

Kim-Meow Liew Hong Shen
Simon See Wentong Cai
Pingzhi Fan Susumu Horiguchi (Eds.)

LNCS 3320

Parallel and Distributed Computing: Applications and Technologies

5th International Conference, PDCAT 2004
Singapore, December 2004
Proceedings

 Springer

Technique for Java	752
del	756
Zhan	763
DP	768
on Semantic Grid	772
HT-Based Peer to	778
ased Networking	782
chanism Without	787
Coefficients Using	791
le	791
Having Specified	795
Li, Patch Beadle ...	795
Sheets	799

Operating Systems

State Machine Based Operating System Architecture for Wireless Sensor Networks	803
<i>Tae-Hyung Kim, Seongsoo Hong</i>	
QoS-Aware Admission Control for Video-on-Demand Services	807
<i>Ilhoon Shin, Y.H. Shin, Kern Koh</i>	
Context Aware Thread Aggregation in Linux	811
<i>Takuya Kondoh, Shigeru Kusakabe</i>	

Parallelizing Compilers

Compiler-Assisted Software DSM on WAN Cluster	815
<i>Junpei Niwa</i>	

Performance Evaluation and Measurements

Performance Analysis of Batch Rekey Algorithm for Secure Group Communications	829
<i>Jun Zhang, Fanyuan Ma, Yingcai Bai, Minglu Li</i>	
Workload Dispatch Planning for Real-Time Fingerprint Authentication on a Sensor-Client-Server Model	833
<i>Yonguha Chung, Daesung Moon, Taehae Kim, Jin-Won Park</i>	
Explaining BitTorrent Traffic Self-Similarity	839
<i>Gang Liu, Mingzeng Hu, Bingxing Fang, Hongli Zhang</i>	
Scheduler Oriented Grid Performance Evaluation	844
<i>Liang Peng, Simon See</i>	

Tools and Environments for Software Development

Integrated Service Component Development Tool on Web Services	848
<i>Woon-Yong Kim, R. Young-Chul Kim</i>	
On Construction of a Large Computing Farm Using Multiple Linux PC Clusters	856
<i>Chao-Tung Yang, Chun-Sheng Liao, Kuan-Ching Li</i>	

Integrated Service Component Development Tool on Web Services

Woon-Yong Kim¹ and R. Young-Chul Kim²

¹ Dept of Computer Science, Kwang-Woon University,
Seoul, 139-701, Korea
wykim@kw.ac.kr

² Dept of Computer Information Comm., Hong-Ik University,
Jochiwon, 339-701, Korea
bob@hongik.ac.kr

Abstract. The web services based on the HTTP protocol and XML are the efficient way to integrate distributed applications on the Internet. The web services contain SOAP, WSDL and UDDI standard, and various technologies related to them are researched now. In this paper, we design and implement an efficient integrated development tool for web services components. Generally a structure of web service components is constructed by combining other components closely. For this, it may be required the efficient mechanism during design of service components. Therefore we propose an integrated model of the functions required and some composition techniques of service components. Then we develop the integrated service component development tool. This automatic tool provides UML design on the GUI and includes some procedures that are creation and registration of WSDL document for accessing the web service component efficiently and that create wrapper class composing with the other services. As a result, this tool will provide the opportunity of faster and stable component development.

1 Introduction

The web services are the interface, which describes the group of operations that access networks through the standard XML messages. One way of web service integrations, the web services easily make system integration through the web [8]. This is recognized as new substitute for the distributed computing models like CORBA, Java RMI, DCOM. The web services are consisted of open-oriented standards with SOAP, WSDL, UDDI on the XML. Various toolkits for web application development are released. These toolkits are providing with API and tools.

In this paper, we develop the integrated automatic development tool for developing the efficient web service components. Generally, the web service component has service structures either with independent-structured service or with other web service components [2]. As a result, the processes of the component design and implementation on the web service should be required the mechanisms during design of service. For supporting them, our tool provides the efficiency of component design and im-

development Tool on

Chul Kim²

oon University,

ong-Ik University,

sa

protocol and XML are the effi-
 e Internet. The web services
 rious technologies related to
 and implement an efficient
 onents. Generally a structure
 ombining other components
 mechanism during design of
 rated model of the functions
 vice components. Then we
 ment tool. This automatic
 as some procedures that are
 accessing the web service
 s composing with the other
 erty of faster and stable

es the group of operations that
 One way of web service integra-
 on through the web [8]. This is
 uting models like CORBA, Java
 1-oriented standards with SOAP,
 eb application development are
 is.
 e development tool for develop-
 the web service component has
 ervice or with other web service
 onent design and implementa-
 anisms during design of service
 y of component design and im-

