New Patterns Discovery for Web Services Composition from Mining Execution Logs

Tossapol Ritcharoenwattu and Wararat Rungworawut

***Abstract*—Web Services have became the key part in the software development rapidly. There are many web services to ready available. The problem is lacks of the method that leads to guideline compose these web services to be successful. Web services composition was referred from different web services in a business application using an engine tools for composition. Each web services composition is wrapped to WSDL to deploy on system. When web services composition can be invoked by user request and recorded each service usage into services execution logs. The execution logs will be recorded success of fail execution of web services composition. The objective of this paper purposed a framework that can guide new patterns of web services for composition using information of exist WSDL and execution logs. Its outcome was collected the results that web services have composed successful or fails. Execution logs are employed to construct data mining model using mining association rules technique to guide how to select web service for composition from frequent of set of services successful. Mining association rules help to extract patterns of the relationship between each web services automatically.**

***Index Terms*—Web services composition, WSDL, execution logs, data mining, mining association rule.**

I. Introduction

An organization has different developing business process including the foundation of system, types of information, or even equipment and initial data that is used for software development. Therefore, web service [1] is created in order to serve for communicating and exchanging information which is used by XML language [2]. As the complexity of business process is started increasing, a simple web service is not satisfied all the needs to achieve the target of business and can be composed to connect different web service working together. The common method of forming a web service composition is used by an engine as Business Process Execution Language (BPEL) [3]. BPEL involved in business workflow, which is executed from composite web services. These composed web services will be wrapped in Web Service Definition Language (WSDL) [4] and deployed for server application. WSDL is used to describe the character of a web services or already composite web services.

The concepts of web services are divided into two types, static web service composition and dynamic web service composition [5]. Static web service composition will defines the service provider to serve the customer requirement in advance. The advantage of static web service is certain in accuracy from desired complex connection of services. For example, an online bookseller needs two services; shipping services for carriers and payment services of financial institutions. The shipping services are composed with payment services to achieve business goals. Therefore, both of services will be wrapped to WSDL file and availability to be able work and play together. In addition, dynamic web service composition can find a service provider automatically. The advantage is able to adjust the service to the user and the environment. However the limitation is the uncertainties of demand which is no support the complex connection of services affect the errors in the service.

There are many researches in web service composition to facilitate the discovery composition patterns. The great number of research is form of dynamic web service composition. The research required to generate the plan the patterns to web service composition automatically in difference technique. In [6] represented how to make web service by using rules mapping. Rules mapping is a way to find the relationship between service and the rule that was made using input and output of service to find composition path effectively when compared to the original path of web service composition and reduce time when web service is desired increasing. Aspect of service mining is applied enhance semantic techniques between web services and construct a semantic relation-based web services registry with complementary functions [7], [8]. A survey of research areas has web service composition using execution logs [9], [10]. Because web services finally are deployed on application server that records the all usage of web services in execution logs. Therefore, it keeps all transaction even single web services or composite web services.

However, our approach is different from the existing approaches in the following aspects: we will present the advantage of properties both of exist WSDL files that are described composition and execution logs to extract new patterns to be composed. As mention, WSDL document describes each web services that have invoked. In addition, execution log describes overall web services composition successfully. Our framework will be combined the information of WSDL and execution logs to construct real association within a set of invoked web services in past time.

This paper is organized as follows. Section II is presented overview of data mining techniques. Section III presents framework for mining web service composition using web service execution logs. The experimental results are shown in Section IV. The conclusions and future work are presented in Section V.

Manuscript received October 20, 2012; revised November 24, 2012.

Tossapol Ritcharoenwattu is with Computer Center at Rajaphat Mahasakham University, Mahasakham, Thailand. (e-mail: cross1308@windowslive.com).

Wararat Rungworawut is with Department of Computer Science, Khon Kaen University, Khon Kaen, Thailand. (e-mail: wararat@kku.ac.th).

