

Research Article

Web Service Discovery with Quality of Service (QoS) Parameters using Bayesian Belief Network

¹A. Gnanasekar and ²R.M. Suresh

¹Department of Computer Science and Engineering, R.M.D. Engineering College,
²Sri Muthukumar Institute of Technology, Anna University, Chennai 601206, India

Abstract: The aim of this study is to present a novel approach which uses the belief net to intelligently infer the best and most relevant web service based on the available Quality of service parameters. Service Oriented Architecture (SOA) is the discovery of relevant Web Services for a specified task is a one of the key challenges. As Web services proliferate, there is a possibility of existing more than one service with same functionalities consequently; most of service discovery method will return several services which meet user given input and output. Finding the mainly relevant and best service is very crucial for the service consumers. In this situation, the non-functional Quality of Service (QoS) parameters present in the service description provided by the service provider play a major responsibility.

Keywords: Bayesian belief network, discovery, quality of service parameters, service oriented architecture, web service

INTRODUCTION

Web Services become indispensable and become a promising technology for the development of new Internet based e-trading systems. For example, a pervasive book purchasing application can be possibly converted into a Web Service. However, there exists a large number of books purchasing services with similar functionalities and become a major problem for the discovery of relevant one.

Web services are self-contained, self-describing, heterogeneous software components that can be deployed into the service registry UDDI (Luc *et al.*, 2004) and accessed over the internet. The Web Service Description Language (WSDL) (Erik *et al.*, 2001) is used as the *de facto* standard for service providers to describe the input and output parameters and the operations of the web services. Therefore web services do not have homogeneous structure and hence heterogeneity arises from different ways to name parameters and describe their processing. On the one hand, the heterogeneity of services is an obstacle for the service discovery and integration and on the other hand, WSDL description is purely syntactic; consequently, UDDI is able to feature only keyword-based matches that often lead to the discovery of more irrelevant services than relevant services.

The lack of any machine interpretable semantics requires human intervention for service discovery. This drawback prevents the use of web services on complex business contexts, where the automation of these

business processes is necessary. Semantic web services had been proposed to overcome the issues such as interface heterogeneity and keyword-based syntactic search. Various semantic descriptions of web services have already been proposed such as Business Process Execution Language for Web Services (BPEL4WS) (Andrews *et al.*, 2003) Web Service Modeling Language (WSML) (Roman *et al.*, 2005), Web Service Modeling Ontology (WSMO) (Bruijn *et al.*, 2005) and Semantic Annotations for Web Service Description Language (SAWSDL) (Kopecky *et al.*, 2007). In this direction, OWL-S coalition has promoted Ontology Web Languages for Services (OWL-S) (David *et al.*, 2004), which enrich WSDL and BPEL4WS with rich semantic annotations. Using OWL-S, web services are extended with an unambiguous description by relating the input and output parameters of the service to common concepts defined in *Web Ontology* (Sean *et al.*, 2004), which serve as the key mechanism to globally define and reference concepts. For the annotation of web services, OWL-S defines four ontologies: Service, Service profile, Service grounding and Service model. The *Service* ontology serves as an upper ontology of the other three containing references to them.

LITERATURE REVIEW

A number of Quality of Service (QoS) based web services discovery and composition mechanisms have

been developed in recent years. A service is a piece of functionality made available by a service provider, in order to deliver its results to a service consumer. There are many reasons for the enterprises to take a web service based Service Oriented Architecture approach. Some of the primary reasons are reusability, scalability, flexibility, interoperability and cost effectiveness (Guinard *et al.*, 2003).

As more and more web services are available, QoS capability is becoming a decisive factor to distinguish the best services (Xu *et al.*, 2010; Tao *et al.*, 2008). The international quality standard describes quality as the totality of features and characteristics of a product or service that bear on its ability to satisfy stated or implied needs. The following views about quality are common, Quality as functionality, Quality as conformance, Quality as reputation. These different views of quality require the QoS to be monitored and measured differently. Quality as a functionality characterizes the design of a service and can only be measured by comparing the service against other services, offering similar functionalities (Kyriakos and Dimitris, 2009). Quality as conformance, on other hand, can be monitored for each service individually and usually requires the user's experience of the service in order to measure the "promise" against the "delivery." Finally, reputation can be regarded as a reference to a service's consistency over time, in contributing both functionality and conformance qualities and can therefore be measured through the other two types of quality over time.

A QoS based model has been developed by Demian Antony and Ananthanarayana (2010) for business driven web service selection. They have proposed the QoS requirement format for the requesters, to specify their complex demands for the web service discovery, using a tree structure. A web service selection algorithm has been presented, which ranks functionally similar web services based on the requester's QoS requirements and preferences. The web service selection algorithm has also been extended to handle the requester's alternative set of QoS constraints having diminished preferences. This study also explores a method to resolve the tie i.e., a web service with the same computed score that may occur among qualitatively competitive web services during the selection process.