2004.

plementation with integrating the functions required for the design and implementation of the web service component. This tool provides UML design and implementation on the GUI, and includes creation and registration of WSDL document for accessing the web service component and wrapper class to use the other services. This tool includes each step for software design, code generating, web service wrapper class creation, web services component creation and registration, and UDDI browser. In this paper, the chapter 2 describes the technologies and toolkits of web services, and the chapter 3 suggests the system structure and the component development process of the web services. The chapter 4 describes characteristics and techniques of integrated development tool on web services, and the chapter 5 is conclusion.

2 Related Researches

2.1 Web Services and Development Toolkit

It is recognized the web service as new technology, which makes system integration on the web environment based on XML, and HTTP possible. It consists of 3 open-oriented standards: the one of elements is SOAP(Simple Object Access Protocol) for the protocol of XML transmission[4], and the second one is WSDL(Web Service Description Language), the language for web service description[9], and the final one is UDDI(Universal Discovery Description Integration), the standard of storehouse for sharing information [6]. Also to solve the security problem which the web service technology does not support, BPEL4WS(Business Process Execution Language for Web Services) specs, WS-Security, WS-Coordination and WS-Transaction specs were released[7]. The architecture of web service is shown in fig. 1.

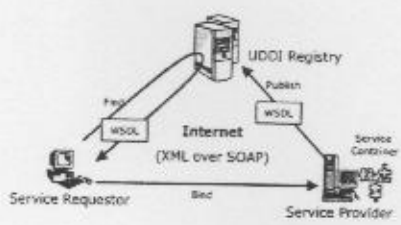


Fig. 1. Web services architecture

Various toolkits for developing web services being released. SUN releases JWSDP(Java Web Services Developer Pack)[5] version 1.3 as web service development pack. IBM starts to develop WSTK(Web Services ToolKit) as AlphaWorks project, and now releases WSTK version 1.2 including ETTK (Emerging Technologies Toolkit)[3]. Also Apache as public software group starts to develop SOAP toolkit, and now Axis version 1.2 Alpha [1]. This paper uses this toolkit for web service development. In addition, UDDI4J module in the ETTK [3] is used for making UDDI browser.

2.2 Web Services Platform

Many platforms on web services are released. Those platforms have their own arrangement methods. The representative platforms are SUN's Java Web Service Development Pack(JWSDP), Cape Clear's CapeConnect, Iona's XMLBus, Apache's Axis, Systinet's Web Applications and Service Platform(WASP), and Mind Electronic's GLUE etc[2]. This paper only uses the platform of JWSDP's environment.

3 Integrated Web Services Component Development Tool

Our tool is providing with two mechanisms by creating new component, and by using the existing web service component during design process. Also, it provides web services component development environment.

3.1 The System Structure of Tool

Our tool is provided with web services component development environment based on the structure of JAVA. Fig. 2 (left) shows structure in this integrated tool. This tool is implemented by JWSDP, which SUN supported, and its XML Parser is Apache Xerces. In addition, UDDI4J in the IBM AlphaWorks's project is used for UDDI browser.

This tool provides GUI environment based on UML 1.3. Also it provides efficiency of design through use-case, class diagram, and collaboration diagram on graphical environment. UDDI browser searches the web service component required at design step, and provides WSDL information required to composite components. The code-generation module creates skeleton code required to generate components through design information. WSC (Web Service Component) wrapper generation module creates wrapper class for the other web service components. The wrapper class provides the class for static and dynamic access. WSC generator module is in charge of WSDL creation that makes the documentation of designed web service component and service description. WSC registration is providing the registration process that new web service components are registered to web service container. The relationships of all the elements are shown in fig. 2 (right).

