Direct trust evaluation model for composite web services (Using Baye's Theorem)

1.Ms. F.Ezhil Mary(Asst.Prof / MCA), 2. Lovelesh Sharma, 3. Hans Raj
SRM University, Kattankulathur Chennai 603203

Abstract:

Web services consist of information, software or other resources, and make them available over the network via standard interfaces and protocols. Complex web services may be created by aggregating other individual web services. This is referred to as a *composite web service*. However,there are a few stubborn problems which exist in its architectures, for example, security. Changes to a composite service need to be well analyzed in order to ensure the trust of it. In this paper, we propose a modified Bayesian based confidence model that gives an explicit probabilistic interpretation of trust for composite web services.

Keywords- Web service; Trust model; Bayesian model; Composite web services

Introduction:

Web services which based on existing Internet protocols and open standards can provide a flexible solution to the problem of application integration. With the help of WSDL (Web Services Description Language), SOAP (Simple Object Access Protocol), and UDDI (Universal Description, Discovery and Integration), web services are becoming very popular in Web applications [1]. Currently, composition of web services is carried out by rchestration [14]. An orchestration is a workflow that combines invocations of individual operations of the web services involved. It is therefore a composition of individual operations, rather than a composition of entire web services. For example online shopping website which is composition of various different web services like the shopping website and different payment websites which are connected to each other to provide aggregated web service to the user. In this paper we will discuss the stubborn problems of composite web services using Bayesian model.

Composite Web Services:

Using our model as a component model for web services, we can use standard web services as atomic components, composite web services as composite components, and use the composition connectors1 as composition operators for web services. This is

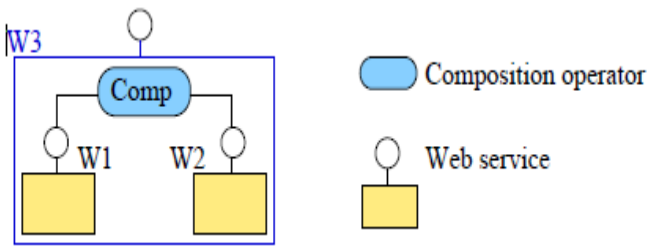


Fig. 1. Composite web services.

illustrated in Fig 1, where two services *W1* and *W2* are composed by a composition operator *Comp* into a composite service *W3*. *W3* is a web service, just like *W1* and *W2*. However, whereas *W1* and *W2* have interfaces described in standard WSDL, *W3* has an interface that cannot be described in standard WSDL, because W3 contains workflow embodied in the composition operator Comp. Therefore, in order to define W3 as a web service, we need to extend standard WSDL in order to incorporate workflow description. Then we need to devise a method to generate its interface in the extended WSDL from the standard WSDL interfaces of W1 and W2. The bank system in Figure 6 can be built as a composite web service composed from standard web services for ATM, BC1, BC2, B1, B2, B3 and B4 (Figure 2).