Huang *et al.* (2009) have proposed an efficient service selection scheme, to help service requesters to select services by considering two different contexts: single QoS-based service discovery and QoS-based optimization of service composition. Based on the QoS measurement metrics, this study proposed multiple criteria decision making and integer programming approaches to select the optimal service. This study has applied Multiple Criteria Decision Making (MCDM) with a Weighted Sum Model (WSM) to help service

requesters to evaluate services numerically. It then transforms the QoS-based optimization of service composition into a mathematical programming problem by deriving the objective functions of the constituent workflow patterns. A QoS Consensus Moderation Approach (QCMA) has been proposed (Wei-Li *et al.*, 2008) in order to perform the QoS consensus and to alleviate the differences on QoS characteristics in the selection of web services. The QCMA is a web service selection mechanism based on the fuzzy QoS consensus for a group of participants. The architecture achieved QoS consensus by including a number of activities such as participants' opinion similarity, QoS term preference ordering and QoS fuzzy scale. The contribution of the QCMA not only included the fuzzy inquiry for service selection, but also offered the features to model the QoS preference consensus, after aggregating sufficient web services. The QCMA is designed for open and dynamic web environment, such that new opinions and preferences as well as new QoS aspects can be modeled flexibly. In the static composition approach, the user or developer manually interprets the requirements for the required composition and the available service capability or functionality and makes a decision regarding how the services can be interweaved to make a value-added service (Sioutas *et al.*, 2009).

A top-down modeling approach for web service based business processes has been proposed (Florian *et al.*, 2008) to capture the functional and non-functional aspects using a choreography language (WS-CDL), which describes the message interactions among the participants. The choreography has been annotated with SLAs for the different partners. An OWL-S service profile ontology based framework is used, for the retrieval of web services based on subsumption relation and structural case-based reasoning, which performs domain-dependant discovery (Georgios and Nick, 2010). The non-functional information which is present in the service profile is also used for service discovery. Bruneo *et al.* (2013) have proposed a method to evaluate nonfunctional properties of a WS-BPEL process, starting from those of the WSEs involved in the process. The method can be applied both at design time, for identifying unsatisfactory performance in the composition design and when the composed WS provider wants to publish its service into public registry for third-party compositions. Mahbub *et al.* (2011) in their work have proposed runtime service discovery framework supports identification of services based on service discovery queries in both classic pull mode and proactive push mode of query execution. Service discovery queries are specified in an XML-based query language, called SerDiQueL and it monitor component to verify if behavioural and contextual conditions in the queries can be satisfied by services. Paliwal *et al.* (2012) in their work have presented an integrated approach for automated service discovery. Specifically,

the approach addresses two major aspects related to semantic-based service discovery: semantic-based service categorization and semantic-based service selection. For service categorization, employ ontology framework and additionally, utilize clustering for accurately classifying the web services based on service functionality. For semantic-based service selection, employ ontology linking and LSI thus extending the indexing procedure from solely syntactical information to a semantic level.

Kun *et al.* (2009) in their work have proposed a model for the discovery of the semantic associations among the web services, by constructing a semantic model to describe their behavior rules based on the qualitative probabilistic network. They have also proposed a distance measure approach to discover the semantic associations among web services. Staikopoulos *et al.* (2010) have proposed a model driven approach for the dynamic adaptation of web services based on ontology-aware service templates. Model-driven engineering raises the level of abstraction from concrete web service implementations to high-level service models, which leads to more flexible and automated adaptations through template designs and transformations. Wu *et al.* (2007) present a Bayesian network based QoS assessment model for web services. That could predict the service capability in various combinations of users' QoS requirements. This approach is used to evaluate the capability of each service and the one with best capability is selected as the binding service. Though it uses Bayesian network classification algorithm for each provider/service to predict the level of QoS, it is computationally complex and is based on probabilities, moreover it just considers local constraints in web service selection and doesn't mention the global constraints.

SEMANTIC WEB SERVICE DISCOVERY

In recent years along with the attention on Service Oriented Architecture (SOA), the academia and the industry are getting closer and closer and the research on SOA is increasing. One of the research areas in SOA is service discovery.

The semantic web has emerged as a solution, extending the current web technologies with well-defined meanings, to the existing resources and services. This is supported by annotations with ontological semantics, through languages that can be interpreted and processed by computers. Ontologies are used to provide a formal and explicit specification to the domain concepts, restrictions, logical relations and properties of a semantic web service. Semantic reasoners use this information to perform automatic analysis and assertions of web services and resources (Ngu *et al.*, 2010). Consequently, semantic web technologies provide additional scope for automated

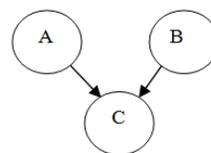


Fig. 1: Example of Bayes' theorem

service analysis, selection and matching, providing automated processing and decision making based on semantic descriptions. These processes can be exploited in the context of service composition and adaptation. The semantic web services which utilize the power of ontologies, would provide an efficient way of matching services. Several semantic description languages have been proposed by the research community for discovery and composition. Nevertheless, the Web Ontology Language for Services (OWL-S) has dominated all other languages. Many different approaches have been proposed in the literature for service discovery and composition. Some of the formal methods for web service composition are Petri Nets, Unified Modeling Language (UML), State charts, Logical Reasoning, Activity diagrams, Process Algebra and Fuzzy Logic. In this study the Belief Net (BN) based novel method is proposed to find the best relevant web service shown in Fig. 1, the user given content will pass through the following refinement steps to extract the annotations (nouns) that are used to discover the services.

Belief net: The Belief Networks (BN) is a 'causal reasoning' tool that has found many uses in a wide variety of applications and is now the mainstay of the AI research field. BNs are based on the laws of probability and in particular conditional and Bayesian probability theory. Bayesian Belief Networks (BBN) provides a mathematically correct and therefore more accurate method of measuring the effects of events on each other. The mathematics involved also allows us to calculate in both directions. So it is possible to find out which event was the most likely cause of another. It defines various events, the dependencies between them and the conditional probabilities involved in those dependencies. A BBN can use this information to calculate the probabilities of various possible causes being the actual cause on event.

