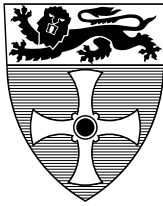


UNIVERSITY OF
NEWCASTLE



University of Newcastle upon Tyne

COMPUTING SCIENCE

Experimenting With Exception Handling Mechanisms Of Web Services
Implemented Using Different Development Kits

A. Gorbenko, A. Mikaylichenko, V. Kharchenko, A. Romanovsky.

TECHNICAL REPORT SERIES

No. CS-TR-1010 March, 2007

Experimenting With Exception Handling Mechanisms Of Web Services Implemented Using Different Development Kits

A. Gorbenko, A. Mikhaylichenko, V. Kharchenko, A. Romanovsky.

Abstract

Achieving high dependability and fault-tolerance in service-oriented architecture (SOA) is an open problem. Exception handling is one of the powerful means for improving the quality of SOA. The paper discusses the results of experimental analysis of the SOA-specific exceptions and factors affecting availability and fault-tolerance of Web Services, implemented using two development kits: JAX-RPC implementation at Sun Microsystems and IBM WebSphere Software Developer Kit for Web Services. We specifically focus on the results of exception propagation and performance analysis. Finally, applications of different error recovery strategies including backward, forward and enhanced forward error recovery, in the context of SOA are briefly discussed.

Bibliographical details

GORBENKO, A., MIKHAYLICHENKO, A., KHARCHENKO, V., ROMANOVSKY, A..

Experimenting With Exception Handling Mechanisms Of Web Services Implemented Using Different Development Kits

[By] A. Gorbenko, A. Mikhaylichenko, V. Kharchenko, A. Romanovsky.

Newcastle upon Tyne: University of Newcastle upon Tyne: Computing Science, 2007.

(University of Newcastle upon Tyne, Computing Science, Technical Report Series, No. CS-TR-1010)

Added entries

UNIVERSITY OF NEWCASTLE UPON TYNE
Computing Science. Technical Report Series. CS-TR-1010

Abstract

Achieving high dependability and fault-tolerance in service-oriented architecture (SOA) is an open problem. Exception handling is one of the powerful means for improving the quality of SOA. The paper discusses the results of experimental analysis of the SOA-specific exceptions and factors affecting availability and fault-tolerance of Web Services, implemented using two development kits: JAX-RPC implementation at Sun Microsystems and IBM WebSphere Software Developer Kit for Web Services. We specifically focus on the results of exception propagation and performance analysis. Finally, applications of different error recovery strategies including backward, forward and enhanced forward error recovery, in the context of SOA are briefly discussed.

About the author

Alexander (Sascha) Romanovsky is a Professor in the CSR. He received a M.Sc. degree in Applied Mathematics from Moscow State University and a PhD degree in Computer Science from St. Petersburg State Technical University. He was with this University from 1984 until 1996, doing research and teaching. In 1991 he worked as a visiting researcher at ABB Ltd Computer Architecture Lab Research Center, Switzerland. In 1993 he was a visiting fellow at Istituto di Elaborazione della Informazione, CNR, Pisa, Italy. In 1993-94 he was a post-doctoral fellow with the Department of Computing Science, the University of Newcastle upon Tyne. In 1992-1998 he was involved in the Predictably Dependable Computing Systems (PDCS) ESPRIT Basic Research Action and the Design for Validation (DeVa) ESPRIT Basic Project. In 1998-2000 he worked on the Diversity in Safety Critical Software (DISCS) EPSRC/UK Project. Prof Romanovsky was a co-author of the Diversity with Off-The-Shelf Components (DOTS) EPSRC/UK Project and was involved in this project in 2001-2004. In 2000-2003 he was in the executive board of Dependable Systems of Systems (DSoS) IST Project. Now he is coordinating Rigorous Open Development Environment for Complex Systems (RODIN) IST Project (2004-2007).

Suggested keywords

EXCEPTION HANDLING,
WEB SERVICES

EXPERIMENTING WITH EXCEPTION HANDLING MECHANISMS OF WEB SERVICES IMPLEMENTED USING DIFFERENT DEVELOPMENT KITS

A. GORBENKO	A. MIKHAYLICHENKO	V. KHARCHENKO	A. ROMANOVSKY
<i>National Aerospace University, Kharkiv, Ukraine</i>	<i>National Aerospace University, Kharkiv, Ukraine</i>	<i>National Aerospace University, Kharkiv, Ukraine</i>	<i>University of Newcastle upon Tyne, UK</i>

Achieving high dependability and fault-tolerance in service-oriented architecture (SOA) is an open problem. Exception handling is one of the powerful means for improving the quality of SOA. The paper discusses the results of experimental analysis of the SOA-specific exceptions and factors affecting availability and fault-tolerance of Web Services, implemented using two development kits: JAX-RPC implementation at Sun Microsystems and IBM WebSphere Software Developer Kit for Web Services. We specifically focus on the results of exception propagation and performance analysis. Finally, applications of different error recovery strategies including backward, forward and enhanced forward error recovery, in the context of SOA are briefly discussed.

CONTENTS

1. Introduction	2
2. Web Services Development Toolkits	3
2.1. NetBeans IDE/Sun Java System Application Server bundle	3
2.2. IBM WebSphere Software Developer Kit for Web Services	3
3. Example of Web Service Implementation and Description	3
4. SOA-specific Errors and Failures	6
5. Analysis of Exception Raising Mechanism and Performance Implications	8
5.1. Exceptions Correspondence Analysis	8
5.2. Exception propagation and performance analysis	10
6. Exception Handling and Fault-tolerant Means	16
Appendix A. Client-Side Program Code Invoking Web Service	19
Appendix B. Exceptions Stack Traces	21

1. Introduction

The objective of the paper is to analyze the features of exception raising mechanisms and performance implications in the service-oriented architecture depending on the Web Services development toolkits used. To improve Web Services dependability we propose a middleware-based architecture which uses redundancy of target Web Services and implements different procedures of exception handling.

Our experimental investigation has been organised as following (see Fig.1). We developed Java class “WSCalc” which performs simple arithmetic operation (multiplication) upon two integers and converts the result to a string (1). Then, we implemented two diverse Web Services using two different development toolkits (2). These testbed services were deployed on the two computers using different application servers: IBM WebSphere and SJS AppServer (section 2). These computers were operated under Windows XP and located in the university’s LAN. Thus, the transfer delays and other network problems affected them in the same way.

In section 3 we present the results of a comparison of generated WSDL descriptions (3). After that, in section 4, we analyse the SOA-specific errors and failures (4) and seed them into our Web Service architecture (5). Section 5 gives the results of analysis and comparison of exception raising mechanism and performance implications (6). In section 6, we discuss possible exception handling and fault-tolerant procedures (7) based on backward, forward and enhanced forward error recovery in middleware-oriented architecture.

The client-side program code and full exception stack traces are given in Appendixes A and B.

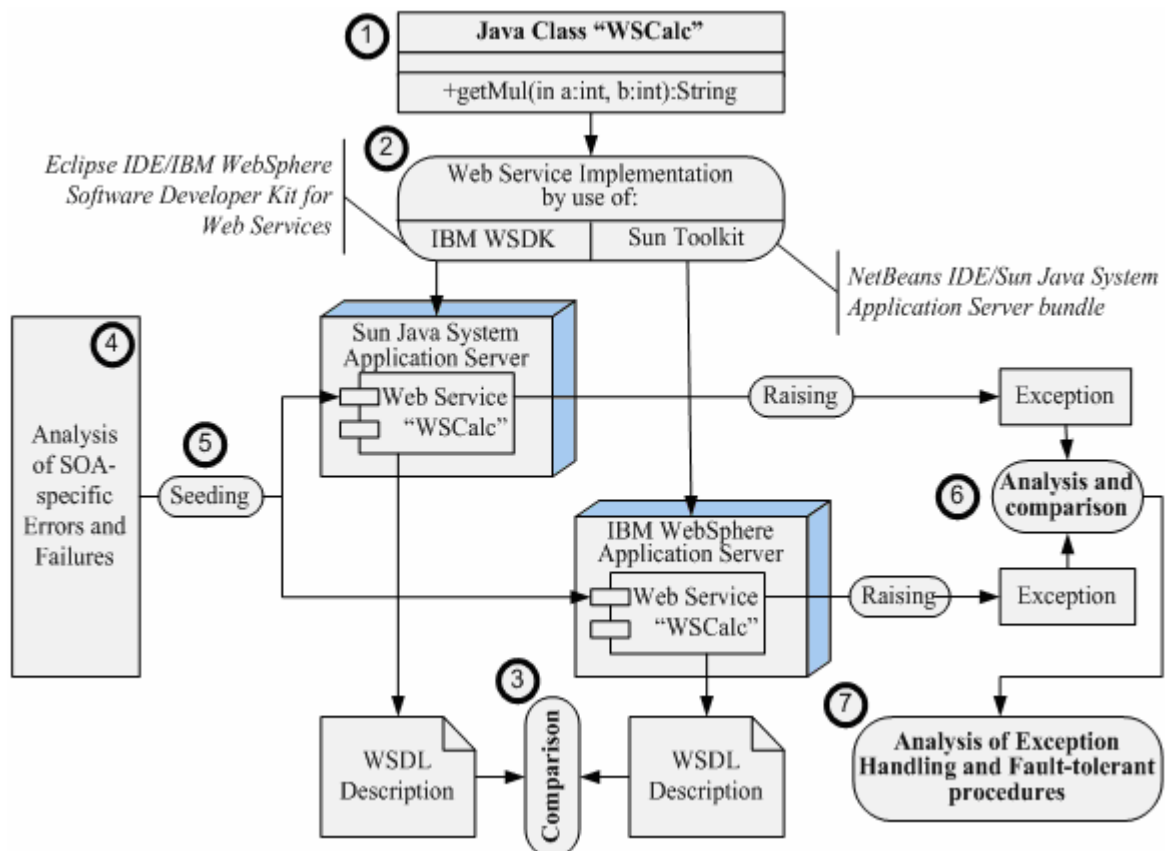


Figure 1. Research engineering process.

2. Web Services Development Toolkits

2.1. NetBeans IDE/Sun Java System Application Server bundle

NetBeans IDE 5.0 ¹ is a powerful integrated development environment for developing applications on Java platform, supporting Web Services technologies through the Java Platform Enterprise Edition (J2EE).

Sun Java System (SJS) Application Server² is the Java EE implementation at Sun Microsystems. NetBeans IDE with SJS Application Server support JSR-109, which is a development paradigm that is suited for J2EE development, based on JAX-RPC (JSR-101).

Web Service functionality in NetBeans IDE is part of an end-to-end set of J2EE features. Also, NetBeans IDE provides wizards to create Web Services and Web Service's clients.

2.2. IBM WebSphere Software Developer Kit for Web Services

IBM WebSphere Software Developer Kit for Web Services, Version 5.1 (WSDK)³ is an integrated kit for creating, discovering, invoking, and testing Web Services. The WSDK V5.1 is based on WebSphere Application Server V5.0.2 and provides support for the following open industry standards: SOAP 1.1, WSDL 1.1, UDDI 2.0, JAX-RPC 1.0, EJB 2.0, Enterprise Web services 1.0, WSDL4J, UDDI4J, WS-Security.

WSDK can be used with the Eclipse IDE. Eclipse provides a graphical interactive development environment, which provides tools for building and testing Java applications. WSDK adds to the standard Eclipse package the tools relating to Web Services, making it suitable for building Web Services. Required level of Eclipse is V2.1.1. The Eclipse package can be freely downloaded from the Eclipse Web site⁴.

Supporting the latest specifications for Web Services WSDK enables to build, test, and deploy Web Services on industry-leading IBM WebSphere Application Server. Functionality of the WSDK V5.1 has been recently incorporated into the IBM WebSphere Studio family of products.

3. Example of Web Service Implementation and Description

The starting point for developing a JAX-RPC Web Service is the coding of a service endpoint interface and an implementation class with public methods that must throw the *java.rmi.RemoteException*.

To analyze features of the exception raising mechanisms in the service-oriented architecture we have developed a testbed Web Service executing simple arithmetic operations. The service endpoint interface and the implementation class are shown in the Fig. 2 and Fig. 3.

```
package ai.xail2.loony.wscalculator;

public interface WSCalcSEI extends java.rmi.Remote {
    public String getMul (int a, int b)
        throws java.rmi.RemoteException;
    ...
}
```

Figure 2. The service endpoint interface describing simple arithmetic operations.

¹ www.netbeans.org

² www.sun.com/software/products/appsrvr_pe/index.xml

³ www-128.ibm.com/developerworks/webservices/wsdk/

⁴ www.eclipse.org

```
package ai.xail2.loony.wscalculator;

public class WSCalculator implements WSCalculatorSEI {
    public String getMul (int a, int b) {
        return new Integer(a * b).toString();
    }
    ...
}
```

Figure 3. The implementation bean class of the simple Web Service providing arithmetic operations.

A service endpoint interface (SEI) is a Java interface declaring service methods that a client can invoke. SEI is used for generating the WSDL specification of the Web Service and the stubs that connect a Web Service client to the JAX-RPC runtime.

NetBeans IDE/SJS AppServer and Eclipse IDE/IBM WSDK support wizards that automatically generate service description (WSDL-file) and deploy Web Service. However, in spite of the fact that both toolkits are based on the open specifications and interfaces we discovered a sufficient number of differences in generated Web Service descriptions (compare Fig. 4 and Fig. 5). They involve description of input and output parameters, definition of used namespaces (default and target namespaces). Some prefixes and namespaces are defined but not used.