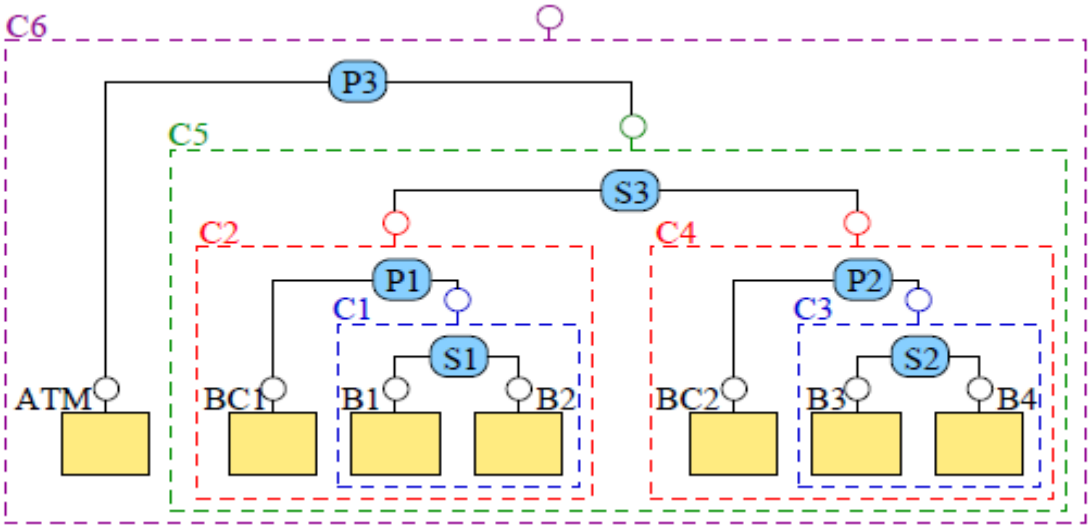


Fig. 2. The bank composite web service.

The composition is hierarchical (composite services are denoted by dotted boxes): B1 and B2 are composed into the composite service C1 by using the selection connector S1; the composite C1 is in turn composed with BC1 using the pipe connector P1, creating the composite C2; similarly B3 and B4 are composed into C3 by using the selection connector S2; the composite C3 is then composed with BC1 using the pipe connector P2, creating the composite C4; the composite C2 is in turn composed with C4 by using the selector connector S3 to create the composite C5; the composite C5 is composed with ATM by using another pipe connector P3, creating the composite C6. The composite service C6 provides all the operations offered by its sub-services.

Defining Composite Web Services:

Composite web services can be defined in two ways: Centralized Orchestration and Decentralized Orchestration,

Centralized Orchestration:

Composite web services may be developed using a specification language such as BPEL4WSand executed by a workflow engine such as Websphere Business Integration Server Foundation Process Choreographer and BPWS4J. Typically, a composite web service specification is executed by a single coordinator node. It receives the client requests, makes the required data transformations and invokes

the component web services as per the specification. We refer to this mode of execution as *centralized* orchestration.

Decentralized Orchestration:

Specifying a composite service using a language like BPEL4WS has interesting ramifications. The specification can be analyzed using techniques such as program analysis , petri-nets, etc. In the *Symphony* project, we analyze a composite service specification for data and control dependences and partition the code can into smaller components that execute at distributed locations. We refer to this mode of execution as *decentralized orchestration*.In *decentralized orchestration* of composite web services, there are multiple engines, each executing a composite web service specification (a portion of the original composite web service specification but complete in itself) at distributed locations. The engines communicate directly with each other (rather than through a central coordinator) to transfer data and control when necessary in an asynchronous manner.

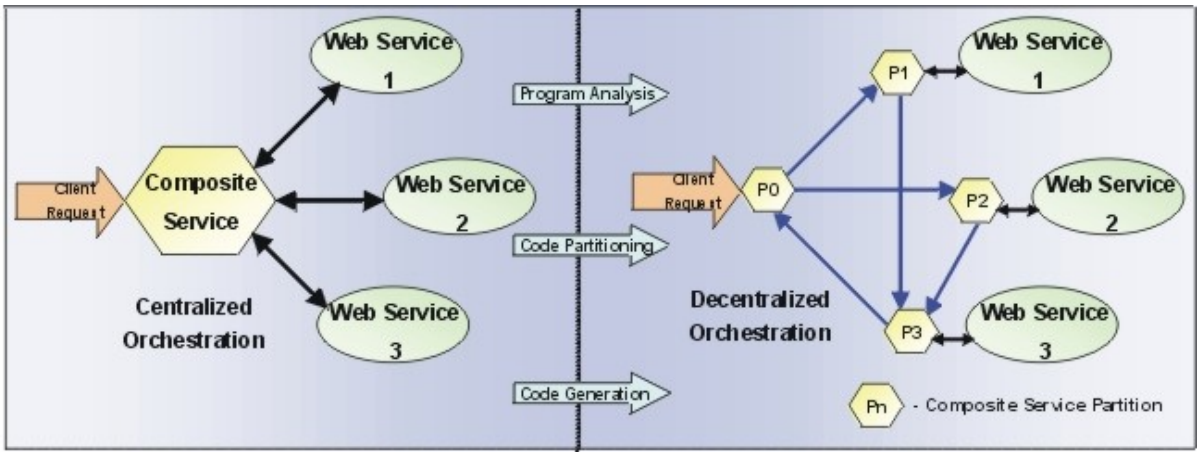


Fig3: Composite web services

WEB SERVICES AND STANDARDS:

