A Context-aware Adaptive Web service Composition Framework

Zhiying Cao, Xiuguo Zhang, Weishi Zhang, Xiong Xie, Jinyu Shi
Information science and technology
Dalian Maritime University
Dalian, China
czy_sophy@163.com

Haotian Xu
Transportation Management College
Dalian Maritime University
Dalian, China

Abstract—**It is a challenge for Web service composition to perceive dynamic changes of external environment and adapt to the changes of business process rapidly. In this paper, a context-aware adaptive Web service composition framework is presented and the functions of main modules are also described in detail. In this framework, BPEL is used to describe Web service composition process and a kind of special Web service called context service is defined to support context-awareness. Agent technology is introduced to enable context service to actively perceive and process contexts of Web service and Web service composition.**

Keywords- Context-awareness; Web Service; Service Composition; Agent; Self-adaptation

I. INTRODUCTION

A Web service is a self-described, loosely coupled, modularized, self-contained and platform-independent software component, which can be discovered and invoked via internet, to implement a particular functionality. In service oriented environment, some Web services are often combined together into a composed Web service to fulfill a complex business requirement, this is Web service composition. At abstract level a composed Web service can be defined as flexible processes composed of abstract Web services. At runtime, it will be implemented by invoking selected services. During the execution of the composed Web services, there are such constraints as the user's preferences, the Quality of Services (Qos) and external environment, which are called context, affecting service selection and process execution. When values of some context information changed, some selected Web services may be unavailable or maladjusted to the current process, the composition processes might be stopped in most existing Web service compositions which are artificial combination with poor flexibility and initiative, i.e., the variation of external environment and the users' requirements are not be considered. In order to solve this problem, a context-aware adaptive Web service composition framework which applies context-aware technology to Web service composition and uses Agents to achieve the self-adaptation of Web service composition. It can provide users with better experience.

The remainder of this paper is organized as follows: The Framework for context-aware Adaptive Web Service Composition is described in detail in Section 2. Section 3 introduces the adaptive composition process. The implementation of the proposed framework is discussed in Section 4. Section 5 reviews the related work on context-ware and adaptive web service composition. Conclusions are drawn in Section 6.

II. A FRAMEWORK FOR ADAPTIVE WEB SERVICE COMPOSITION

As shown in Figure 1, the adaptive composition framework of Web services that we proposed consists of the following three modules: the module of Web service composition, the module of context awareness and the module of adaptive management. In this system, BPEL is used for describing, parsing and implementing Web Service composition, the module of context awareness perceives the change of context and passes context values to the module of adaptive management in time, and the module of adaptive management updates BPEL process by selecting appropriate solution from the library of policies after it received the change of context values.

Contexts represent special parameters, which do not belong to any service but make the whole process of service composition more efficient and meet users' personalized requirements.

Contexts are non-functional attributes which are used to describe the user's preference and external environment that their changes will affect service execution. It is considered as the interaction among the user, the environment and the services. It can be used to describe environment information of an entity, that can be such a physical entity as users, locations and so on, or a virtual entity as software, Internet connection and so on. Context values are ubiquitous and often changed.

There are many classification methods for context. Under the pervasive computing, contexts are generally classified into user context, service context and physical context.

User context trace and record the user's status, mark the location, time and preferences of users while Web services are selected and executed. Service context describe the attributes and runtime state of the Web service, the Web service execution constraints, the response time of service itself, the reliability, completion rates of execution, current number of running instances and the maximum number of permitted running instances, and so on. Physical context

978-1-4799-6023-1/15 $31.00 © 2015 IEEE
DOI 10.1109/CICT.2015.68

describe the impact on service of the external environment, internet speed, weather, road conditions and so on.

In this paper, we assume that contexts involved in BPEL processes are known and controlled.