The basic idea in BBN is that the problem domain is modeled as a set of nodes interconnected with arcs to form a directed acyclic graph. Each node represents a random variable, or uncertain quantity, which can take two or more possible values. The arcs signify the existence of direct influences between the linked variables and the strength of each influence is quantified by a forward conditional probability. Belief networks are used to develop knowledge based applications in domains which are characterized by inherent uncertainty. Increasingly, belief network

techniques are being employed to deliver advanced knowledge based systems to solve real world problems. For better understanding of the Bayesian Network it is ideal to understand the Bayesian Probability Theorem (Murray's Web Page, 2013):

Bayesian probability theorem: The well known product Rule of Probability for independent event is:

$$p(AB) = p(A) \times p(B) \tag{1}$$

where, $p(AB)$ means the probability of A and B happening. This is actually a special case of the following Product Rule for dependent events:

$$p(AB) = p(A) \times p(B|A) \tag{2}$$

$$p(AB) = p(B) \times p(A|B) \tag{3}$$

where, $p(A|B)$ means the probability of A given that B has already occurred. From (2) and (3), since $p(A) \times p(B|A) = p(B) \times p(A|B)$, we have $p(A|B) = (p(A) \times p(B|A)) / p(B)$ which is the simpler version of Bayes' Theorem. This equation gives us the probability of A happening given that B has happened. Whereas $p(B) = p(B|A) \times p(A) + p(B|\sim A) \times p(\sim A)$.

Chaining Bayes' theorem: Let us assume that a third event, C, has happened and wanted to Calculate $p(AB)$. This is written as $p(AB|C)$. We can use the Product Rule $p(AB) = p(A|B) \times p(B)$ [by (3)]:

$$p(AB|C) = p(A|C) \times p(B|AC) \tag{4}$$

$$p(AB|C) = p(B|C) \times p(A|BC) \tag{5}$$

Therefore we have, $p(A|BC) = (p(A|C) \times p(B|AC)) / p(B|C)$ this gives us the probability of A happening given that B and C have happened. This is often quoted as $p(H|EC) = (p(H|C) \times p(E|HC)) / p(E|C)$, where $p(H|EC)$ is the probability of Hypothesis H given Evidence E in Context C. By using the product rule we can chain several probabilities together. For instance, to find the probability of H given that E_1, E_2 and C have happened:

$$p(H | E_1 E_2 C) = (p(H | C) \times p(E_1 E_2 | HC)) / p(E_1 E_2 | C) \tag{6}$$

Therefore, to find the probability of H given that E_1, E_2, E_3 and C have happened:

Table 1: Probabilities of sum of all state should be 1

A		B	
True	False	True	False
$p(A) = 0.1$	$p(\sim A) = 0.9$	$p(B) = 0.4$	$p(\sim B) = 0.6$

$$p(H | E_1 E_2 E_3 C) = p(H | C) \times p(E_1 E_2 E_3 | HC) / p(E_1 E_2 E_3 | C) \tag{7}$$

An example of Bayes' theorem: In Fig. 1 we assume the following probabilities for A and B. It is to be noticed that, the sum of the probabilities of all the state should be 1 as shown in Table 1.

As well as for C, when dependencies converge, there may be several conditional probabilities to fill in though some can be calculated from others because the probabilities for each state should sum to 1 as shown in Table 2.

Calculating initialized probabilities: Using the probabilities from Table 1 and 2, the 'initialized' probability of C can be calculated by summing the various combinations in which C is true and breaking those probabilities down into known probabilities:

$$\begin{aligned} p(C) &= p(CAB) + p(C \sim AB) + p(CA \sim B) + p(C \sim A \sim B) \\ &= p(C | AB) \times p(AB) + \\ &\quad p(C | \sim AB) \times p(\sim AB) + \\ &\quad p(C | A \sim B) \times p(A \sim B) + \\ &\quad p(C | \sim A \sim B) \times p(\sim A \sim B) \\ &= p(C | AB) \times p(A) \times p(B) + \\ &\quad p(C | \sim AB) \times p(\sim A) \times p(B) + \\ &\quad p(C | A \sim B) \times p(A) \times p(\sim B) + \\ &\quad p(C | \sim A \sim B) \times p(\sim A) \times p(\sim B) \\ &= 0.518 \end{aligned}$$

So as a result of the conditional probabilities, C has a 0.518 chance of being true in the absence of any other evidence.

