Journal of Southeast University (English Edition) Vol. 24, No. 3, pp. 289 – 292 Sept. 2008 ISSN 1003—7985

MAS-based dynamic web service composition formal model

Xu Donghong Qi Yong Hou Di Shen Linfeng Du Xiaozhi Wang Gongzhen

(School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China)

Abstract: Applying dynamic web services composition is one of the important schemas for solving service-oriented architecture (SOA) and service-oriented computing (SOC). For implementing dynamic web services composition, the multi agent system (MAS) is applied to web services composition. First, the essentials of the MAS and web services composition are analyzed and their relationship is discussed. Secondly, an MAS-based architecture is designed for dynamic web services composition, and it is named as CSMWC. A Jade tool is used to implement the major components. The architecture can primarily implement syntactic level dynamic web services composition, and it gives a basis for semantic level dynamic web services composition. For specifying the correctness of the architecture, by using pi-calculus, the architecture of the MAS is formally described, and its dynamic properties and adaptability are reasoned. Finally, it demonstrates the idea proposed by the Pi4SOA tool.

Key words: dynamic web service composition; MAS (multi agent system); pi-calculus; process calculus; Pi4SOA

Web services composition is becoming the major methodological and technological solution for service-oriented computing (SOC). It can provide mechanisms for describing and composing services. Composing multiple web services can present uniform proprietary interfaces and data formats with standard web-message infrastructures based on XML technology. The method can be applied to enterprises and make them easily and rapidly establish the relation to accomplish common tasks. Meanwhile, it can provide more benefits for users. Currently, several ongoing technologies and protocols provide standards and ways to effectively compose web services distributed across different domains. These standards and protocols respectively support local services definition, advertisement, registration and global interaction and coordination. Although these standards are extremely valuable in highlighting the key innovative features of the SOC approach and some standards have worked their way into vendor products, several limiting factors of the services protocols have been identified from experience. The composition standard BPEL4WS can only specify the workflow and has no mechanism to verify the correctness of the composition. CDL provides the specifications of composition, but it cannot guarantee the dynamic attributes for the implementation of the composition. In addition, both ways are based on an assumption that there are enough services to be used and composed. However, the users' requirements and service contexts change continuously. Meanwhile, new services are deployed and old services are removed at run-time. The design-time composition mechanism does not meet the dynamic requirements. It is better to use a multi-agent system to develop towards an automation services composition.

A web services composition has two basic aspects: orchestration and choreography[1-2]. Orchestration refers to an executable business process that can interact with both internal and external web services. However, in this schema, the services can only perform static and passive behaviors. In an open real-time environment, service providers should enact, maintain and quit services according to requirements. So, this schema cannot meet the requirement of a dynamic and agile composition. Choreography is more collaborative and allows each of the involved services to describe its part in the interaction. Choreography tracks the services message sequence among multiple parties sources and guarantees overall composition. During the interaction procedure, some new problems such as the temporality of services emergence, services selection and services missing can emerge. When they occur, engineers will have to spend more time and attention on service change, which will increase management costs.

The MAS has become a hot research topic regarding automation services composition[3-5]. The MAS is concerned with utilizing multiple agents to accomplish common tasks. In the MAS, a single agent cannot comprehend all of the information, and it cannot have all the resources. An agent must interact with the others in order to support each other in abilities and resources. The major application of the MAS is to let services compositions have more dynamic and active behaviors to satisfy customer requirements. According to the conformance between the requirements of composition and the functions of the MAS, the combination of the MAS and the composition techniques can help engineers and customers make decisions, alleviate the burden of the composition knowledge and a minimum of development intervention and increase the composition among the different types of services. To develop the MAS-based composition, we design a novel MAS structure called CSMWC (collaborative structures of the MAS for web services composition). We use Jade to implement the framework, and employ pi-calculus[6-8] to describe and reason the behavior of the framework. Applying a formal method to analyze the roles and interaction behavior in CSMWC can not only grasp the interaction traits, but also provide the reliable agent interaction mechanism in designing and developing the MAS. Moreover, applying the formal method can guarantee the proper interaction, cooperation and communication in implementing proper composition goals. We use Pi4SOA[9] to verify some properties of the composition.