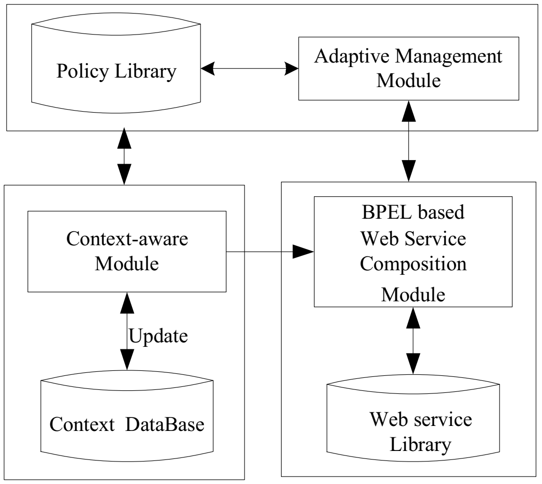

Figure 1. The context-aware adaptive composition framework

A. Web Service Composition Module

The Web service composition module in the adaptive composition framework is responsible for two jobs: one is gathering the endpoints of the relevant Web services and saving them into a database, some Web services may be provided by service providers, some are gleaned online automatically by capturing applications or retrieved from search engine. Figure 2 describes the process about gathering endpoints of Web services in detail. The other is describing Web service composition processes according to BPEL standard.

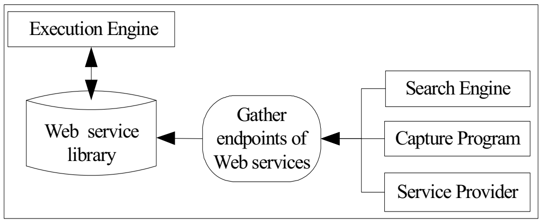

Figure 2. Process of gathering endpoints of Web services

In this paper choreography[1] is used to implement Web service composition, it can reduce the complexity of interaction between services and decrease the occurrence of errors. As the central coordinators, BPEL execution engine coordinates communication between services and makes the whole process well underway.

But BPEL specification can't meet with requirements for dynamic adaptive composition of Web services. Although aspect-based A04BPEL[2] specification and policy-based ActiveBPEL[3] specification are proposed in academia, it still can not realize adaptable web service composition. Business process described in BPEL only specifies the executable order of operations, Shared data and dependences of Web services[4], it can't dynamically perceive changes of context information and adjust their behavior dynamically according to that, so we define a special kind of Web services called context service. This service is responsible for perceiving and processing context information regularly, which make composite Web services have the ability of adaptive to the changes of context.

The Context service perceives the current values of contexts, compares with corresponding previous values stored in the database, and notify the execution engine to response if any context information is changed.

B. Context-aware Module

When the BPEL process is executed, the context-aware module is responsible for two things: one is to gather and check context information, as shown in Figure 3, the other is to perceive changes of the context value the process associated, and inform the adaptive management module to evaluate and respond to the changes.

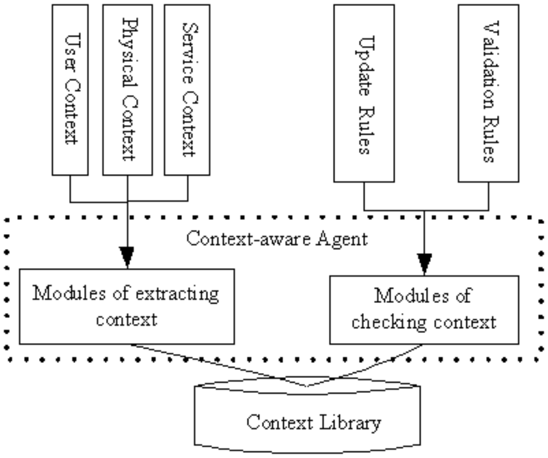

Figure 3. Context manager

Due to the universality, uncertainty and correlation of the context, it is difficult to give all examples of the context. In this paper, based on the classification method mentioned above, context can be classified into service contexts and service composition contexts according to the role and time they work in the composition process. Service context, often used before service instantiation, can help service composition framework to select service and optimize service composition according to context. Service composition contexts work while a composite service is performed. When perceiving the changes of contexts, service composition may need to make some adjustment as adding, deleting, or replacing a service, or fundamentally changing the whole combination process, that is to say, each service in the composition need to be re-instantiated, the composition process need to be reorganized and re-implemented.