Calculating revised probabilities: If we know that C is true, we can calculate the 'revised' probabilities of A or B being true (and therefore the chances that they caused C to be true), by using Bayes' Theorem with the initialized probability:

$$\begin{aligned} p(B|C) &= (p(C|B) \times p(B)) / p(C) \\ &= ((p(C|AB) \times p(A) + p(C|\sim AB) \times p(\sim A)) \times p(B)) / p(C) \\ &= ((0.8 \times 0.1 + 0.5 \times 0.9) \times 0.4) / 0.518 \\ &= 0.409 \end{aligned}$$

Table 2: Probabilities of each state should be 1

A		True				False			
B		True		False		True		False	
C	True	$p(C AB) = 0.8$		$p(C A \sim B) = 0.6$		$p(C \sim AB) = 0.5$		$p(C \sim A \sim B) = 0.5$	
	False	$p(\sim C AB) = 0.2$		$p(\sim C A \sim B) = 0.4$		$p(\sim C \sim AB) = 0.5$		$p(\sim C \sim A \sim B) = 0.5$	

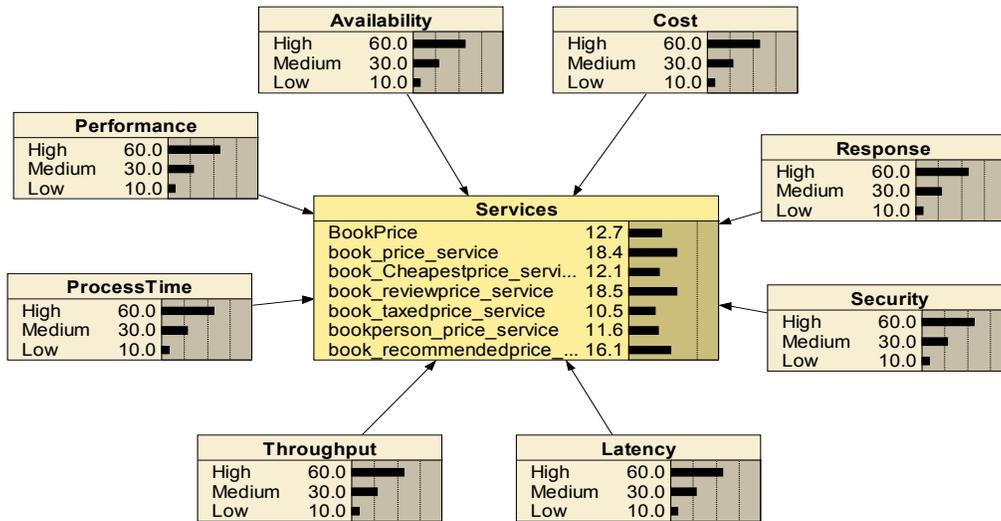


Fig. 2: A belief net for book purchase example

$$\begin{aligned}
 p(A|C) &= (p(C|A) \times p(A)) / p(C) \\
 &= ((p(C|AB) \times p(B) + p(C|A \sim B) \times p(\sim B)) \times p(A)) / p(C) \\
 &= ((0.8 \times 0.4 + 0.6 \times 0.6) \times 0.1) / 0.518 \\
 &= 0.131
 \end{aligned}$$

Therefore, if C is true then B is more likely to be the cause than A.

Belief net based service discovery: A motivating example: The Semantic web services used for the examples were obtained from the OWL-S (version 1.1) service retrieval test collection, OWL-S TC (version 4.0) (Matthias *et al.*, 2009). This dataset consists of 1083 semantic web services from nine different domains namely, education, food, communication, medical care, economy, weapons, geography, travel and simulation. OWL-TC provides a set of 42 test queries covering all the nine domains:

Motivating example 1-book purchase: Suppose a user wants to know the price of a particular text book. The user may give “Title” as input and he/she want its “Price” as output. Any service discovery algorithm will return more than one service which matches the given input and the output and let us assume that there are seven services from OWL-S TC test collection have been returned by the discovery algorithm, namely BookPrice, book_price_service, book_Cheapestprice_service, book_reviewprice_service, book_taxedprice_service, bookperson_price_service, book_recommendedprice_service. However, the user may be in trouble selecting the best service among these services. So, the ultimate option is to consider any one or more of the non-functional QoS parameters such as availability, cost, performance, response time, process

time, security, throughput and execution time etc to select a best and relevant service. That is, a user may want to select a service which satisfies some or all of the above mentioned QoS parameters. Some user may want to select a service with best performance whereas another user may want to select a service which may meet two more parameters such as response time and cost. It is evident that the each user’s requirement may be different and complex. To solve this type of complex and inconsistent state of user requirement, a belief net based service selection method is proposed to infer and select the best service which the user wants.

Construction of belief net: Suppose a user each node in the belief net must have a relation stored at each node, which expresses the value of that node in terms of its parents (or as a constant if the node has no parents). The node may be deterministic or probabilistic. If the node is probabilistic, then the relation must provide a probability for each state of the child, for each possible configuration of parent values. Based on this hypothesis, a belief net for the QoS parameters such as availability, cost, response time, security, latency, throughput, process time and performance is constructed for the Book Purchase example and is shown in Fig. 2. The excerpts of the probability data distributed to all the possible conditional probabilities are shown in Fig. 3.

Suppose a user want to select a best service with high security, performance and processing time and low latency and cost. The probabilities of the services are inferred by the Belief Net is as shown in Fig. 4. In this case, the probability of book_price_service is 23.5% followed by the probability achieved by book_reviewprice_service with 20.6%. That is based on the probability distribution of various dependency

