Journal of Theoretical and Applied Information Technology
20th February 2013. Vol. 48 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

SIMULATION SERVICE MANAGEMENT BUS IN SERVICE ORIENTED RUN-TIME INFRASTRUCTURE

ZHITENG WANG, HONGJUN ZHANG, RUI ZHANG, LUYUAN ZHANG, BAOYU XU
PLA University of Science &Technology, Nanjing 210007, China

ABSTRACT

In this paper, Simulation Service Management Bus (SSMB) is proposed to intergrade and reuse the simulation service resource in Service Oriented Run-time Infrastructure (SORTI). Its functions include integration service, load balancing, communication security, and so on. Its main functions and the whole frame are elaborated. key technologies which include service registry and record, integration service, load balancing, majorization of service composition, thread evictor, communication encryption are elaborated respectively.

Keywords: *Simulation Service Management Bus, Simulation Service, Integration Service, Load Balancing, Communication Security*

1. INTRODUCTION

As the SOA (Service-Oriented Architecture) is paid close attention by many researchers in recent years [1-7]. It is a component model which can be defined as service according to business needs and can be taken as reusable shared resource. It also provides the design method for application software development and has become a standard for design, development, application and management distributed service unit. It needs a basal architecture to flexibly integrate and link heterogeneous service. EAI (Enterprise Application Integration) [8] want to intergrade heterogeneous system to achieve interactive, share data and cooperate in isolated system. However, it ignored the process design and integration for enterprise leading to many problems such as a lack of flexibility, insufficient openness, unreasonable granularity rules, excessively coupling with original system architecture and so on. It has so many problems that it can't support SOA.

Service Bus is a practical formula for SOA. It is a flexible basal architecture to intergrade the application and service [9][10][11] and it combine with the method event-driven and oriented service to provide service management method and the inactive service function in distributed heterogeneous environment. It is able to eliminate differences in different technologies and coordinate different services to communicate with each other. ESB (Enterprise Service Bus) has been paid specific attention by more and more companies and study research communities. Many famous IT enterprises act as a participator in this study such as IBM, BEA, Microsoft, Oracle and so on. Some companies have developed independent and faultless ESB such as IBM's websphere, Microsoft's Inigo platform, and BEA's Aqualogic Service Bus and so on. In practice, many problems such as load balancing, reliability, asynchronous communication, Service decoupling and so on have been met. Hence, many researchers start the research of improving the whole performance of SOA by taking improvement in ESB [12][13] [14] [15][16][17]. However, there is little research about how to adapt the ideal of ESB and SOA to apply them in SORTI.

SSMB is an import part in the SORTI, which support the synchronous or asynchronous call the simulation services by multiple protocols. It also manages the distributed simulation services. The core function of the SSMB includes the service registry and record, integration service, load balancing, majorization of service composition, thread evictor, communication encryption and so on.

2. AN OVERVIEW OF S SORTI

The whole architecture of SORTI can be shown in the figure 1.

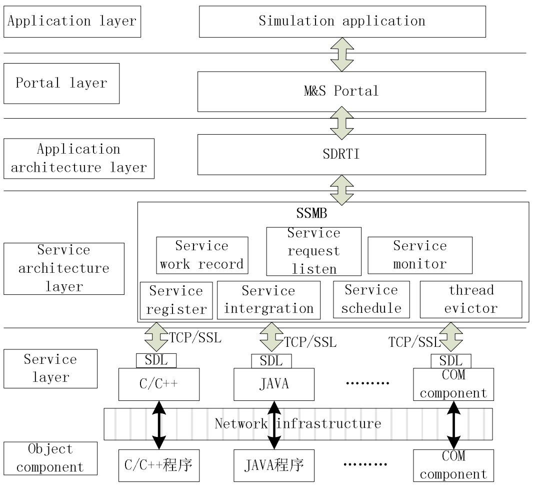


Fig.1.The Whole Architecture Of SORTI

It can be seen from the picture that the DSRTI have five layers: application layer, portal layer, application architecture layer, service architecture layer, service layer. Application layer is to solve the problem of such as SOA's application mode, simulation service visit, and interactive interface. Portal layer's users include service developers, application developers, service consumer and system administrator. Service developers register the developed service to the system though the portal. Application developers can query registered service from portal. Service consumers can directly call services though portal. System administrator can manage service though portal. Application architecture layer's core is to build the standard of service-oriented simulation application and supply infrastructure to support simulation development. Service architecture layer called SSMB is able to manage simulation service. Its main function includes service register, service integration, majorization of service composition and so on.