A good starting point for understanding the web services paradigm is to consider the stated goals, as found in the literature and the standards communities. The basic motivation of standards such as SOAP and WSDL is to allow a high degree of exibility in combining web services to create more complex ones, often in a dynamic fashion. The current dream behind UDDI is to enable both manual and automated discovery of web services, and to facilitate the construction of composite web services. Building on these, the BPEL standard provides the basis for manually specifying composite web services using a procedural language that coordinates the activities of other web services. The underlying structure for the web services

paradigm will most likely be guided by already established standards and practices. Some of the current standards are illustrated by the layered structure shown in Figure 4. Briey, web services interact by passing XML data, with types speci_ed using XML Schema. SOAP can be used as the communication protocol, and the i/o signatures for web services are given by WSDL. All of these can be

de_ned before binding web services to each other. Behavioral descriptions of web services can be de_ned using higher levelstandards such as BPEL, WSCDL, BPML, DAML-S, etc.

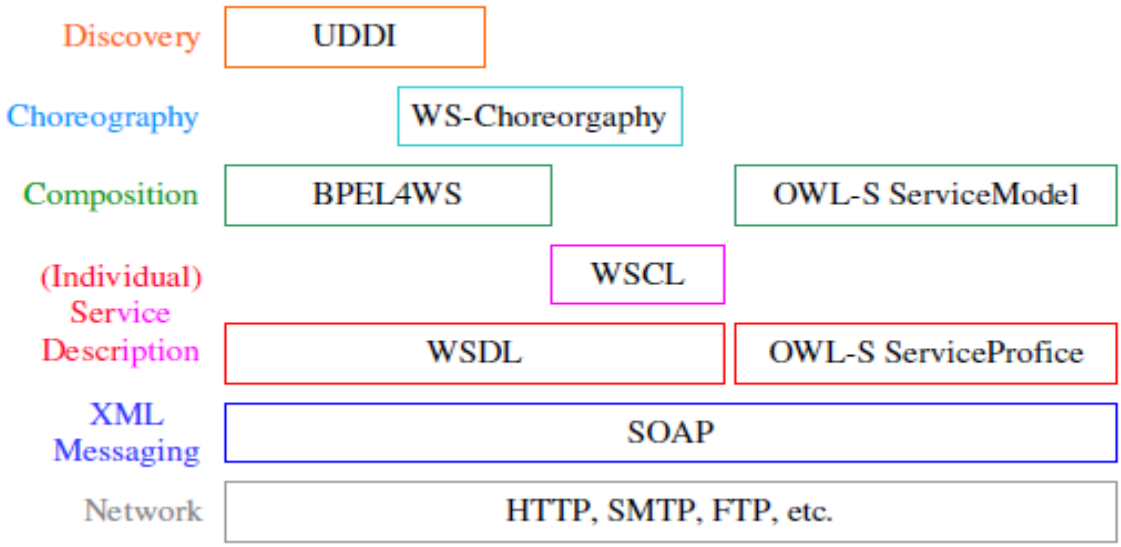


Figure 4: Web Service Standards Stack

In Figure 4, XML messaging and Network layers provide the foundation for interoperations or interactions between services.