Other differences are in the description of input and output parameters. As it will be shown below, these differences have effect on exception raising.

```

<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions targetNamespace="http://wscalcaloony.xai12.ai"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:apachesoap="http://xml.apache.org/xml-soap"
  xmlns:impl="http://wscalcaloony.xai12.ai"
  xmlns:intf="http://wscalcaloony.xai12.ai"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:wsdlssoap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <wsdl:types>
    <schema elementFormDefault="qualified"
      targetNamespace="http://wscalcaloony.xai12.ai"
      xmlns="http://www.w3.org/2001/XMLSchema">
      <element name="getMul">
        <complexType> <sequence>
          <element name="a" type="xsd:int"/>
          <element name="b" type="xsd:int"/>
        </sequence> </complexType>
      </element>
      <element name="getMulResponse">
        <complexType> <sequence>
          <element name="getMulReturn" nillable="true" type="xsd:string"/>
        </sequence> </complexType>
      </element>
    </schema> </wsdl:types>
    <wsdl:message name="getMulRequest">
      <wsdl:part element="intf:getMul" name="parameters"/>
    </wsdl:message>
    <wsdl:message name="getMulResponse">
      <wsdl:part element="intf:getMulResponse" name="parameters"/>
    </wsdl:message>
    <wsdl:portType name="WScalc">
      <wsdl:operation name="getMul">
        <wsdl:input message="intf:getMulRequest"
          name="getMulRequest"/>
        <wsdl:output message="intf:getMulResponse"
          name="getMulResponse"/>
      </wsdl:operation>
    </wsdl:portType>
    <wsdl:binding name="WScalcSoapBinding" type="intf:WScalc">
      <wsdlsoap:binding style="rpc"
        transport="http://schemas.xmlsoap.org/soap/http"/>
      <wsdl:operation name="getMul">
        <wsdlsoap:operation soapAction=""/>
        <wsdl:input name="getMulRequest">
          <wsdlsoap:body use="literal"/>
        </wsdl:input>
        <wsdl:output name="getMulResponse">
          <wsdlsoap:body use="literal"/>
        </wsdl:output>
      </wsdl:operation>
    </wsdl:binding>
    <wsdl:service name="WScalcService">
      <wsdl:port binding="intf:WScalcSoapBinding" name="WScalc">
        <wsdlsoap:address
          location="http://localhost:6080/wscalcaloony/services/WScalc"/>
      </wsdl:port>
    </wsdl:service>
  </wsdl:definitions>

```

Figure 4. WSDL description generated by the Eclipse IDE/IBM WSDK.


```

<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:tns="urn:WSCalc/wsdl"
  xmlns:ns2="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  name="WSCalc" targetNamespace="urn:WSCalc/wsdl">
  <types/>
  <message name="WSCalcSEI_getMul">
    <part name="int_1" type="xsd:int"/>
    <part name="int_2" type="xsd:int"/> </message>
  <message name="WSCalcSEI_getMulResponse">
    <part name="result" type="xsd:string"/> </message>
  <portType name="WSCalcSEI">
    <operation name="getMul" parameterOrder="int_1 int_2">
      <input message="tns:WSCalcSEI_getMul"/>
      <output message="tns:WSCalcSEI_getMulResponse"/>
    </operation>
  </portType>
  <binding name="WSCalcSEIBinding" type="tns:WSCalcSEI">
    <soap:binding
      transport="http://schemas.xmlsoap.org/soap/http"
      style="rpc"/>
    <operation name="getMul">
      <soap:operation soapAction=""/>
      <input>
        <soap:body
          encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
          use="encoded" namespace="urn:WSCalc/wsdl"/>
      </input>
      <output>
        <soap:body
          encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
          use="encoded" namespace="urn:WSCalc/wsdl"/>
      </output>
    </operation>
  </binding>
  <service name="WSCalc">
    <port name="WSCalcSEIPort" binding="tns:WSCalcSEIBinding">
      <soap:address
        location="http://loony.xail2.ai:8080/WS/WSCalc"
        xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"/>
    </port>
  </service>
</definitions>

```

Figure 5. WSDL description generated by NetBeans IDE/SJS AppServer.

4. SOA-specific Errors and Failures

In terms of the fundamental concepts of dependability [2], threats to computer systems include errors, faults and failures. An error is that part of the system state that may cause a subsequent failure: a failure occurs when an error reaches the service interface and alters the service.

A fault is the hypothesized cause of an error. Faults are usually classified into three major fault classes: design faults, physical faults and interaction faults [2].

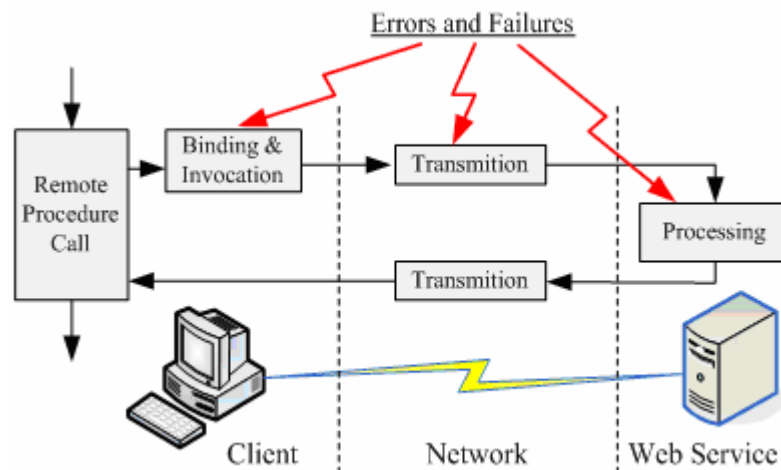


Figure 6. Classification of the SOA-specific errors and failures.

The main stages of the Web Services interaction are 1) service binding, 2) service invocation, 3) XML messages transferring, and 4) requests processing. In our work we have experimented with 18 types of the SOA-specific errors and failures occurring during these stages (see Table 1). We omitted the stages of service discovering and description using the UDDI registry because they are carried only once before the sequences of other interactions.

Table 1. SOA-specific errors and failures.

№	Type of error/failure	Error/failure domain
1.	Network connection break-off	Network and system failures
2.	Domain Name System (DNS) is down	
3.	Lost of packet with client request or service response	
4.	Remote host unavailable (off-line or unreachable)	
5.	Application Server is down	
6.	Suspension of Web Service during transaction (for example, getting into a loop)	Service errors and failures
7.	System run-time error (like “Stack overflow” or “Lack of memory”)	
8.	Application run-time error (for example, “Operand Type Mismatch”, “Product overflow”)	
9.	Application error causing user-defined exception (exception defined by service developer)	
10.	Error in Target Name Space	Client-side binding errors
11.	Error in Web Service name	
12.	Error in service port name	
13.	Error in service operation’s name	
14.	Output parameter type mismatch	
15.	Input parameter type mismatch	
16.	Error in name of input parameter	
17.	Mismatching of number of input service parameters	
18.	Web Service style mismatching (“Rpc” instead of “Document”)	

Network failures are unavoidable in the global service-oriented architecture. We analysed network connection break-off at the client-side and remote host unavailability when it is off-line or unreachable due to the network failures. Common-case network failures are down state of DNS or packets lost due to the network congestion. Besides, the operation of Web Service depends on the operation of the system software like web-server, application server and database management system. In our work we analysed failures occurring when the application servers (WebSphere or SJS AppServer) were shouted down.

Client errors in early binding or dynamic interface invocation (DII) (like “Error in Target Name Space”, “Error in Web Service name”, etc.) occur because of the change of invocation parameters, inconsistency between WSDL-description or/and service interface.

Finally, the service failures are connected with program faults and run-time errors causing system- or user-defined exceptions. System run-time errors like “Stack overflow” or “Lack of memory” result in the exceptions at the system level as a whole. Operation “Division by zero” is also caught and generates an exception at the system level but it is easier to simulate such system error than other ones.

An example of the application run-time errors are “Operand type mismatch”, “Product overflow” and “Index out of bounds”. In our experiments we simulated the “Operand type mismatch” error and also hangs of the Web Service due to its program getting into a loop and error causing user-defined exception (exception defined by a programmer during Web Service development).

5. Analysis of Exception Raising Mechanism and Performance Implications

To analyze features of exception raising mechanisms and performance implications in SOA-architecture depending on Web Services development toolkit used, we seeded errors in the testbed services and client applications, and also simulated network failures.

5.1. Exceptions Correspondence Analysis

We were experimenting with simple Web Services executing arithmetic operations (see Fig. 2 and 3) which were deployed on two application services: SJS AppServer and IBM WebSphere.

The results of our experiments with the Web Services exceptions are shown in Table 2. This table describes a relationship between errors/failures and the exceptions raised at the top level on different application platforms.

Table 2. Exceptions raised by different types of errors and failures.

№	Type of error/failure	Exception	
		Sun Microsystems WS Toolkit	IBM WS Toolkit (WSDK)
1.	Network connection break-off	HTTP transport error: java.net.UnknownHostException: loony.xai12.ai	faultCode: {http://websphere.ibm.com/webservices/} Server.generalException
2.	Domain Name System (DNS) is down	HTTP transport error: java.net.UnknownHostException: loony.xai12.ai	faultCode: {http://websphere.ibm.com/webservices/} Server.generalException
3.	Lost of packet with client request or service response	<i>Waiting for response during too much time (more than 2 hours) without exception</i>	faultCode: {http://websphere.ibm.com/webservices/} Server.generalException faultString: java.io. InterruptedIOException: Read timed out

№	Type of error/failure	Exception	
		Sun Microsystems WS Toolkit	IBM WS Toolkit (WSDK)
4.	Remote host unavailable (off-line)	HTTP Status-Code 404: Not Found - /WS/ WSCalc	faultCode: {http://websphere.ibm.com/webservices/} HTTP faultString: (404)Not Found
5.	Application Server (IBM WebSphere or SJS AppServer) is down	HTTP transport error: java.net.ConnectException: Connection refused: connect	faultCode: {http://websphere.ibm.com/webservices/} Server.generalException
6.	Suspension of Web Service during transaction	<i>Waiting for response during too much time (more than 2 hours) without exception</i>	faultCode: {http://websphere.ibm.com/webservices/} Server.generalException faultString: java.io. Interrupted IOException:Read timed out
7.	System run-time error (“Division by Zero”)	java.rmi.ServerException: JAXRPC.TIE.04: Internal Server Error (JAXRPCTIE01: caught exception while handling request: java.lang. ArithmeticException: / by zero)	faultCode: {http://websphere.ibm.com/webservices/} Server.generalException faultString: java.lang.ArithmeticException: / by zero
8.	Calculation run-time error (“Operand Type Mismatch”)	java.rmi.ServerException: JAXRPC.TIE.04: Internal Server Error (JAXRPCTIE01: java.lang. NumberFormatException: For input string: "578ER")	faultCode: {http://websphere.ibm.com/webservices/} Server.generalException faultString: java.lang.Number FormatException: 5ER
9.	Application error causing user-defined exception	java.rmi.RemoteException: ai.xai12.loony.exception. UserException	faultCode: {http://websphere.ibm.com/webservices/} Server.generalException faultString: (13)UserException
10.	Error in Target Name Space	java.rmi.RemoteException: JAXRPCTIE01: unrecognized operation: {urn:WSStREring/wsdl} gluingstring	<i>OK - Correct output without exception</i>
11.	Error in Web Service name	<i>OK - Correct output without exception</i>	<i>OK - Correct output without exception</i>
12.	Error in service port name	<i>OK - Correct output without exception</i>	<i>OK - Correct output without exception</i>
13.	Error in service operation's name	java.rmi.RemoteException: JAXRPCTIE01: unrecognized operation: {urn:WSNumeric/wsdl}getMRTult	faultCode: {http://websphere.ibm.com/ webservices/}Server.generalException faultString: WWS3277E: Error: No such operation 'getMRTult'
14.	Output parameter type mismatch	java.rmi.RemoteException: JAXRPCTIE01: unexpected element type: expected= {http://www.w3.org/ 2001/XMLSchema}string, actual= {http://schemas.xmlsoap.org/soap/ encoding/}int	<i>OK - Correct output without exception</i>
15.	Input parameter type mismatch	java.rmi.RemoteException: JAXRPCTIE01: unexpected element type: expected= {http://schemas. xmlsoap.org/soap/encoding/}int, actual={http://www.w3.org/2001/ XMLSchema}string	faultCode: {http://websphere.ibm.com/webservices/} Server.generalException faultString: org.xml.sax.SAXException: Bad types
16.	Error in name of input parameter	java.rmi.RemoteException: JAXRPCTIE01: unexpected element name: expected=Integer_2, actual=IntegERer_1	<i>OK - Correct output without exception</i>

№	Type of error/failure	Exception	
		Sun Microsystems WS Toolkit	IBM WS Toolkit (WSDK)
17.	Mismatching of number of input service parameters	java.rmi.RemoteException: JAXRPCTIE01: deserialization error: unexpected XML reader state. expected: END but found: START: Integer_2	faultCode: {http://websphere.ibm.com/webservices/} Server.generalException faultString: java.lang.NullPointerException
18.	Web Service style mismatching (“Rpc” instead of “Document”)	java.rmi.RemoteException: JAXRPCTIE01: unexpected encoding style: expected=http://schemas.xmlsoap.org/ soap/encoding/	faultCode: {http://schemas.xmlsoap.org/soap/envelope/} Client faultString: JAXRPCTIE01: caught exception while handling request: unexpected encoding style: expected= http://schemas.xmlsoap.org/ soap/encoding/

As it was discovered, some errors and failures cause the same exception so we were not able to define the precise exception cause. There are several groups of such errors and failures (see Table 2): 1 and 2 (Sun); 3 and 6 (Sun); 4 and 5 (Sun); 1, 2 and 5 (IBM); 3 and 6 (IBM).

Some client-side binding errors (11 – “Error in Web Service name”, 12 – “Error in service port name”) neither raise exceptions nor affect the service output. This happens because the Web Service is actually invoked by the address location, whereas the service name and the port name are used only as supplementary information.

Moreover, the Web Service, developed by using IBM WSDK and deployed on the IBM WebSphere application server, tolerates such binding errors inside: 10 - “Error in Target Name Space”, 14 - “Output parameter type mismatch”, and 16 - “Error in name of input parameter”.

These abilities are provided by features of the WSDL description and a built-in function of automatic type conversion. Errors in the name of the input parameter are tolerated because checking the order of parameters has a priority over the coincidence of parameter names in the IBM implementation of Web Services.

