Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 10 (2012) 909 – 914

Procedia Computer Science

International Workshop on the Adaptation of Web Services

An Approach to Service Adaptation for Exploratory Application Construction

Wanghu Chen[a,*], Jing Li[a], Ji Zhang[b]

[a]College of Mathematics and Information Sciences,Northwest Normal University,Lanzhou 730070, China
[b]College of Life Science,Northwest Normal University,Lanzhou 730070, China

Abstract

Constructing applications with services shared by third parties is popularly adopted. Because of the autonomy of such services, mismatches are inevitable when to construct or execute such applications. An approach to service adaptation is proposed. Business process is introduced to support exploratory application construction. Adaptors for service adaptation are realized based on algebra operations of message documents satisfying constraints of the specific tree syntax. Experiment and evaluation show that the approach can be adaptive to heterogeneous of services and the variability of requirements, and can support exploratory application constructing efficiently.

Keywords: service adaptation; business process;adaptor; Syntax Tree

1. Main text

In some domains such as Bioinformatics and the Industry of Chinese Traditional Medicine, constructing applications through assembling services shared by third parties is widely adopted. E.g., in Bioinformatics, several autonomy research institutes have provided much more than 3000 Web services to share their rare resources such as genes databases and sequence analyzing software for users [1]. Therefore, small research groups or researchers can develop their applications based on these shared Web services instead of possessing all resources necessary themselves.

Workflows are the most popular measures to assemble such services into complex applications. To do so, services selected must cooperate well so as to fulfill the expected tasks. However, several challenges will still be faced during such a course. (1) Mismatches are usually inevitable for Web services to cooperate because their dynamic change and heterogeneity. Services with same functions may be described with different paradigms such as WSDL and REST. (2) The course for users to construct application may be exploratory [2]. That is to say, it needs to interactively develop, debug and run an

* Corresponding author. Tel.: +86-13893607357; fax: +86-0931-7971297.
E-mail address: chenwh@nwnu.edu.cn.

1877-0509 © 2012 Published by Elsevier Ltd. Open access under CC BY-NC-ND license.
doi:10.1016/j.procs.2012.06.121

application to solve current problems because the indetermination and variability of requirements makes it difficult to go through the usual constraints of the edit-compile-run-debug cycle. That is to say, it is hard to construct a fixed application with specific processing logic beforehand.

Above all, applications need to be reconfigured and optimized frequently according to current requirements. When a service assembled in an application altered, mismatches may occur among the new service for replacement and others. Thus, the new service may not be able to cooperate well with others have been assembled in the application. Therefore, an effective approach to service adaptation is essential to construct such applications.

The paper would focuses on mismatches among services and research a novel approach to service adaption. The rest sections of the paper are arranged as follows. First, the business process for exploratory application construction is discussed in section 2. Then, the adaptation of service in exploratory application construction is analyzed in section 3. In section 4, experiments results and evaluation are given to show the availability and efficiency of the approach proposed in the paper. At last, main related works are analyzed and some conclusions are given in section 5 and 6 respectively.

2. Business Process Enabling Exploratory Application Construction

It is not a good choice to construct applications with under-level services directly because they may change dynamically. Moreover, developers of exploratory applications are usually domain experts and are also users of the application at the same time. In our former work [3], business processes are proposed to constructed applications for user. Business process uses a kind of special services for end-users called Business Services, which can be understood by end-users and are stable relatively as a kind of abstractions to business activities [4]. Business Services will be mapped to a cluster of services that can fulfill the expected business function based on automatic matchmaking. When a business process is executed, Business Services as functional process nodes will be replaced by the most proper services. Business Service is produced based on domain knowledge and can be understood and handled by users for its rich business semantics. Thus, it is more feasible for user to construct exploratory applications by taking advantages of Business Processes.

When a segment of business process is constructed and to be executed, functional nodes will be replaced by under-level services. According to results got currently, users can modify the process or change the services picked for the specific process node before. Therefore, it can support users to realize exploratory application construction. The details would not be discussed in the paper.

It is mentioned above that each functional process node possesses a cluster of services. These services may be in different paradigms such as WSDL, REST and even Web pages. Therefore, it is necessary to describe such services in a unified way to support their cooperation.

It can be found that though these services may be in different paradigms, their inputs/outputs are all pieces of XML documents in running time. Therefore, their inputs/outputs schemas can be represented as some structures of tree type. Tree Syntax is an ideal measure to describe the inputs/outputs schemas of such services in a unified way.