Due to the universality, uncertainty and correlation of the context, it is difficult to give all examples of the context. In this paper, based on the classification method mentioned above, context can be classified into service contexts and service composition contexts according to the role and time they work in the composition process. Service context, often

used before service instantiation, can help service composition framework to select service and optimize service composition according to context. Service composition contexts work while a composite service is performed. When perceiving the changes of contexts, service composition may need to make some adjustment as adding, deleting, or replacing a service, or fundamentally changing the whole combination process, that is to say, each service in the composition need to be reinstantiated, the composition process need to be reorganized and re-implemented.

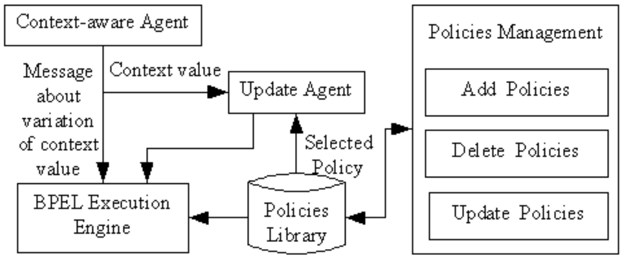

Figure 4. Structure of adaptive management module

C. Adaptive Management Module

As shown in Figure 4, in the adaptive management module, Update Agent is responsible for realizing the adaptive combination of Web services. When receiving message about variation of context value from context-aware Agent in the context-aware module, Update Agent will search the most suitable policy from a policy library and sent it to BPEL execution engine which will change the composition process according to it.

The policies in the library are put in advance according to the variation values of associate contexts. As changing Internet environment and user requirement will change the context value, it is need to add, delete or update the policy in the library to offer better solutions for users.

1) Policy Library

The policy library is a collection of policies. A policy is a plan or method chosen to guide and make a current and future decision. At present there are many policy description languages in the literature, such as the logic-based language, the rule-based language, the event-based language and so on. In this paper, the event-based policy description language is used. The basic syntax of this language is If Condition Then Action. In our context, Condition is a value of the context information, and Action is a solution associate with a context value.

A solution is an execution plan EP of a composition web service, which consists of a set of web services. In this paper, web services registered in UDDI will be classified into service categories by its function, e.g., services with the function of buying tickets belong to category A, and services with the function of querying weather forecast belong to category B. For the services in the service composition execution plan, it is important to mark which category it belongs to, because the weight of the service selected from one category is different from that from others. Each service in a solution is indicated with s_{iA} , where A is the name of a category which is the collection of Web services completing a specific task, and *i* is the index of task in the process specification of a composition web service.

Table 1 gives some examples of service composition EPs. For example, the EP (s_{1A}, s_{2A}, s_{3B}) is composed of three web services, and the services carry out task t_1 and t_2 is belong to set A, and the services complete task t_3 is belong to set B.

TABLE I. THE REPRESENTATION FORMAT OF WEB SERVICE IN THE POLICY LIBRARY

Service Categories / Sequence No.	A	B
1	s_{1A}	s_{3B}, s_{4B}
2	$\mathbf{s_{1A}, s_{2A}}$	s_{3B}
3	s_{3A}, s_{5A}	s_{1B}, s_{2B}, s_{6B}

The same context information may have multiple context values, each of which is correspond to some solutions. As mentioned above, initial policies in policy library were manually input, so the context value perceived and its corresponding solutions may be found or not in the library. If it exists, the best solution will be selected; otherwise, it means that a new context value appeared without corresponding solutions, then it will be compared with existing context values in the library and a solution of a proximate value will be chosen to generate a new one for it. If the new solution is adopted and functioning, a new policy including the new context value and the new solution will be stored into the library.