II. Applied Association Mining with Web Services Composition

Association mining was proposed in [11] to search for underlying relationships in the data or to find frequent pattern, association, correlation or other structures in the Item or Transaction Object in database, relational database, and other issues. We use Apriopi algorithm is the stage where the building techniques frequent item sets. The algorithm of Apriori is applied with web services composition. There are two-step processes as follows:

1) Find all frequent web service set: each of these web services will occur at least as frequently as a pre-determined minimum support count.
2) Generate association rules from the frequent web services which have composed: These rules must satisfy minimum support and minimum confidence.

Applied Association Mining with WS Algorithm

- Web services is a set of web services composition
  Let WS composition = {ws_{11}, …, ws_{ij}} where, ws_{ij} is the web service domain i that has composed j.
- An Mining association rule is define as follows,
- For example,

$$ws_{11} \Rightarrow ws_{21}$$

where,
$ws_{11} \subset$ WS composition, $ws_{21} \subset$ WS composition.

- Find all possible association rules with equal and greater than min support and confidence as pre-define.

Support is probability that a transaction contains ws_{ik} has composed with ws_{jl}.

$$\sup(ws_{ik} \Rightarrow ws_{jl}) = \frac{\#ws_{ik}\ has\ composed\ with\ \ ws_{jl}}{\#All\ transation} \quad (1)$$

Confidence is conditional probability that a transaction having ws_{ik} also composed with ws_{jl} absolutely.

$$conf(ws_{ik} \Rightarrow ws_{jl}) = \frac{\#ws_{ik}\ has\ composed\ with\ \ ws_{jl}}{\#all\ ws_{ik}\ composition} \quad (2)$$

Both of support and confidence value is between 0 and 1. Web service composition is good when the value is nearby 1.

III. Purposed Framework

We purposed a framework of web service composition that extracts new patterns for service composition to expected results successfully using mining association rule techniques. There are 3 main parts of our framework. Firstly, web services has ever composed that will be collected to web service server. When users request composition a set of web services in a business domain, then the application server collected a set of web services that ever have composed and deployed. Therefore, a set of web services composition is extracted from WSDL and the results of set web services composition are complete and success transaction from execution logs. These documents need to construct mining association rule. Thus, the data is transformed for rule creation. The transformed data will be processed by applied association mining with web services composition that are equal and greater pre-define minimum support and confidence.