Therefore, services in different paradigms can be defined as the model $S=<P, I, O>$, where, P describes the functions of a service based on actions or entities from the domain ontology. I and O denote the inputs/outputs of a service. To be noted, I and O are some kinds of abstraction to features of a domain action. They can be represented based on Tree Syntax [5].

Above all, applications of users will be business processes and each of their functional nodes would possess a cluster of candidate services. Because heterogeneous services have been described with a unified model, therefore, the cooperation between these services is feasible.

3. Adaptation of Services

To execute an application, each functional node should be replaced by a specific service according to current requirement and QoSs constraints. However, because the diversity of candidate services for a specific functional node, mismatches would exist inevitably. At the same time, when the service picked for a process node becomes disabled, mismatches also would occur when to replace it by a new one. So does the case when a service picked for the process node altered. These will hinder services from normal cooperation.

Mismatches mentioned above may include that of interface schema and content of service messages. Obviously, service adaptation is necessary to avoid all these mismatches. Therefore, adaptor would be introduced to maintain the cooperation of services. Adaptors are inserted between each pair of functional nodes of the business process.

An adaptor can be defined as $Adaptor = (\{s_i\}, \{OP_i\}, \{s'_i\})$, where, $\{s_i\}$ and $\{s'_i\}$ are services and $\{OP_i\}$ is a set of operators for adaption.

The schema of a service interface is described with Tree Syntax and the messages delivered to or produced by a service are trees consistent to the syntax. Therefore, it can be applied both to SOAP messages and RESTful messages, for their messages can all be considered as pieces of XML documents. Taking advantages of research achievements about Tree Syntax [6], messages of services can be defined as document trees as follows.

Let E be a set, each element of which is the 2-tuple of a parameter of a service and its semantic, D demotes element types of a message and basic data types, and ε denotes a special tree NULL, a message of a service can be defined as follows.

- ε is message of a service;
- $a(\omega)$ is tree message, where $a \in E$, $\omega \in D$ and D is a set of basic data types;
- $a(t_1\, t_2 \dots\, t_k)$, k≥1 is service message, where $a \in E$, $t_1, t_2, \dots, t_k$ are all service messages;
- No other service messages existing.

Thus, operator for service adaptation can be realized based on the algebra operations of Tree Documents [6]. The location of nodes in a tree document is denoted by paths in XPATH. For example, the path, //*program.fasta* means all sub-nodes of the node *program*.

3.1. Schema adaptors

Let m is a service message, e is a path expression on E, c is a condition expression like *path*=<*value*> and $a \in E$, operators for message schema adaption can be defined as follows.

$s(m, e, c)$, selection operator: Return message corresponding to the sub-tree of m that satisfies $e=c$.

$p(m, e)$, projection operator: Remove paths that not covered by path expression e, and return the message.

$c(m, a)$, deletion operator: Remove node a from the tree corresponding to message m, and let it refers to its former parent.

$r(m_1, m_2, e)$, replacement operator: Replace all sub-tree of m_1 satisfying e with m_2.

Operators for service adaptation above can be realized with query operators of Tree Syntax.

3.2. Message adaptors

Let m is service message, e is a path expression on E, c is a condition expression like *path*=<*value*>, operators for the adaptation of service messages can be defined as follows.

$u(m_1, m_2, e)$, union operator: Get sub-messages of $m_1$and m_2 according to the path e, and return the union of two sub-messages.

$i(m_1, m_2, e)$, intersect operator: Get sub-messages of $m_1$and m_2 according to the path e, and return the intersect of two sub-messages.

d(m_1, m_2, *e*), complementary operator: Get sub-messages of $m_1$and *m2* according to the path *e*, and return the complementary of two sub-messages.

It can be certificated that operators above is complete to adapting service messages and their schemas according to the expected message schema or document tree.

Thus automatic service matchmaking would be used to find the mismatches between two Web service to cooperate and proper adaptors would be inserted between them to eliminate mismatches. Of course, because adaptors are based on some kinds of algebra operation, they can be composed to solve complex mismatches.

3.3. Usage of adaptors

Adaptors mentioned above can be used to construct and adjust business processes. Some patterns are listed as follows.

Trim (*s*, *e*), removing some redundant elements in the schema of the service *s* based on the path expression *e*.

