

Approach to adaptive service matchmaking

Chen Wanghu^{1,2,3} Han Yanbo¹ Wang Jing^{1,3} Liu Chen^{1,3} Wang Jianwu^{1,3}

(¹Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China)

(²College of Mathematics and Information Science, Northwest Normal University, Lanzhou 730070, China)

(³Graduate University, Chinese Academy of Sciences, Beijing 100039, China)

Abstract: To make service matchmaking more adaptive to various service requests and diverse web services, an adaptive approach—ASMA is proposed to service matchmaking based on temporal logic model-checking. The approach is based on the proposed abstract service model, ASM-TL, which addresses some important constraints for identifying capabilities of web services, such as service inner constraints and invocation constraints, and also has a virtual process model for describing service behavioral properties. By treating service requests as temporal logic conditions and web services as temporal models, ASMA does service matchmaking through model checking. Therefore, ASMA makes service matchmaking more accurate and more adaptive to the variety of service requests and the diversity of web services. The approach has been applied to the problem solving environment (PSE) for bioinformatics research. Applications show that the approach is suitable for dynamic environments.

Key words: service matchmaking; service model; model checking; temporal logic

More and more web services become available on the Internet. According to Ref. [1], about 3 000 such services are accessible in the biomedical area. According to our study, only web services about gene sequence similarity search provided by several authoritative bioinformatics organizations such as EBI, NCBI, and DDBJ etc. can amount to 50. At the same time, most bioinformatics research groups or individuals lack some rare resources such as sequencing devices and gene databases. Therefore, it is necessary and also feasible for such research groups or individual researchers to build a problem solving environment (PSE) system with these shared web services.

Fig. 1 illustrates our PSE for bioinformatics research. Researchers use business services^[2], a kind of service on the business level for end-users to access concrete service resources transparently, to build their dynamic experiment processes. To make business services accessible to concrete software resources transparently, proper web services should be bound to business services by means of service virtualization^[2]. Therefore, it is important to locate web services having consistent capabilities with given requests represented in

business services. Service matchmaking is the essential means to such end.

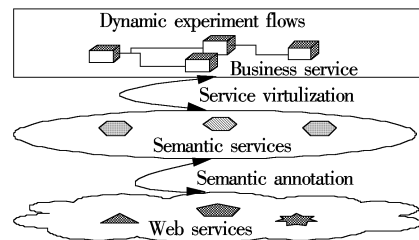


Fig. 1 Architecture of PSE

Definition 1 Service matchmaking represents the process of locating web services with capabilities consistent with service requests in a set of advertised services.

To do service matchmaking, some difficulties exist at present, such as the variety of constraints of service requests on the capabilities of services expected and the diversity of web services having similar functions. Service requests may include information on interfaces, special inner constraints, or temporal constraints on service behavioral properties. At the same time, according to the survey of web services in bioinformatics, capabilities of web services with similar functions may be diversified in inner constraints, invocation constraints and behavioral properties. For example, WS-ClustalW, an EBI web service, must use the specific tool, blast, to do a sequences similarity search, but such constraints are not exposed through service interfaces. Relations of input parameters and the value of some

Received 2007-05-18.

Foundation items: The National High Technology Research and Development Program of China (863 Program) (No. 2006AA12Z202), the National Natural Science Foundation of China (No. 90412010).

Biographies: Chen Wanghu (1973—), male, graduate; Han Yanbo (corresponding author), male, doctor, professor, yhan@ict.ac.cn.

special parameters can affect service capabilities. To do sequence search, some web services (e. g. , WSBlas) have only one atomic operation, but others (e. g. , WSWuBlas) may take interactions of several operations.

This paper proposes ASMA, an approach of service matchmaking adaptive to the variety of service requests and the diversity of web services. The approach is based on our abstract service model, ASM-TL, which addresses some important constraints for identifying capabilities of web services and also provides a virtual process model for describing service behavioral properties. By treating service requests as temporal logic conditions and web services as temporal models, ASMA does service matchmaking through model checking. Moreover, ASMA can map the information of service requests to the corresponding web service messages, and specify some invocation constraints. Therefore, the approach is suitable for the dynamic environment.

1 ASM-TL: Abstract Service Model

First, some necessary formal definitions, such as service Profile, Parameter, Inputs/Outputs, Operation and Constraints are given as follows.

Profile: $:= (\text{name}, \text{desc}, \text{cat}, \text{prov}, \text{exts})$, where elements, desc, cat, and prov correspond to service description, category and provider, and exts is an extensible parameter list.