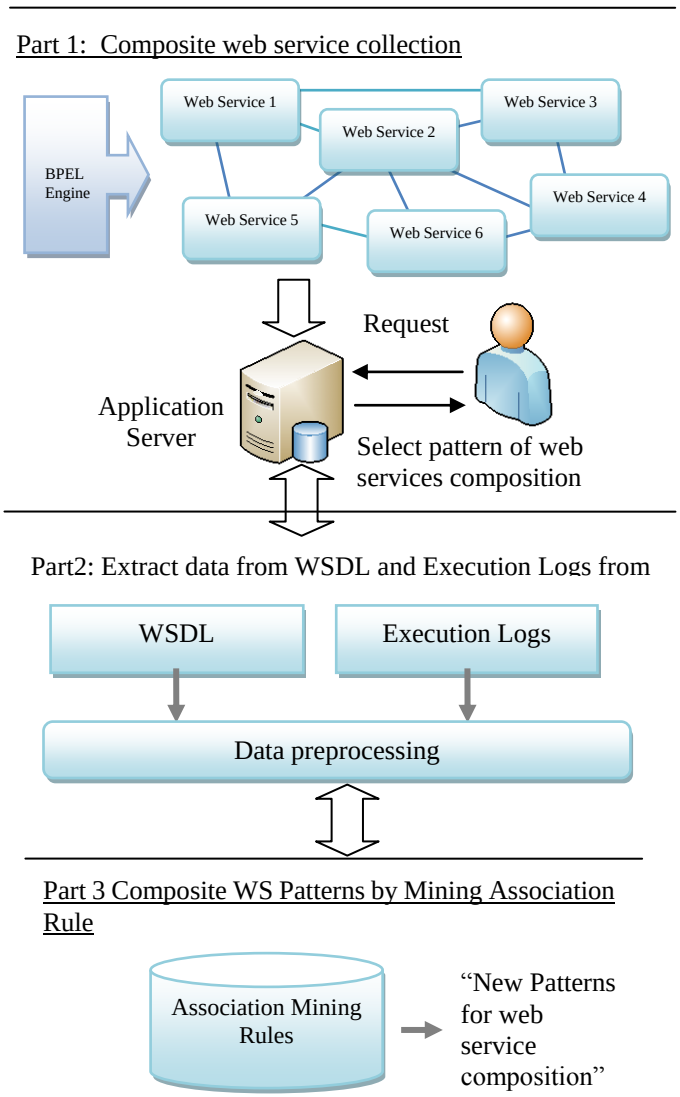

Fig. 1. Framework of web services composition from mining execution logs

A. Extract Data from WSDL and Execution Logs

The WDSL is used to describe the character of web services composition. We assume that the WDSL file is not enough to analyze the best rule to be success. At this step, execution log files show that is success or fail. On the other hand, it is no significant that web service had been composed or not. In this case, general execution log file is combined in application server.

The process was divided into 7 levels on [#|yyyy-mm-ddThh:mm:ss.SSS-Z|LogLevel|ProductName-Version|LoggerName|Key Value Pairs|Message|#]. The results showed that the application server will record logs of the same instance next to each others. It starts from the web service composition in the pattern that is successful and each services recorded was served successfully or not. If the services are successful, log file will show the recorded in

each other respectively.

Example of web services composition:

|2555-07-27T01:20:42.303+0700|FINER|sun-appserver2.1|com.sun.jbi.engine.bpel.com.sun.jbi.common.qos.messaging.MessagingChannel|_ThreadID=19;_ThreadName=sun-bpel-engine-thread-5;ClassName=com.sun.jbi.common.qos.messaging.BaseMessagingChannel;MethodName=removeTemplates;Process Instance Id=192.168.85.100:-5f783760:138c2f9c3ab:-7f89;**Service Assembly Name=travelcomptest1**;BPEL Process Name=travelservice1;_RequestID=f7992a9f-0295-41c2-999d-1b6363bc7bf8;|Removed template 192.168.85.100:-5f783760:138c2f9c3ab:-7f89:-2000003:-0 from mTemplatesMap, now have: |

Example of airline service:

|2555-07-27T01:21:09.122+0700|FINE|sun-appserver2.1|com.sun.jbi.httpsoapbc.OutboundMessageProcessor|_ThreadID=12;_ThreadName=HTTPBC-OutboundReceiver-2;ClassName=com.sun.jbi.httpsoapbc.util.DebugLog;MethodName=debugLog;Context=travelcomptest1-sun-http-binding-{http://airline1.provider.com/}bookAirline1;_RequestID=2c9579be-837d-4c3e-90c7-351e4cd884b5;|SOAP response:

Example of hotel service:

|2555-07-27T01:21:09.637+0700|FINE|sun-appserver2.1|com.sun.jbi.httpsoapbc.OutboundMessageProcessor|_ThreadID=14;_ThreadName=HTTPBC-OutboundReceiver-3;ClassName=com.sun.jbi.httpsoapbc.util.DebugLog;MethodName=debugLog;Context=travelcomptest1-sun-http-binding-{http://hotel1.provider.com/}BookHotel1;_RequestID=47568a30-765c-4fe7-a2b7-4c9907e93166;|SOAP response:

Example of car service:

|2555-07-27T01:21:10.162+0700|FINE|sun-appserver2.1|com.sun.jbi.httpsoapbc.OutboundMessageProcessor|_ThreadID=13;_ThreadName=HTTPBC-OutboundReceiver-4;ClassName=com.sun.jbi.httpsoapbc.util.DebugLog;MethodName=debugLog;Context=travelcomptest1-sun-http-binding-{http://car1.provider.com/}BookCar1;_RequestID=9006247b-dd6a-4189-ab2c-7c21d1f6c606;|SOAP response:

Example of map service:

|2555-07-27T01:21:10.682+0700|FINE|sun-appserver2.1|com.sun.jbi.httpsoapbc.OutboundMessageProcessor|_ThreadID=17;_ThreadName=HTTPBC-OutboundReceiver-1;ClassName=com.sun.jbi.httpsoapbc.util.DebugLog;MethodName=debugLog;Context=travelcomptest1-sun-http-binding-{http://map1.provider.com/}bookmap1;_RequestID=fb7380ec-ec90-4010-a927-86a0940d6aa1;|SOAP response:

Fig. 2. Example of execution log.