5.2. Exception propagation and performance analysis

Table 2 includes exceptions raised at the top level on client’s side. However, a particular exception can be wrapped dozens of times before it finally propagates to the top. This process takes time and significantly reduces performance of exception handling in service-oriented architecture. Examples of the stack trace in SOA-architecture, developed by using different toolkits (NetBeans IDE and Eclipse IDE/IBM WSDK) and deployed on different application servers (SJS AppServer and IBM WebSphere), are given below (Fig. 7 – 11).

```

HTTP Status-Code 404: Not Found - /WS/WSCalc
at com.sun.xml.rpc.client.http.HttpClientTransport.
    checkResponseCode(HttpClientTransport.java:302)
at com.sun.xml.rpc.client.http.HttpClientTransport.
    ectForResponse(HttpClientTransport.java:252)
at com.sun.xml.rpc.client.http.HttpClientTransport.
    invoke(HttpClientTransport.java:88)
at com.sun.xml.rpc.client.StreamingSender._send(StreamingSender.java:92)
at com.sun.xml.rpc.client.dii.CallInvokerImpl.
    doInvoke(CallInvokerImpl.java:79)
at com.sun.xml.rpc.client.dii.BasicCall.invoke(BasicCall.java:482)
at ai.loony.xai12.wstest.InvokeWS.invoke(InvokeWS.java:125)
at ai.loony.xai12.wstest.InvokeWS.invokeByVector(InvokeWS.java:75)
at wstest.Main.main(Main.java:42)

```

Figure 7. Stack trace of network failure “Host unavailable”, raised in client application developed in NetBeans IDE by using JAX-RPC implementation at Sun Microsystems.

```

WebServicesFault
  faultCode: {http://websphere.ibm.com/webservices/}HTTP
  faultString: (404)Not Found
  faultActor: null
  faultDetail: null: WSWS3192E: Error: return code: 404
Error 404: File not found: services/WSCalc (404)Not Found
  at com.ibm.ws.webservices.engine.transport.http.HTTPSender.
    readFromSocket(HTTPSender.java:773)
  at com.ibm.ws.webservices.engine.transport.http.
    HTTPSender.invoke(HTTPSender.java:157)
  at com.ibm.ws.webservices.engine.PivotHandlerWrapper.
    invoke(PivotHandlerWrapper.java:217)
  at com.ibm.ws.webservices.engine.WebServicesEngine.
    invoke(WebServicesEngine.java:258)
  at com.ibm.ws.webservices.engine.client.Connection.
    invokeEngine(Connection.java:680)
  at com.ibm.ws.webservices.engine.client.Connection.
    invoke(Connection.java:604)
  at com.ibm.ws.webservices.engine.client.Connection.
    invoke(Connection.java:434)
  at com.ibm.ws.webservices.engine.client.Call.invoke(Call.java:1381)
  at ai.loony.xail2.testwsibm.InvoceWS.invoice(InvoceWS.java:127)
  at ai.loony.xail2.testwsibm.InvoceWS.invoiceByVector(InvoceWS.java:77)
  at testwsibm.Main.main(Main.java:44)

```

Figure 8. Stack trace of network failure “Host unavailable”, raised in client application developed in Eclipse IDE by using IBM WSDK.

```

java.rmi.ServerException: JAXRPC.TIE.04: Internal Server Error (JAXRPC.TIE.01:
caught exception while handling request: java.lang.ArithmeticException: / by zero)
  at com.sun.xml.rpc.client.StreamingSender._raiseFault(StreamingSender.java:497)
  at com.sun.xml.rpc.client.StreamingSender._send(StreamingSender.java:294)
  at com.sun.xml.rpc.client.dii.CallInvokerImpl.
    doInvoke(CallInvokerImpl.java:79)
  at com.sun.xml.rpc.client.dii.BasicCall.invoke(BasicCall.java:482)
  at ai.loony.xail2.wstest.InvoceWS.invoice(InvoceWS.java:125)
  at ai.loony.xail2.wstest.InvoceWS.invoiceByVector(InvoceWS.java:75)
  at wstest.Main.main(Main.java:42)

```

Figure 9. Stack trace of system service failure “Division by Zero”, raised in service application developed by using JAX-RPC implementation at Sun Microsystems and deployed on SJS AppServer.

```

WebServicesFault
  faultCode: {http://websphere.ibm.com/webservices/}Server.generalException
  faultString: java.lang.ArithmeticException: / by zero
  faultActor: null
  faultDetail:
stackTrace: <![CDATA[java.lang.ArithmeticException: / by zero
  at ai.xail2.loony.ws.WSCalc.getMul(Mult.java:20)
  at java.lang.reflect.Method.invoke(Native Method)
  at com.ibm.ws.webservices.engine.providers.java.RPCProvider.
    invokeMethod(RPCProvider.java:421)
  at com.ibm.ws.webservices.engine.providers.java.
    RPCProvider.processRequestResponse(RPCProvider.java:313)
  at com.ibm.ws.webservices.engine.providers.java.
    RPCProvider.processMessage(RPCProvider.java:262)
  at com.ibm.ws.webservices.engine.providers.java.
    JavaProvider.invoke(JavaProvider.java:289)
  at com.ibm.ws.webservices.engine.PivotHandlerWrapper.
    invoke(PivotHandlerWrapper.java:217)
  at com.ibm.ws.webservices.engine.handlers.WrappedHandler.
    invoke(WrappedHandler.java:61)
  at com.ibm.ws.webservices.engine.PivotHandlerWrapper.
    invoke(PivotHandlerWrapper.java:217)

```

```

at com.ibm.ws.webservices.engine.PivotHandlerWrapper.
    invoke(PivotHandlerWrapper.java:217)
at com.ibm.ws.webservices.engine.WebServicesEngine.
    invoke(WebServicesEngine.java:258)
at com.ibm.ws.webservices.engine.transport.http.
    WebServicesServlet.doPost(WebServicesServlet.java:835)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:760)
at com.ibm.ws.webservices.engine.transport.http.
    WebServicesServletBase.service(WebServicesServletBase.java:341)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:853)
at com.ibm.ws.webcontainer.servlet.StrictServletInstance.
    doService(StrictServletInstance.java:110)
at com.ibm.ws.webcontainer.servlet.StrictLifecycleServlet.
    _service(StrictLifecycleServlet.java:174)
at com.ibm.ws.webcontainer.servlet.IdleServletState.
    service(StrictLifecycleServlet.java:313)
at com.ibm.ws.webcontainer.servlet.StrictLifecycleServlet.
    service(StrictLifecycleServlet.java:116)
at com.ibm.ws.webcontainer.servlet.ServletInstance.
    service(ServletInstance.java:283)
at com.ibm.ws.webcontainer.servlet.ValidServletReferenceState.
    dispatch(ValidServletReferenceState.java:42)
at com.ibm.ws.webcontainer.servlet.ServletInstanceReference.
    dispatch(ServletInstanceReference.java:40)
at com.ibm.ws.webcontainer.webapp.WebAppRequestDispatcher.
    handleWebAppDispatch(WebAppRequestDispatcher.java:948)
at com.ibm.ws.webcontainer.webapp.WebAppRequestDispatcher.
    dispatch(WebAppRequestDispatcher.java:530)
at com.ibm.ws.webcontainer.webapp.WebAppRequestDispatcher.
    forward(WebAppRequestDispatcher.java:176)
at com.ibm.ws.webcontainer.srt.WebAppInvoker.
    doForward(WebAppInvoker.java:79)
at com.ibm.ws.webcontainer.srt.WebAppInvoker.
    handleInvocationHook(WebAppInvoker.java:201)
at com.ibm.ws.webcontainer.cache.invocation.CachedInvocation.
    handleInvocation(CachedInvocation.java:71)
at com.ibm.ws.webcontainer.cache.invocation.CacheableInvocationContext.
    invoke(CacheableInvocationContext.java:114)
at com.ibm.ws.webcontainer.srp.ServletRequestProcessor.
    dispatchByURI(ServletRequestProcessor.java:186)
at com.ibm.ws.webcontainer.oselister.OSELListenerDispatcher.
    service(OSELListener.java:334)
at com.ibm.ws.webcontainer.http.HttpConnection.
    handleRequest(HttpConnection.java:56)
at com.ibm.ws.http.HttpConnection.
    readAndHandleRequest(HttpConnection.java:610)
at com.ibm.ws.http.HttpConnection.run(HttpConnection.java:431)
at com.ibm.ws.util.ThreadPool$Worker.run(ThreadPool.java(Compiled Code))
]]>
java.lang.ArithmeticException: / by zero
at com.ibm.ws.webservices.engine.xmlsoap.builders.
    SOAPFaultBuilder.createFault(SOAPFaultBuilder.java:227)
at com.ibm.ws.webservices.engine.xmlsoap.builders.
    SOAPFaultBuilder.endElement(SOAPFaultBuilder.java:155)
at com.ibm.ws.webservices.engine.events.P2DConverter.
    endElement(P2DConverter.java:405)
at org.apache.xerces.parsers.AbstractSAXParser.
    endElement(AbstractSAXParser.java:569)
at org.apache.xerces.impl.XMLNamespaceBinder.
    handleEndElement(XMLNamespaceBinder.java:853)
at org.apache.xerces.impl.XMLNamespaceBinder.
    endElement(XMLNamespaceBinder.java:643)
at org.apache.xerces.impl.dtd.XMLDTDValidator.
    handleEndElement(XMLDTDValidator.java:3003)
at org.apache.xerces.impl.dtd.XMLDTDValidator.endElement(XMLDTDValidator.java:931)
at org.apache.xerces.impl.XMLDocumentFragmentScannerImpl.
    handleEndElement(XMLDocumentFragmentScannerImpl.java:1147)
at org.apache.xerces.impl.XMLDocumentFragmentScannerImpl.
    scanEndElement(XMLDocumentFragmentScannerImpl.java:988)
at

```

```

org.apache.xerces.impl.XMLDocumentFragmentScannerImpl$FragmentContentDispatcher.
    dispatch(XMLDocumentFragmentScannerImpl.java:1448)
at org.apache.xerces.impl.XMLDocumentFragmentScannerImpl.
    scanDocument(XMLDocumentFragmentScannerImpl.java:333)
at org.apache.xerces.parsers.StandardParserConfiguration.
    parse(StandardParserConfiguration.java:525)
at org.apache.xerces.parsers.StandardParserConfiguration.
    parse(StandardParserConfiguration.java:581)
at org.apache.xerces.parsers.XMLParser.parse(XMLParser.java:147)
at org.apache.xerces.parsers.AbstractSAXParser.parse(AbstractSAXParser.java:1158)
at javax.xml.parsers.SAXParser.parse(Unknown Source)
at com.ibm.ws.webservices.engine.encoding.DeserializationContextImpl.
    parse(DeserializationContextImpl.java:256)
at com.ibm.ws.webservices.engine.SOAPPart.getAsSOAPEnvelope(SOAPPart.java:698)
at com.ibm.ws.webservices.engine.Message.getSOAPEnvelope(Message.java:440)
at com.ibm.ws.webservices.engine.client.Connection.
    invokeEngine(Connection.java:700)
at com.ibm.ws.webservices.engine.client.Connection.invoke(Connection.java:604)
at com.ibm.ws.webservices.engine.client.Connection.invoke(Connection.java:434)
at com.ibm.ws.webservices.engine.client.Call.invoke(Call.java:1381)
at ai.loony.xai12.testwsibm.InvoceWS.invoce(InvoceWS.java:127)
at ai.loony.xai12.testwsibm.InvoceWS.invoceByVector(InvoceWS.java:77)
at testwsibm.Main.main(Main.java:44)

```

Figure 10. Stack trace of system service failure “Division by Zero”, raised in service application developed in Eclipse IDE by using IBM WSDK and deployed on IBM WebSphere.

```

...
catch (Exception e) {
    e.printStackTrace();
}

```

Figure 11. Example of client Java-code that prints the stack trace.

The results of exception propagation and performance analysis are represented in Table 3. This table includes a number of exceptions stack traces (length of exceptions propagation chains) and propagation delays (min, max and average values). The performance of exception raising mechanisms has been monitored at the university LAN on heterogeneous server platforms.

The first row of the table corresponds to the correct service output without any exceptions. The rows, marked in bold, correspond to the cases of correct service outputs without exceptions in spite of seeded errors. It is evident from the table, that exceptions propagation delay is several times greater than working time.

As can be seen from Table 3, the IBM implementation of the Web Service has almost twice as good a performance as that of the service implemented in the Sun technology. However, exception propagation delay of the Web Service, developed with NetBeans IDE using JAX-RPC implementation of Sun Microsystems, was two times shorter than the delay we had when we used IBM WSDK. It explains the fact that exceptions propagation chain in the IBM implementation of the Web Service is, usually, much longer (compare Fig. 7 and 8).

It is unacceptable that the service client, developed using the Sun WS toolkit, does not raise any exception even after more than 2 hours of waiting in cases of service suspension or packets loss.

Analysis of the exception stack trace and propagation delay can help in identifying the source of the exception.

For example, failures 1 - “Network connection break-off” and 2 - “Domain Name System (DNS) is down” raise the same top-level exception “*HTTP transport error: java.net.UnknownHostException: loony.xai12.ai*”. However, if we use Sun WS toolkit we will be able to distinguish these failures by comparing numbers of the stack traces (38 vs. 28).

If we use IBM WSDK we are able to distinguish between failure 5 – “Application Server is down” and failures 1 and 2 on the basis of analysis of the exception propagation delay (it is one order greater).

Table 3. Performance analysis of exceptions propagation mechanism in SOA-architecture depending on the Web Services development toolkit used.

WS Development Toolkit		NetBeans IDE (Sun)			IBM WSDK				
№	Type of error/failure	no of stack traces	exception's propagation delay, ms			no of stack traces	exception's propagation delay, ms		
			min	max	av.		min	max	av.
	Without Error/Failure	0	40	210	95	0	15	120	45
1.	Network connection break-off	38	10	30	23	16	10	40	28
2.	Domain Name System (DNS) is down	28	16	32	27	16	15	47	34
3.	Lost of packet with client request or service response	-		>7200 000		15	3005 03	3006 61	3006 22
4.	Remote host unavailable (off-line)	9	110	750	387	11	120	580	350
5.	Application Server is down	9	70	456	259	16	100	550	287
6.	Suspension of Web Service during transaction (getting into a loop)	-		>7200 000		15	3005 33	3007 71	3006 42
7.	System run-time error (“Division by Zero”)	7	90	621	250	62	120	551	401
8.	Calculation run-time error (“Operand Type Mismatch”)	4	90	170	145	63	130	581	324
9.	Application error causing user-defined exception	4	100	215	175	61	150	701	366
10.	Error in Target Name Space	4	100	281	180	0	10	105	38
11.	Error in Web Service name	0	40	120	80	0	10	125	41
12.	Error in service port name	0	30	185	85	0	15	137	53
13.	Error in service operation name	4	90	270	150	58	190	511	380
14.	Output parameter type mismatch	14	80	198	160	0	15	134	48
15.	Input parameter type mismatch	4	80	190	150	76	90	761	305
16.	Error in name of input parameter	4	70	201	141	0	10	150	47
17.	Mismatching of number of input service parameters	4	80	270	160	61	130	681	350
18.	Web Service style mismatching	4	70	350	187	58	90	541	298

Figure 12 shows the classification of errors and failures taking into consideration their sources, consequences and the influence on the exceptions raised by them.

