A Tool to Synthesize Intelligible State Machine Models from Choreography using Petri Nets*

Toshiyuki Miyamoto[1] and Hiroyuki Oimura[1]

Graduate School of Engineering, Osaka University,
Suita, Osaka 565-0871, Japan
miyamoto@eei.eng.osaka-u.ac.jp

Abstract. Application of service-oriented architecture, which builds the entire system by a combination of independent software components, to a wide variety of computer systems is expected. The problem to synthesize state machine models of the services from a communication diagram representing the overall specifications of service interaction is known as the choreography realization problem. It should be minded on automatic synthesis that software models should be simple to be understood easily by software engineers. We have proposed a method to synthesize hierarchical state machine models for the choreography realization problem in the last PNSE. In this paper, we present a prototypical tool for the method.

In recent years, the internationalization of activities and information technology in the enterprise has intensified competition between companies. Under such circumstances, service-oriented architecture (SOA)[6] has been attracting attention as the architecture of information systems in the enterprise. In SOA, an information system is built by composing independent software units called services.

In SOA, the problem to synthesize the concrete model from an abstract specification is known as the choreography realization problem[5]. In which the abstract specification, called *choreography*, is defined as a set of interactions among services, which are given in a dependency relation of messages sent and received; the concrete model is called the *service implementation* which defines the behavior of the service. This paper utilizes the communication diagram and the state machine of UML 2.x[4] to describe the choreography and the service implementation, respectively.

Bultan and Fu formally introduced the choreography realization problem in [2]. They used collaboration diagrams of UML1.x and showed some conditions for a given choreography to be realizable. In addition, they showed a method to represent the service implementation as the state space in which a state was defined as a set of unsent messages, and they also showed a method to map to a set of finite state machines. However, it is not intelligible because the number of states increases exponentially as the number of messages increases.

* This work was supported by KAKENHI (23500045).

Table 1. Number of simple states(NSS), transitions(NT), and guards(NG) of synthesized state machines.

ex	service	message	projection			CSCB		
			NSS	NT	NG	NSS	NT	NG
1	4	10	32	52	0	11	32	4
2	3	4	7	10	0	7	10	0
3	4	6	15	23	0	8	15	0
4	4	7	17	25	0	7	20	0
5	5	7	15	22	0	10	18	0
6	6	12	48	88	0	16	36	1
7	5	10	23	32	0	16	27	4

Antonio et al. have experimentally evaluated the relationship between metrics and intelligibility of the state machines by measuring time to understand state machines[1]. According to the result, state machines are intelligible the smaller the following metrics: the number of simple states (NSS), the number of transitions (NT), and the number of guards (NG).

In [3], we have proposed a method to synthesize hierarchical state machines by using Petri nets from a choreography defined by single communication diagram, where the method is called the CSCB method. So far, we have developed a prototypical tool of the CSCB method and the projection method in [2], and evaluated the CSCB method on several examples. Table 1 shows the results, and the CSCB method is better than the projection method in term of several metrics for the intelligibility by Antonio et al. We, however, found some points to be improved in the algorithm and the intelligibility.

We are going to import the prototypical tool into an UML modeling tool as a plug-in and extend the CSCB method so as to synthesize more intelligible machines.

References

1. Antonio Cruz-Lemus, J., Genero, M., Piattini, M.: Metrics for UML Statechart Diagrams. In: Genero, M., Piattini, M., Calero, C. (eds.) Metrics for Software Conceptual Models, pp. 237–272. Imperial College Press, London (2005)
2. Bultan, T., Fu, X.: Specification of realizable service conversations using collaboration diagrams. Service Oriented Computing and Applications 2(1), 27–39 (2008)
3. Miyamoto, T., Hasegawa, Y.: A Petri Net Approach to Synthesize Intelligible State Machine Models from Choreography. In: International Workshop on Petri Nets and Software Engineering 2012. pp. 222–236 (Jun 2012)
4. OMG: Unified modeling language, http://www.uml.org/
5. Su, J., Bultan, T., Fu, X., Zhao, X.: Towards a theory of web service choreographies. In: Proceedings of the 4th international conference on Web services and formal methods. pp. 1–16 (2008)
6. Thomas, E.: Service-Oriented Architecture. Prentice Hall (2004)