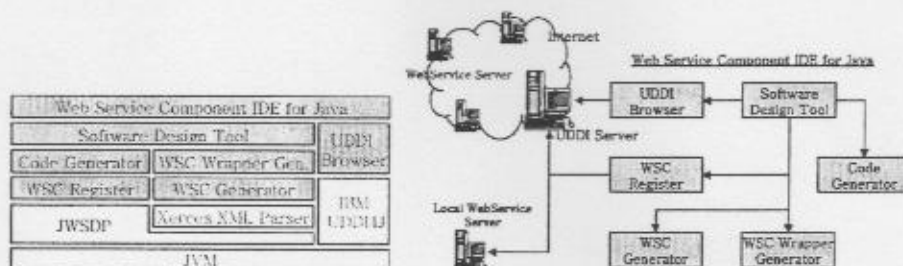


Fig. 2. Structure (left) and relationship of elements (right) in our development tool

3.2 The Web Services Component Development Process

The development process of web services component consists of the web services component design process, the code generation process, WSDL document and web service creation, and the registration process that registers web service component into the web server container. The web service component development process in our integrated tool is shown in fig. 3.

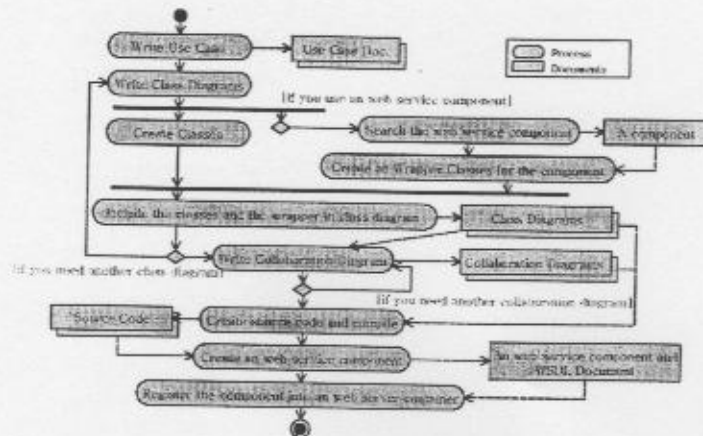


Fig. 3. Process of web services component development

Our process of web services starts from defining the web services component. The functions of component are defined by Use case doc., and class diagram is created based on this defined documentation. If the usage of other web service is necessary locally, the information of web services component will be gained by searching UDDI, and wrapper class will be created. The class diagram is composed of original class and wrapper class. Also collaboration model required for the service can be composed. The next step is generating source code. The generated source code will be the service-runnable class file after compiling and building. Through this information, WSDL required in the web services is created, and also composed as one component possible for web service. This component can be serviced through the registration on web server container. Our development tool includes the above steps. We may give a faster and stable component development chance to the developer.

4 Implementation of the Integrated Web Services Component Development Tool

Our tool environment consists of the component design, wrapper class creation, code generation, and web service component creation and registration tool. This tool provides the interface environment that manages and runs components per each project.

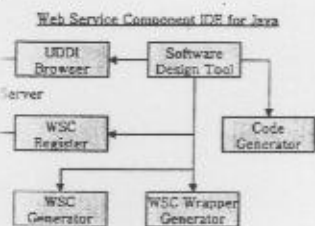
platforms have their own architecture. SUN's Java Web Service Development Kit, IBM's XMLBus, Apache's Axis, Microsoft's ASP, and Mind Electronic's SDP's environment.

Development Tool

new component, and by using the process. Also, it provides web

development environment based on the integrated tool. This tool is supported, and its XML Parser is based on the ApacheWorks's project is used for

XML 1.3. Also it provides efficient and collaboration diagram on the web service component required to composite components. The wrapper class is required to generate components. The wrapper class is composed of original class and wrapper class. Also collaboration model required for the service can be composed. The next step is generating source code. The generated source code will be the service-runnable class file after compiling and building. Through this information, WSDL required in the web services is created, and also composed as one component possible for web service. This component can be serviced through the registration on web server container. Our development tool includes the above steps. We may give a faster and stable component development chance to the developer.



ht) in our development tool

4.1 Web Services Component Design Tool

Our design tool provides the creation and management of Use case, class diagram and collaboration diagram based on UML 1.3. The Use case part composes the description of web service component. The class diagram part, as a core component of web services development, is in charge of designing static web services component models with original classes and classes created from the wrapper class. The collaboration diagram represents method-calling relationships on objects. These class models created by this tool are used as the information for source code automatic generation. They provide with consistency between design and implementation, so that they can provide stable web component creation and management. Fig. 4 shows Use case document, class and collaboration diagram for web services component model.