Received 2008-04-15.

Biographies: Xu Donghong (1978—), male, graduate; Qi Yong (corresponding author), male, doctor, professor, qiy@mail.xjtu.edu.cn.

Foundation items: The National Natural Science Foundation of China (No. 90612014), IBM China Research Lab Joint Project.

Citation: Xu Donghong, Qi Yong, Hou Di, et al. MAS-based dynamic web service composition formal model[J]. Journal of Southeast University (English Edition), 2008, 24(3): 289 – 292.

1 The CSMWC and the Formal Analysis

The aim of designing the CSMWS is to enable the composition to be easily reconstructed, to support distributed compositions, and to facilitate customers to interact with each other. The role type of each agent is important for applying the CSMWS. Although different agents can implement the common tasks by collaboration, the status of each agent is not peer due to their different abilities and responsibilities in CSMWS. In order to increase the applicability of agents, we adopt a self-adaptive two-layer structure: one layer is an agent stub and the other is an agent. We will separately and formally describe the agent stub and the protocol in this structure.

Definition 1 Agentstub $\triangleq (\mathrm{vx})\,\overline{\mathrm{SA}}\langle \mathrm{Req}\rangle.\,\mathrm{AS}(\mathrm{Abil, Respon, Acc, RoleModel, RoleRelation}).\,\tau \mid \overline{P}$.

The interaction protocol denotes interaction procedure. It decides which type of agent can attend the interaction, adopt which rule, transits the message, and fulfills the responsibility by collaboration. The formal description of the protocol is as follows:

Definition 2 Proto $\triangleq (\mathrm{vid})\,\overline{A_iA_j}\langle \mathrm{Interid, Purpo, Act, Commit}_i\rangle.\,[\mathrm{Commit}_i, \mathrm{Purpose}_i = \mathrm{Commit}_j, \mathrm{Purpose}_j].\,\overline{A_jA_i}\langle \mathrm{Interid, Accept}\rangle \mid \mathrm{Proto}$.

The relationships among agent, agent stub and protocols are illustrated in Fig. 1.

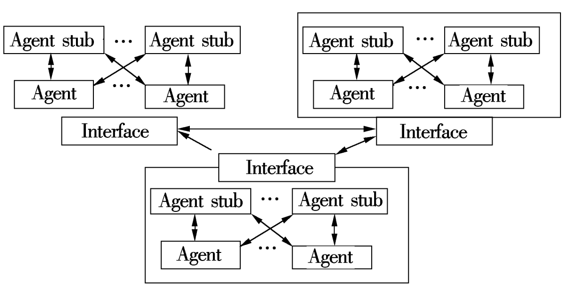

Fig. 1 The relationship of agent, agent stub and protocol

In order to answer the questions and challenges mentioned, we introduce an MAS-based framework for dynamic web services composition, which is the collaborative structure of MAS for web services composition. In the interactive structure, we adopt three types of agents: user service agent, composite service agent and services agent. The framework is illustrated in Fig. 2. We analyze and formally describe our designed framework as follows.

1.1 User service agent (USA)

USAs interact with users, and provide special application domains and some smart GUI. In the MAS, a USA sends service request information to an SA to accomplish certain goals. Some of these goals come from the requirements themselves and some are from the contexts. The behavior processes of the USA are formally described by pi-calculus as follows:

Definition 3 $\mathrm{CQS}(\mathrm{req, resp}) \triangleq (s)\,\overline{\mathrm{qc}}\langle\{\}\rangle.\,\mathrm{rc}(s)\,\mathrm{Req}A(s) \mid \mathrm{DBMulticast}$.