2) Adaptive Composition Algorithm

BPEL Execution Engine executes execution plans of composition web services and generates a log file called ProcessLog to record what services have been invoked. For example, during the execution of a solution ($s_{1A}, s_{2C}, s_{3A}, s_{4B}$) , the change of a context value is perceived while s_{3A} is invoked, the execution will be suspended and an execution path (s_{1A}, s_{2C}) indicated with S_{ex} has been saved in ProcessLog. In order to make the composition adapt to the changes of context value, Update Agent will search the most suitable solution from the policy library. The search algorithm is as following:

(1) To find out a set of solutions indicated with S associate with the context value

(2) To get the execution path indicated with S_{ex} from ProcessLog

(3) To find out a subset of S indicated with S' in which the beginning section of each solution is same with S_{ex}

(4) If S' exists, to find out the optimum solution from it; otherwise, a solution of a proximate value with the context value being processed will be chosen to generate a new solution for it.

In this paper, we assume that n quality dimensions such as execution time, price, reliability, etc. are considered while estimating a service. A quality dimension can be classified as positive and negative attribute. If the higher the value the higher the quality, a quality attribute is positive, otherwise, if the higher the value the lower the quality, it is negative. The optimum solution is the one which has the maximum aggregated score. The algorithm for searching the optimum solution is as following:

(1) To find out all corresponding service categories in S'

(2) For each service in S', to find out the service category to which it belongs

(3) For each service category A in S', to calculate the maximum raw value denoted with $\mathrm{U}_{\mathrm{k}}^{\max}(s_A)$ and minimum raw value denoted with $\mathrm{U}_{\mathrm{k}}^{\min}(s_A)$ of the quality attribute k (0<k<n)

(4) For each solution in S',

① To normalize raw value to a score for each attribute of each web service using a formula, as defined in (1):

$$u_k^{'}(s_{\mathrm{iA}}) = \frac{U_{\mathrm{k}}^{\max}(s_A) - U_{\mathrm{k}}(s_{iA})}{U_{\mathrm{k}}^{\max}(s_A) - U_{\mathrm{k}}^{\min}(s_A)} \bullet p_k \tag{1}$$

where $\mathrm{U}_k(s_{iA})$ indicates the raw value of, p_k indicates the weight associated with, and $u_k^{'}(s_{\mathrm{iA}})$ indicates the score of the kth quality attribute of service s_{iA} in the solution,

② To evaluate the aggregate score associated with each service by applying the Simple Additive Weighting technique. The score is calculated as

$$\boldsymbol{W(S_{iA}) = \sum_{k=1}^{n} u_k^{'}(s_{\mathrm{iA}}) \bullet x} \tag{2}$$

where with x=1 while the kth attribute is positive, x=-1 otherwise, and $\boldsymbol{W(S_{iA})}$ denotes the aggregate score of service s_{iA} in the solution.

③To evaluate the overall aggregate score associated with it. The calculated way is as defined in (3):

$$C(S_{\mathrm{j}}) = \sum_{i=1}^{m} W(\mathrm{s}_{\mathrm{iA}}) \tag{3}$$

where m is the number of services including in the solution.

(5) To find out the optimum solution with the maximum aggregated score.

III. ADAPTIVE COMPOSITION PROCESS

A. Web Service Adaptive Composition Process

The web service adaptive composition process is as shown in Figure 5. A choreographed composition process document in BPEL is parsed by BPEL parser and then it is performed by the BPEL execution engine. During its execution, Search Agent will select a best concrete Web service in UDDI for each abstract web service in the process, and Context-aware Agent will monitor values of context information. If any context value is changed, Context-aware Agent will suspend the execution of the composition process and interact with Update Agent which will select the corresponding solution of the context value in the policy library and update composition process according to it, and send the new process to the execution engine and continue with the execution.