Throughput	Latency	Security	Process Time	Cost	Availability	Performance	Response TL...	BookPrice	book princ...	book Chea...	book revi...	book taxa...	bookpersa...	book reco...
High	High	High	High	High	High	High	High	5.405	24.324	18.919	16.216	2.703	10.811	21.622
High	High	High	High	High	High	High	Medium	5.405	24.324	18.919	16.216	2.703	10.811	21.622
High	High	High	High	High	High	High	Low	5.405	24.324	18.919	16.216	2.703	10.811	21.622
High	High	High	High	High	High	Medium	High	5.405	24.324	18.919	16.216	2.703	10.811	21.622
High	High	High	High	High	High	Medium	Medium	5.405	24.324	18.919	16.216	2.703	10.811	21.622
High	High	High	High	High	High	Medium	Low	5.405	24.324	18.919	16.216	2.703	10.811	21.622
High	High	High	High	High	High	Low	High	5.405	24.324	18.919	16.216	2.703	10.811	21.622
High	High	High	High	High	High	Low	Medium	20	14.286	8.571	11.429	5.714	17.143	22.857
High	High	High	High	High	High	Low	Low	20	14.286	8.571	11.429	5.714	17.143	22.857
High	High	High	High	High	Medium	High	High	20	14.286	8.571	11.429	5.714	17.143	22.857
High	High	High	High	High	Medium	High	Medium	20	14.286	8.571	11.429	5.714	17.143	22.857
High	High	High	High	High	Medium	High	Low	20	14.286	8.571	11.429	5.714	17.143	22.857
High	High	High	High	High	Medium	Medium	High	20	14.286	8.571	11.429	5.714	17.143	22.857
High	High	High	High	High	Medium	Medium	Medium	20	14.286	8.571	11.429	5.714	17.143	22.857
High	High	High	High	High	Medium	Medium	Low	20	14.286	8.571	11.429	5.714	17.143	22.857
High	High	High	High	High	Medium	Low	High	5.405	24.324	18.919	16.216	2.703	10.811	21.622
High	High	High	High	High	Medium	Low	Medium	5.405	24.324	18.919	16.216	2.703	10.811	21.622
High	High	High	High	High	Medium	Low	Low	5.405	24.324	18.919	16.216	2.703	10.811	21.622
High	High	High	High	High	Low	High	High	5.405	24.324	18.919	16.216	2.703	10.811	21.622
High	High	High	High	High	Low	High	Medium	5.405	24.324	18.919	16.216	2.703	10.811	21.622
High	High	High	High	High	Low	High	Low	5.405	24.324	18.919	16.216	2.703	10.811	21.622

Fig. 3: Excerpts of the conditional probability for the book purchase example

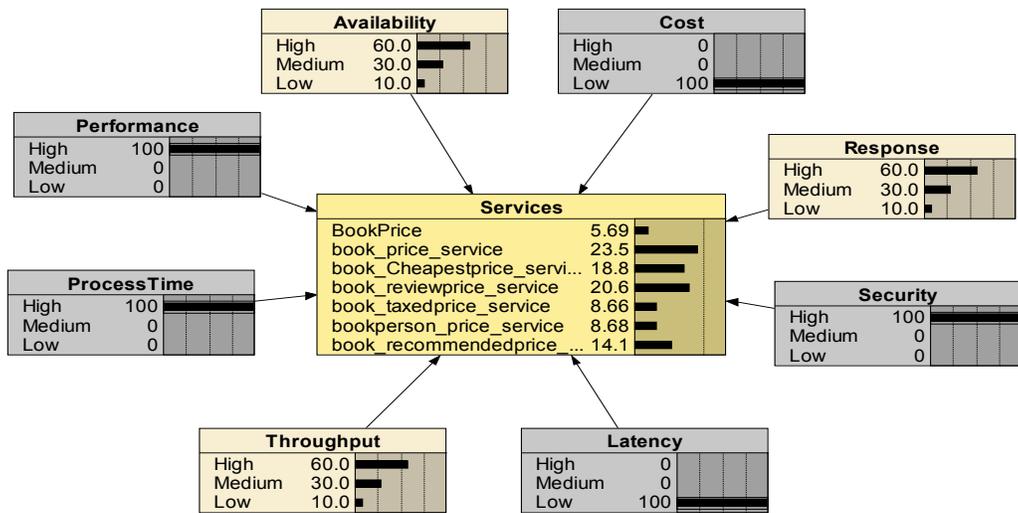


Fig. 4: A compiled belief net shows the probabilities of user given QoS options for book purchase example

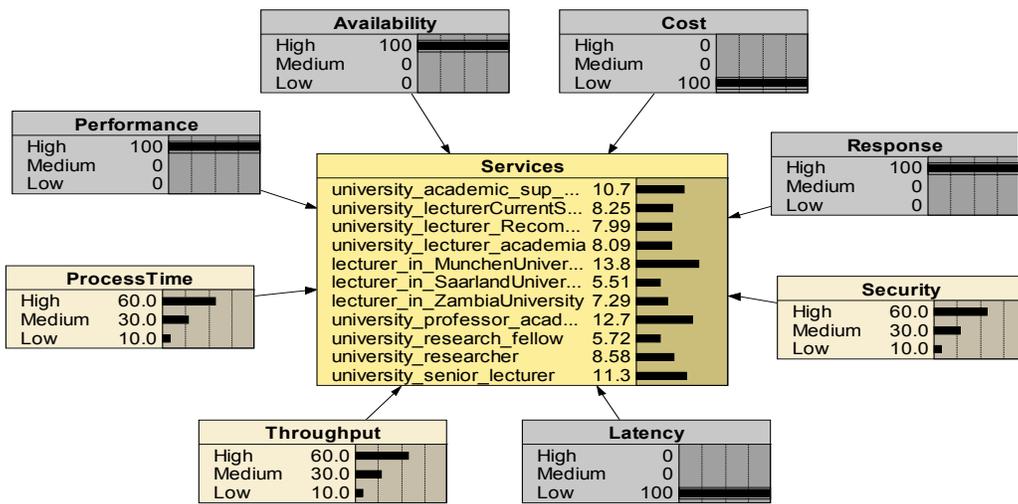


Fig. 5: A compiled belief net to show the probabilities of user given QoS options for finding a best professor in a university example