Merge ({s_i}, *e*), merging the output messages of services in the set {s_i} according to *e*.

Extract (*s*, *e*, *c*), extracting a segment of messages for the output message of service *s* according to the path expression *e* and the condition *c*.

Filter (*s*, *c*), filtering the output messages of a service *s* according to the condition *c*.

Transform (*s*, *e*), transform the schema of the message of a service *s* according to the condition *c*.

4. Experiments and Evaluation

To verify the availabilities of the approach proposed in the paper, Web services in the Industry of Chinese Traditional Medicine are taken. Some of these services are supplied by third organizations and deployed in the remote servers.

A segment of a simple business process is represented in Fig. 1(a). The nodes A, B, C, D and E denote Business Services. It can be found that the business process permits users to do supervising and adjustment. Each functional node of the process possesses several candidate services. E.g., the candidate services for the node A even amount to 50.

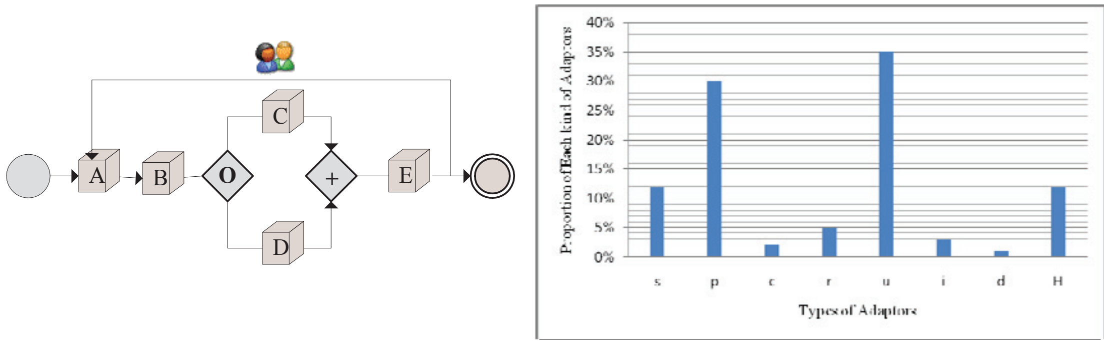

Fig.1 (a) Business Process of the scenario (b) The analysis of application of adaptors.

Experiment results show that service adaptation have played important roles. According to the segment of the business process, service cooperation occurred between each pair of service adjacent in the process. The cooperation points are denoted as AB, BC, BD, CE and DE. Taking AB as an example, each candidate service for A and B may cooperate to fulfill the expected task. During the course, adaptors proposed in the paper may need to be inserted to make cooperation feasible. Based on matchmaking, the table 1 illustrated the ratio of cooperation that adaptors must act on. It shows the adaptor is essential to enable available cooperation.

The segment of process is executed repeatedly for many times. The average proportion of cooperation points that need adaptors to act on can amount to 21.86%. According to the statistics results, the proportion of each kind of adaptors in all adaptors applied are showed in Fig. 1(b). The Notations *s*, *p*, *c*, *r*, *u*, *i* and *d* represents each type of adaptors proposed respectively. *H* means to do adaption by hands. Because the characteristic of data types of the Web services, the application of *c*, *r*, *i* and *d* are largely less than that of others. The adaptor *p* and *u* are used in high frequency to fulfill the adaption of service messages. At the same time, it can be found, based on the adaptors, about 12% of the adaption would be done with the surveillance of users. It is important in constructing exploratory application just-in-time.

Table 1. Applications of adaptors for service adaptation

Cooperation Points	AB	BC	BD	CE	DE
Amount of pairs of Web services may Cooperate	450	225	102	750	360
Amount of mismatching Web services pairs	126	49	19	104	98
Ratios of Web service pairs need to be adapted	28%	21.7%	18.6%	13.8%	27.2%

5. Related Work

Workflows in business-level are often used to improve service cooperation. A multi-view cooperation model is introduced in [7], and an approach to construct abstract business process for cooperation based on the hierarchical task network planning is introduced in [8]. Reference [2] discussed an approach to end-user-oriented exploratory service composition.

Workflow related approaches to service cooperation have aroused wide concerns. Reference [9] introduced an approach to bottom-up workflow interoperating called CoopFlow. A solution of service collaboration based on distributed coordination model is proposed in [10]. In [11] the Hierarchical Task Network (HTN) is used to enhance service cooperation. The functional requirements of users are fulfilled by atomic services and the constraint satisfaction problem (CSP) is applied to find such services.