3. KEY TECHNOLOGIES IN SSMB

In SSMB, key technologies include service record register and record, service integration, majorization of service composition, thread evictor, load balancing, encrypted communications. They will be elaborated described in the following.

3.1 Simulation Registration And Work Record

Three parts, which include simulation consumers, simulation registry, simulation provider, play import role in the field of M&S. Service provider publish simulation service to service registry with Service Description Language (SDL). Service requester publish request to service registry. The function of simulation service registry is achieved by SSMB which can query required service in simulation service list of the service registry when it find required service it will call the simulation service by address, port and protocol in related description in SDL and return executive result to service requester. It can be shown in the figure 2. This mode can make sure that simulation consumer doesn't need to communicate with simulation service publisher directly but to communicate by SSMB. It decreases the coupling degree between consumer and publisher.

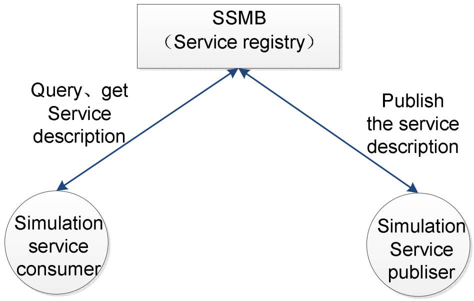
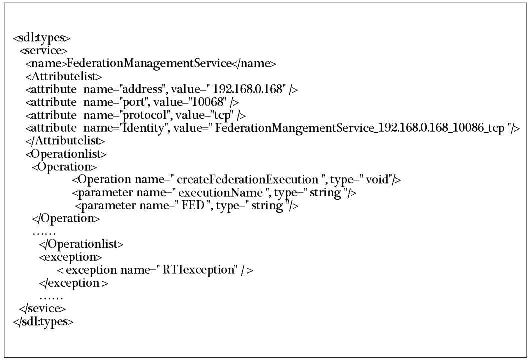

SSMB
（Service registry）
Query、get Service description
Publish the service description
Simulation service consumer
Simulation Service publiser

Fig.2. Simulation Service Invocation Ins SORTI

In fact, the definition of service in the service-oriented not only include web service but include many other components like EJB, JMS, JMS, JavaBean, COM/DCOM、CORBA、ICE, all of this component can be call service[18]. Service in M&S is also generalized concept, which includes Web Service and other the encapsulation of services. Web Services Description Language (WSDL) is special for Web service. In order to describe generalized service in a general formal, we use Service Description Language (SDL) which is simple and readable. For example, creating federation Execution in Federation Management service is can be described with SDL as follow code. From the code, it can be seen that many important elements like service name, service identify, operation name, parameter name, address, port, protocol is described in SDL. The attribute list of service include address, port, protocol, identify. The identify is the combination of service name, address, port and protocol with underline, which is important element to distinguish different service and is able to quickly locate the service with this element. SSMB can record the work effect of the service with unique identify of different service. Credit rating of service with different level is evaluated by the efficient of service such as the probability of completing task, service reliability,

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

delaying time and so on, which is important consider factor for service schedule. In service registry, different services are sorted with credit rating by service identify in a queue. The related SDL can be called by service identify where need.



```
<sdl:types>
  <service>
    <name>FederationManagementService</name>
    <Attributelist>
    <attribute  name="address", value=" 192.168.0.168" />
    <attribute  name="port", value="10068" />
    <attribute  name="protocol", value="tcp" />
    <attribute  name="Identity", value=" FederationMangementService_192.168.0.168_10086_tcp "/>
    </Attributelist>
    <Operationlist>
      <Operation>
            <Operation name=" createFederationExecution ", type=" void"/>
            <parameter name=" executionName ", type=" string "/>
             <parameter name=" FED ", type=" string "/>
      </Operation>
      ……
        </Operationlist>
        <exception>
            < exception name=" RTIexception" / >
        </exception >
        ……
  </sevice>
</sdl:types>
```