Table 3: Probabilities of different services for book purchase service here

Test case	Cost	Response	Security	Latency	Throughput	Process time	Performance	Availability
1	L	-	H	L	-	H	H	-
2	L	H	H	L	H	H	H	M
3	M	H	M	L	H	M	H	H
4	L	H	-	M	-	-	-	M
5	-	-	-	-	H	H	H	H
6	-	H	H	-	-	L	H	-
7	L	-	-	-	-	M	M	H
8	L	-	-	L	H	-	H	H
9	-	-	H	L	-	-	H	H
10	M	M	-	M	H	-	M	-

Test case	Book Price	Book_price_service	Book_Cheapestprice_service	Book_reviewprice_service	Book_taxedprice_service	Bookperson_price_service	Book_recommendedprice_service
1	5.69	23.5	18.8	20.6	8.66	8.68	14.1
2	7.89	23.7	18.4	21.1	5.26	10.5	13.2
3	5.41	24.3	18.9	16.2	2.70	10.8	21.6
4	9.88	21.4	15.1	16.1	8.23	10.6	18.7
5	7.80	22.9	17.5	16.7	3.64	11.6	19.9
6	12.2	17.6	12.2	17.1	8.58	13.4	18.9
7	14.0	16.3	10.9	15.7	8.30	14.2	20.5
8	5.92	24.2	18.8	17.2	3.23	10.8	19.9
9	7.13	22.8	17.4	17.8	6.93	10.0	17.9
10	11.0	12.0	7.89	21.5	15.9	13.5	18.3

H = High, L = Low and M = Medium

Table 4: Probabilities of different services for best professor in a university example here

Test case	Cost	Response	Security	Latency	Throughput	Process time	Performance	Availability	University_academic_sup_staff	University_lecturer_CurrentSem
1	L	H	-	L	-	-	H	H	10.7	8.25
2	L	H	H	L	H	H	H	H	5.96	7.55
3	L	M	H	L	M	L	H	H	10.6	11.5
4	M	H	-	L	-	L	H	-	13.2	7.92
5	L	-	-	L	H	L	H	-	9.74	9.56
6	-	H	M	-	H	-	H	H	10.7	8.74
7	-	-	H	M	M	M	H	H	7.65	13.3
8	M	H	-	-	-	L	H	H	6.42	8.93
9	M	H	M	L	-	-	H	-	10.2	8.97
10	M	-	H	-	M	-	H	-	6.53	8.05

Test case	University_lecturer_Recommend	University_lecturer_academia	Lecturer_in_MunichUniversity	Lecturer_in_SaarlandUniversity	Lecturer_in_ZambiaUniversity	University_professor_academia	University_research_fellow	University_researcher	University_senior_lecturer
1	7.99	8.09	13.8	5.51	7.29	12.7	5.72	8.58	11.3
2	6.71	14.7	18.2	3.75	4.72	15.7	7.82	2.41	12.5
3	19.8	1.56	15.7	0.65	10.8	8.75	11.3	6.05	3.27
4	9.13	10.1	5.65	7.4	9.85	10.9	8.09	9.29	8.48
5	9.02	9.82	11.1	6.91	9.57	7.83	7.97	7.48	11.0
6	10.9	8.56	6.03	8.71	9.98	9.35	9.56	8.69	8.81
7	11.6	12.6	8.54	7.30	8.18	6.53	8.97	8.01	7.37
8	9.46	9.5	10.4	9.71	9.66	8.13	9.78	10.1	7.91
9	7.23	11.8	9.8	7.74	9.75	8.65	8.25	7.23	10.3
10	7.77	10.9	9.22	10.4	11.1	8.33	8.18	10.1	9.50

H = High, L = Low and M = Medium

convergences, the BN inferred that for the given preference book_price_service is a better choice than any other services. Here the user ignored or not interested in the remaining QoS such as response, availability and throughput.

Motivating example 2-find a best professor in a university: In the same manner, to find a best professor from a University, there are 11 services with same functionality have been taken from OWL-S TC, such as:

university_academic_sup_staff,
university_lecturerCurrentSem,
university_lecturer_Recommend,
university_lecturer_academia,
lecturer_in_MunichUniversity,

lecturer_in_SaarlandUniversity,
lecturer_in_ZambiaUniversity,
university_professor_academia,
university_research_fellow,
university_researcher, university_senior_lecturer.

A user wants to find a best professor from his academic area of interest and he now in a need of a web service with high response, high performance, high availability and low cost, low latency. The probabilities of the 11 services for the given requirement are as shown in Fig. 5. In this case the lecturer_in_Munich University service attained 13.8% when compared to the nearest low of 12.7% probability of university_professor_academia service. The various possible probabilities of the Book Purchase example and finding a Professor in a University example are given in Table 3 and 4, respectively.

