An approach for verifying Web Service composition using Interface Automata

Mr.R.Saravanan1, Mrs N. Danapaguiame2, K.Rajalakshmi3, R.Nivetha3, M.Vedhavizhi3
1 Assistant Professor, M.E, Department of Information Technology
2 Research Scholars, Department of Computer Science Engineering
3 B.Tech, Department of Information Technology
Sri Manakula Vinayagar Engineering College
Puchucherry,India
Email id: r.saravanan26@gmail.com

Abstract— Web services is one of the key technologies for cloud computing. A single web service can offer only limited function to which does not provide efficiency for business application. Thus the composition of different web services provides abundant functions for distributing applications efficiently. A verification algorithm is designed to verify the sequence flow of the composed service. In the existing system, composition and algorithm was used to verify the composition. It fails to determine the reachability of services and the representation of services in automata. Thus in the proposed system, a modified interface automata is used to achieve the composition of services and also a verification algorithm is designed to validate the execution sequence of services.

Index Terms— Web service; Web service composition; Modified Interface Automata.

1 Introduction

A Web service is a software function provided at a network address over the web or the cloud. Interoperability has highest priority. A Web service is a method of communication between two electronic devices over the World Wide Web. It is a software system designed to support interoperable machine-to-machine interaction over a network. Web services describes a standardized way of integrating Web-based applications using the XML, SOAP, WSDL and UDDI open standard over an Internet protocol backbone. It has an interface described in a machine-process able format. Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards. Web service composition is done in order to provide abundant functions.

It provides an open, standard-based approach for connecting web services together to create higher level business processes. Interface automata used to model connectors in component-based systems which can be used not only in design and documentation but also in validation, in particular, for checking that interfaces of two components are compatible. It is an effort to build a distributed computing platform for the Web. It enables universal interoperability, dynamic binding and efficiently support both open and more constrained environments.

Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards. Web service composition is done in order to provide abundant functions. It provides an open, standard-based approach for connecting web services together to create higher level business processes. Composition refers to the way something is build, the new term at the moment is mash-up which basically means utilizing a variety of different services in a composite application. It provides an open, standards-based approach for connecting web services together to create higher-level businesses processes. The standards are designed to reduce the complexity required to compose web services, hence reducing time and costs, and increase the overall efficiency in businesses. So that functionality of disparate application can be used in one application. The basic requirement is to invoke the services in a synchronous manner, manage exception and transactional integrity, provide dynamic, flexible and adaptable framework. It is used to provide a clear separation between the process logic and the web services used. It has the ability to compose higher-level services from existing processes and .It is used to achieve reliability, scalability and adaptability required by IT environment. The major issues in web service composition are service discovery, service coordination and management, security, accountability, testing, adaptiveness, fault tolerance and scalability, reliability and transaction, uniform information exchange infrastructure. Interface automata are light weight formalism that captures temporal aspects of software component interfaces. Conventional type systems specify interfaces in terms of values and domains.

An automata-based language to capture both input assumptions about the order in which the methods of a component are called, and output guarantees about the order in which the component calls external methods. The formalism supports automatic compatibility checks between interface models, and thus constitutes a type system for component interaction. Unlike traditional uses of automata, this formalism is based on an optimistic approach to composition, and on an alternating approach to design refinement. According to the optimistic approach, two components are compatible if there is some environment that can make them work together. According to the alternating approach, one interface refines another if it has weaker input assumptions, and stronger output guarantees.

This project provides a web service composition model. Since single web service fails to provide abundant functions, the web services are integrated in order to provide various function requirements. In order to verify whether the service process generated satisfies the function requirements, a verification algorithm was used to validate the execution sequence of the services. It also proposes an algorithm to find whether the desire service is obtained or not through a Reachability algorithm. It uses states and links to represent the flow of execution and the modified interface automata.

It provides efficiency and consistency of service process through,

- Reachability
- Elimination of Bareness problem
- Modified interface automata

The related works are discussed in sec. 2 and proposed concepts are introduced in sec. 3, the working model are discussed in sec 4,the conclusion and future research are summarized in sec.6.

2 Related work

There are 3 kinds of approaches for describing and verifying services composition formally including

1. Approach based on Petri nets.
2. Approach based on process algebra.
3. Approach based on automata.

Petri nets are often used by researchers for implementing, analyzing and verifying services composition. The composition verification based on Petri nets is to use some theoretical methods of Petri nets, such as Reachability tree and reduction, to model and verify services. The valid form of the combination of Petri icons to combine and model service execution processes [1]. An algebraic method for describing the interrelationships between processes, process algebra is very theoretical, and mainly includes communicating system calculus, communicating sequential processes, and pi calculus [2]. Pi calculus is most widely used and studied in the field of web services composition verification. An automatic synthetic method to automatically integrate semantic web services based on pi calculus, and to describe web services formally with pi calculus model. Alfaro presented interface automata in 2001, applications based on interface automata are more and more widely used. There are many research results in the service computing field. For example, Reference [7] introduces an interface automata model with extended label attributes. The main function of that model is to cope with the case when there are mismatch data which is generated by calls between services.