3.2 The Integration Of Simulation

The integration of simulation is able to integrate many service publishers which supply similar class or similar simulation service, which not only facilitate classifying and managing simulation service but also facilitate simulation service schedule. In process of integration of simulation service, the first step is to abstract same or similar class and simulation from the SDL and name them with unified name which must be generalized and special for the abstract service. The abstract service name is necessary for SSMB to query and match related simulation service for service requester. The abstract named service will be sorted in a queue and many simulation services join the queue according to itself function and classification with related service identify. When service requester applies for calling the simulation to SSMB, SSMB can match the applied service with abstract service name in the service queen. When complete the matching the SSMB will choose the front service identify queued in the service identify queue with high credit rating, and then call related simulation service with SDL. Its process can be shown in figure 3.

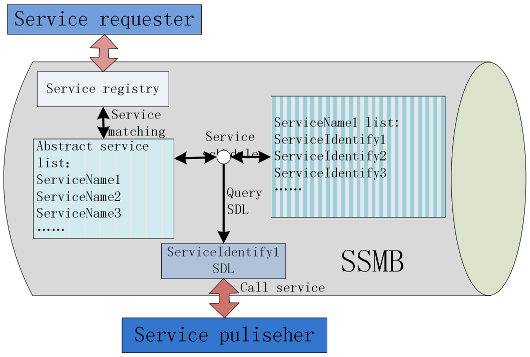


Fig.3. The Integration Of Simulation Service

3.3 Majorization Of Service Composition

In the process of calling simulation service, a service combination with specific function is unavoidable to cope with a large number of simulation services with similar or identical functions, which usually form amounts of different service combinations solutions. How to select the optimal solution from multiple combinations solutions belongs to combinatorial optimization problems, which is a NP problem. which is shown as Figure 4.

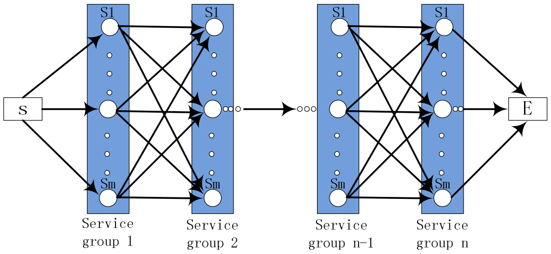


Fig.4.The Process Of Service Combination

In the Figure 4, S represents start point and E represents end point, both of them are virtual position. Service group represents a service group which is composed of services with same interface and function supplied by service publishers. Evaluating indicator of simulation service which include Performance, Reliability, Availability and Reputation are taken as constraint condition to evaluate the quality of simulation. Specific services are chosen from the related service groups and the composition solution is composed of these services. The global optimal service composition solution is searched under the specific constraint condition of qualify of simulation service and allocated weighting of evaluated indicator. In this article, each service composition process is coded as a chromosome. The quality of simulation service evaluated by the global composition solution is taken as objective function which is to be optimized by quantum optimized algorithm. Quantum rotate gate is used to optimize the composition solution until get the global optimizing composition solution. The process of quantum optimized algorithm can be shown in the figure 5.

