A Secure Process-Service Model

Shuiguang Deng, Zhaohui Wu, Zhen Yu, and Lican Huang

College of Computer Science, Zhejiang University, Hangzhou 310027, PRC {dengsg, wzh, yz,lchuang}@zju.edu.cn

Abstract. Encapsulating processes into process-services is a hot topic nowadays. Time management is an important issue for service providers to ensure the successful execution of process-services, and time information is also concerned by process-service consumers. Due to the security and secrecy factors in businesses, service providers are not willing to publish all information in process-services out. Thus process-services present as black boxes with only interfaces to consumers. As a result it is hard for consumers to engage in time management. We propose a secure process-service model, in which a processservice is divided into a public part and a private part.

1 Introduction

E-services have been announced as the next wave of internet-based business application that will dramatically change the use of the Internet [1]. The emergence of technologies and standards supporting the development of web services has unleashed a wave of opportunities for enterprises to form alliance by encapsulating processes into services and composing different services [2]. We give those services, which focus on processes, a name "process-services". Due to the security and secrecy factors in businesses, service providers tend to hide the details of the process from service consumers. Thus process-services present as black boxes with only interfaces to consumers. But on the contrary, process-service consumers want to know that information in order to use process-services well. How to deal with the contradiction between process-service providers and consumers is the focus of this paper. In our opinion, a process-service can be divided into two parts: a process-service body and a processservice declaration. The former is private to providers and contains all the details of the process information including its structure and time constraints. The latter, published to consumers, is abstracted from the former and contains some necessary information about the process in the process-service for consumers.

2 A Process-Service Model

Process is the center focus of a process-service, which achieves a special target through accomplishing serials of activities, between which there are structure and time constraints. In this section, we first introduce some basic elements and time constraints in process-service, and then present a process-service model with two parts: process-service body and process-service declaration.

2.1 Basic Elements and Time Constraints in Process-Service

Definition 1 (Activity). An activity is defined as a 2-tuple *<id, duration>* where *id* is the identity of the activity, *duration* is the execution time of the activity.

Definition 2 (Dependency). A dependency is defined as a 2-tuple, *<prev, succ>* which means activity *succ* must be executed after activity *prev*.

Definition 3 (Lower Time Constraint). A lower time constraint *LConstraint* is defined as a 5-tuple, $\langle src, P1, des, P2, limitation \rangle$, where *src* and *des* are activities, *P1* and *P2* are from the set $\{b, e\}$, *b* represents the beginning time of the activity and *e* represents the end time of the activity, *limitation* represents a period of time.

A *LConstraint* means that the distance between the beginning time (or end time) of the activity *src* and the beginning time (or end time) of the activity *des* is greater than *limitation* time units.

Definition 4 (Upper Time Constraint). A upper time constraint *UConstraint* is defined as a 5-tuple, *<src, P1, des, P2, limitation>* which means that the *distance* between the beginning time (or end time) of the activity *src* and the beginning time (or end time) of the activity *des* is smaller than *limitation* time units.

2.2 Process-Service Body and Process-Service Declaration

In order to make process-service not only satisfy the security and secrecy requirements, and also provide enough structure and time information for consumers, we deem a process-service should have two parts. One is a process-service body, and the other is a process-service declaration. The former, containing the detail information about the realization of the process-service, is private to the service provider; and the latter, abstracted from the former according to a time equivalence principle, is public to service consumers.

Definition 5 (*Process-Service Body or PSB*). A process-service body is defined as a 5-tuple, <*ActSet, DepSet, In, Out, Constraints*>, where *ActSet* is a set of activities, *DepSet* is a set of dependencies, *In/Out* is a set of identities of the input/output activities which are all called interface activities. *Constraints* is a set of time constraints.

Definition 6 (Activity Time Assignment). For a *PSB* and *time:Identity*×{*b,e*} \rightarrow *R*, if the following conditions are satisfied, the map time is called an activity time assignment of the process-service body.