After that, the name of web service composition and each services will be extracted to compare with WSDL document (exported by JBI Descriptor type) which is the regulation of each service assemblies is needed. Our framework will search for service assemblies and use the name to in the storage area of the server automatically as Fig.3. The WSDL document is shown that are ordered by airline1, hotel1, car1 and map1 respectively. On the other hand, the travelcomtest3 is unsuccessful; it will be not recorded in execution log file. The map3 service is missing value in the table when the data is processed by mining association rule, it will be handled by fill in the most service concern with map3 which ever has invoked successfully.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<jbi xmlns="http://java.sun.com/xml/ns/jbi"
xmlns:ns1="http://enterprise.netbeans.org/bpel/travelservice1/travelservice1"
xmlns:ns2="http://airline1.provider.com/"
xmlns:ns3="http://hotel1.provider.com/"
xmlns:ns4="http://map1.provider.com/"
xmlns:ns5="http://car1.provider.com/" version="1.0">
  <services binding-component="true">
    <provides endpoint-name="wsairline3Port"
interface-name="ns2:wsairline3" service-name="ns2:wsairline3Service"/>
    <provides endpoint-name="wshotel3Port" interface-name="ns3:wshotel3"
service-name="ns3:wshotel3Service"/>
    <provides endpoint-name="wsmap3Port" interface-name="ns4:wsmap3"
service-name="ns4:wsmap3Service"/>
     <provides endpoint-name="wscar3Port" interface-name="ns6:wscar3"
service-name="ns6:wscar3Service"/>
    <consumes endpoint-name="travelservice3Port"
interface-name="ns5:travelservice3PortType"
service-name="ns5:travelservice3Service"/>
  </services>
</jbi>
```

Fig. 3. WDSL of a web services composition.

From WSDL and execution log file is transformed and combined documents in order to analyze as association rule. There are contents that need to use in mining association rule such as instance name from execution logs in a row and invoked web service name. For example, a set of instance of travelcomptest3 is composed and ordered by airline3, hotel3 and car3 that have invoked successfully. Therefore, the execution logs from application server that will be transformed to data transformation file in Table I.

TABLE I: EXAMPLE OF EXTRACT WEB SERVICE COMPOSITION.

Instance	Airline	Hotel	Car Rental	Map
Travelcomptest1	airline1	hotel1	car1	map1
Travelcomptest3	airline3	hotel3	car3	-

B. WS composition by Association Mining

In this paper, we develop a tool for web services composition testing to create new pattern. This case study, we try to demonstrate in domain of travel agency. This tool can input the name of services to define pattern of web services. In addition, it can discover the pattern of web services composition when it has chance fails in percentage of each service randomly in Fig. 4.

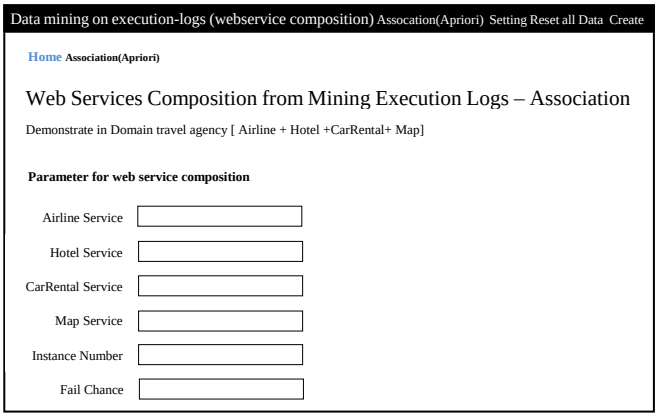

Fig. 4. A tool for web service composition by association mining