An interface automata model with the function descriptions of semantic operations[3]. But it does not include the processing for service messages, and has not an algorithm to validate the business function of services. A services composition algorithm based on interface automata model. The algorithm uses interface automata model to model services, and the algorithm judges if the services composition can satisfy the practical demand according to specific business requirements [4]. An algorithm to realize the transformation between the BPEL processes to the interface automata model [5].

The algorithm can combine the existing service process execution language with interface automata. A service verification algorithm based on extended finite automata. The reference extends finite automata, using them as a tool to describe services formally [6]. After integrating services, deadlocks and live locks created can be processed with the tool. Besides, the reference gives an algorithm to judge whether the services composition matches the function requirements of user system. The behaviors of software components with interface automata are tested [7], which is a lightweight tool for formal modeling, on interface automata model. The algorithm uses interface automata model to model services, and the algorithm judges if the services composition can satisfy the practical demand according to specific business requirements.

Among these 3 kinds of composition verification methods, Petri nets can describe dynamic behaviors of services naturally and directly. But with the increasing in the number of services and the expanding of the size of services composition, the size of the Reachability tree would be huge, and time complexity of the verification algorithm would be worse. Among 3 kinds of methods, process algebra has the most strong formal expression ability and theoretical basis. But its graphical expression ability is also very weak, and too abstract to be easily understood by business user. Services composition models based on automata have the strong graphical expression ability. The drawbacks of all these models were overcome by automata. It is used to represent the services in the form of states and verify the attainable of the services. The verification of composition using automata uses deterministic approach to verify the composition but fail to check the traffic of services [9].

3 Reasearch proposal

In the existing system, single service does not provide the required function for the distributed application, web service composition was introduced. Web service composition is performed to provide more function requirements to business processes. In order to verify the composition, an verification algorithm is used. This algorithm verifies the execution sequence of the processes. Interface automata are used to provide the semantic descriptions of web service.

The graphical description of services in interface automata is more visual. The drawbacks in the existing system:

- The algorithm does not explain the flow through an real time application.
- The system uses only the deterministic approach.
- There is no solution provided for the occurrence of traffic in the path.

In order to overcome the drawbacks of the existing system, we propose a new system where the modified interface automata approach is used. The composition is verified by using a testing process, Modified Interface Automata (MIA).It verifies the composition and also represent the services in the form of states and path. The solutions for the Reachability problem and Bareness problem are given. The proposed system consists of many features where it is efficiently used for web service composition. The proposed system has also got binary compatibility for the existing system it also supports the verification algorithm of the web service.

The composition of the web service is verified by using Modified Interface Automata.

It includes two algorithms namely:

- Reachability algorithm
- Bareness algorithm

A.REACHABILITY ALGORITHM:

Reachability: Reachability defines whether there is path between any two nodes (i.e.) whether we are able to reach to any node from the existing node. This algorithm contains the states, path and a Boolean variable. If the Boolean variable condition is satisfied then the particular path is reached and the composition is correct else the path is not obtained and the composition is incorrect.

```
Algorithm 1: Reachability algorithm
Input: States, Final state and the path
Initialization: qMIA0, qMIA1 and Fs
1. while (path is not empty)
qMIA0 qMIA1
path. put(qMIA1)
2. if(fs in fMIA) reach=”true”
Output: Displays whether the path is attained
fs=Final state.
qMIA0 , qMIA1=states or nodes of automata.
    qMIA=modified interface automata.
```

B.BARENESS ALGORITHM

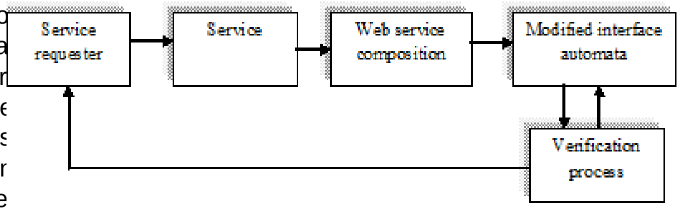
Bareness problem: It defines whether the language is accepted by the automata or not. This algorithm is used to overcome the occurrence of traffic in the network. If the state is present and it is final state then the problem is overcome. If the traffic occurs then different path is used to reach the final state. It uses non-deterministic automata in order to choose a new path to overcome the traffic.