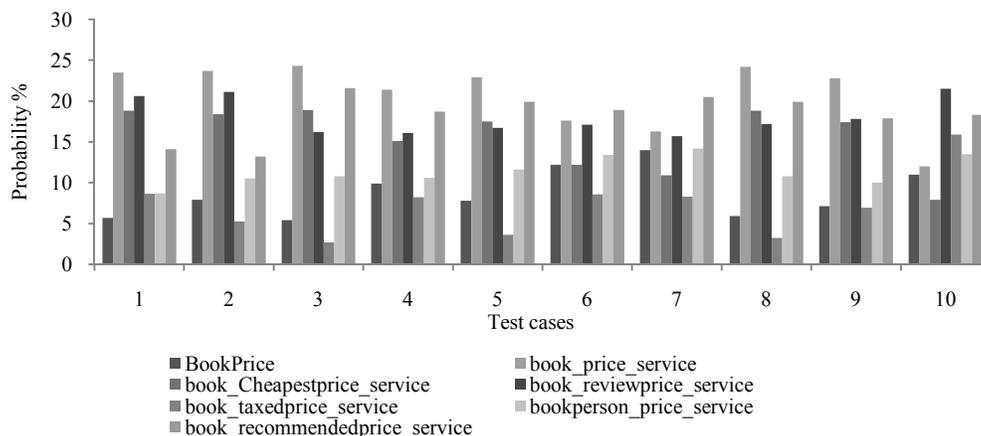


Fig. 6: The probabilities of the services of book purchase example

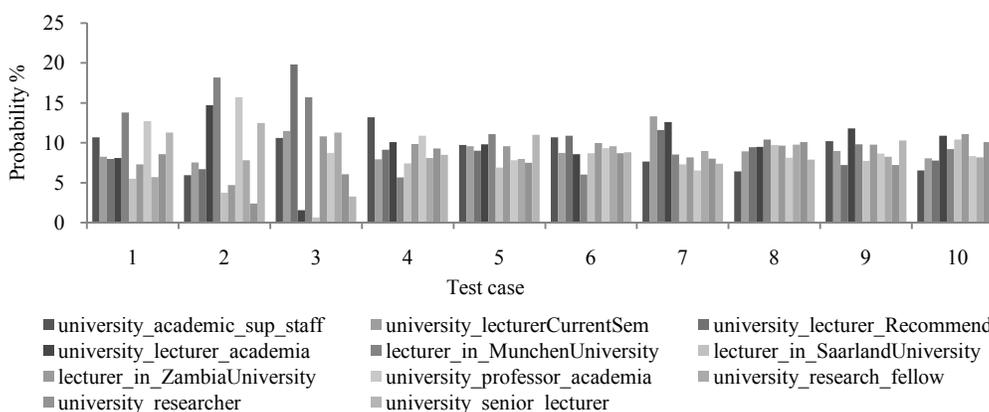


Fig. 7: The probabilities of the services of finding a best professor in a university example

DISCUSSION

The BN have been compiled several times with several QoS requirements for both Book Purchase and Finding a Best Professor in a University examples and the result obtained is shown in Table 3 and 4. The graphical representation is shown in Fig. 6 and 7. It is to be noticed that, for each different requirement, the probability of the services differ according to the probability distribution given to the QoS parameters. For the Book Purchase example bookperson_price_service is having a highest probability than any other services except test case numbers 6, 7 and 10. In the test cases 6 and 7, book_recommendedprice_service is having the probability of 18.9% and 20.5%, respectively, whereas in test case number 10, it is book_reviewprice_service which attained the probability of 21.5%. However, bookperson_price_service got 17.6, 16.3 and 12.0% probability for test cases 6, 7 and 10, respectively. However, in test case 10, the QoS requirement consists of medium availability and ignored the performance. Hence book_review price_service have attained 21.5% and the service bookperson_price_service attained only

12.0%. The reason why the service bookperson_price_service have achieved high probability for many test cases is that the QoS parameters such as availability and performance are selected high for this test case. But in test case 6, the user has ignored the availability and has demanded high performance and in test case 7 and 10, the user has demanded medium performance and high availability. One thing we can pointed out here is in the Book Purchase example, if any user demanded high performance and availability in addition to other QoS parameters, then it is bookperson_price_service which would be a better choice than any other service. Similarly, in the Find a best Professor in a University example as well the probability of the services varies for different service QoS requirements and the probabilities attained by the different service for different requirements is shown in Table 4.

CONCLUSION

In this study a novel approach using belief network is proposed. The study proved that, when any service algorithms or approaches discover more services, then the belief network based approach will infer a best

service which the user intended to use. One of the main advantages of this approach is that, the users have the flexibility of selecting any of these QoS parameter that they are desired. One of the limitations of this study is that, the construction of the belief net work is manual and in future this will be automated.