Parameter: $:= (n, t, s)$, where n , t and s correspond to service name, type and semantics of the parameter.

Inputs: $:= \{p \mid p \text{ is a parameter}\}$.

Outputs: $:= \{p \mid p \text{ is a parameter}\}$.

Constraints: $:= P(x) \mid \neg P(x) \mid P(x, y, z, \dots) \mid P \Theta Q$. Here, $P(x)$ is a predicate; x , y and z are parameters with semantics; P and Q are constraints; Θ represents arithmetic operators such as conjunction, disjunction, and implicate.

Operation: $:= (\text{ins}, \text{outs}, \text{cons}, \text{coor}, \text{NFPs})$, where ins, outs, cons, and NFPs are corresponding to inputs, outputs, constraints and nonfunctional properties of the web service.

Especially, a coordinator denoted by coor is attached to each operation. The coordinator can be considered as a candidate of an operation interacting with other ones or clients, such as end-users and other services.

Definition 2 A coordinator can be denoted as coordinator: $:= (\text{peers}, \text{pre}, \text{post})$, where peers represent other coordinators interacting with the current one

when the operation it is bound to has been done, and pre/post corresponds to the precondition/post-condition when entering/leaving the current coordinator.

Definition 3 The abstract service model, ASM-TL, is a 3-tuple:

$WS: := (\text{profile}, \text{opers}, \text{NFPs})$, where profile corresponds to service profile, and opers is a set of service operations. NFPs means nonfunctional properties of a service.

It is worth noting that ASM-TL has a special virtual process model for describing behavioral properties, which is constructed dynamically.

Example 1 Fig. 2 illustrates the behaviors of the service WUBlast provided by EBI for sequence similarity search. The run operation submits a task. Then the CheckStatus operation checks the status of the task. Finally, if the status is true, the GetResults operation will return all the candidate results for the operation poll to pick the expected. The solid arrow denotes interaction relations, and the dashed arrow denotes data flows.

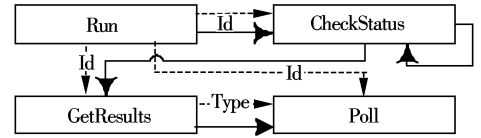


Fig. 2 Behavior of WUBlast

Coordinators are bound to each operation (see Fig. 3). In Fig. 3,

$C_0 = (\{C_1\}, \text{true}, \text{true})$

$C_1 = (\{C_1, C_2\}, \text{status} = \text{false}, \text{status} = \text{true})$

$C_2 = (\{C_3\}, \text{status} = \text{true}, \text{TRUE})$

$C_3 = (\text{null}, \text{TRUE}, \text{TRUE})$

where status reflects whether the operation is executed successfully; TRUE represents a special condition that always can be satisfied.

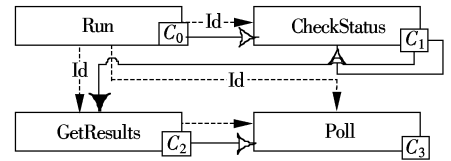


Fig. 3 Simplified concept model of WUBlast

Definition 4 The virtual process model of ASM-TL can be denoted as Process = (S, S_0, R, AP, L) , where S is a set of states; S_0 is a set of initial states; AP is a set of propositions and their negatives; L is a signing function. A state $s_i \in S$ represents once occurrence (virtual) of a service operation O_i . Given states s_i and s_j , $(s_i, s_j) \in R$ if and only if the coordinator C_i bound to O_i has at least one peer-coordinator C_j bound to O_j .

Example 2 The virtual process model for the web service, WUBlast, can be shown as follows, and to

be simplified, let s_i denote the state corresponding to the occurrence of an operation O_i .

$$S = \{s_0, s_1, s_2, s_3\}$$

$$S_0 = \{s_0\}$$

$$AP = \{\text{status} = \text{true}, \text{status} = \text{false}\}$$

$$R = \{(s_0, s_1), (s_1, s_1), (s_1, s_2), (s_2, s_3)\}$$

$$L(s_1) = (\text{status} = \text{false}) \wedge L(s_2) (\text{status} = \text{true})$$

The process model corresponds to a Kripke model^[3]. Algorithm 1 shows how to construct the virtual process model automatically.