The formal description of process Req$A$ in definition 3 is as follows:

Definition 4 $\mathrm{Req}A(\mathrm{req\text{-}agt\text{-}resp, req, reqcofn}) \triangleq (\mathrm{vx})\,\overline{\mathrm{rc}}\langle w, \mathrm{fltinfMp}, r\rangle.\,w(x, \mathrm{reqconf}).\,\mathrm{req\text{-}agt\text{-}resp}\langle \mathrm{true}\rangle$.

The implementation is described in Fig. 3.

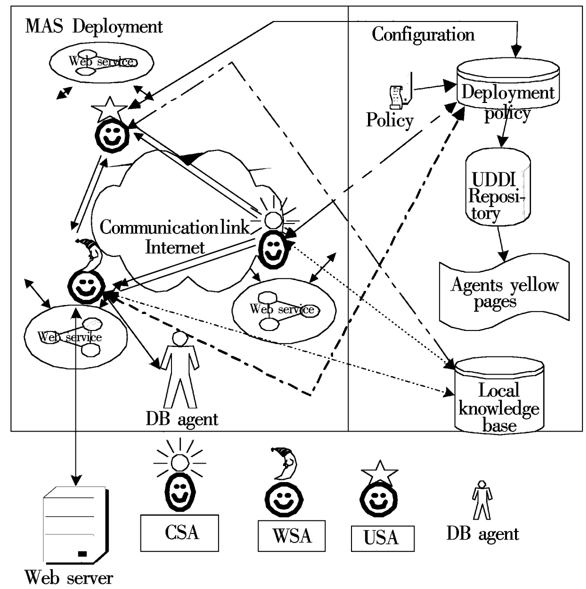

Fig. 2 MAS deployment and configuration

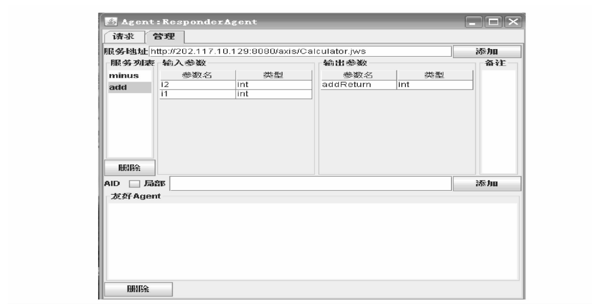

Fig. 3 The implementing structure of USA

1.2 Web services agent (WSA)

The role of the WSA is to offer web services to other services. However, not only WSAs do not actively want to know which type of agent that needs their services, but also it is unnecessary for WSAs to care to whom they will provide the services. Only when there is requested information, will the USA respond. In many scenarios, there are no direct relationships between the WSA and the USA; the requested information is transported by the CSA. If it wants other agents to query and search them, they should apply UDDI to the register and announce their service concepts and the service types which they can provide. The register services (RS), declare services (DS), and providing services agent (PSA) are respectively defined as follows:

Definition 5 $\mathrm{RS} \triangleq\, !(s)\,\overline{\mathrm{Register}C}\langle \mathrm{id}\rangle \ldots S$.

Definition 6 $\mathrm{DS}(r, s) \triangleq (\mathrm{vid})\,[s = \mathrm{nil}]\,\overline{\mathrm{id}}\,|\,(\mathrm{id}.\,o + !(\mathrm{vp})\,\overline{\mathrm{service}}\langle p\rangle.\,\overline{p}\langle r\rangle.\,\overline{r}\langle s\rangle$.

Definition 7 $\mathrm{PSA} \triangleq S_{\mathrm{id}}(x, y).\,\overline{\mathrm{XC}}\langle s\rangle$.

The implementation is shown in Fig. 4.