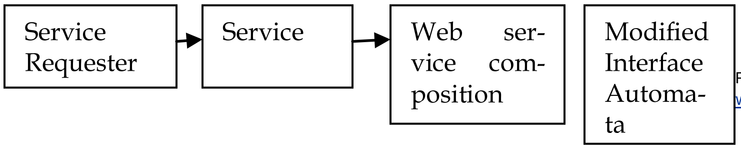
```
Algorithm 1: Bareness algorithm
Input: States, Final states, Automata parameters and the path
Initialization: States and path
1. Path.put(qMIA0)
2. while (path is not empty)
3. qMIA0 qMIA1
4. Path.put (qMIA1)
5. if(qMIA1 is in Q & qMIA1 is in fs)then

No Bareness problem
Output: The path is reached overcoming the
Bareness problem
fs=Final state.
    Q=Total states.
```

4 ARCHITECTURE

The architecture design consists of the Service requester who
ma
req
tior
pro
rea
ser
ove
eps
tior
the





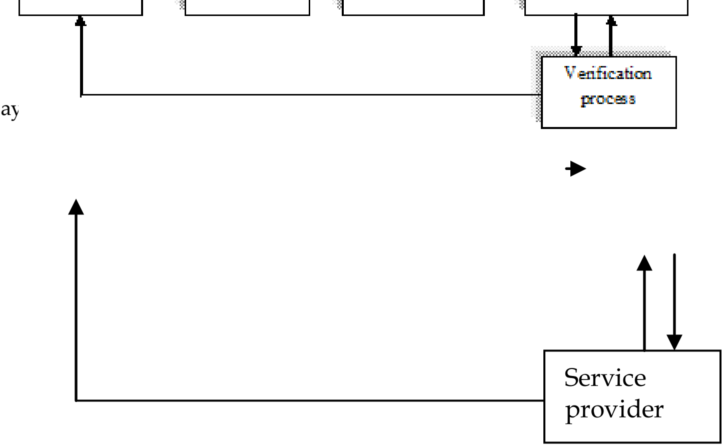


Fig 1: Proposed system architecture

5 CONCLUSION

Thus the Web service composition using Modified Interface Automata provides the composition of different services so that user can access easily. It also verifies whether the composition is valid by using the Interface automata. It uses Reachability algorithm to check the flow of the service. By using Bareness algorithm, the traffic is overcome. If there is a occurrence of traffic then by using non-deterministic automata concepts, a different path is chosen so that the particular service is reached without any disturbances. Web service composition can be done by new technique Different automata concepts can be used to perform composition and its verification process.

The extension of interface automata model with semantic of web service in the future. Our future work also deals with providing the dynamic re-composition of web service based on the network traffic and tool can be used to perform the sequence flow of each service and the status of the web services.

display equations to create better flow in a paragraph. If display equations do not fit in the two-column format, they will also be reformatted. Authors are strongly encouraged to ensure that equations fit in the given column width.

REFERENCES

[1] Peiyun Zhang, Bo Huang, Yamin Sun, “Petri-net-based description and verification of web services composition model”. Journal of System Simlulation, 2007, vol. 19, pp. 2872-2876.

[2] Yong-Lian Wang, Xue-Li Yu, “Formalization and verification of automatic composition based on Pi-calculus for semantic web service”, Proc. of the 2009 Second International Symposium on Knowledge Acquisition and Modeling.

[3] Tao Liu, Guosun Zeng, “Detecting and Resolving Mismatches between Pairs of Services”, Proc. of the 2010 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT), pp. 624-628.

[4] Samir Chouali, Sebti Mouelhi, Hassan Mountassir, “Adapting Component Behaviours using Interface Automata”, Proc. of the 2010 36th EUROMICRO Conference on Software Engineering and Advanced Applications, 2010, pp. 234-239.

[5] Liangming Li, Zhijian Wang, Longye Tang, “Research on the testing of component functional behavior”, Journal of Chinese Computer System, 2010, vol. 31, pp. 686-690.

[6] Huancheng Su, Zhiqiu Huang, Linyuan Liu, “Interface automatabased formal model for BPEL4WS Web service composition”. Application Research of Computers, 2009, vol. 26, pp. 1774-1777.

[7] Lihui Lei, Zhenhua Duan, “An extended deterministic finite automata based method for the verification of composite web services”. Journal of Software, 2007, vol. 18: 2980-2990.

[8] Liangming Li, Lei Liu, Zhijian Wang, “Research on interface automata testing”, Proc. of the International Conference on Computer Science and Software Engineering, CSSE, 2008, 2008, vol. 2, pp.415-419 .

[9] Jianhua Li, Hongyu Zhang,”*A Web Services Composition Model and its Verification Algorithm Based on Interface Automata”,* 2011 International Joint Conference of IEEE TrustCom-11/IEEE ICESS-11/FCST-11.