The step of usage tools is started to define parameter for web service composition in Fig.4. For example, it needs the service name for each serves. After that, tools will create test cases that mean create pattern web services composition as possible by user acceptance the percentage of chance fails. When the web service combine to preprocess data for association mining, the tools will be replace missing value to fill in services that are unsuccessful by the most probability service successful to eliminate noise data in Fig. 5.

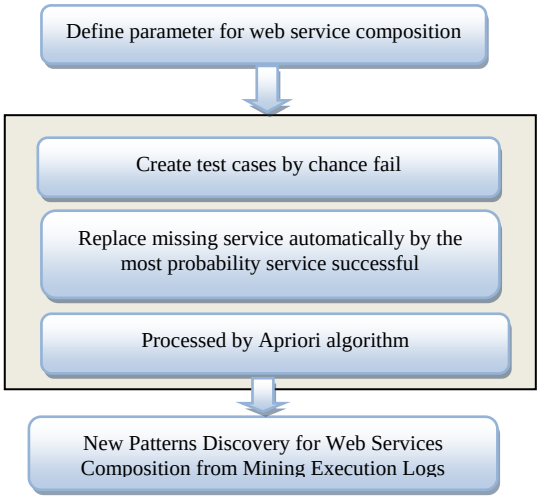

Fig. 5. The step of new pattern discovery for WS composition tool.

IV. Experimental Results

Our web service composition from mining execution log tool is experimented with original 5 service names. In addition, the number of instance for testing is overall 2,500 instances in difference name. It is made fail chances that are 5%, 10%, 15%, 20% and 25% respectively. The experimental results are shown the pattern of web service composition that are introduced the most probability selection of web service composition by support and confidence value. For example, pattern of web service composition is airline5->hotel5->car5->map5 by support 0.51 that means the opportunity composed this pattern successfully 51%. It proofed by history of web service composition 2,500 instances. Moreover, the confidence value is confidence ordered by this pattern; it ever was successful 97% in 2,500 instances. However, each service has fail chance 5%. Therefore, the best pattern is appeared ID5 that is airline3->hotel3->car3->map3 but it has chance fail 15% .

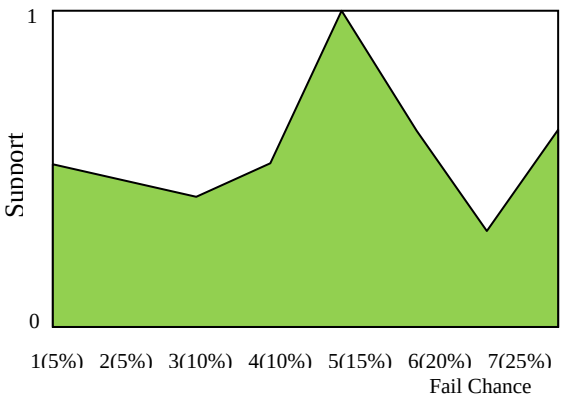

Fig. 6. Support value of 8 patterns of WS composition.

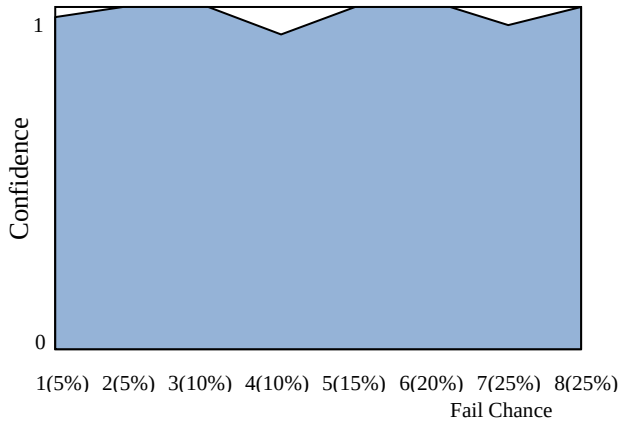

Fig. 7. Confidence value of 8 patterns of WS composition.

TABLE II: The Experimental Results of Pattern

ID	Fail	Pattern of WS composition	Sup	Conf	Type of pattern
1	5%	airline5->hotel5->car5->map5	0.51	0.97	Old patterns
2	5%	airline2->hotel2->car2->map2	0.46	1.00	Old patterns
3	10%	airline4->hotel4->car4->map4	0.41	1.00	Old patterns
4	10%	airline5->hotel5->car5->map5	0.52	0.92	Old patterns
5	15%	airline3->hotel3->car3->map3	1.00	1.00	Old patterns
6	20%	airline4->hotel4->car4->map4	0.63	1.00	Old patterns
7	25%	airline1->hotel1->car4->map1	0.31	0.95	New patterns
8	25%	airline1->hotel1->car1->map1	0.62	1.00	Old patterns

Discussion, our proposed method can introduce new pattern of web service composition. The ID7 is a new pattern that is airline1->hotel1->car4->map1 with support value for 31% and confidence value of 95%. This pattern is low support but high confidence in order to hotel1-> car4 is new patterns. However, the confidence value shown high correlation between them. This is advantage of web service composition because in real-life composition between services needs to discovery a new service or pattern to replace when web service is failure. Therefore, it found that the new patterns are appeared when has chance fail between 20% and 25%.

V. Conclusion

This paper presents a framework for new patterns discovery for web services composition from mining execution logs that have all the challenging aspect of real-life. Exist files are execution logs and WSDL files that can be contribute to use for analyze web services composition. These file can be solve to problem of web service composition. The mining associations between web services are very strong although web services have chance fail which refer the real environment resources. Moreover, the method is more flexible and robustness because it is developed as a tool for mining web service composition using mining association rules technique to guide web service successful. Mining association rules help to extract patterns of the relationship between each web services automatically. In future work, the tools can further analyze in

