, but we directly received an reply.
06:29:37.296	Info	WS-Addressing Spoofing	'ReplyTo' attack failed.
06:29:37.296	Info	WS-Addressing Spoofing	Trying to attack using 'To' method
06:29:40.312	Info	WS-Addressing Spoofing	Web-Server does not send anything to local server, but we directly received an reply.
06:29:40.312	Info	WS-Addressing Spoofing	Changing WSA Version from 200508 to 200408
06:29:43.343	Info	WS-Addressing Spoofing	Web-Server does not send anything to local server, but we directly received an reply.
06:29:43.343	Info	WS-Addressing Spoofing	'To' attack failed.
06:29:43.343	Info	WS-Addressing Spoofing	Trying to attack using 'FaultTo' method (request will have empty SOAP Body)
06:29:46.406	Info	WS-Addressing Spoofing	Web-Server does not send anything to local server, but we directly received an reply.
06:29:46.406	Info	WS-Addressing Spoofing	Changing WSA Version from 200508 to 200408
06:29:49.421	Info	WS-Addressing Spoofing	Web-Server does not send anything to local server, but we directly received an reply.
06:29:49.421	Info	WS-Addressing Spoofing	'FaulTo' attack failed.

Fig.3 SBA Results: Behavioral Information

The interaction in the <<results_package>> refines the query interaction by replacing query messages by bound messages (stereotyped as <<bound_message>>) corresponding to bound operations. When no operation is found, the query message is not modified.

V. CONCLUSION

In this paper, the framework allows the designer to analyze the results of a query and select candidate operations to become bound operations. After the designer selects a particular service from the returned candidates, the structural model in the results package is automatically updated with concrete data of the chosen service, and the interaction is modified to reflect the binding of the services and operations.

It also addresses the various challenges that common vulnerable in service detection, in particular allowing service detection to be impelled by design decisions taken during the development of SCS systems and fulfills the lack of processes and tools to assist the engineering of complex and dependable SCS. Together with industrial partners, we are conducting large-scale experiments of our framework taking into consideration different types of service specifications ranging from structural, to semantic and behavioral aspects that ranked based on QOS.

REFERENCES

[1] R. Feri, G.D.M serugendo and J. Barata, “designing self-oganization for evolvable assembly systems,” Proc second IEEE int’l Conf.self adaptive and self organizing systems,pp.97-106,2008.

[2] L. Beresi, E.D. Nitto, and c. Ghezzi,” Towards Open-World Software Issues and Challenges,”computer,vol 39,no. 10,pp.36-43,0ct 2006.

[3] Calinescu, R.; Grunske, L.; Kwiatkowska, M.; Mirandola, R.; Tamburrelli, G.; , "Dynamic QoS Management and Optimization in Service-Based Systems," Software Engineering, IEEE Transactions on , vol.37, no.3, pp.387-409, May-June 2011; doi: 10.1109/TSE.2010.92.

[4] M.A. Serhani, R. Dssouli, A. Hafid, and H.A. Sahraoui, “A QoS Broker Based Architecture for Efficient Web Services Selection,”Proc. IEEE Int’l Conf. Web Services, pp. 113-120, 2005.

[5] R. Calinescu, “Reconfigurable Service-Oriented Architecture for Autonomic Computing,” Int’l J. Advances in Intelligent Systems,vol. 2, no. 1, pp. 38-57, June 2009.

[6] N. Sato and K.S. Trivedi, “Stochastic Modeling of Composite Web Services for Closed-Form Analysis of Their Performance and Reliability Bottlenecks,” Proc. Int’l Conf. Service Oriented Computing, pp. 107-118, 2007.

[7] T.Darling and M.A.Shyaman “A Markov Decision Model For Intruder Detection in Web Services” ,IEEE Conf on Decision and Control Sidney, December 2009.

[8] M.L. Puterman, Markov Decision processes Wiley,1994.

[9] Klein and A. Bernstein, “Toward High-Precision Service Retrieval,” IEEE Internet Computing, vol. 8, no. 1, pp. 30-36, Jan./ Feb. 2004.

[10] J. Garofalakis, Y. Panagis, E. Sakkopoulos, and A. Tsakalidis, “Web Service Discovery Mechanisms: Looking for a Needle in a Haystack,” Proc. Int’l Workshop Web Eng., Hypermedia Development and Web Eng. Principles and Techniques: Put Them in Use, 2004.

[11] T. Gardner, “UML Modelling of Automated Business Processes with a Mapping to BPEL4WS,” Proc. Second European Workshop OO and Web Services, 2004.

D. MEENAKSHI M.Tech Student, CSE Department, JNTUACEA, Anantapur, Andhra Pradesh, India.

R. RAJA SEKHAR M.Tech,(Ph.D.),Assistant Professor, CSE Department, JNTUACEA, Anantapur, Andhra Pradesh, India.