1.3 Composite service agent (CSA)

The role of the CSA in the MAS is the connector. The

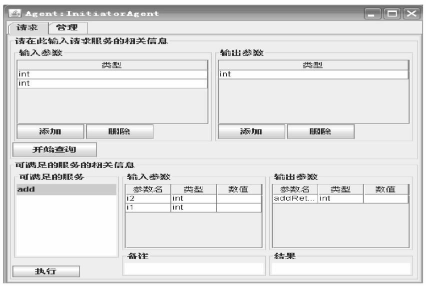

Fig. 4 The implementing structure of services provider agent

CSA is responsible for monitoring and managing deployment. It ensures that the appropriate WSAs can be involved and collaborate according to specific requirements. Ontological information written in OWL is converted to RDF triples and loaded to the KB. The ontology (Onto) and the register service (CSR) are respectively defined in definitions 8 and 9.

Definition 8 $\mathrm{Onto}(c) \triangleq (\mathrm{vc})\,\mathrm{sen}(\mathrm{tok}, \mathrm{val}, \mathrm{ran}).\,\mathrm{Sen}(c)$.

Definition 9 $\mathrm{CRS} \triangleq \mathrm{Re}g(x).\,\overline{\mathrm{XC}}\langle y \rangle.\,S$.

The framework is implemented by using Jade[10] and Hopitool. Jade is a development environment for the MAS. Its structure is shown in Fig. 5.

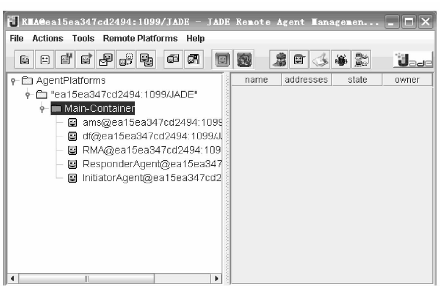

Fig. 5 The development environment of Jade

Hopitool can exchange XML format into Jade code. The implementation procedure is illustrated in Fig. 6. Due to the limitation of space, we omit the explicit designing procedure.

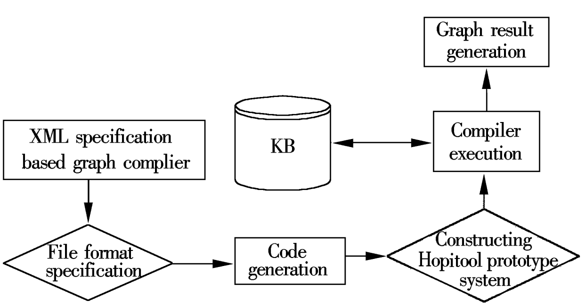

Fig. 6 From pi-calculus specification to implementation

2 Tool for verifying the framework

Pi4SOA is the foundation that is used to establish communication and collaboration between industrial and academic organizations which are interested in the application of process algebra and other concurrency formalisms to software development for web services and other distributed architectures. It starts from the theory-engineering collaboration within W3C's WS-CDL working group based on pi-calculus. Building complex service composition without introducing unintended consequences is a real challenge. The tool helps to reduce such complexity. It does not necessarily guarantee that erroneous situations cannot occur due to inappropriately specified interactions.

3 Conclusion and Future Work

Web services composition is an emerging and promising area, which involves important technological challenges. Some of the main challenges are to correctly describe web services, to adequately compose them, and to discover suitable web services. In this paper, we present a novel MAS-based framework for dynamic services composition, and adopt simple pi-calculus to describe and reason the procedure of composition behavior. We advocate the application of pi-calculus as an abstract representation and mean to describe, compose and reason the composition system[11-13]. We especially illustrate our approach for the orchestration of dynamic adaptation behavior and choreography interaction behavior.

We intend to find a mechanism for automatic composition of web services in our future work. We also plan to examine the support for exception handling during the composition process. Furthermore, we will work on the formalization of connections between abstract and concrete descriptions. Some ideas in this direction will be further explored by experiment studies.

References

[1] Blake M Brian. Forming agents for business process orchestration[C]//*Proc of the 37th Hawaii International Conference on System Sciences*. Washington, DC: IEEE Computer Society, 2004: 202 - 211.