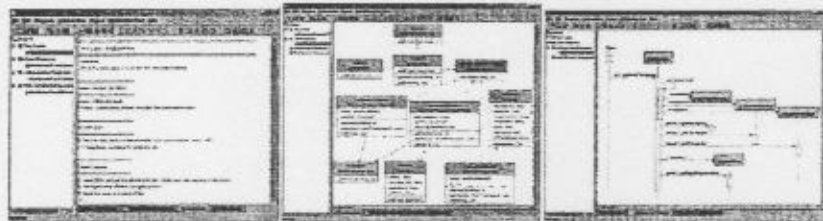


Fig. 4. Use case view, web services component static model and collaboration diagram

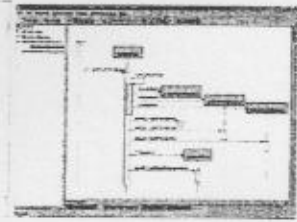
4.2 Web Services Wrapper Class Creation Tool

The web services component can compose of not only web services client program, but also other web services. To use the web services component when developing a program, many steps are required. WSDL document registered UDDI should be referenced, and class file should be created based on this WSDL document, and then this file applies to the program. Generally, the way to access web services is classified three mechanisms; static stub, dynamic proxy which create object on runtime, and DII(Dynamic Invocation Interface). The wrapper class creation tool provides 3 types introduced above. Fig. 5 shows the activity of wrapper class creation.



Fig. 5. Web services wrapper class creation activity

Use case, class diagram and collaboration diagram as a core component of web services wrapper class. The collaboration on objects. These class models source code automatic generation. implementation, so that they can be implemented. Fig. 4 shows Use case services component model.



Model and collaboration diagram

web services client program, component when developing a registered UDDI should be re- WSDL document, and then this access web services is classified create object on runtime, and class creation tool provides 3 types of class creation.

Use Case diagram

collaboration activity

As a user use one of three different mechanisms, developers may concentrate on only service domain because of hiding functional part to access the service. The process of web services wrapper class creation is classified several parts; selection process, wrapper class creation process, and wrapper class registration. Fig. 6 shows the dialog when class creation tool is selected and UDDI Browser.

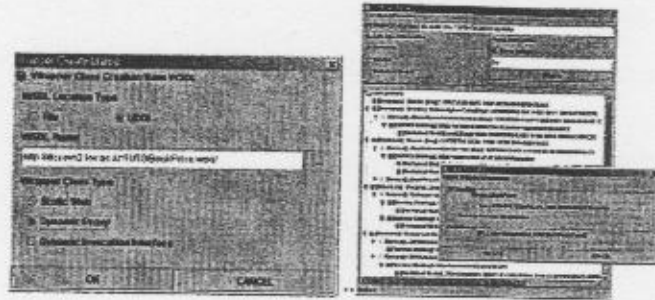


Fig. 6. Wrapper class creation view(left) and UDDI browser(right)

This dialog (left) can create the desired wrapper classes with selecting WSDL document required wrapper class creation and wrapper class type. When WSDL is acquired by UDDI, the WSDL document can be selected with UDDI browser. The UDDI browser searches on views of business, service, and t_Model. Through the chosen WSDL documentation, wrapper class can be created as one of static stub, dynamic proxy, and DII. These wrapper classes will be discussed as followed.

4.2.1 Wrapper Class Creation for Static Stub

The wrapper class for static stub is made of the structure that uses stub created before runtime. This wrapper class has structure shown as fig. 7 (left). The wrapper class for static stub is in charge of acquiring stub object and making it available.

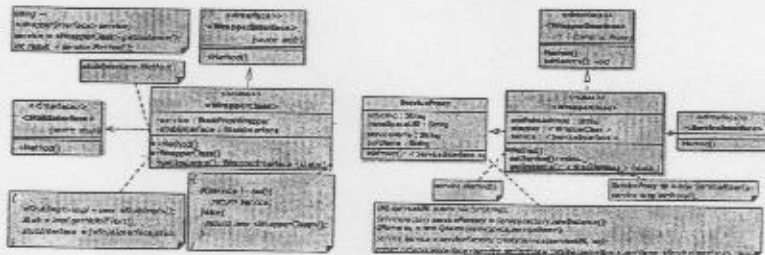


Fig. 7. Wrapper class for stub (left) and for structure based on Dynamic proxy (right)

4.2.2 Creation of Wrapper Class to Access Web Services on Dynamic Proxy

The static stub about web services depends on the implementation environment. To overcome this limit, the dynamic proxy, which creates web service object during

runtime, is using for it. This creation of dynamic proxy is done through WSDL documentation. This wrapper class is in charge of management of web service information and creation of dynamic proxy. Fig. 7 (right) shows structure based on dynamic proxy.