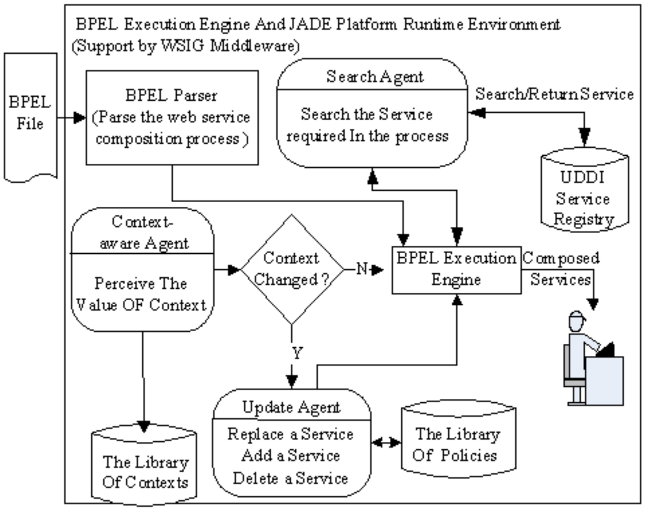

Figure 5. The Web service adaptive composition process

In the process is executed, if the execution is suspended while Context-aware Agent perceives the changes of context values, it is necessary to save the scene result performed already in order to resume execution when called again.

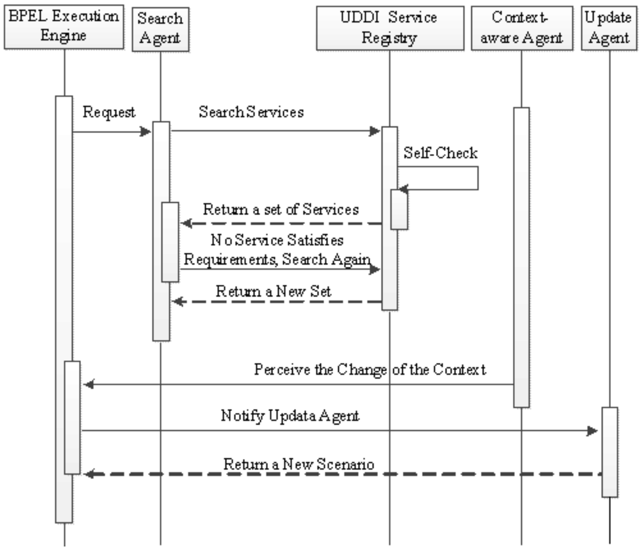

Figure 6. The interactive sequence between Agents

B. Interact between Agents

While the composition process of Web service is carried out, Agents interact with each other by encapsulated ACL messages which will be converted into SOAP messages by WSIG middleware in JADE. The interactive sequence among Agent is as shown in Figure 6. When the BPEL execution engine invokes a Web service, it sends a request to Search Agent to look up it dynamically in the UDDI registry and receives the search results. At the same time, Context-aware Agent will use its initiative to aware the outside change of the context value, pass the perceived change to

BPEL Execution Engine, and then the BPEL Execution Engine will call Update Agent in response to changes.

IV. IMPLEMENTATION OF THE ADAPTIVE WEB SERVICE COMPOSITION FRAMEWORK

The main function modules of this context-aware adaptive web service composition framework, including the design of BPEL process, the parse and execution of the BPEL document, Search Agent, Context-aware Agent and Update Agent are implemented using WSIG technology and Java language.

To deploy the system, the Agents developed with Java should be transited and packaged into web services by WSIG middleware, and these web services and the context service should be exported into War files and be deployed to the webapps directory of Tomcat web server. The BPEL process files which describe the composition specification of web services also should be deployed to webapps\ode\WEB-INF\processes directory and be parsed by ODE of Tomcat web server. Then lauch http://localhost:8080/ode/ and you will find the deployed system which can be executed in the Web service explore mode of Eclipse.