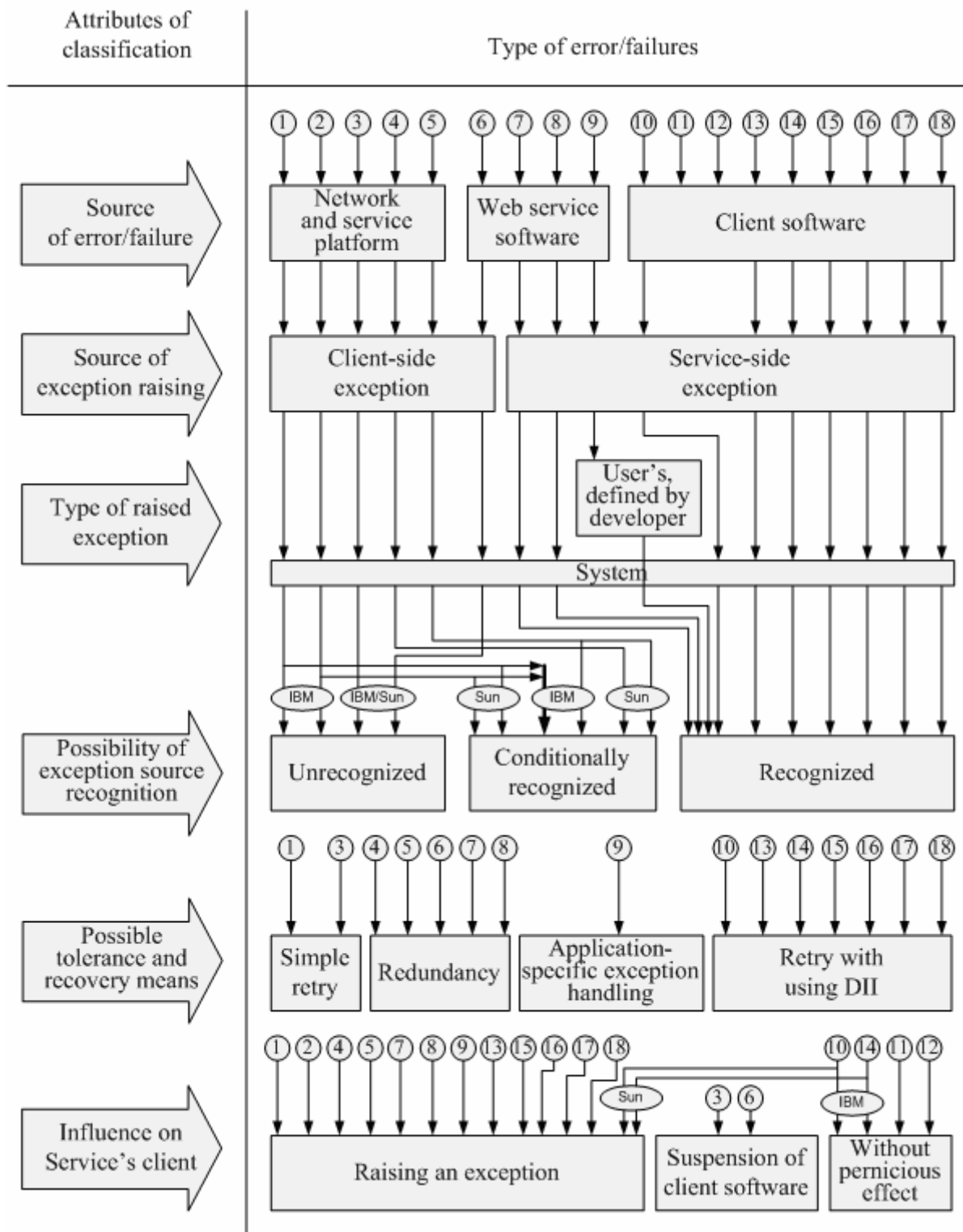


Figure 12. Classification of the SOA-specific errors and failures.

6. Exception Handling and Fault-tolerant Means

Paper [1] discusses two main classes of error recovery, applicable in service-oriented architecture: *backward* (based on rolling system components back to the previous correct state) and *forward error recovery* (which involves transforming system components into any correct state). The latter is usually application-specific and relies on the *exception handling mechanisms*.

Knowing about the exact reasons and sources of exceptions, allows us to try to tolerate some of them. Several types of errors resulting in exceptions can be effectively handled on the client side, whereas other ones should be handled on the service side. Exceptions handling of the client side errors in early binding procedures may include retry with the help of dynamic invocation interface (DII). Transient network failures can be tolerated by simple retry. In other cases redundancy and majority voting should be used.

However, exception handling complicates implementation of the Web Service and client application. Therefore, most of the modern forward error recovery means in SOA are based on using middleware.

For example, when a flight booking Web Service returns an exception, alternative transport means can be proposed to the user as a replacement of the flight (e.g., a train ticket to a nearby city and renting a car). This process is illustrated on the Fig. 13.

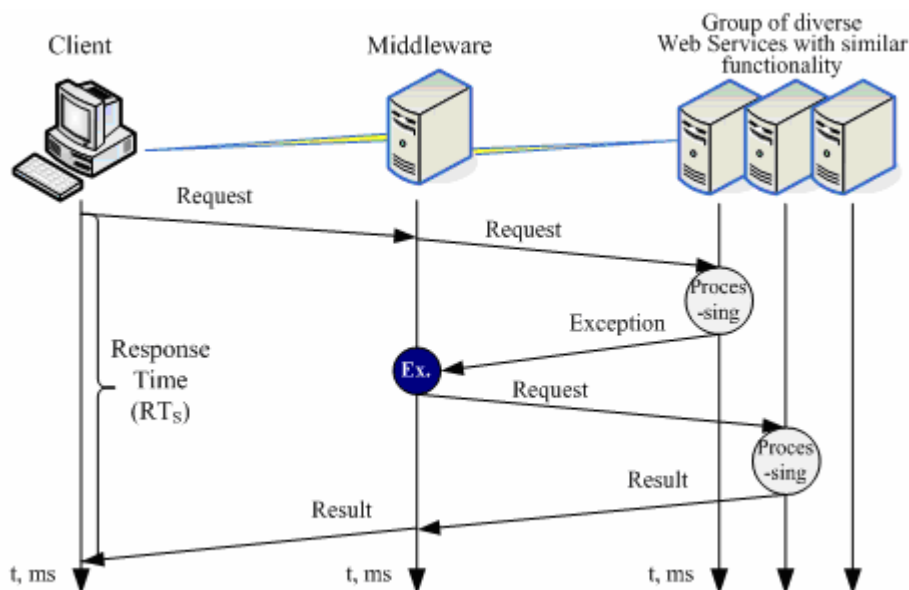


Figure 13. Exception handling with forward error recovery.

The diverse Web Services are executed sequentially (the order of execution can be predefined or randomly chosen). The subsequent releases are only executed if the response, received from the previous releases, is evidently incorrect (service reports an exception).

We can apply an enhanced forward errors recovery procedure which provides better performance on the basis of concurrent requests processing by several diverse Web Services with similar functionality (Fig. 14). All available diverse Web Services are executed concurrently and the fastest response is returned to the client. Thus, the architecture reduces average service response time.

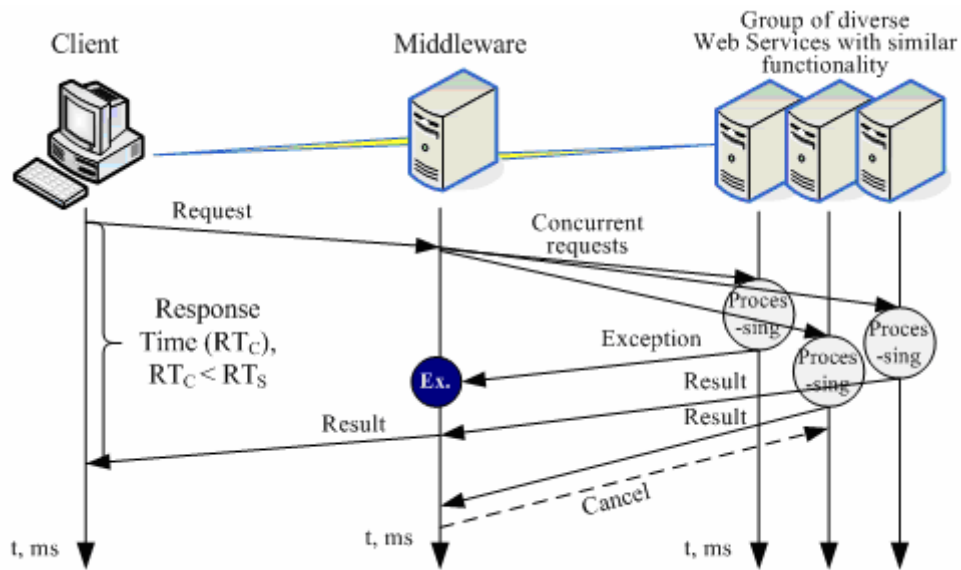


Figure 14. Exception handling with concurrent forward error recovery.

Conclusion

Exception handling is widely used as the basis of forward error recovery procedure in service-oriented architecture. Effectiveness of exception handling depends on the features of raising exceptions and propagation mechanisms.

In our work we have experimented with Web Services, implemented by use of different development kits: 1) JAX-RPC implementation at Sun Microsystems and 2) IBM WebSphere Software Developer Kit for Web Services. We have performed compatibility analysis of exception propagation mechanisms and performance implications.

Modern technologies of fault-tolerance and dependability improvement in service-oriented architectures are middleware-oriented and use natural redundancy and diversity of existing Web Services with similar functionality.

Acknowledgments

Alexander Romanovsky is partially supported by the EC RODIN project.

References

1. F. Tartanoglu, V. Issarny, A. Romanovsky, N. Levy. *Coordinated Forward Error Recovery for Composite Web Services* / Proc. of 22nd Symposium on Reliable Distributed Systems (SRDS), Florence, Italy, pp. 167-176 (2003).
2. Pat. P.W. Chan, Michael R. Lyu, Miroslaw Malek. *Making Services Fault Tolerant* / In D. Penkler, M. Reitenspiess, and F. Tam (Eds.): *Service Availability*, International Service Availability Symposium (ISAS'2006), LNCS 4328. Berlin, Heidelberg: Springer-Verlag, pp. 43–61 (2006).
3. A. Gorbenko, V. Kharchenko, P. Popov, A. Romanovsky. *Dependable Composite Web Services with Components Upgraded Online* / In R. de Lemos et al. (Eds.): *Architecting Dependable Systems III*, LNCS 3549. Berlin, Heidelberg: Springer-Verlag, pp. 92 – 121 (2005).
4. A. Gorbenko, V. Kharchenko, A. Romanovsky. *On Composing Dependable Web Services Using Undependable Web Components* / *International Journal of Simulation and Process Modelling (IJSPM)*, 18 pages (accepted for publishing in 2006).
5. P. Townend, P. Groth, J. Xu. *A Provenance-Aware Weighted Fault Tolerance Scheme for Service-Based Applications* / Proc. of 8th IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC), pp. 258-266 (2005).
6. W. T. Tsai, Y. Chen, R. Paul, N. Liao, and H. Huang. *Cooperative and Group Testing in Verification of Dynamic Composite Web Services* / Proc. Workshops and Fast Abstracts of 28th Annual International Computer Software and Applications Conference (COMPSAC), pp. 170-173 (2004).
7. M. Brambilla, C. Tziviskou. *Fundamentals Of Exception Handling Within Workflow-Based Web Applications* / *Journal of Web Engineering (JWE)*, Vol. 4, Issue 1, pp. 38-56 (2005).
8. M. Brambilla, S. Ceri, S. Comai, C. Tziviskou. *Exception Handling in Workflow-Driven Web Applications* / Proc. of WWW International Conference, Chiba (Japan), pp. 170 – 179 (2005).
9. J. Miller, A. Sheth, K. Kochut and Z. Luo. *Recovery Issues in Web-Based Workflow* / Proc. of 12th International Conference on Computer Applications in Industry and Engineering (CAINE), Atlanta, Georgia, pp. 101-105 (1999).
10. A. Gorbenko, V. Kharchenko, O. Tarasyuk. *Vertical and Horizontal Composition in Service-Oriented Architecture* / Proc. of 5th International Conference “Internet-Education-Science” (IES'2006), Vinnytsia (Ukraine), pp. 328-331 (2006).

APPENDIX A. Client-side program code invoking Web Service

```
// The instance method invoking Web Service by using DII
// (Dynamic Invocation Interface)
// Input parameters:
// String aLog - the name of the log file
// String aMessage - the message to be written in the log file
// String aTargetEndpointAddress - the URI of the Web Service
// String aTargetNameSpace - the Target Name Space
// String aServiceName - the service's name
// String aPortName - service's port name
// String aOperationName - the service's operation name
// QName aReturnType - the type of result
// QName aINType1 - the type of the 1st input parameter
// QName aINType2 - the type of the 2nd input parameter
// String aINName1 - the name of the 1st input parameter
// String aINName2 - the name of the 2nd input parameter
// Object[] aValues - the set of input parameters' values

    public Object invoke(String aLog, String aMessage, String
aTargetEndpointAddress,
        String aTargetNameSpace, String aServiceName, String aPortName,
        String aOperationName, QName aReturnType,
        QName aINType1, QName aINType2, String aINName1, String
aINName2,
        Object[] aValues) {
    String ENCODING_STYLE_PROPERTY =
"javax.xml.rpc.encodingstyle.namespace.uri";
    String URI_ENCODING = "http://schemas.xmlsoap.org/soap/encoding/";

    ServiceFactory sf;
    Log log = new Log(aLog);
    Object title = null; // Object returning service response

    try {
        sf = ServiceFactory.newInstance();
        Service serv = sf.createService(new
QName(aTargetNameSpace, aServiceName));
        Call call = serv.createCall(new
QName(aTargetNameSpace, aPortName));
        call.setTargetEndpointAddress(aTargetEndpointAddress);
        call.setProperty(Call.SOAPACTION_USE_PROPERTY, new
Boolean(true));
        call.setProperty(Call.SOAPACTION_URI_PROPERTY, "");
        //call.setProperty(ENCODING_STYLE_PROPERTY, URI_ENCODING);
        call.setProperty(Call.ENCODINGSTYLE_URI_PROPERTY,
URI_ENCODING);
        call.setProperty(Call.OPERATION_STYLE_PROPERTY, "rpc");
        call.setReturnType(aReturnType);
        call.setOperationName(new
QName(aTargetNameSpace, aOperationName));
        call.addParameter(aINName1, aINType1, ParameterMode.IN);
        call.addParameter(aINName2, aINType2, ParameterMode.IN);
        SimpleDateFormat formatter =
            new SimpleDateFormat("yyyy.MM.dd hh:mm:ss:SSS");

        //Start of the Service Invocation
        Date dtStart = new Date();
        try {
            title = call.invoke(aValues);
```

```

        log.println("Service response: " + title.toString());
    } catch (RemoteException e1) {
        log.println("Invocation Error");
        log.println(aMessage);
        log.println(e1);
        System.out.println("Invocation Error");
    } catch (Exception e){
        log.println("Invocation Error");
        log.println(aMessage);
        log.println(e);
        System.out.println("Invocation Error");
    }
    //End of the Service Invocation
    Date dtFinish = new Date();
    sf = null;
    String strStart = formatter.format(dtStart);
    String strFinish = formatter.format(dtFinish);
    log.println(aMessage);
    log.println("The start time: " + strStart);
    String strDelta =
        new Long(dtFinish.getTime() - dtStart.getTime()).toString();
    log.println("The end time: " + strFinish);
    log.println("Service fetch time: " + strDelta);
    log.println("");
    return title;
} catch (ServiceException e) {
    log.println(aMessage);
    log.println(e);
    log.println("Client initiation error");
    log.println("");
}
return null;
}
}