[2] Xu Donghong, Qi Yong, Hou Di, et al. SpiG4WSC: a calculus for secure services composition[C]//*Proc of* 2008 *IEEE International Conference on Services Computing*. Washington, DC: IEEE Computer Society, 2008, to appear.

[3] Maamar Zakaria, Mostefaoui Soraya Kouadri, Yahyaoui Hamdi. Toward an agent-based and context-oriented approach for web services composition [J]. *IEEE Transactions on Knowledge and Date Engineering*, 2005, **17**(5): 686 - 697.

[4] Zhao Xinpei, Li Mingshu, Wang Qing, et al. An agent-based self-adaptive software process model [J]. *Journal of Software*, 2004, **15**(3): 348 - 359. (in Chinese)

[5] Ishikawa F, Yoshioka N, Tahara Y, et al. Mobile agent system for web services integration in pervasive networks[C]//*Proc of the First International Workshop on Ubiquitous Computing*. Porto: INSTICC Press, 2004: 38 - 47.

[6] Overdick Hagen, Puhlmann Frank, Weske Mathias. Towards a formal model for agile service discovery and integration [C]//*Proc of the ICSOC Workshop on Dynamic Web Processes*. Washington, DC: IEEE Computer Society, 2005: 75 - 83.

[7] Liao Shouyi, Lu Hongwei, Chen Jian, et al. Research on conceptual framework for agent-based modeling and simulation [J]. *Journal of System Simulation*, 2006, **1**(18): 616 - 620. (in Chinese)

[8] Xu Wei, Jin Beihong, Li Jing. A mobile agent-based fault-tolerant model for composite web service[J]. *Chinese Journal of Computers*, 2005, **128**(4): 558 - 567. (in Chinese)

[9] Ross-Talbot Steve. Pi4 techniques foundationhttp[EB/OL]. (2004-05-12)[2008-04-01]. http://pi4soa. sourceforge. net.

[10] Bellifemine Fabio, Caire Giovanni, Rimassa Giovanni, et al.

Java agent development framework: an open source platform for peer-to-peer based applications[EB/OL]. (2003-09-18) [2008-04-01]. http://jade.tilab.com/.

[11] Guidi Claudio, Lucchi Roberto, Mazzara Manuel. A formal framework for web services coordination[J]. *Electronic Notes in Theoretical Computer Science*, 2007, **180**(2): 55－70.

[12] Lucchi Roberto, Mazzara Manuel. A pi-calculus based semantics for WS-BPEL[J]. *Journal of Logic and Algebraic Programming*, 2007, **70**(1): 96－118.

[13] Wirsing Martin, De Nicola Rocco, Gilmore Stephen, et al. SENSORIA process calculi for service-oriented computing [C]//*Lecture Notes in Computer Science*. Springer-Verlag, 2007, **4661**: 30－50.

基于 MAS 系统的动态 web 服务组合形式化建模研究

徐东红　齐　勇　候　迪　沈林峰　杜小智　王功震

(西安交通大学电子与信息工程学院,西安 710049)

摘要:动态服务组合是解决 SOA 和 SOC 的重要方案之一. 针对动态服务组合的实现,把 MAS 技术应用到服务组合中. 首先对 MAS 和 web 服务组合的本质进行分析,讨论了 MAS 和 web 服务组合的关系. 其次,构建了基于 MAS 的 web 服务组合框架,命名为 CSMWC,并用 Jade 实现了该框架. 此框架初步实现了语法级别的动态服务组合,并为语义级别的动态服务组合打下基础. 为说明构建框架的正确性,应用 pi 演算对构建的 MAS 体系结构进行形式化描述,同时对 web 服务组合中重要的性质如动态性、适应性进行了推理. 最后给出了应用基于 pi 演算的验证工具 Pi4SOA 进行验证的过程.

关键词:动态 web 服务组合;MAS;pi 演算;进程代数;Pi4SOA

中图分类号:TP301