V. RELATED WORKS

With the rapid development of Web service composition, a growing number of adaptive approaches to solve problems arising in the implementation of Web service composition are proposed.

A. Charfi and M. Mezini[2] propose an aspect-oriented Web service composition approach in which the aspect-oriented idea and BPEL are combined together to help the service manager to get services more conveniently. Because of BPEL with poor dynamic and modularity, they extend BPEL to A04BPEL so that it makes Web service composition adaptive in the external environment. Reflection is a Self description and self control mechanism. With the development of adaptive technology, reflection mechanism is also used in adaptive composition of Web service.

American Carnegie Mellon University applies context-awareness to mobile service. Mobile terminal automatically collects the user information and ultimately provides appropriate services for users[5].

N.C. Narendra and Srinivas Gundugola propose a workflow operation rule which uses context information to work adaptively in 2006, and explain the adaptive workflow how to decide and adjust itself behavior by the adaptive behavior of Web service instances[6].

ZOU Wen-Ke and MENG Xiang-Wu apply the semantic Web and context-awareness technology to mobile service[7]. They extract the context information from intelligent mobile services and develop user profile ontology, context information ontology, situation ontology, service ontology and so on.

Zhang Xiangyu designs a environment adaptive reflection framework of Web service composition running which supports multi-strategy[8].

Maamar, Z et al. associate agent, Web service and context to complete adaptive component based composition by agent, Web and context modeling[9].

VI. CONCLUSIONS

In this paper, we have presented an adaptive combination framework for the composition of web services using context-aware perceiving and agent techniques, in which the composition process is described in BPEL, a context service, which is implemented with agent technique, is defined to perceive values of the context information, and an adaptive algorithm is proposed to dynamically select an optimum composition scenario while the context value is changed. It can improve the accuracy and effectiveness of Web service composition and make it more personalized. But using WSIG middleware to transmit message between web services and agents will, to some extent, affected performance of Agents and web services, how to solve this problem to improve the efficiency of the composition needs to be studied in the future.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China (Grant No. 61272172, 51179020) and the Provincial Undergraduate Innovation and Entrepreneurship Training Program (Grant No. 2014SJG041).

REFERENCES

[1] FCurbera,YGoland,J Klein,et al. Business Process Execution Language for Web Services[EB/OL]. http://dev2dev.bea.com/techtrack/BPEL4WS.jsp (last visit: 12th March 2007)

[2] A. Charfi, M. Mezini. Aspect-Oriented Web Service Composition with AO4BPEL. Proceedings of the 2nd European Conference on Web Services, Erfurt, Germany, 2004:168-182.

[3] LLC ActiveBPEL: The Open Source BPEL Engine[EB/OL]. http://www.activebpel.org (last visit: 2th September 2006).

[4] Esfandiari Babak, Tosic V., Towards a Web service composition management framework. IEEE International Conference on Web service(ICWS) , 2005:1-8.

[5] Sadeh N, Gandon F, Kwon O.B.. Ambient Intelligence: The MyCampus Experience. Technical Report CMU-ISR1-05-123, Carnegie Mellon University, July 2005.

[6] N.C. Narendra, Srinivas Gundugola. Automated context-aware adaptation of web service executions. IEEE International Conference on Computer Systems and Applications, 2006:179-187.

[7] ZOU Wen-Ke, MENG Xiang-Wu. Context-aware Intelligent Mobile Service Based on Semantic Web Technologies[J]. Computer Science, 2006, 33(9):30-33.

[8] Zhang Xiangyu. Research on Multi-Strategy Runtime Environment Adaptation for Composite Services[D]. ShenYang: Northeastern University, 2010.

[9] Maamar, Z.; Mostefaoui, S.K.; Yahyaoui, H.. Toward an agent-based and context-oriented approach for Web services composition. IEEE Transactions on Knowledge and Data Engineering, 2005, 17(5):686-697.