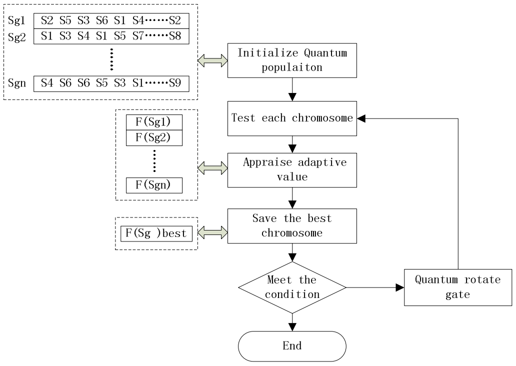


Fig.5. The Process Of Quantum Optimized Algorithm

3.4 Thread Evictor

SSMB establish listener thread according to multiple service requests and create a thread to calls simulation services according to the needs of the service requester requirements. If Simulation services create a listener thread for each simulation, not only to the system memory consumption and not all service requests arrive simultaneously. So it can establish a listening queue for services that are called frequently, which could accept the listener thread queue client connections and process the request message. Through the dynamic adjustment of the size of the queue threads can adapt the ever increasing service requests. However, when there are a large number of service requests, with the thread increasing also brought a large number of the system memory consumption, even to the point of unbearable. Streamlining is therefore a need for the member of thread queues, hence a thread evictor concept presented here is used to solve this problem. In accordance with the “recently the least uses” order to maintain the thread queue members online, least recently used end of the thread in the queue, the most recently used up to the thread at the head of the queue. In other words, the "latest" the thread is always in the head, "the oldest" is always at the end of the thread. The Length of the queue can be configured to decide the number of listening threads stored in a queue if the queue is full, evictor will remove the least recently used thread in the end of thread queue, and the removed thread will sleep and be awaken regularly to change or update status information, and then go into hibernation again. It can be shown in the figure 6.

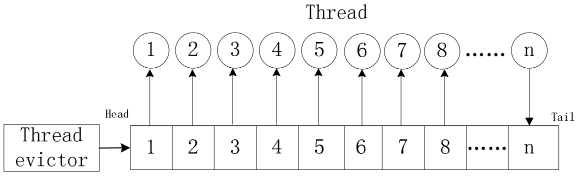


Fig.6.Thread Evictor

3.5 Loading Balancing

If SSMB is a node, when it deal with a large amounts of interactive data in the same time can result in lowering worked efficiency or leading to crashes. Hence, one mater many slaves mode for load balancing in SSMB is proposed to address this bottleneck in this article. The master node is refers to the SSMB’s node, and the slave node is able to backup master node. The slave node may be disposed in the different node according to the system actual load needed. It will backup essential data in master node and detect master node load state in real time. When master node load is overweight, some duties of master node can be allocated to the slave node dynamically. When the master node crashes, the slave which has least load will instead. Load balance modes provide random mode, adaptive mode, polling algorithm mode to be chosen: random mode is select a node randomly; adaptive mode is select a node which has least load; polling mode select the node by the default order. In practical applications, this mode can be selected flexibly. One mater many slaves mode can greatly improve the stability and efficiency of SSMB. Likewise, for the simulation service also can uses one mater many slaves mode, when a master node works several node with similar function monitor the maser in real-time and directly instead of master node real-time according when needed.

3.6 Communication Encryption

Communication security is also an important consideration for distributed simulation application. Protecting sensitive information, ensuring its integrity, inspection of the identity of the communicating units, these capabilities on a service-oriented simulation application are essential for developing secure applications. The communication protocol of Secure Socket Layer (SSL) is important secure network communications protocol which is communication protocol for SDRTI. Federate will establish SSL handshake when connect to SSMB. In typical handshake process, it will verify the digital certificates of various communication sides and exchanges the symmetrical key for encryption conversation communication. Once the handshake is success, SSL message authentication codes are used to ensure data integrity, so as to allow federate, SSMB and simulation services reasonably believe their communication is security. It can be shown in the figure 7.

The special support layer directly faces the different simulation users and the application domain, its function service is closely connected with the special request of different simulation application domain and provide the simulation component for the special-purpose simulation system, for example: Information processing simulation component, direction decision-making simulation component and so on. Simulation application commonality supported layer provides common features that you want to build a simulation application system services, including simulation configuration control, simulation, simulation, simulation clock management monitor repetition of the record, protocol conversion, graphics display, geo-information services, resource management, and so on.

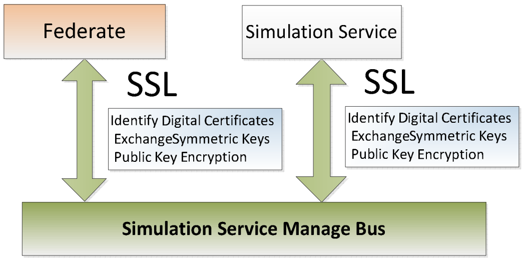


Fig.7. Communication Encryption

4. CONCLUSIONS

SSMB is important to utilize SOA in distributional simulation. This article in view of the insufficiency that traditional service bus, has used service registry and record, integration service, load balancing, majorization of service composition, thread evictor, and communication encryption. It has achieved important functionality including integration service, load balancing, majorization of service composition, thread evictor, and communication encryption and so on. To a certain extent, it overcomes many shortcomings of traditional service bus, and entirely improves the performance of management simulation service bus. Next step, data conversion, quality of service issues, will be researched.