4.2.3 Creation of Wrapper Class Based on Dynamic Calling Interface

The stub or dynamic proxy needs service interface information before running, but Dynamic Invocation Interface can access the service without service name and procedure information until runtime. However, these methods need more informative procedures than other methods. Fig. 8 shows the wrapper class that provides DII access.



Fig. 8. Wrapper class structure for access DII

The wrapper class creation tool provides with creating the desired wrapper class automatically based on the WSDL information and class structure of desired wrapper type. In addition, it provides class information for creating component through the process of registering them to the class diagram. The web services developer can hide the functional requirements for access. So the developer can construct the stable component from the service point of view.

4.3 Code Generation Management Tool

The code generation management tool creates source code and provides environment for code modification, compilation and run based on class diagram and collaboration diagram made for web services component. Fig.9 (left) shows source code management tool. This tool is related with class diagram, and it composes design information related directly with code information. It provides efficient management with consistency between design and implementation.

4.4 Web Services Component Creation and Registration Tool

The web service component creation and registration tool creates WSDL document, and registers the component to the Tomcat on the JWSDP. The WSDL document is created as setting up several options, which are interface for service, package information, service registration location, service name, WSDL document name and WSDL style information. The created WSDL document and running classes are integrated, and registered to the web server container. This is shown in fig. 9 (right).

proxy is done through WSDL management of web service (http) shows structure based on

Calling Interface

information before running, but without service name and methods need more informative class that provides DII access.

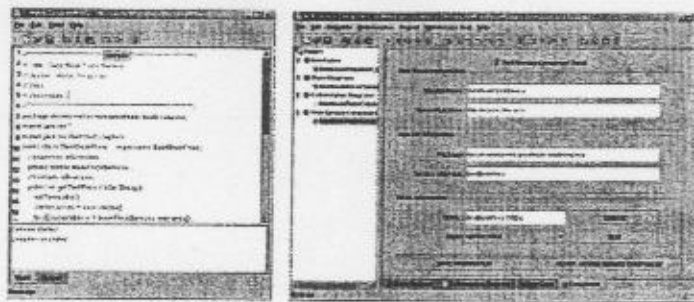


Fig. 9. Source code management (left) and component creation and registration tool (right)

5 Conclusion

Nowadays, Web service technologies have many standards like SOAP, WSDL, UDDI. They should be applied quickly for changing existing user oriented and application oriented web environment and replacing the existed distributed environment like CORBA, DCOM, Java RMI. Many specs. are modified consistently.

To solve this problem, we suggest to our integrated development tool, which may provide the efficient environment for faster and stable component development on web services. In this paper, we show to design and implement our integrated development tool for supporting efficient development on web services. Our integrated environment include as follows. Web services component design tool based on UML 1.3, Web service wrapper class creation tool, Code generation management tool, web services component creation and registration tool. As a result, we believe to develop faster and stable service component. We need consistent studies for supporting evaluated specs and additional web service functions.

References

1. Apache Web Services - Axis, "http://ws.apache.org/axis/index.html"
2. Harvey M. Deitel, Paul U. Deitel, "Java Web Services For Experienced Programmers", Prentice Hall, 2003
3. IBM AlphaWorks Emerging Technologies ToolKit, "http://alphaworks.ibm.com/tech/ettk", April 8 2003
4. Simple Object Access Protocol Spec. <http://www.w3.org/TR/soap12>
5. Sun Java Web Services Developer Pack (JWSDP), "http://java.sun.com/webservices/jwsdp/index.jsp"
6. UDDI Technical White Paper, http://www.uddi.org/pubs/Iru_UDDI_Technical_White_Paper.pdf
7. W3C Web Services Activity, <http://www.w3.org/2002/ws>
8. Web Services Architecture "http://www.w3.org/TR/2004/NOTE-ws-arch-20040211"
9. Web Services Description Languages (WSDLv2.0), <http://www.w3.org/TR/wsdl20>

ation Tool

ool creates WSDL document, DP. The WSDL document is interface for service, package. WSDL document name and tent and running classes are This is shown in fig. 9 (right).