```

APPENDIX B. Exceptions stack traces

FAILURE: Network connection break-off	
EXCEPTION STACK TRACE: Sun Microsystems WS Toolkit	EXCEPTION STACK TRACE: IBM WS Toolkit (WSDK)
<pre> HTTP transport error: java.net.UnknownHostException: loony.xai12.ai at com.sun.xml.rpc.client.http.HttpClientTransport.invoke(HttpClientTransport.java:140) at com.sun.xml.rpc.client.StreamingSender._send(StreamingSender.java:92) at com.sun.xml.rpc.client.dii.CallInvokerImpl.doInvoke(CallInvokerImpl.java:79) at com.sun.xml.rpc.client.dii.BasicCall.invoke(BasicCall.java:482) at ai.loony.xai12.wstest.InvoceWS.invoce(InvoceWS.java:125) at ai.loony.xai12.wstest.InvoceWS.invoceByVector(InvoceWS.java:75) at wstest.Main.main(Main.java:42) at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method) at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39) at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25) at java.lang.reflect.Method.invoke(Method.java:585) at com.intellij.rt.execution.application.AppMain.main(AppMain.java:86) CAUSE: java.net.UnknownHostException: loony.xai12.ai at java.net.PlainSocketImpl.connect(PlainSocketImpl.java:177) at java.net.Socket.connect(Socket.java:507) at java.net.Socket.connect(Socket.java:457) at sun.net.NetworkClient.doConnect(NetworkClient.java:157) at sun.net.www.http.HttpClient.openServer(HttpClient.java:365) at sun.net.www.http.HttpClient.openServer(HttpClient.java:477) at sun.net.www.http.HttpClient.<init>(HttpClient.java:214) at sun.net.www.http.HttpClient.New(HttpClient.java:287) at sun.net.www.http.HttpClient.New(HttpClient.java:299) at sun.net.www.protocol.http.HttpURLConnection.getNewHttpClient(HttpURLConnection.java:792) at sun.net.www.protocol.http.HttpURLConnection.plainConnect(HttpURLConnection.java:744) at sun.net.www.protocol.http.HttpURLConnection.connect(HttpURLConnection.java: </pre>	<pre> WebServicesFault faultCode: {http://websphere.ibm.com/webservices/}Server.generalException faultString: java.net.UnknownHostException: loony.xai12.ai faultActor: null faultDetail: stackTrace: java.net.UnknownHostException: loony.xai12.ai at java.net.InetAddress.getAllByName0(InetAddress.java:591) at java.net.InetAddress.getAllByName0(InetAddress.java:560) at java.net.InetAddress.getByName(InetAddress.java:469) at java.net.Socket.&lt;init&gt;(Socket.java:121) at com.ibm.ws.webservices.engine.components.net.DefaultSocketFactory.create(DefaultSocketFactory.java:145) at com.ibm.ws.webservices.engine.transport.http.HTTPSender.getSocket(HTTPSender.java:225) at com.ibm.ws.webservices.engine.transport.http.HTTPSender.invoke(HTTPSender.java:143) at com.ibm.ws.webservices.engine.PivotHandlerWrapper.invoke(PivotHandlerWrapper.java:217) at com.ibm.ws.webservices.engine.WebServicesEngine.invoke(WebServicesEngine.java:258) at com.ibm.ws.webservices.engine.client.Connection.invokeEngine(Connection.java:680) at com.ibm.ws.webservices.engine.client.Connection.invoke(Connection.java:604) at com.ibm.ws.webservices.engine.client.Connection.invoke(Connection.java:434) at com.ibm.ws.webservices.engine.client.Call.invoke(Call.java:1381) at ai.loony.xai12.testwsibm.InvoceWS.invoce(InvoceWS.java:127) at ai.loony.xai12.testwsibm.InvoceWS.invoceByVector(InvoceWS.java:77) at testwsibm.Main.main(Main.java:44) </pre>

<pre> 669) at sun.net.www.protocol.http.HttpURLConnection.getOutputStream(HttpURLConnection.java:836) at com.sun.xml.rpc.client.http.HttpClientTransport.writeMessageToConnection(HttpClientTransport.java:382) at com.sun.xml.rpc.client.http.HttpClientTransport.invoke(HttpClientTransport.java:86)) at com.sun.xml.rpc.client.StreamingSender._send(StreamingSender.java:92) at com.sun.xml.rpc.client.dii.CallInvokerImpl.doInvoke(CallInvokerImpl.java:79) at com.sun.xml.rpc.client.dii.BasicCall.invoke(BasicCall.java:482) at ai.loony.xai12.wstest.InvoceWS.invoce(InvoceWS.java:125) at ai.loony.xai12.wstest.InvoceWS.invoceByVector(InvoceWS.java:75) at wstest.Main.main(Main.java:42) at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method) at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39) at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25) at java.lang.reflect.Method.invoke(Method.java:585) at com.intellij.rt.execution.application.AppMain.main(AppMain.java:86) </pre>	
FAILURE: Domain Name System (DNS) is down	
EXCEPTION STACK TRACE: Sun Microsystems WS Toolkit	EXCEPTION STACK TRACE: IBM WS Toolkit (WSDK)
See exception stack trace of "Network connection break-off" failure	See exception stack trace of "Network connection break-off" failure
FAILURE: Lost of packet with client request or service response	
EXCEPTION STACK TRACE: Sun Microsystems WS Toolkit	EXCEPTION STACK TRACE: IBM WS Toolkit (WSDK)
<i>Waiting for response during too much time (more than 2 hours) without exception</i>	<pre> WebServicesFault faultCode: {http://websphere.ibm.com/webservices/}Server.generalException faultString: java.io.InterruptedIOException: Read timed out faultActor: null faultDetail: stackTrace: java.io.InterruptedIOException: Read timed out at java.net.SocketInputStream.socketRead(Native Method) at java.net.SocketInputStream.read(SocketInputStream.java:113) at java.io.BufferedInputStream.fill(BufferedInputStream.java:202) at java.io.BufferedInputStream.read(BufferedInputStream.java(Compiled Code)) at com.ibm.ws.webservices.engine.transport.http.HTTPSender.readHeadersFromSocket(HTTPSender.java(Compiled Code)) at com.ibm.ws.webservices.engine.transport.http.HTTPSender.invoke(HTTPSender.java:156) at com.ibm.ws.webservices.engine.PivotHandlerWrapper.invoke(PivotHandlerWrapper.java:217) at com.ibm.ws.webservices.engine.WebServicesEngine.invoke(WebServicesEngine.java:258) at com.ibm.ws.webservices.engine.client.Connection.invokeEngine(Connection.java:680) at com.ibm.ws.webservices.engine.client.Connection.invoke(Connection.java:604) </pre>

	<pre> at com.ibm.ws.webservices.engine.client.Connection.invoke(Connection.java:434) at com.ibm.ws.webservices.engine.client.Call.invoke(Call.java:1381) at ai.loony.xai12.testwsibm.InvoiceWS.invoice(InvoiceWS.java:127) at ai.loony.xai12.testwsibm.InvoiceWS.invoiceByVector(InvoiceWS.java:77) at testwsibm.Main.main(Main.java:44) </pre>
FAILURE: Host unavailable (off-line)	
EXCEPTION STACK TRACE: Sun Microsystems WS Toolkit	EXCEPTION STACK TRACE: IBM WS Toolkit (WSDK)
<pre> HTTP Status-Code 404: Not Found - /WS/WSNumeric at com.sun.xml.rpc.client.http.HttpClientTransport.checkResponseCode(HttpClientTransport.java:302) at com.sun.xml.rpc.client.http.HttpClientTransport.connectForResponse(HttpClientTransport.java:252) at com.sun.xml.rpc.client.http.HttpClientTransport.invoke(HttpClientTransport.java:88) at com.sun.xml.rpc.client.StreamingSender._send(StreamingSender.java:92) at com.sun.xml.rpc.client.dii.CallInvokerImpl.doInvoke(CallInvokerImpl.java:79) at com.sun.xml.rpc.client.dii.BasicCall.invoke(BasicCall.java:482) at ai.loony.xai12.wstest.InvoiceWS.invoice(InvoiceWS.java:125) at ai.loony.xai12.wstest.InvoiceWS.invoiceByVector(InvoiceWS.java:75) at wstest.Main.main(Main.java:42) </pre>	<pre> WebServicesFault faultCode: {http://websphere.ibm.com/webservices/}HTTP faultString: (404)Not Found faultActor: null faultDetail: null: WSW3192E: Error: return code: 404 Error 404: File not found: services/Mult87 (404)Not Found at com.ibm.ws.webservices.engine.transport.http.HTTPSender.readFromSocket(HTTPSender.java:773) at com.ibm.ws.webservices.engine.transport.http.HTTPSender.invoke(HTTPSender.java:157) at com.ibm.ws.webservices.engine.PivotHandlerWrapper.invoke(PivotHandlerWrapper.java:217) at com.ibm.ws.webservices.engine.WebServicesEngine.invoke(WebServicesEngine.java:258) at com.ibm.ws.webservices.engine.client.Connection.invokeEngine(Connection.java:680) at com.ibm.ws.webservices.engine.client.Connection.invoke(Connection.java:604) at com.ibm.ws.webservices.engine.client.Connection.invoke(Connection.java:434) at com.ibm.ws.webservices.engine.client.Call.invoke(Call.java:1381) at ai.loony.xai12.testwsibm.InvoiceWS.invoice(InvoiceWS.java:127) at ai.loony.xai12.testwsibm.InvoiceWS.invoiceByVector(InvoiceWS.java:77) at testwsibm.Main.main(Main.java:44) </pre>
FAILURE: Application Server is down	
EXCEPTION STACK TRACE: Sun Microsystems WS Toolkit	EXCEPTION STACK TRACE: IBM WS Toolkit (WSDK)
<pre> HTTP transport error: java.net.ConnectException: Connection refused: connect at com.sun.xml.rpc.client.http.HttpClientTransport.invoke(HttpClientTransport.java:140) at com.sun.xml.rpc.client.StreamingSender._send(StreamingSender.java:92) at com.sun.xml.rpc.client.dii.CallInvokerImpl.doInvoke(CallInvokerImpl.java:79) at com.sun.xml.rpc.client.dii.BasicCall.invoke(BasicCall.java:482) at ai.loony.xai12.wstest.InvoiceWS.invoice(InvoiceWS.java:125) at ai.loony.xai12.wstest.InvoiceWS.invoiceByVector(InvoiceWS.java:75) at wstest.Main.main(Main.java:45) CAUSE: java.net.ConnectException: Connection refused: connect at java.net.PlainSocketImpl.socketConnect(Native Method) at java.net.PlainSocketImpl.doConnect(PlainSocketImpl.java:333) at java.net.PlainSocketImpl.connectToAddress(PlainSocketImpl.java:195) at java.net.PlainSocketImpl.connect(PlainSocketImpl.java:182) </pre>	<pre> See exception stack trace of "Network connection break-off" failure </pre>

<pre> at java.net.Socket.connect(Socket.java:507) at java.net.Socket.connect(Socket.java:457) at sun.net.NetworkClient.doConnect(NetworkClient.java:157) at sun.net.www.http.HttpClient.openServer(HttpClient.java:365) at sun.net.www.http.HttpClient.openServer(HttpClient.java:477) at sun.net.www.http.HttpClient.<init>(HttpClient.java:214) at sun.net.www.http.HttpClient.New(HttpClient.java:287) at sun.net.www.http.HttpClient.New(HttpClient.java:299) at sun.net.www.protocol.http.HttpURLConnection.getNewHttpClient(HttpURLConnection.java:792) at sun.net.www.protocol.http.HttpURLConnection.plainConnect(HttpURLConnection.java:744) at sun.net.www.protocol.http.HttpURLConnection.connect(HttpURLConnection.java:669) at sun.net.www.protocol.http.HttpURLConnection.getOutputStream(HttpURLConnection.java:836) at com.sun.xml.rpc.client.http.HttpClientTransport.sendMessageToConnection(HttpClientTransport.java:382) at com.sun.xml.rpc.client.http.HttpClientTransport.invoke(HttpClientTransport.java:86)) at com.sun.xml.rpc.client.StreamingSender._send(StreamingSender.java:92) at com.sun.xml.rpc.client.dii.CallInvokerImpl.doInvoke(CallInvokerImpl.java:79) at com.sun.xml.rpc.client.dii.BasicCall.invoke(BasicCall.java:482) at ai.loony.xai12.wstest.InvoceWS.invoce(InvoceWS.java:125) at ai.loony.xai12.wstest.InvoceWS.invoceByVector(InvoceWS.java:75) at wstest.Main.main(Main.java:45) </pre>	
FAILURE: Suspension of web service during transaction	
EXCEPTION STACK TRACE: Sun Microsystems WS Toolkit	EXCEPTION STACK TRACE: IBM WS Toolkit (WSDK)
<p><i>Waiting for response during too much time (more than 2 hours) without exception</i></p>	<pre> WebServicesFault faultCode: {http://websphere.ibm.com/webservices/}Server.generalException faultString: java.io.InterruptedIOException: Read timed out faultActor: null faultDetail: stackTrace: java.io.InterruptedIOException: Read timed out at java.net.SocketInputStream.socketRead(Native Method) at java.net.SocketInputStream.read(SocketInputStream.java:113) at java.io.BufferedInputStream.fill(BufferedInputStream.java:202) at java.io.BufferedInputStream.read(BufferedInputStream.java(Compiled Code)) at com.ibm.ws.webservices.engine.transport.http.HTTPSender.readHeadersFromSocket(HTTPSender.java(Compiled Code)) at com.ibm.ws.webservices.engine.transport.http.HTTPSender.invoke(HTTPSender.java:156) </pre>