Protecting Consumer Data in Composite Web Services :

The increasing number of linkable vendor-operated databases present unique threats to customer privacy and security intrusions, as personal information communicated in online transactions can be misused by the vendor. Existing privacy enhancing technologies fail in the event of a vendor operating against their stated privacy policy, leading to loss of customer privacy and security. Anonymity may not be applicable when transactions require identification of participants Many vendors have shown poor security of customer databases, leading to intrusions, loss of customer privacy and even identity theft [internetnews.com, 2003]. When back-end customer databases are copied, sold or linked with databases of other vendors, the wealth of available customer information rapidly increases. In some cases, customers trust a vendor with personal information, however the information is collected for processing by other (untrusted) parties along the chain, as seen in outsourcing and supply chain management [Medjahed et al., 2003].

SCENARIOS: HOW ONLINE TRANSACTIONS AFFECT CUSTOMER PRIVACY

3.1 Scenario 1: Online brokers

A customer uses on online bookseller web service as the vendor to locate a textbook. After finding a suitable match, the customer decides to purchase the package from the vendor. Current practices require customers to log into the vendor's website with a previously established account that probes for customer identity information. SSL/TLS is used for encrypting credit card information, which is generally handled by a payment gateway, not the vendor. The

vendor redirects customers to a payment gateway, and once payment is complete, the payment gateway returns an outcome to the
vendor. Despite what may be stated within the vendor's privacy policy, SSL/TLS does not prevent the vendor from disclosing consumer spending habits to other parties.

3.2 Scenario 2: Composite web services:

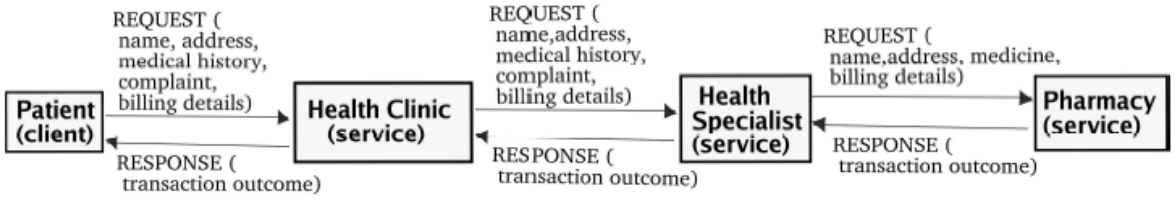


Figure 5. Composite web services

A customer seeks medication by lodging a request to an online health
clinic and must log in for identification. As with Scenario 1, the previously
established account may require a number of personally identifiable
customer attributes deemed private in nature. The health clinic is a front-end only, outsourcing medical knowledge to a specialist back-end service, as shown in Figure 5. Furthermore, if medicine is required, the specialist
outsources prescription services to a pharmacy. The customer may not be
aware of multiple vendors operating to fulfil their transaction. Each of these
back-end services will request customer details from the front-end service to perform their business activity, possibly without customer knowledge.
Privacy policies of back-end services may be independent to the health clinic privacy policy agreed to by the customer.

Bayesian Model:

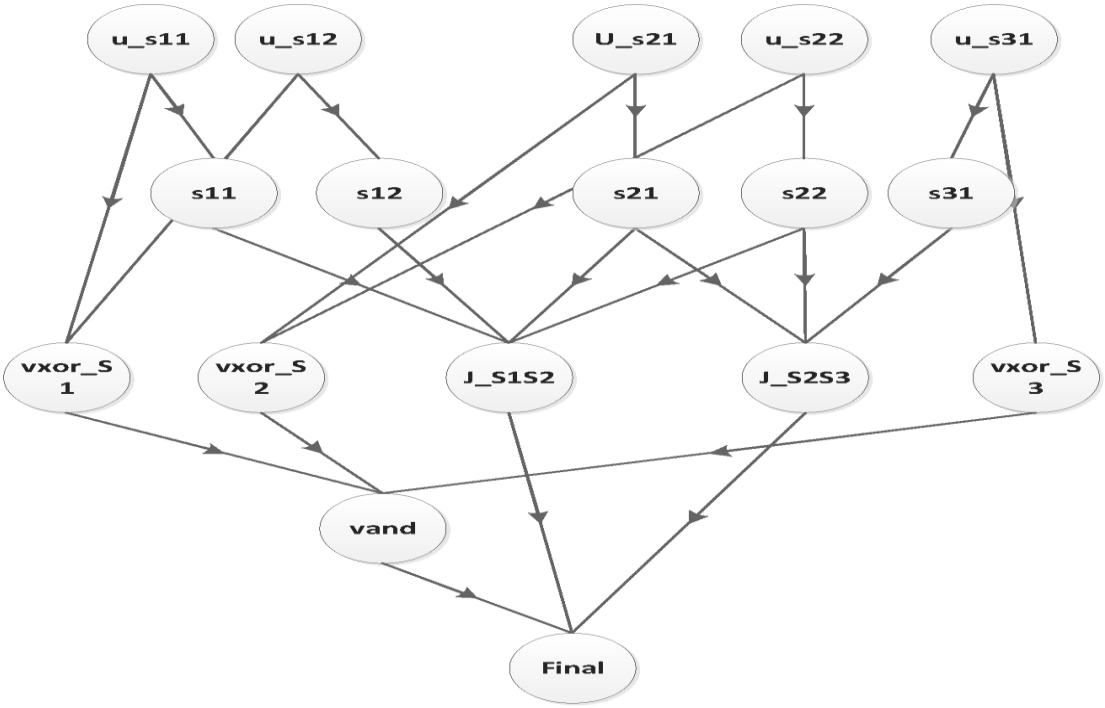
Bayesian model provides the means to compactly represent the joint probability of a set of variables. It provides a systematic and localized method for structuring probabilistic information about a situation into a coherent whole. It also provides a suite of algorithms that allow one to automatically derive many implications of this information, which can form the basis for important conclusions and decisions about the corresponding situation (Darwiche 2010). It is a compact representation of a probability distribution that is usually too large to be handled using traditional specifications from probability and statistics such as tables and equations
(Jensen and Nielsen 2007). A Bayesian Model for a set of variables is formally defined as a pair of (G; _), where: G is a directed acyclic graph (DAG) over variables Z, called the model structure, and _ is a set of Conditional Probability Tables (CPTs), one for each variable in Z, called the model parameterization (Darwiche 2009). The parameter G representing the DAG in the Bayesian model, encodes the qualitative part of the
model, showing how variables influence their descendant variables and how each variable is conditionally independent of its non-descendants given the state of its parents. On the other hand, the parameter _, represents the quantitative parameters of the model, which are described in a manner which is consistent with the Markovaian property between

each variable and its parent (Darwiche 2009) (Jensen and Nielsen 2007).

Using the Component Trust value for combination of web services that we previously measured, we can compute the global trust value of the composition. For this, we provide the required procedure for modeling the web service composition as a Bayesian model, by incorporating the previously explained trust values into the conditional probability tables (CPTs) of the nodes in the model. This model, which was inspired by the model for Reliability Block Diagrams, has some advantages: providing a graphical model depicting the dependency between the trust of the services and invocations and benefiting from the set of algorithms and queries available for Bayesian probabilities are among them. Consider a composition consisting of three service classes S1, S2, and S3, having 2, 2 and 1 candidates each that are invoked sequentially. First, for each of the candidates s11 to s21, we add a variable representing whether the service candidate is invoked in the composition or not. These nodes, shown as u s11, etc. in Fig. 2 , act as the root nodes of our model and have two outcomes: Used and Not used. We evenly distribute the probability of candidates of one class being invoked. So, for the two services s11 and s12
of class S1, each have the probability of 0:5 for being invoked. If class S had n service candidates, then P(u Si = Invoked) = 1=n. Obviously, the probability of the candidates not being invoked would be the complement of them being invoked, P(u Si = NotInvoked) = 1 □ P(u Si = Invoked). Next we add nodes representing the trust value of the service candidates and add an incoming link from the root nodes. These variables, shown as S11, etc. in Fig. 2, take two values, Trusted and Not trusted. Based on the value of the preceding node, it will either report the trust value of that candidate or 0 if that candidate is not invoked; P(s11 = Trusted j u s11 = Invoked) = Tr(s11), P(s11 = Trusted j u s11 = NotInvoked) = 0. The probability value for the case that the node in not trusted, i.e. s11 = NotT rusted would be the complement of it beingtrusted. Having the variables for all candidates in the composition, we then add a node representing the trust of the joint invocation. This node, shown as J S1S2 and representing
the composition of service classes S1 and S2, would have links incoming from all service candidates of S1 and S2. The output would be either Trusted or NotTrusted and for each pair of service candidates sij and si0j0, the value for their CR is inserted into the CPT of this node as the trusted value. For candidates from the same service class, the trusted value of the node would be zero. So far we can compute the trust value of a simple composition, consisting of only two service classes. For cases where the size of the composition is greater, we would have to add nodes to join the joint invocation nodes, J SiSj. Either more nodes would be added between every two J i, or one node would be added between all J is. They would have an AND CPT, giving a Trusted value for only the case where all inputs are Trusted. Also, we add nodes showing the validity of the selected service candidates. The nodes vxor Si shown in Fig. 2, act as an XOR function to ensure that no two services from the same class are selected.