- i. $\forall activity \in Service.ActSet$, time(activity.id.b) + activity.duration = time(activity.id,e)
- ii. ∀dependency∈ Service.DepSet , time(dependency.prev.id,e)≤time(dependency.succ.id,b)
- iii. $\forall LConstraint \in Service. Constraint s$ time(LConstraint.des.id, LConstraint.P2) $-time(LConstraint.src.id, LConstraint.P1) \ge LConstraint.dis \tan ce$
- iv. $\forall UConstraint \in Service. Constraint s$ time(UConstraint.des.id, UConstraint.P2) $-time(UConstraint.src.id, UConstraint.P1) \leq UConstraint.dis \tan ce$

Definition 7 (Interface Time Equivalent Principle) If two PSBs service1 and service2 fulfill the following conditions, they are interface time equivalent.

- i. Service1.In = Service2.In; Service1.Out = Service2.Out
- ii. For any activity time assignment of Service1, time1, there exists an activity time assignment of Service2, time2, and they fulfill the conditions:
 (∀id ∈ Service1.In)(time1(id,b) = time2(id,b)), (∀id ∈ Service1.Out)(time1(id,e) = time2(id,e))
- iii. For any activity time assignment of Service2, time2, there exists an activity time assignment of Service1, time1. They fulfill the conditions:
 (∀id ∈ Service2.In)(time1(id,b) = time2(id,b)), (∀id ∈ Service2.Out)(time1(id,e) = time2(id,e))

Definition 8 (*Process-Service Declaration or PSD*) For a PSB, its PSD is a simple PSB that is interface time equivalent with the original PSB. This is expressed as *PSD=Declare (PSB)*. PSD is abstracted from its relative PSB and is open to process-service consumers and have some simple but essential structure and time information.

3 Algorithms to Automatically Generate PSD from a PSB

This section illustrates an example to automatically generate a corresponding PSD from a PSB shown in the left of fig. 1 using the algorithm introduced in [3]. The algorithm is $O(n^3)$ time complexity.



Fig. 1. An example of process-service body

A comparison between the PSD and the corresponding PSB in the above fig. 1 shows that a PSD is deeply abstracted from its relative PSB and hides much detail information from consumers. If open the PSD to consumers instead of a black box with only interfaces, consumers know time constraints information between interface activities.

4 Conclusion

More and more enterprises pay attention to adopt web services to encapsulate processes. But web service model only emphasize on the publication of input/output interfaces and ignore processes in services. As a result, a process-service presents like a black box with only interfaces to consumers. It prevents consumers from learning more information about services. DAML-S [4] is another alternative model to describe process-service. But time information of processes in services is not included. Moreover it exposes the whole processes to public. However business enterprises are not willing to do that due to the consideration on security and secrecy factors. There are the same problems in WSFL [5], XLANG [6] and BPEL4WS [7]. We divide a process-service into two parts: a process-service body and a process-service declaration. This model not only satisfies the security and secrecy requirements from providers, but also provides enough information for consumers to engage in time management.

Acknowledgement. This work is supported by the National High Technology Development 863 Program of China under Grant No.2001AA414320 and No.2001AA113142; the Key Research Program of Zhejiang province under Grant No. 2003C21013.

References

- 1. Fabio Casati, M. C. Shan, et al. E-Service –Guest editorial. The VLDB Journal 10(1):1
- 2. Fabio Casati, M. C. Shan. Dynamic and adaptive composition of e-services. Information system 26, 3, 143-162.
- Shuiguang Deng, Zhen Yu, Zhaohui Wu. Research of Time Model in Service-Based Workflow. To be appeared in Journal of Zhejiang University (ENGINEERING SCIENCE) 2004.
- 4. Ankolekar, M.Burstein, et al. DAML-S: Web Service Description for the Semantic Web. First International Semantic Web Conference, June, 2002
- Frank Leymann. Web Services Flow Language. http://www-4.ibm.com/software/solutions/ webservice/pdf/ WSFL.pdf, May 2001
- Satish Thatte. XLANG: Web Services for Business Process Design. Microsoft Corporation 2001
- BEA Systems, IBM, Microsoft, SAP AG and Siebel Systems, Business Process Execution Language for Web Services, May 2003