	<pre> at com.ibm.ws.webservices.engine.PivotHandlerWrapper.invoke(PivotHandlerWrapper.java:217) at com.ibm.ws.webservices.engine.WebServicesEngine.invoke(WebServicesEngine.java:258) at com.ibm.ws.webservices.engine.client.Connection.invokeEngine(Connection.java:680) at com.ibm.ws.webservices.engine.client.Connection.invoke(Connection.java:604) at com.ibm.ws.webservices.engine.client.Connection.invoke(Connection.java:434) at com.ibm.ws.webservices.engine.client.Call.invoke(Call.java:1381) at ai.loony.xai12.testwsibm.InvoceWS.invoce(InvoceWS.java:127) at ai.loony.xai12.testwsibm.InvoceWS.invoceByVector(InvoceWS.java:77) at testwsibm.Main.main(Main.java:44) </pre>
FAILURE: System error during processing (like "Division by Zero")	
EXCEPTION STACK TRACE: Sun Microsystems WS Toolkit	EXCEPTION STACK TRACE: IBM WS Toolkit (WSDK)
<pre> java.rmi.ServerException: JAXRPC.TIE.04: Internal Server Error (JAXRPC:TIE01: caught exception while handling request: java.lang.ArithmeticException: / by zero) at com.sun.xml.rpc.client.StreamingSender._raiseFault(StreamingSender.java:497) at com.sun.xml.rpc.client.StreamingSender._send(StreamingSender.java:294) at com.sun.xml.rpc.client.dii.CallInvokerImpl.doInvoke(CallInvokerImpl.java:79) at com.sun.xml.rpc.client.dii.BasicCall.invoke(BasicCall.java:482) at ai.loony.xai12.wstest.InvoceWS.invoce(InvoceWS.java:125) at ai.loony.xai12.wstest.InvoceWS.invoceByVector(InvoceWS.java:75) at wstest.Main.main(Main.java:42) </pre>	<pre> WebServicesFault faultCode: {http://websphere.ibm.com/webservices/}Server.generalException faultString: java.lang.ArithmeticException: / by zero faultActor: null faultDetail: stackTrace: <![CDATA[java.lang.ArithmeticException: / by zero at ai.xai12.loony.ws85.Mult85.getMult(Mult85.java:20) at java.lang.reflect.Method.invoke(Native Method) at com.ibm.ws.webservices.engine.providers.java.RPCProvider.invokeMethod(RPCProvider.java:421) at com.ibm.ws.webservices.engine.providers.java.RPCProvider.processRequestResponse(RPCProvider.java:313) at com.ibm.ws.webservices.engine.providers.java.RPCProvider.processMessage(RPCProvider.java:262) at com.ibm.ws.webservices.engine.providers.java.JavaProvider.invoke(JavaProvider.java:289) at com.ibm.ws.webservices.engine.PivotHandlerWrapper.invoke(PivotHandlerWrapper.java:217) at com.ibm.ws.webservices.engine.handlers.WrappedHandler.invoke(WrappedHandler.java:61) at com.ibm.ws.webservices.engine.PivotHandlerWrapper.invoke(PivotHandlerWrapper.java:217) at com.ibm.ws.webservices.engine.PivotHandlerWrapper.invoke(PivotHandlerWrapper.java:217) at com.ibm.ws.webservices.engine.WebServicesEngine.invoke(WebServicesEngine.java:258) at com.ibm.ws.webservices.engine.transport.http.WebServicesServlet.doPost(WebServicesServlet.java:835) at javax.servlet.http.HttpServlet.service(HttpServlet.java:760) at com.ibm.ws.webservices.engine.transport.http.WebServicesServletBase.service(WebServicesServletBase.java:341) at javax.servlet.http.HttpServlet.service(HttpServlet.java:853) at com.ibm.ws.webcontainer.servlet.StrictServletInstance.doService(StrictServletInstance.java:110) at com.ibm.ws.webcontainer.servlet.StrictLifecycleServlet._service(StrictLifecycleServlet.java:174) at com.ibm.ws.webcontainer.servlet.IdleServletState.service(StrictLifecycleServlet.java:313) at com.ibm.ws.webcontainer.servlet.StrictLifecycleServlet.service(StrictLifecycleServlet.java:116) at com.ibm.ws.webcontainer.servlet.ServletInstance.service(ServletInstance.java:283) at com.ibm.ws.webcontainer.servlet.ValidServletReferenceState.dispatch(ValidServletReferenceState.java:42) at com.ibm.ws.webcontainer.servlet.ServletInstanceReference.dispatch(ServletInstanceReference.java:40) at com.ibm.ws.webcontainer.webapp.WebAppRequestDispatcher.handleWebAppDispatch(WebAppRequestDispatcher.j ava:948) at com.ibm.ws.webcontainer.webapp.WebAppRequestDispatcher.dispatch(WebAppRequestDispatcher.java:530) at com.ibm.ws.webcontainer.webapp.WebAppRequestDispatcher.forward(WebAppRequestDispatcher.java:176) at com.ibm.ws.webcontainer.srt.WebAppInvoker.doForward(WebAppInvoker.java:79) </pre>

	<pre> at com.ibm.ws.webcontainer.srt.WebAppInvoker.handleInvocationHook(WebAppInvoker.java:201) at com.ibm.ws.webcontainer.cache.invocation.CachedInvocation.handleInvocation(CachedInvocation.java:71) at com.ibm.ws.webcontainer.srp.ServletRequestProcessor.dispatchByURI(ServletRequestProcessor.java:182) at com.ibm.ws.webcontainer.oselister.OSELlistenerDispatcher.service(OSELlistener.java:334) at com.ibm.ws.webcontainer.http.HttpConnection.handleRequest(HttpConnection.java:56) at com.ibm.ws.http.HttpConnection.readAndHandleRequest(HttpConnection.java:610) at com.ibm.ws.http.HttpConnection.run(HttpConnection.java:431) at com.ibm.ws.util.ThreadPool\$Worker.run(ThreadPool.java(Compiled Code))]]> java.lang.ArithmeticException: / by zero at com.ibm.ws.webservices.engine.xmlsoap.builders.SOAPFaultBuilder.createFault(SOAPFaultBuilder.java:227) at com.ibm.ws.webservices.engine.xmlsoap.builders.SOAPFaultBuilder.endElement(SOAPFaultBuilder.java:155) at com.ibm.ws.webservices.engine.events.P2DConverter.endElement(P2DConverter.java:405) at org.apache.xerces.parsers.AbstractSAXParser.endElement(AbstractSAXParser.java:569) at org.apache.xerces.impl.XMLNamespaceBinder.handleEndElement(XMLNamespaceBinder.java:853) at org.apache.xerces.impl.XMLNamespaceBinder.endElement(XMLNamespaceBinder.java:643) at org.apache.xerces.impl.dtd.XMLDTDValidator.handleEndElement(XMLDTDValidator.java:3003) at org.apache.xerces.impl.dtd.XMLDTDValidator.endElement(XMLDTDValidator.java:931) at org.apache.xerces.impl.XMLDocumentFragmentScannerImpl.handleEndElement(XMLDocumentFragmentScannerImpl.java:1147) at org.apache.xerces.impl.XMLDocumentFragmentScannerImpl.scanEndElement(XMLDocumentFragmentScannerImpl.java:988) at org.apache.xerces.impl.XMLDocumentFragmentScannerImpl\$FragmentContentDispatcher.dispatch (XMLDocumentFragmentScannerImpl.java:1448) at org.apache.xerces.impl.XMLDocumentFragmentScannerImpl.scanDocument(XMLDocumentFragmentScannerImpl.java:333) at org.apache.xerces.parsers.StandardParserConfiguration.parse(StandardParserConfiguration.java:525) at org.apache.xerces.parsers.StandardParserConfiguration.parse(StandardParserConfiguration.java:581) at org.apache.xerces.parsers.XMLParser.parse(XMLParser.java:147) at org.apache.xerces.parsers.AbstractSAXParser.parse(AbstractSAXParser.java:1158) at javax.xml.parsers.SAXParser.parse(Unknown Source) at com.ibm.ws.webservices.engine.encoding.DeserializationContextImpl.parse(DeserializationContextImpl.java:256) at com.ibm.ws.webservices.engine.SOAPPart.getAsSOAPEnvelope(SOAPPart.java:698) at com.ibm.ws.webservices.engine.Message.getSOAPEnvelope(Message.java:440) at com.ibm.ws.webservices.engine.client.Connection.invokeEngine(Connection.java:700) at com.ibm.ws.webservices.engine.client.Connection.invoke(Connection.java:604) at com.ibm.ws.webservices.engine.client.Connection.invoke(Connection.java:434) at com.ibm.ws.webservices.engine.client.Call.invoke(Call.java:1381) at ai.loony.xai12.testwsibm.InvoceWS.invoce(InvoceWS.java:127) at ai.loony.xai12.testwsibm.InvoceWS.invoceByVector(InvoceWS.java:77) at testwsibm.Main.main(Main.java:44) </pre>
FAILURE: Calculation error during processing (for example, "Operand Type Mismatch")	
EXCEPTION STACK TRACE: Sun Microsystems WS Toolkit	EXCEPTION STACK TRACE: IBM WS Toolkit (WSDK)
java.rmi.ServerException: JAXRPC.TIE.04: Internal Server Error (JAXRPC:TIE01:	WebServicesFault

```
java.lang.NumberFormatException: For input string: "578ER"  
at com.sun.xml.rpc.client.dii.BasicCall.invoke(BasicCall.java:497)  
at ai.loony.xai12.wstest.InvoceWS.invoce(InvoiceWS.java:125)  
at ai.loony.xai12.wstest.InvoceWS.invoceByVector(InvoiceWS.java:75)  
at wstest.Main.main(Main.java:42)
```

```
faultCode: {http://websphere.ibm.com/webservices/}Server generalException  
faultString: java.lang.NumberFormatException: 5ER  
faultActor: null  
faultDetail:  
stackTrace: <![CDATA[  
java.lang.NumberFormatException: 5ER  
at java.lang.Integer.parseInt(Integer.java:Compiled Code)  
at java.lang.Integer.<init>(Integer.java:558)  
at ai.xai12.loony.ws84.Mult84.getMult(Mult84.java:20)  
at java.lang.reflect.Method.invoke(Native Method)  
at com.ibm.ws.webservices.engine.providers.java.RPCProvider.invokeMethod(RPCProvider.java:421)  
at com.ibm.ws.webservices.engine.providers.java.RPCProvider.processRequestResponse(RPCProvider.java:313)  
at com.ibm.ws.webservices.engine.providers.java.RPCProvider.processMessage(RPCProvider.java:262)  
at com.ibm.ws.webservices.engine.providers.java.JavaProvider.invoke(JavaProvider.java:289)  
at com.ibm.ws.webservices.engine.PivotHandlerWrapper.invoke(PivotHandlerWrapper.java:217)  
at com.ibm.ws.webservices.engine.handlers.WrappedHandler.invoke(WrappedHandler.java:61)  
at com.ibm.ws.webservices.engine.PivotHandlerWrapper.invoke(PivotHandlerWrapper.java:217)  
at com.ibm.ws.webservices.engine.PivotHandlerWrapper.invoke(PivotHandlerWrapper.java:217)  
at com.ibm.ws.webservices.engine.WebServicesEngine.invoke(WebServicesEngine.java:258)  
at com.ibm.ws.webservices.engine.transport.http.WebServicesServlet.doPost(WebServicesServlet.java:835)  
at javax.servlet.http.HttpServlet.service(HttpServlet.java:760)  
at com.ibm.ws.webservices.engine.transport.http.WebServicesServletBase.service(WebServicesServletBase.java:341)  
at javax.servlet.http.HttpServlet.service(HttpServlet.java:853)  
at com.ibm.ws.webcontainer.servlet.StrictServletInstance.doService(StrictServletInstance.java:110)  
at com.ibm.ws.webcontainer.servlet.StrictLifecycleServlet._service(StrictLifecycleServlet.java:174)  
at com.ibm.ws.webcontainer.servlet.IdleServletState.service(StrictLifecycleServlet.java:313)  
at com.ibm.ws.webcontainer.servlet.StrictLifecycleServlet.service(StrictLifecycleServlet.java:116)  
at com.ibm.ws.webcontainer.servlet.ServletInstance.service(ServletInstance.java:283)  
at com.ibm.ws.webcontainer.servlet.ValidServletReferenceState.dispatch(ValidServletReferenceState.java:42)  
at com.ibm.ws.webcontainer.servlet.ServletInstanceReference.dispatch(ServletInstanceReference.java:40)  
at  
com.ibm.ws.webcontainer.webapp.WebAppRequestDispatcher.handleWebAppDispatch(WebAppRequestDispatcher.java:948)  
at com.ibm.ws.webcontainer.webapp.WebAppRequestDispatcher.dispatch(WebAppRequestDispatcher.java:530)  
at com.ibm.ws.webcontainer.webapp.WebAppRequestDispatcher.forward(WebAppRequestDispatcher.java:176)  
at com.ibm.ws.webcontainer.srt.WebAppInvoker.doForward(WebAppInvoker.java:79)  
at com.ibm.ws.webcontainer.srt.WebAppInvoker.handleInvocationHook(WebAppInvoker.java:201)  
at com.ibm.ws.webcontainer.cache.invocation.CachedInvocation.handleInvocation(CachedInvocation.java:71)  
at com.ibm.ws.webcontainer.srp.ServletRequestProcessor.dispatchByURI(ServletRequestProcessor.java:182)  
at com.ibm.ws.webcontainer.oselister.OSelisterDispatcher.service(OSelister.java:334)  
at com.ibm.ws.webcontainer.http.HttpConnection.handleRequest(HttpConnection.java:56)  
at com.ibm.ws.http.HttpConnection.readAndHandleRequest(HttpConnection.java:610)  
at com.ibm.ws.http.HttpConnection.run(HttpConnection.java:431)  
at com.ibm.ws.util.ThreadPool$Worker.run(ThreadPool.java:Compiled Code)  
]]>  
  
java.lang.NumberFormatException: 5ER  
at com.ibm.ws.webservices.engine.xmlsoap.builders.SOAPFaultBuilder.createFault(SOAPFaultBuilder.java:227)
```