Algorithm 1 Converting-to-process

Input: ws = (profile, ops, NFPs)

Return: Process = (S, S₀, R, AP, L)

Procedure:

ops = { O_i | $O_i \in \text{ws. ops}$ }

for each $O_i \in \text{ws. ops}$ do

$S = S \cup \{O_i\}$

$c = O_i.$ coor

peers = { c_i | $c_i \in c.$ peers}

for each $c_i \in c.$ peers do $R = R \cup (c, c_i)$

end for

for each $s \in S$ do

$L(s) = L(s) \cup s.$ pre

for each $s_i \in S$ do

if $(s, s_i) \in R$ then $L(s_i) = L(s_i) \cup s.$ post

end for

if not exists $(s_i \in S \text{ and } (s_i, s) \in R)$

then $S_0 = S_0 \cup \{s\}$

end for

return (S, S₀, R, AP, L)

End algorithm.

2 Description of Service Requests

Service requests represent conditions that services are expected to satisfy. Some examples of service requests can be ① Giving a genomic sequence, then returning the sequences similarity research reports, ② Doing sequence similarity search, then returning the results by e-mail, and ③ Giving two genomic sequences, then doing sequences alignment and returning the alignment reports by e-mail. Therefore, we treat service requests as some temporal properties services that are expected to be satisfied. Therefore, we describe service requests in a unified way as temporal logic conditions.

3 Service Matchmaking

3.1 Semantic matching for service interface

Let In_1 denote an input parameter of the service request and In_2 denote a parameter of the web service advertised, their matching level is denoted as L_{in} . Similar to L_{in} , L_{out} and L_{nfp} denote the matchmaking result of output and NFP. The matching level is decided accord-

ing to their semantic relations such as equal, subsume and containing etc. We will not discuss semantic matching approaches in detail in this paper. Then, the matching result of service inputs can be denoted as L_{ins}

$= \prod_{i=1}^n L_{in}^i$, and the matching results of outputs and nonfunctional properties denoted by L_{outs} and L_{nfp} can be assessed in the same way. Notice that in this paper the matchmaking is based on a consistent semantic foundation. In Ref. [3] we introduced an approach to obtain consistent ontologies.

Let Interface = (Inputs, Outputs, NFPs), and given two interfaces I_1 and I_2 , the matching level can be represented as $L = \alpha L_{ins} + \beta L_{outs} + \gamma L_{nfp}$, where $0 \leq L_{ins} \leq 1$, $0 \leq L_{outs} \leq 1$ and $0 \leq L_{nfp} \leq 1$ are matching levels of inputs, outputs and NFPs, α, β and γ denote the weights of inputs, outputs and NFPs, which can affect the similarities of service requests and advertised web services.

3.2 Service matchmaking algorithm

Let R denote service request, W denote a web service and $P = \{S, S_0, AP, R, L\}$ represents a process model. The service matchmaking algorithm is shown as follows.

Algorithm 2 Service-matchmaking

Input: R, W

$P = \text{Converting-to-process}(W)$

$S = \text{SAT_E}(R, P)$

if $S \supseteq P.$ S_0 then

Construct process segment matched.

End algorithm.

Given a request denoted by ϕ and a process model denoted by P , the extended SAT algorithm^[1], which determines states of P satisfying the conditions ϕ is illustrated in algorithm 3.

Algorithm 3 SAT_E(ϕ, P)

Case

$\phi = \text{true}$: return S

$\phi = \text{false}$: return null

ϕ is atomic: return $\{s \in S \mid \phi \vdash L(s)\}$

$\phi = \neg \phi_1$: return $S - \text{SAT_E}(\phi_1)$

$\phi = \phi_1 \wedge \phi_2$: return $\text{SAT_E}(\phi_1) \cap \text{SAT_E}(\phi_2)$

$\phi = \phi_1 \vee \phi_2$: return $\text{SAT_E}(\phi_1) \cup \text{SAT_E}(\phi_2)$

$\phi = \phi_1 \rightarrow \phi_2$: return $\text{SAT_E}(\neg \phi_1 \vee \phi_2)$

$\phi = AX\phi_1$: return $\text{SAT_E}(\neg EX\neg \phi_1)$

$\phi = EX\phi_1$: return $\text{SAT_E}_{EX}(\phi_1)$

$\phi = EX(\phi_1 \cup \phi_2)$: return $\text{SAT_E}_{EU}(\phi_1, \phi_2)$

$\phi = EF\phi_1$: return $\text{SAT_E}(\text{true } U \phi_1)$

$\phi = EG\phi_1$: return $\text{SAT_E}(\neg AF\neg \phi_1)$

End Case

End algorithm

In the algorithm, the interface matchmaking approach above is used to identify $\phi \vdash L(s)$. A, E, F, X, G and U are all CTL temporal connectives^[4]. SAT_

$E_{EX}(\phi_1)$ and $SAT_{-}E_{EU}(\phi_1, \phi_2)$ can be computed according to $SAT_{EX}(\phi_1)$ and $SAT_{EU}(\phi_1, \phi_2)$ in Ref. [4].