ACKNOWLEDGEMENTS

This work was supported by National Natural Science Foundation of China (Grant no. 70791137).

REFERENCES:

[1] N.Bieberstein, S.Bose, M. Fiammante, K.Jones and R.Shah, Service Oriented Architecture Compass: Business Value, Planning and Enterprise Roadmap. Upper Saddle River, NJ: Pearson Education, Inc, 2006.

[2] M.B. Blake, Decomposing composition: Service oriented software engineers, *IEEE Software (Nov.)*, Vol.9, No.8, 2007, pp.68-77.

[3] J. Bosch, Service orientation in the enterprise, *IEEE Computer (Nov.)* Vol.12, No.3, 2007, pp.51-56.

[4] F. Curbera, Component contracts in Service Oriented Architectures, *IEEE Computer (Nov.)*, Vol.6, No.7, 2007, pp.74-80.

[5] M. Kaiser,Towards the realization of policy-oriented enterprise management, *IEEE Computer (Nov.)* , Vol.8, No.3,2007, pp.51-56.

[6] O. Nano, A. Zisman, Realizing service-centric software systems, *IEEE Software (Nov.) Vol.5,* No.3, 2007, pp.28-30.

[7] M.P. Papazoglou, et al., Service-oriented computing: State of the art and research challenges, *IEEE Computer (Nov.),* Vol.5, No.9, 2007, pp. 38-45.

[8] Linthicum, D. Enterprise application integration. Massachusetts, USA: Addison-Wesley, 2000.

[9] ChapPell Danel.Enterprise Service Bus[C].0'Reilly, 2004.

[10] G.Ziyaeva, E. Choi and D.Min, "Content-Based Intelli-gent Routing and Message Processing in Enterprise Service Bus," *2008 International Conference on Conver-gence and Hybrid Information Technology, Washington,* Vol.13, No.3,2008, pp.28-29.

[11] K.Ueno and M. Tatsubori, "Early Capacity Testing of an Enterprise Service Bus," *IEEE International Conference on Web Services,Chicago,*Vol.18,No.22, 2006, pp.70 9-716

[12] Marek Psiuk,Tomasz Bujok, and Krzysztof Zieliski,Member,IEEE, Enterprise Service Bus MonitoringFramework for SOA Systems. *IEEE TRANSACTIONS ON SERVICES COMPUTING,* Vol.5, No.3, 2012 JULY-SEPTEMBER, pp.56-79.

[13] Xiaoqiang,Liu and Yingpei Wu.The Learning Service Bus Based on SOA. *Proceedings of the World Congress on Engineering and Computer Science,* Vol.11, No.4,2008, pp.22-24.

[14] FU Ning, ZHOU Xingshe, WANG Kaibo, ZHAN Tao. NpuESB:a BPM Environment based on Enterprise Service Bus. *2009 International Conference on Information Technology and Computer Science,* Vol.7, No.4, 2009, pp.465-469.

[15] Jianwei Yin, Hanwei Chen, Shuiguang Deng, and Zhaohui Wu. *A Dependable ESB Framework for Service Integration. IEEE INTERNET COMPUTING*, Vol.10, No.3, 2008, pp.26-33.

[16] Dongjin Yu, Daqiang Yan.Towards the Integration of Enterprise Service Bus with UDDI Server: A Case Study. *Proceedings of 2011 International Conference on System Science and Engineering, Macau, China-June* Vol.6, No.7, 2011, pp.28-31.

[17] Gabriela Gheorghe, Stephan Neuhaus, and Bruno Crispo. xESB: An Enterprise Service Bus for Access and Usage Control Policy Enforcement. *International Federation for Information Processing 2010*. AICT 321, 2010, pp.63-78.

[18] Jameela Al-Jaroodi, Nader Mohamed. Service-oriented middleware: A survey. *Journal of Network and Computer Applications*, Vol.5, No.6, 2012, pp.211–220.

[19] Web Services Description Language (WSDL) Version 1.2[EB/OL].[2003-3-3] [2006-4-1].

［20］Gareymr, Johnsonds. computers and intractability: a guide to the theory of NP completeness.New York: FREEMAN W H and company,1979.