	<pre> at com.ibm.ws.webservices.engine.xmlsoap.builders.SOAPFaultBuilder.endElement(SOAPFaultBuilder.java:155) at com.ibm.ws.webservices.engine.events.P2DCConverter.endElement(P2DCConverter.java:405) at org.apache.xerces.parsers.AbstractSAXParser.endElement(AbstractSAXParser.java:569) at org.apache.xerces.impl.XMLNamespaceBinder.handleEndElement(XMLNamespaceBinder.java:853) at org.apache.xerces.impl.XMLNamespaceBinder.endElement(XMLNamespaceBinder.java:643) at org.apache.xerces.impl.dtd.XMLDTDValidator.handleEndElement(XMLDTDValidator.java:3003) at org.apache.xerces.impl.dtd.XMLDTDValidator.endElement(XMLDTDValidator.java:931) at org.apache.xerces.impl.XMLDocumentFragmentScannerImpl.handleEndElement(XMLDocumentFragmentScannerImpl.java:1147) at org.apache.xerces.impl.XMLDocumentFragmentScannerImpl.scanEndElement(XMLDocumentFragmentScannerImpl.java:988) at org.apache.xerces.impl.XMLDocumentFragmentScannerImpl\$FragmentContentDispatcher.dispatch(XMLDocumentFragmentScannerImpl.java:1448) at org.apache.xerces.impl.XMLDocumentFragmentScannerImpl.scanDocument(XMLDocumentFragmentScannerImpl.java:333) at org.apache.xerces.parsers.StandardParserConfiguration.parse(StandardParserConfiguration.java:525) at org.apache.xerces.parsers.StandardParserConfiguration.parse(StandardParserConfiguration.java:581) at org.apache.xerces.parsers.XMLParser.parse(XMLParser.java:147) at org.apache.xerces.parsers.AbstractSAXParser.parse(AbstractSAXParser.java:1158) at javax.xml.parsers.SAXParser.parse(Unknown Source) at com.ibm.ws.webservices.engine.encoding.DeserializationContextImpl.parse(DeserializationContextImpl.java:256) at com.ibm.ws.webservices.engine.SOAPPart.getAsSOAPEnvelope(SOAPPart.java:698) at com.ibm.ws.webservices.engine.Message.getSOAPEnvelope(Message.java:440) at com.ibm.ws.webservices.engine.client.Connection.invokeEngine(Connection.java:700) at com.ibm.ws.webservices.engine.client.Connection.invoke(Connection.java:604) at com.ibm.ws.webservices.engine.client.Connection.invoke(Connection.java:434) at com.ibm.ws.webservices.engine.client.Call.invoke(Call.java:1381) at ai.loony.xai12.testwsibm.InvoiceWS.invoce(InvoiceWS.java:127) at ai.loony.xai12.testwsibm.InvoiceWS.invoceByVector(InvoiceWS.java:77) at testwsibm.Main.main(Main.java:44) </pre>
FAILURE: Application error raising user exception	
EXCEPTION STACK TRACE: Sun Microsystems WS Toolkit	EXCEPTION STACK TRACE: IBM WS Toolkit (WSDK)
<pre> java.rmi.RemoteException: ai.xai12.loony.exception.MyException at com.sun.xml.rpc.client.dii.BasicCall.invoke(BasicCall.java:497) at ai.loony.xai12.wstest.InvoiceWS.invoce(InvoiceWS.java:125) at ai.loony.xai12.wstest.InvoiceWS.invoceByVector(InvoiceWS.java:75) at wstest.Main.main(Main.java:42) </pre>	<pre> WebServicesFault faultCode: {http://websphere.ibm.com/webservices/} Server.generalException faultString: (13) MyException faultActor: null faultDetail: stackTrace: <![CDATA[(13)Май Экспешен at ai.xai12.loony.ws77.Mult77.getMult1(Mult77.java:36) at java.lang.reflect.Method.invoke(Native Method) at com.ibm.ws.webservices.engine.providers.java.RPCProvider.invokeMethod(RPCProvider.java:421) </pre>

```

at com.ibm.ws.webservices.engine.providers.java.RPCProvider.processRequestResponse(RPCProvider.java:313)
at com.ibm.ws.webservices.engine.providers.java.RPCProvider.processMessage(RPCProvider.java:262)
at com.ibm.ws.webservices.engine.providers.java.JavaProvider.invoke(JavaProvider.java:289)
at com.ibm.ws.webservices.engine.PivotHandlerWrapper.invoke(PivotHandlerWrapper.java:217)
at com.ibm.ws.webservices.engine.handlers.WrappedHandler.invoke(WrappedHandler.java:61)
at com.ibm.ws.webservices.engine.PivotHandlerWrapper.invoke(PivotHandlerWrapper.java:217)
at com.ibm.ws.webservices.engine.PivotHandlerWrapper.invoke(PivotHandlerWrapper.java:217)
at com.ibm.ws.webservices.engine.WebServicesEngine.invoke(WebServicesEngine.java:258)
at com.ibm.ws.webservices.engine.transport.http.WebServicesServlet.doPost(WebServicesServlet.java:835)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:760)
at com.ibm.ws.webservices.engine.transport.http.WebServicesServletBase.service(WebServicesServletBase.java:341)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:853)
at com.ibm.ws.webcontainer.servlet.StrictServletInstance.doService(StrictServletInstance.java:110)
at com.ibm.ws.webcontainer.servlet.StrictLifecycleServlet._service(StrictLifecycleServlet.java:174)
at com.ibm.ws.webcontainer.servlet.IdleServletState.service(StrictLifecycleServlet.java:313)
at com.ibm.ws.webcontainer.servlet.StrictLifecycleServlet.service(StrictLifecycleServlet.java:116)
at com.ibm.ws.webcontainer.servlet.ServletInstance.service(ServletInstance.java:283)
at com.ibm.ws.webcontainer.servlet.ValidServletReferenceState.dispatch(ValidServletReferenceState.java:42)
at com.ibm.ws.webcontainer.servlet.ServletInstanceReference.dispatch(ServletInstanceReference.java:40)
at
com.ibm.ws.webcontainer.webapp.WebAppRequestDispatcher.handleWebAppDispatch(WebAppRequestDispatcher.java:948)
at com.ibm.ws.webcontainer.webapp.WebAppRequestDispatcher.dispatch(WebAppRequestDispatcher.java:530)
at com.ibm.ws.webcontainer.webapp.WebAppRequestDispatcher.forward(WebAppRequestDispatcher.java:176)
at com.ibm.ws.webcontainer.srt.WebAppInvoker.doForward(WebAppInvoker.java:79)
at com.ibm.ws.webcontainer.srt.WebAppInvoker.handleInvocationHook(WebAppInvoker.java:201)
at com.ibm.ws.webcontainer.cache.invocation.CachedInvocation.handleInvocation(CachedInvocation.java:71)
at com.ibm.ws.webcontainer.srp.ServletRequestProcessor.dispatchByURI(ServletRequestProcessor.java:182)
at com.ibm.ws.webcontainer.oselistener.OSEListenerDispatcher.service(OSEListener.java:334)
at com.ibm.ws.webcontainer.http.HttpConnection.handleRequest(HttpConnection.java:56)
at com.ibm.ws.http.HttpConnection.readAndHandleRequest(HttpConnection.java:610)
at com.ibm.ws.http.HttpConnection.run(HttpConnection.java:431)
at com.ibm.ws.util.ThreadPool$Worker.run(ThreadPool.java(Compiled Code))
]]>

(13) MyException
at com.ibm.ws.webservices.engine.xmlsoap.builders.SOAPFaultBuilder.createFault(SOAPFaultBuilder.java:227)
at com.ibm.ws.webservices.engine.xmlsoap.builders.SOAPFaultBuilder.endElement(SOAPFaultBuilder.java:155)
at com.ibm.ws.webservices.engine.events.P2DConverter.endElement(P2DConverter.java:405)
at org.apache.xerces.parsers.AbstractSAXParser.endElement(AbstractSAXParser.java:569)
at org.apache.xerces.impl.XMLNamespaceBinder.handleEndElement(XMLNamespaceBinder.java:853)
at org.apache.xerces.impl.XMLNamespaceBinder.endElement(XMLNamespaceBinder.java:643)
at org.apache.xerces.impl.dtd.XMLDTDValidator.handleEndElement(XMLDTDValidator.java:3003)
at org.apache.xerces.impl.dtd.XMLDTDValidator.endElement(XMLDTDValidator.java:931)
at
org.apache.xerces.impl.XMLDocumentFragmentScannerImpl.handleEndElement(XMLDocumentFragmentScannerImpl.java:1147)

```


	<pre> at org.apache.xerces.impl.XMLDocumentFragmentScannerImpl.scanEndElement(XMLDocumentFragmentScannerImpl.java:988) at org.apache.xerces.impl.XMLDocumentFragmentScannerImpl\$FragmentContentDispatcher.dispatch(XMLDocumentF ragmentScannerImpl.java:1448) at org.apache.xerces.impl.XMLDocumentFragmentScannerImpl.scanDocument(XMLDocumentFragmentScannerImpl.ja va:333) at org.apache.xerces.parsers.StandardParserConfiguration.parse(StandardParserConfiguration.java:525) at org.apache.xerces.parsers.StandardParserConfiguration.parse(StandardParserConfiguration.java:581) at org.apache.xerces.parsers.XMLParser.parse(XMLParser.java:147) at org.apache.xerces.parsers.AbstractSAXParser.parse(AbstractSAXParser.java:1158) at javax.xml.parsers.SAXParser.parse(Unknown Source) at com.ibm.ws.webservices.engine.encoding.DeserializationContextImpl.parse(DeserializationContextImpl.java:256) at com.ibm.ws.webservices.engine.SOAPPart.getAsSOAPEnvelope(SOAPPart.java:698) at com.ibm.ws.webservices.engine.Message.getSOAPEnvelope(Message.java:440) at com.ibm.ws.webservices.engine.client.Connection.invokeEngine(Connection.java:700) at com.ibm.ws.webservices.engine.client.Connection.invoke(Connection.java:604) at com.ibm.ws.webservices.engine.client.Connection.invoke(Connection.java:434) at com.ibm.ws.webservices.engine.client.Call.invoke(Call.java:1381) at ai.loony.xai12.testwsibm.InvoiceWS.invoice(InvoiceWS.java:127) at ai.loony.xai12.testwsibm.InvoiceWS.invoiceByVector(InvoiceWS.java:77) at testwsibm.Main.main(Main.java:44) </pre>
FAILURE: Error in Target Name Space (in case of using DII)	
EXCEPTION STACK TRACE: Sun Microsystems WS Toolkit	EXCEPTION STACK TRACE: IBM WS Toolkit (WSDK)
<pre> java.rmi.RemoteException: JAXRPC1E01: caught exception while handling request: unrecognized operation: {urn:WSStrEring/wsd}gluingstring at com.sun.xml.rpc.client.dii.BasicCall.invoke(BasicCall.java:497) at ai.loony.xai12.wstest.InvoiceWS.invoice(InvoiceWS.java:125) at ai.loony.xai12.wstest.InvoiceWS.invoiceByVector(InvoiceWS.java:75) at wstest.Main.main(Main.java:42) </pre>	<i>Correct output without exception</i>
FAILURE: Error in service operation's name (in case of using DII)	
EXCEPTION STACK TRACE: Sun Microsystems WS Toolkit	EXCEPTION STACK TRACE: IBM WS Toolkit (WSDK)
<pre> java.rmi.RemoteException: JAXRPC1E01: caught exception while handling request: unrecognized operation: {urn:WSNumeric/wsd}getMultInteERgerToInteger at com.sun.xml.rpc.client.dii.BasicCall.invoke(BasicCall.java:497) at ai.loony.xai12.wstest.InvoiceWS.invoice(InvoiceWS.java:125) at ai.loony.xai12.wstest.InvoiceWS.invoiceByVector(InvoiceWS.java:75) at wstest.Main.main(Main.java:42) </pre>	<pre> WebServicesFault faultCode: {http://websphere.ibm.com/webservices/}Server.generalException faultString: WWS3277E: Error: No such operation 'getMRTult' faultActor: null faultDetail: stackTrace: <![CDATA[WebServicesFault faultCode: {http://websphere.ibm.com/webservices/}Server.generalException faultString: WWS3277E: Error: No such operation 'getMRTult' faultActor: null faultDetail: WWS3277E: Error: No such operation 'getMRTult' </pre>

```
at com.ibm.ws.webservices.engine.providers.java.RPCProvider.processMessage(RPCProvider.java:168)
at com.ibm.ws.webservices.engine.providers.java.JavaProvider.invoke(JavaProvider.java:289)
at com.ibm.ws.webservices.engine.PivotHandlerWrapper.invoke(PivotHandlerWrapper.java:217)
at com.ibm.ws.webservices.engine.handlers.WrappedHandler.invoke(WrappedHandler.java:61)
at com.ibm.ws.webservices.engine.PivotHandlerWrapper.invoke(PivotHandlerWrapper.java:217)
at com.ibm.ws.webservices.engine.PivotHandlerWrapper.invoke(PivotHandlerWrapper.java:217)
at com.ibm.ws.webservices.engine.WebServicesEngine.invoke(WebServicesEngine.java:258)
at com.ibm.ws.webservices.engine.transport.http.WebServicesServlet.doPost(WebServicesServlet.java:835)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:760)
at com.ibm.ws.webservices.engine.transport.http.WebServicesServletBase.service(WebServicesServletBase.java:341)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:853)
at com.ibm.ws.webcontainer.servlet.StrictServletInstance.doService(StrictServletInstance.java:110)
at com.ibm.ws.webcontainer.servlet.StrictLifecycleServlet._service(StrictLifecycleServlet.java:174)
at com.ibm.ws.webcontainer.servlet.IdleServletState.service(StrictLifecycleServlet.java:313)
at com.ibm.ws.webcontainer.servlet.StrictLifecycleServlet.service(StrictLifecycleServlet.java:116)
at com.ibm.ws.webcontainer.servlet.ServletInstance.service(ServletInstance.java:283)
at com.ibm.ws.webcontainer.servlet.ValidServletReferenceState.dispatch(ValidServletReferenceState.java:42)
at com.ibm.ws.webcontainer.servlet.ServletInstanceReference.dispatch(ServletInstanceReference.java:40)
at
com.ibm.ws.webcontainer.webapp.WebAppRequestDispatcher.handleWebAppDispatch(WebAppRequestDispatcher.java:948)
at com.ibm.ws.webcontainer.webapp.WebAppRequestDispatcher.dispatch(WebAppRequestDispatcher.java:530)
at com.ibm.ws.webcontainer.webapp.WebAppRequestDispatcher.forward(WebAppRequestDispatcher.java:176)
at com.ibm.ws.webcontainer.srt.WebAppInvoker.doForward(WebAppInvoker.java:79)
at com.ibm.ws.webcontainer.srt.WebAppInvoker.handleInvocationHook(WebAppInvoker.java:201)
at com.ibm.ws.webcontainer.cache.invocation.CachedInvocation.handleInvocation(CachedInvocation.java:71)
at com.ibm.ws.webcontainer.cache.invocation.CacheableInvocationContext.invoke(CacheableInvocationContext.java:114)
at com.ibm.ws.webcontainer.srp.ServletRequestProcessor.dispatchByURI(ServletRequestProcessor.java:186)
at com.ibm.ws.webcontainer.oselistener.OSEListenerDispatcher.service(OSEListener.java:334)
at com.ibm.ws.webcontainer.http.HttpConnection.handleRequest(HttpConnection.java:56)
at com.ibm.ws.http.HttpConnection.readAndHandleRequest(HttpConnection.java:610)
at com.ibm.ws.http.HttpConnection.run(HttpConnection.java:431)
at com.ibm.ws.util.ThreadPool$Worker.run(ThreadPool.java(Compiled Code))
]]>

WSWS3277E: Error: No such operation 'getMRTult'
at com.ibm.ws.webservices.engine.xmlsoap.builders.SOAPFaultBuilder.createFault(SOAPFaultBuilder.java:227)
at com.ibm.ws.webservices.engine.xmlsoap.builders.SOAPFaultBuilder.endElement(SOAPFaultBuilder.java:155)
at com.ibm.ws.webservices.engine.events.P2DConverter.endElement(P2DConverter.java:405)
at org.apache.xerces.parsers.AbstractSAXParser.endElement(AbstractSAXParser.java:569)
at org.apache.xerces.impl.XMLNamespaceBinder.handleEndElement(XMLNamespaceBinder.java:853)
at org.apache.xerces.impl.XMLNamespaceBinder.endElement(XMLNamespaceBinder.java:643)
at org.apache.xerces.impl.dtd.XMLDTDValidator.handleEndElement(XMLDTDValidator.java:3003)
at org.apache.xerces.impl.dtd.XMLDTDValidator.endElement(XMLDTDValidator.java:931)
at
org.apache.xerces.impl.XMLDocumentFragmentScannerImpl.handleEndElement(XMLDocumentFragmentScannerImpl.java:1147)
```