4 Application

ASMA has been applied to the PSE for bioinformatics research. Users represent personalized service requests with business services and web services are annotated by semantics. Fig. 4 shows the principles of ASMA in the PSE.

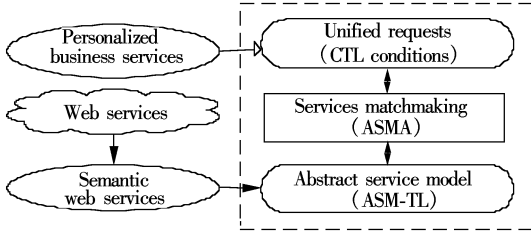


Fig. 4 Services matchmaking process

In the application, 47 web services provide by five authoritative organizations for sequence similarity search and alignment have been involved. The enacted business services have various capabilities, though they are all about service similarity search and alignment.

Fig. 5 shows a segment of service matchmaking results, in which B means business services, I corresponds to a path of a web service's process satisfying the request, and S means the operations comprised by the path. Fig. 6 shows such a path of service WUBlast corresponding to the business service, sequences searching-blast. The arrow represents interaction relations of operations, and the label on it represents data flows that were dynamically made during service

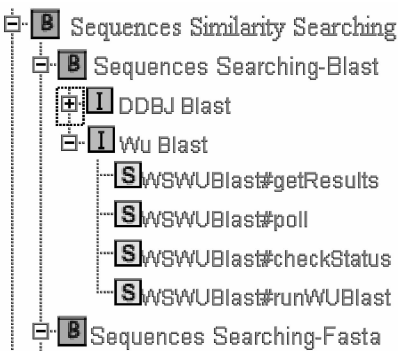


Fig. 5 Segment of service matchmaking

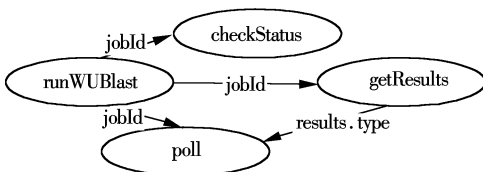


Fig. 6 Matched path of WUBlast

matchmaking. It is worth noting that, for some web services (e. g. , WSClusal), several paths satisfying the request may be found.

The application shows that the matchmaking approach is more accurate as well as more adaptive to the variety of service requests and the diversity of web services. It is effective in dynamic environments.

5 Analysis

Refs. [5 – 6] proposed approaches to assessing the similarity of two web services described in WSDL and DAML-S, respectively. In Ref. [1], an approach of service matchmaking based on description logic reasoning was introduced. Ref. [7] relied on to subsume relations of semantics of inputs/outputs to do service matchmaking. Approaches proposed in Refs. [1, 6 – 7] mainly focused on service interfaces matchmaking and cared little about behavioral properties of web services.

Ref. [8] enhanced assessing service similarities based on the semantics of process nodes' inputs and outputs. In Ref. [9], a service matchmaking approach based on PI-calculus extended with DL was proposed. In Ref. [10], the authors provided some feasible approaches and solutions for comparing two service capabilities based on an abstract finite-state machine. Approaches in Refs. [9 – 10] provided a novel measure to do matchmaking, but both focused on two process matchmaking. Compared to these approaches, ASMA treats service requests and web services as temporal logic conditions and temporal models, respectively. Thus, service matchmaking is done through model checking. Therefore, ASMA is more adaptive to the variety of service requests and the diversity of web services.

At the same time, specifications such as WSDL, WSDL-S^[11], and OWL-S are all used to describe web services. Ref. [12] proposed a description logic framework for describing web services, but did not concern behavior properties of services. Ref. [13] proposed a process model with no description of interfaces, messages transferring and special constraints. Ref. [14] proposed an abstract model based on FSM. The goal of ASM-TL is to enable adaptive service matchmaking. Therefore, it addresses some important factors to affect service matchmaking and provides a virtual process model, which can reflect the temporal properties of service behavior. Therefore, ASMA is more accurate in assessing service capabilities according to service request.

6 Conclusion and Future Work

The approach proposed in this paper is effective in

assessing service capabilities according to service requests in dynamic environments. Moreover, it is adaptive to various service requests and diverse web services. In the future, we will enhance the service matching approach by taking into account composition and aggregation based on ASM-TL.