REFERENCES

- Andrews, T., F. Curbera, H. Dholakia, Y. Golland, J. Klein, F. Leymann, K. Liu, D. Roller, D. Smith, S. Thatte, I. Trickovic and S. Weerawarana, 2003. Business process execution language for web services version 1.1. Retrieved from: <https://www.oasis-open.org/committees/download.php/2046/BPELV1-1May52003Final.pdf>. (Accessed on: April 16, 2013)
- Bruijn, J., H. Lausen, A. Polleres and D. Fensel, 2005. The Web Service Modeling Ontology WSMO. Retrieved from: <http://www.w3.org/Submission/WSMO/>. (Accessed on: April 20, 2013)
- Bruneo, D., S. Distefano, F. Longo and M. Scarpa, 2013. Stochastic evaluation of QoS in service-based systems. *IEEE T. Parall. Distr.*, 24(10): 2090-2099.
- David, M., B. Mark, H. Jerry, L. Ora, M. Drew, M. Sheila, N. Srini, P. Massimo, P. Bijan, P. Terry, S. Evren, S. Naveen and S. Katia, 2004. OWL-S Semantic Markup for Web Services. Retrieved from: <http://www.w3.org/Submission/2004/SUBM-OWL-S-20041122/>. (Accessed on: May 2, 2013)
- Demian Antony, D. and V.S. Ananthanarayana, 2010. Dynamic selection mechanism for quality of service aware web services. *Lect. Notes Bus. Inf.*, 4(1): 23-60.
- Erik, C., C. Francisco, M. Greg and W. Sanjiva, 2001. WSDL Web Services Description Language. Retrieved from: <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>. (Accessed on: January 6, 2013)
- Florian, R., M. Anton and D. Schahram, 2008. Top-down business process development and execution using quality of service aspects. *Lect. Notes Bus. Inf.*, 2(4): 459-475.
- Georgios, M. and B. Nick, 2010. Structural and role-oriented web service discovery with taxonomies in OWL-S. *IEEE T. Knowl. Data En.*, 22(2): 278-290.
- Guinard, D., V. Trifa, S. Karnouskos, P. Spiess and D. Savio, 2003. Interacting with the SOA-based internet of things: Discovery, query, selection and on-demand provisioning of web services. *IEEE T. Serv. Comput.*, 3(3): 223-235.
- Huang, A.F.M., C.W. Lan and S.J.H. Yang, 2009. An optimal QoS-based web service selection scheme. *Inform. Sciences*, 179(9): 3309-3322.
- Ngu, A.H.H., M.P. Carlson, Q.Z. Sheng and P. Hye-Young, 2010. Semantic-based mashup of composite applications. *IEEE T. Serv. Comput.*, 3(1): 2-15.
- Kopecky, J., T. Vitvar, C. Bournez and J. Farrell, 2007. SAWSDL: Semantic annotations for WSDL and XML schema. *IEEE Internet Comput.*, 11(6): 60-67.
- Kun, Y., L. Weiyi, W. Xiaoling, Z. Aoying and L. Jin, 2009. Discovering semantic associations among Web services based on the qualitative probabilistic network. *Expert Syst. Appl.*, 36(5): 9082-9094.
- Kyriakos, K. and P. Dimitris, 2009. Mixed-integer programming for QoS based web service matchmaking. *IEEE T. Serv. Comput.*, 2(2): 122-139.
- Luc, C., H. Andrew, R. Claus and R. Tony, 2004. Universal Description Discovery and Integration (UDDI) Version 3.0.2. Retrieved from: <http://www.uddi.org/pubs/uddi-v3.0.2-20041019.htm>. (Accessed on: May 5, 2013)
- Mahbub, K., G. Spanoudakis and A. Zisman, 2011. A monitoring approach for runtime service discovery. *Automat. Softw. Eng. J.*, 18(2): 117-161.
- Matthias, K., F. Benedikt and S. Katia, 2009. OWLS-MX: A hybrid semantic web service matchmaker for OWL-S services. *J. Web Semant.*, 7(2): 121-133.
- Murray's Web Page, 2013. Retrieved from: <http://www.murrayc.com/learning/AI/bbn.shtml>. (Accessed on: March 4, 2013)
- Paliwal, A.V., B. Shafiq, J. Vaidya, H. Xiong and N. Adam, 2012. Semantics-based automated service discovery. *IEEE T. Serv. Comput.*, 5(2): 260-275.
- Roman, D., U. Keller, H. Lausen, J. Bruijn, R. Lara, M. Stollberg, A. Polleres, D. Fensel and C. Bussler, 2005. Web service modeling ontology. *Appl. Ontol. J.*, 1(1): 77-106.
- Sean, B., H. Frank, H. Jim, H. Lan, L. Deborah, F. Peter and A. Lynn, 2004. OWL Web Ontology Language. Retrieved from: <http://www.w3.org/TR/owl-ref/>. (Accessed on: May 14, 2013)
- Sioutas, S., E. Sakkopoulos, C. Makris, B. Vassiliadis, A. Tsakalidis and B. Triantafillou, 2009. Dynamic web service discovery architecture based on a novel peer based overlay network. *J. Syst. Software*, 82(5): 809-824.
- Staikopoulos, A., O. Cliffe, R. Popescu, J. Padget and S. Clarke, 2010. Template-based adaptation of semantic web services with model-driven engineering. *IEEE T. Serv. Comput.*, 3(2): 116-130.
- Tao, H., L. Lei and W. Jun, 2008. Efficient approach for web services selection with multi-QoS constraints. *Int. J. Coop. Inf. Syst.*, 17(3): 349-371.

- Wei-Li, L., L. Chi-Chun, C. Kuo-Ming and Y. Muhammad, 2008. Consumer-centric QoS-aware selection of web services. *J. Comput. Syst. Sci.*, 74(2): 211-231.
- Wu, G., J. Wei, X. Qiao and L. Li, 2007. A Bayesian network based Qos assessment model for web services. *Proceeding of the IEEE International Conference on Services Computing (SCC'2007)*, pp: 498-505.
- Xu, B., S. Luo and Y. Yan, 2010. Efficient composition of semantic web services with end-to-end QoS optimization. *Tsinghua Sci. Technol.*, 15(6): 678-686.