other data mining method such as decision tree for pattern discovery in difference web service compositions.

REFERENCES

[1] W3C. (2004). World Wide Web Consortium, Web Service Architecture. [Online]. Available: http://www.w3.org/TR/ws-arch.

[2] W3C. (2003). World Wide Web Consortium, Extensible Markup Language (XML). [Online]. Available: http://www.w3.org/XML.

[3] OASIS. (2007). Web Services Business Process Execution Language. [Online]. Available: http://www.oasis-open.org/committees/wsbpel.

[4] W3C. (2003). World Wide Web Consortium, Web Services Description Language (WSDL). [Online]. Available: http://www.w3.org/TR/wsdl.

[5] A. Alamri, M. Eid, and A.E. Saddik, "Classification of the state-of-the art dynamic web service composition techniques," in *International Journal of Web and Grid Services*, vol. 2, no. 2, 2006, pp. 148-166.

[6] T. Song and C. Wei, "Automatic composition of web services based on rules mapping," in *Proc. of the 2nd international Asia conference on Informatics in control*, Automation and Robotic, Wuhan, China, 2010, pp. 313-318.

[7] L. Jing-Shan, L. Hua-Ming, H. Zi-Feng, and X. Zhi-Wei, "A dynamic service composition method based on semantic interface description in pervasive computing," *Journal of Computer Research and Development*, vol. 41, no. 7, 2004 , pp. 1124-1134.

[8] S. Bayati, A. F. Nejad, S. Kharazmi, and A. Bahreininejad, "Using association rule mining to improve semantic web services composition performance," in *Proc. of the 2nd International Conference on Computer Control & Communication*, Karachi, Pakistan, 2009, pp.1-5.

[9] S. Ogasawara, K. Tayama, and T. Yamamura, "Estimating Structures of Business Process Models from Execution Logs," presented at the *10th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2006)*, Hong Kong, China, October 16-20, 2006.

[10] R. Tang and Y. Zou, "An Approach for Mining Web Service Composition Patterns from Execution Logs," presented at the International Symposium on Web Systems Evolution (WSE 2010), Timisoara, Romania, September 17-18, 2010.

[11] R. Agrawal, T. Imielinski, and A. N. Swami, "Association Rules between Sets of Services in Large Databases," presented at ACM SIGMOD international conference on Management of data, Washington DC, USA, May 25-28, 1993.

Tossapol Ritcharoenwattu received bachalor degrees in computer science from Rajaphat Mahasarakham Institute, Thailand in 2002. He works with Computer Center at Rajaphat Mahasakham University. His research interests include the web service composition and data mining.

Wararat Rungworawut received the Ph.D. degree in computer science from Chulalongkorn University, Thailand in 2010. She works with Department of Computer Science at Khon Kaen University, Khon Kaen, Thailand. Her research interests include the data mining, web services, applied ontology.