References

- [1] Hull D, Zolin E, Bovykin A, et al. Deciding semantic matching of stateless services[C]//*Proc of the 21st National Conference on Artificial Intelligence*. California: AAAI Press, 2006: 16 – 20.
- [2] Han Yanbo, Wang Hongcui, Wang Jianwu, et al. An end-user-oriented approach to exploratory service composition[J]. *Journal of Computer Research and Development*, 2006, **43** (11): 1895 – 1903. (in Chinese)
- [3] Chen Wanghu, Liu Chen, Li Houfu, et al. Approach to dynamically forming semantic infrastructure for virtual organizations[J]. *Chinese Journal of Computers*, 2006, **29** (7): 1127 – 1136. (in Chinese)
- [4] Huth M, Ryan M. *Logic in computer science: modeling and reasoning about system* [M]. 2nd ed. England: Cambridge University Press, 2004: 207 – 243.
- [5] Wang Yiqiao, Stroulia E. Semantic structure matching for assessing web-service similarity [C]//*First International Conference on Service-Oriented Computing*. Trento, Italy, 2003: 194 – 207.
- [6] Wu Jian, Wu Zhaohui. Similarity-based web service matchmaking[C]//*Proceedings of IEEE International Conference on Services Computing (SCC'05)*. Los Alamitos: IEEE Computer Society, 2005: 287 – 294.
- [7] Paolucci M, Kawamura T, Payne T, et al. Semantic matching of web services capabilities[C]//*Proc of the First International Semantic Web Conference (ISWC 2002)*. Berlin: Springer-Verlag, 2002: 333 – 347.
- [8] Bansal S, Vidal J M. Matchmaking of web services based on the DAML-S service model[C]//*Proc of the Second International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'03)*. New York: ACM Press, 2003: 926 – 927.
- [9] Agarwal S, Aankolekar A. Automatic matchmaking of web services[C]//*Proc of the 15th International Conference on World Wide Web (WWW 2006)*. New York: ACM Press, 2006: 1057 – 1058.
- [10] Gao Xiang, Yang Jian, Papazoglou M P. The capability matching of web services[C]//*IEEE Fourth International Symposium on Multimedia Software Engineering (MSE'02)*. Los Alamitos: IEEE Computer Society, 2002: 56 – 63.
- [11] Patil A, Oundhakar S, Sheth A, et al. METEOR-S web service annotation framework [C]//*Proc of the World Wide Web Conference (WWW 2004)*. New York: ACM Press, 2004: 553 – 562.
- [12] Li Lei, Horrocks I. A software framework for matchmaking based on semantic web technology[C]//*Proc of the 12th International Conference on the World Wide Web (WWW 2003)*. New York: ACM Press, 2003: 331 – 339.
- [13] Berardi D, Calvanese D, Giacomo D, et al. Automatic composition of e-services that export their behavior[C]//*First International Conference on Service-Oriented Computing*. Trento, Italy, 2003: 43 – 58.
- [14] Berardi D, Calvanese D, Giacomo D, et al. Automatic composition of transition-based semantic web services with messaging[C]//*Proc of the 31st International Conference on Very Large Data Bases (VLDB 2005)*. New York: ACM Press, 2005: 613 – 624.

一种适应性 web 服务匹配方法

陈旺虎^{1,2,3} 韩燕波¹ 王 菁^{1,3} 刘 晨^{1,3} 王建武^{1,3}

(¹ 中国科学院计算技术研究所, 北京 100080)

(² 西北师范大学数学与信息科学学院, 兰州 730070)

(³ 中国科学院研究生院, 北京 100039)

摘要:为提高服务匹配对请求多样性和服务能力描述差异性的适应能力,提出了一种基于时态逻辑模型检验的适应性服务匹配方法——ASMA. 该方法基于一个抽象服务模型 ASM-TL, ASM-TL 包含了影响服务匹配的服务内部约束和调用约束等重要因素,以及一个描述服务行为属性的虚拟过程模型. ASMA 将服务请求统一描述为一种时态约束条件,基于 ASM-TL 将服务转换为一种时态模型,并引入模型检验的思想进行服务匹配. 因此,ASMA 可提高服务匹配的准确度,并很好地适应服务请求的多样性和服务能力描述的差异性. 该方法已应用到生物信息领域的问题求解环境(PSE)中. 应用表明,该方法对动态环境具有良好的适应能力.

关键词:服务匹配;服务模型;模型检验;时态逻辑

中图分类号:TP311