Its incoming links are from all nodes u sij from a class Si and the CPT has the value 1 for cases where only one of the u sij has value Selected. Finally we link all validity nodes vxor Si and the joint node to a new AND node to
ensure that services selection in all classes are valid. The trust value of the composition would then be the joint marginal for the final node, Pr(Final = Trusted; e), where the evidence is the valuation for all u sij of all service
classes si. It should be noted that the probabilistic service invocation type, where a few services are invoked alternatively using a certain probability value could also be shown in our model, by inserting their selection probability in

the CPT of u sij nodes, instead of assigning equal selection chances. Calculating the posterior marginals for variables in a model is one of the basic queries in Bayesian models. There are a few algorithms and approaches to infer the probability of the whole model given the CPT of each variable. Inference by variable elimination, inference by factor elimination and inference by conditioning are the main approaches for solving the problem. Each of these approaches have



their own benefits, yet they differ mainly in their space and time complexity. While the details of these approaches are out of the scope of this paper, we will use them in our experiments and compare their performance

Conclusion:

In this paper, we reviewed the concept of trust for web services and identified the context of services' invocation in the composition as one of the points that cannot be addressed using current solutions. We introduced the notion of Component Trust which uses the frequency of invocation of a pair or group of services as the measure representing their reputation. Next we described a procedure for modeling a service composition as a Bayesian model. By integrating the Component Reputation and service trust values into the model, we can then compute the global trust of the composition using the posterior marginal query of the model. We carried out experiments using simulated test cases and showed the results obtained using the SamIam tool. Although computing the global trust value of the composition was not a heavy task, however choosing the service execution with the highest trust value is a relatively complex process which Bayesian model queries such as Maximum A Posteriori (MAP) to our model. Other matters such as the unwillingness of some clients to provide the actual frequency of invocations, or the disjunction of the service trust computation method and the Component Reputation method could be mentioned as

the shortcomings ofthe current paper which we plan to study in our future work.

Acknowledge and References:

[1]. Zhenmei Yu .et.al, "Small-World based Trust Evaluation Model for Web Service", 978-0-7695-4719-0/12 $26.00 © 2012 IEEE DOI 10.1109/CSSS.2012.175

[2]. Shanshan Qi .et.al, "A Trust Impact Analysis Model for Composite Service Evolution", 1530-1362/12 $26.00 © 2012 IEEE DOI 10.1109/APSEC.2012.30

[3]. R.Joseph Manoj .et.al, "A Literature Review on Trust Management in Web Services Access Control", DOI : 10.5121/ijwsc.2013.4301

[4]. Mohammad Reza Motallebi .et.al, "Component Trust forWeb Service Compositions", AAAI Technical Report SS-12-04 Intelligent Web Services Meet Social Computing

[5]. Symphony: Decentralized Orchestration of Composite Web Services http://researcher.ibm.com/researcher/view_project.php?id=3802