As to service adaptation, some works focus on composite services adapting by substituting its out-of-dated sub-services based on the similarity of services [12]. Reference [13] focus on message mismatches of services and proposed several adaptation patterns. The work also researches how to integrate such pattern to do adaptation. Reference [14] analyzed behavioral mismatches of services based on dependence graph. Reference [15] discussed the approach to adaptation evaluation. The objective is to maximize the quality of satisfaction of users and environment, and to optimize the optional service selection. Reference [16] discussed both mismatches of messages and process, and proposed adaption patterns corresponding.

Compared to works above, the paper mainly focuses on the change and heterogeneity of services as well as the indetermination of requirement in exploratory application construction to research an available approach to service adaptation.

6. Conclusion

An approach to service adaptation for exploratory application construction is proposed. Experiments show that the approach is available in a heterogeneous services environment to support exploratory application construction, even if users' requirements are indeterminate or variable.

Acknowledgements

Support for this work has been provided by National Sciences Foundation of China awards 61040022.

References

[1]D. Hull, E. Zolin, A. Boyvkin, Deciding semantic matching of stateless services, Proceedings of the National Conference on Artificial Intelligence, AAAI Press, 2006, pp. 16-20.

[2] S. Yan, Y. Han, J. Wang, C. Liu and G. Wang, A user-steering exploratory service composition approach, Proceedings of IEEE International Coference on Service Computing, pp. 309-316.

[3]J. Wang, Y. Han, S. Yan, W. Chen, G. Ji, VINCA4Science: A personal workflow system for e-Science, Proceedings of the International Conference on Internet Computing in Science and Engineering, 2008, pp. 444 – 451.

[4]Y. Han, P. Zhang, W. Chen, C. Zhang, J. Wang, Business-Oriented virtualization of Web services: A case study, Proceedings of IEEE International Symposium on Service-oriented System Engineering, 2008, pp.275-280.

[5]W. Chen, J. Li, Agile services enabling transparent resources sharing in virtual organizations, Proceedings of International Conference on Systems and Informatics, 2012.

[6]B. Chidlovskii, Using regular Tree Automata as XML schemas, Proceedings of IEEE Advances in Digital Libraries, 2000, pp.89–104.

[7]H. Lin, S. Liu, Y. Fan, Service-Oriented enterprise cooperation: Modeling, Method and System, Proceedings of IEEE Asia-Pacific Conference on Services Computing, 2008, pp.1032-1037.

[8]L. Cui, J. Tian, H. Wang, Q. Li, Service cooperation in PaaS platform based on planning method, Proceedings of the 15th International Conference on Computer Supported Cooperative Work in Design, 2011, pp.367-375.

[9]S. Tata, K. Klai, N. O. Ahmed, A bottom-up approach to workflow cooperation for short-term virtual enterprises, IEEE Transctions on Service Computing, vol. 1, Apr. 2008, pp.214-228.

[10]X. Qiao, J. Wei, T. Huang, Service collaboration approach based on decentralized mediation model, Journal of Software, vol. 20, Jun. 2009, pp.1470-1482.

[11]I. Paik, D. Maruyama, Automatic web services composition using combining HTN and CSP, Proceedings of International Conference on Computer and Information Technology, 2007, pp.206-211.

[12]C. Lu, R. Chow, A light weight substitution-based adaptation approach, Proceeding of IEEE International Conference on Information Reuse and Integration, 2011, 490 - 491.

[13] Zhe Shan, Integrated service adaptation, Proceedings of World Congress on Services, 2010, 140 – 143.

[14]B. Wu, S. Deng, J. Wu, Y. Li, L. Kuang, Service behavioral adaptation based on dependency graph, Proceedings of IEEE Conference on Asia-Pacific Services Computing, 2008, 1276 - 1281.

[15]B. Zhang, Y. Shi, Y. Chen, A policy-based adaptation method for service composition, Proceedings of International Symppsium on Pervasive Computing and Applications, 2006, 619 – 624.

[16] S. Zhe, A. Kumar, P. Grefen, Towards integrated service adaptation: A new approach combining message and control flow adaptation, Proceedings of IEEE International Conference on Web Services, 2010, 385 – 392.