	<pre> at org.apache.xerces.impl.XMLDocumentFragmentScannerImpl.scanEndElement(XMLDocumentFragmentScannerImpl.java: 988) at org.apache.xerces.impl.XMLDocumentFragmentScannerImpl\$FragmentContentDispatcher.dispatch(XMLDocumentF ragmentScannerImpl.java:1448) at org.apache.xerces.impl.XMLDocumentFragmentScannerImpl.scanDocument(XMLDocumentFragmentScannerImpl.ja va:333) at org.apache.xerces.parsers.StandardParserConfiguration.parse(StandardParserConfiguration.java:525) at org.apache.xerces.parsers.StandardParserConfiguration.parse(StandardParserConfiguration.java:581) at org.apache.xerces.parsers.XMLParser.parse(XMLParser.java:147) at org.apache.xerces.parsers.AbstractSAXParser.parse(AbstractSAXParser.java:1158) at javax.xml.parsers.SAXParser.parse(Unknown Source) at com.ibm.ws.webservices.engine.encoding.DeserializationContextImpl.parse(DeserializationContextImpl.java:256) at com.ibm.ws.webservices.engine.SOAPPart.getAsSOAPEnvelope(SOAPPart.java:698) at com.ibm.ws.webservices.engine.Message.getSOAPEnvelope(Message.java:440) at com.ibm.ws.webservices.engine.client.Connection.invokeEngine(Connection.java:700) at com.ibm.ws.webservices.engine.client.Connection.invoke(Connection.java:604) at com.ibm.ws.webservices.engine.client.Connection.invoke(Connection.java:434) at com.ibm.ws.webservices.engine.client.Call.invoke(Call.java:1381) at ai.loony.xai12.testwsibm.InvoceWS.invoce(InvoceWS.java:127) at ai.loony.xai12.testwsibm.InvoceWS.invoceByVector(InvoceWS.java:77) at testwsibm.Main.main(Main.java:44) </pre>
FAILURE: Output parameter type mismatch (in case of using DII)	
EXCEPTION STACK TRACE: Sun Microsystems WS Toolkit	EXCEPTION STACK TRACE: IBM WS Toolkit (WSDK)
<pre> unexpected element type: expected={http://www.w3.org/2001/XMLSchema}string, actual={http://schemas.xmlsoap.org/soap/encoding/}int at com.sun.xml.rpc.encoding.SerializerBase.verifyType(SerializerBase.java:128) at com.sun.xml.rpc.encoding.SimpleTypeSerializer.deserialize(SimpleTypeSerializer.java:114) at com.sun.xml.rpc.encoding.AttachmentSerializer.deserialize(AttachmentSerializer.java:219) at com.sun.xml.rpc.encoding.ReferenceableSerializerImpl.deserialize(ReferenceableSerializerI mpl.java:155) at com.sun.xml.rpc.encoding.soap.SOAPResponseSerializer.doDeserialize(SOAPResponseSer ializer.java:291) at com.sun.xml.rpc.encoding.ObjectSerializerBase.deserialize(ObjectSerializerBase.java:192) at com.sun.xml.rpc.encoding.ReferenceableSerializerImpl.deserialize(ReferenceableSerializerI mpl.java:155) at com.sun.xml.rpc.client.dii.CallInvokerImpl_readFirstBodyElement(CallInvokerImpl.java:2 </pre>	<pre> Correct output without exception </pre>

<pre>85) at com.sun.xml.rpc.client.StreamingSender._send(StreamingSender.java:215) at com.sun.xml.rpc.client.dii.CallInvokerImpl.doInvoke(CallInvokerImpl.java:79) at com.sun.xml.rpc.client.dii.BasicCall.invoke(BasicCall.java:482) at ai.loony.xai12.wstest.InvoceWS.invoce(InvoceWS.java:125) at ai.loony.xai12.wstest.InvoceWS.invoceByVector(InvoceWS.java:75) at wstest.Main.main(Main.java:42)</pre>	
FAILURE: Input parameter type mismatch	
EXCEPTION STACK TRACE: Sun Microsystems WS Toolkit	EXCEPTION STACK TRACE: IBM WS Toolkit (WSDK)
<pre>java.rmi.RemoteException: JAXRPC1E01: caught exception while handling request: unexpected element type: expected={http://schemas.xmlsoap.org/soap/encoding/}int, actual={http://www.w3.org/2001/XMLSchema}string at com.sun.xml.rpc.client.dii.BasicCall.invoke(BasicCall.java:497) at ai.loony.xai12.wstest.InvoceWS.invoce(InvoceWS.java:125) at ai.loony.xai12.wstest.InvoceWS.invoceByVector(InvoceWS.java:75) at wstest.Main.main(Main.java:42)</pre>	<pre>WebServicesFault faultCode: {http://websphere.ibm.com/webservices/}Server.generalException faultString: org.xml.sax.SAXException: Bad types (class java.lang.String -> class java.lang.Integer) faultActor: null faultDetail: stackTrace: <![CDATA[org.xml.sax.SAXException: Bad types (class java.lang.String -> class java.lang.Integer) at com.ibm.ws.webservices.engine.xmlsoap.builders.RequestResponseProcessor.onStartChild(RequestResponseProcess or.java:324) at com.ibm.ws.webservices.engine.events.DEventProcessor.onSimpleChild(DEventProcessor.java:212) at com.ibm.ws.webservices.engine.events.P2DConverter.flush(P2DConverter.java:724) at com.ibm.ws.webservices.engine.events.P2DConverter.endElement(P2DConverter.java:391) at sax.SAX2DocumentEntityParserBase.endElementAction(Unknown Source) at util.DocumentEntityParserBase.endElementEvent(Unknown Source) at com.ibm.xml.b2b.scan.latin.LatinWFCDocumentScanner.scanEndElement(Unknown Source)(Compiled Code) at com.ibm.xml.b2b.scan.latin.LatinWFCDocumentScanner.scanContent(Unknown Source) at com.ibm.xml.b2b.scan.latin.LatinWFCDocumentScanner.scanDocument(Unknown Source) at sax.latin.LatinWFCSAX2DocumentEntityParser.scanDocument(Unknown Source) at util.DocumentEntityParserBase.parse(Unknown Source) at sax.SAX2DocumentEntityParserBase.parseEntity(Unknown Source) at sax.SAX2DocumentEntityParserBase.parse(Unknown Source) at javax.xml.parsers.SAXParser.parse(Unknown Source) at com.ibm.ws.webservices.engine.encoding.DeserializationContextImpl.parse(DeserializationContextImpl.java:256) at com.ibm.ws.webservices.engine.SOAPPart.getAsSOAPEnvelope(SOAPPart.java:698) at com.ibm.ws.webservices.engine.Message.getSOAPEnvelope(Message.java:440) at com.ibm.ws.webservices.engine.handlers.jaxrpc.JAXRPCSOAPHandler.checkSOAPSemantics(JAXRPCSOAPHandler.jav a:218) at com.ibm.ws.webservices.engine.handlers.jaxrpc.JAXRPCSOAPHandler.invokeServerRequestHandler(JAXRPCSOAPHandler.jav a:188) at com.ibm.ws.webservices.engine.handlers.jaxrpc.JAXRPCHandler\$1.invoke(JAXRPCHandler.java:232) at com.ibm.ws.webservices.engine.PivotHandlerWrapper.invoke(PivotHandlerWrapper.java:203) at com.ibm.ws.webservices.engine.handlers.WrappedHandler.invoke(WrappedHandler.java:61) at com.ibm.ws.webservices.engine.PivotHandlerWrapper.invoke(PivotHandlerWrapper.java:217) at com.ibm.ws.webservices.engine.PivotHandlerWrapper.invoke(PivotHandlerWrapper.java:217)</pre>

```
at com.ibm.ws.webservices.engine.WebServicesEngine.invoke(WebServicesEngine.java:258)
at com.ibm.ws.webservices.engine.transport.http.WebServicesServlet.doPost(WebServicesServlet.java:835)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:760)
at com.ibm.ws.webservices.engine.transport.http.WebServicesServletBase.service(WebServicesServletBase.java:341)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:853)
at com.ibm.ws.webcontainer.servlet.StrictServletInstance.doService(StrictServletInstance.java:110)
at com.ibm.ws.webcontainer.servlet.StrictLifecycleServlet._service(StrictLifecycleServlet.java:174)
at com.ibm.ws.webcontainer.servlet.IdleServletState.service(StrictLifecycleServlet.java:313)
at com.ibm.ws.webcontainer.servlet.StrictLifecycleServlet.service(StrictLifecycleServlet.java:116)
at com.ibm.ws.webcontainer.servlet.ServletInstance.service(ServletInstance.java:283)
at com.ibm.ws.webcontainer.servlet.ValidServletReferenceState.dispatch(ValidServletReferenceState.java:42)
at com.ibm.ws.webcontainer.servlet.ServletInstanceReference.dispatch(ServletInstanceReference.java:40)
at
com.ibm.ws.webcontainer.webapp.WebAppRequestDispatcher.handleWebAppDispatch(WebAppRequestDispatcher.java:948)
at com.ibm.ws.webcontainer.webapp.WebAppRequestDispatcher.dispatch(WebAppRequestDispatcher.java:530)
at com.ibm.ws.webcontainer.webapp.WebAppRequestDispatcher.forward(WebAppRequestDispatcher.java:176)
at com.ibm.ws.webcontainer.srt.WebAppInvoker.doForward(WebAppInvoker.java:79)
at com.ibm.ws.webcontainer.srt.WebAppInvoker.handleInvocationHook(WebAppInvoker.java:201)
at com.ibm.ws.webcontainer.cache.invocation.CachedInvocation.handleInvocation(CachedInvocation.java:71)
at
com.ibm.ws.webcontainer.cache.invocation.CacheableInvocationContext.invoke(CacheableInvocationContext.java:114)
at com.ibm.ws.webcontainer.srp.ServletRequestProcessor.dispatchByURI(ServletRequestProcessor.java:186)
at com.ibm.ws.webcontainer.oselistener.OSELListenerDispatcher.service(OSELListener.java:334)
at com.ibm.ws.webcontainer.http.HttpConnection.handleRequest(HttpConnection.java:56)
at com.ibm.ws.http.HttpConnection.readAndHandleRequest(HttpConnection.java:610)
at com.ibm.ws.http.HttpConnection.run(HttpConnection.java:431)
at com.ibm.ws.util.ThreadPool$Worker.run(ThreadPool.java(Compiled Code))
]]>

org.xml.sax.SAXException: Bad types (class java.lang.String -> class java.lang.Integer)
at com.ibm.ws.webservices.engine.xmlsoap.builders.SOAPFaultBuilder.createFault(SOAPFaultBuilder.java:227)
at com.ibm.ws.webservices.engine.xmlsoap.builders.SOAPFaultBuilder.endElement(SOAPFaultBuilder.java:155)
at com.ibm.ws.webservices.engine.events.P2DConverter.endElement(P2DConverter.java:405)
at org.apache.xerces.parsers.AbstractSAXParser.endElement(AbstractSAXParser.java:569)
at org.apache.xerces.impl.XMLNamespaceBinder.handleEndElement(XMLNamespaceBinder.java:853)
at org.apache.xerces.impl.XMLNamespaceBinder.endElement(XMLNamespaceBinder.java:643)
at org.apache.xerces.impl.dtd.XMLDTDValidator.handleEndElement(XMLDTDValidator.java:3003)
at org.apache.xerces.impl.dtd.XMLDTDValidator.endElement(XMLDTDValidator.java:931)
at org.apache.xerces.impl.XMLDocumentFragmentScannerImpl.handleEndElement(XMLDocumentFragmentScannerImpl.java:1147)
at org.apache.xerces.impl.XMLDocumentFragmentScannerImpl.scanEndElement(XMLDocumentFragmentScannerImpl.java:988)
at
org.apache.xerces.impl.XMLDocumentFragmentScannerImpl$FragmentContentDispatcher.dispatch(XMLDocumentFragmentScannerImpl.java:1448)
at org.apache.xerces.impl.XMLDocumentFragmentScannerImpl.scanDocument(XMLDocumentFragmentScannerImpl.java:333)
at org.apache.xerces.parsers.StandardParserConfiguration.parse(StandardParserConfiguration.java:525)
```

	<pre> at org.apache.xerces.parsers.StandardParserConfiguration.parse(StandardParserConfiguration.java:581) at org.apache.xerces.parsers.XMLParser.parse(XMLParser.java:147) at org.apache.xerces.parsers.AbstractSAXParser.parse(AbstractSAXParser.java:1158) at javax.xml.parsers.SAXParser.parse(Unknown Source) at com.ibm.ws.webservices.engine.encoding.DeserializationContextImpl.parse(DeserializationContextImpl.java:256) at com.ibm.ws.webservices.engine.SOAPPart.getAsSOAPEnvelope(SOAPPart.java:698) at com.ibm.ws.webservices.engine.Message.getSOAPEnvelope(Message.java:440) at com.ibm.ws.webservices.engine.client.Connection.invokeEngine(Connection.java:700) at com.ibm.ws.webservices.engine.client.Connection.invoke(Connection.java:604) at com.ibm.ws.webservices.engine.client.Connection.invoke(Connection.java:434) at com.ibm.ws.webservices.engine.client.Call.invoke(Call.java:1381) at ai.loony.xai12.testwsibm.InvoiceWS.invoce(InvoiceWS.java:127) at ai.loony.xai12.testwsibm.InvoiceWS.invoceByVector(InvoiceWS.java:77) at testwsibm.Main.main(Main.java:44) </pre>
FAILURE: Error in name of input parameter	
EXCEPTION STACK TRACE: Sun Microsystems WS Toolkit	EXCEPTION STACK TRACE: IBM WS Toolkit (WSDK)
<pre> java.rmi.RemoteException: JAXRPCTIE01: caught exception while handling request: unexpected element name: expected=Integer_2, actual=IntegERer_1 at com.sun.xml.rpc.client.dii.BasicCall.invoke(BasicCall.java:497) at ai.loony.xai12.wstest.InvoiceWS.invoce(InvoiceWS.java:125) at ai.loony.xai12.wstest.InvoiceWS.invoceByVector(InvoiceWS.java:75) at wstest.Main.main(Main.java:42) </pre>	<i>Correct output without exception</i>
FAILURE: Mismatching of number of input service parameters	
EXCEPTION STACK TRACE: Sun Microsystems WS Toolkit	EXCEPTION STACK TRACE: IBM WS Toolkit (WSDK)

```

java.rmi.RemoteException: JAXRPC1E01: deserialization error: unexpected XML
reader state. expected: END but found: START: Integer_2
at com.sun.xml.rpc.client.dii.BasicCall.invoke(BasicCall.java:497)
at ai.loony.xai12.wstest.InvoiceWS.invoke(InvoiceWS.java:125)
at ai.loony.xai12.wstest.InvoiceWS.invokeByVector(InvoiceWS.java:75)
at wstest.Main.main(Main.java:42)

```

```

WebServicesFault
faultCode: {http://websphere.ibm.com/webservices/}Server.generalException
faultString: java.lang.NullPointerException
faultActor: null
faultDetail:
stackTrace: <![CDATA[
java.lang.NullPointerException
at ia.xai12.loony.ws73.Mult73.getMult(Mult73.java:17)
at java.lang.reflect.Method.invoke(Native Method)
at com.ibm.ws.webservices.engine.providers.java.RPCProvider.invokeMethod(RPCProvider.java:421)
at com.ibm.ws.webservices.engine.providers.java.RPCProvider.processRequestResponse(RPCProvider.java:313)
at com.ibm.ws.webservices.engine.providers.java.RPCProvider.processMessage(RPCProvider.java:262)
at com.ibm.ws.webservices.engine.providers.java.JavaProvider.invoke(JavaProvider.java:289)
at com.ibm.ws.webservices.engine.PivotHandlerWrapper.invoke(PivotHandlerWrapper.java:217)
at com.ibm.ws.webservices.engine.handlers.WrappedHandler.invoke(WrappedHandler.java:61)
at com.ibm.ws.webservices.engine.PivotHandlerWrapper.invoke(PivotHandlerWrapper.java:217)
at com.ibm.ws.webservices.engine.PivotHandlerWrapper.invoke(PivotHandlerWrapper.java:217)
at com.ibm.ws.webservices.engine.WebServicesEngine.invoke(WebServicesEngine.java:258)
at com.ibm.ws.webservices.engine.transport.http.WebServicesServlet.doPost(WebServicesServlet.java:835)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:760)
at com.ibm.ws.webservices.engine.transport.http.WebServicesServletBase.service(WebServicesServletBase.java:341)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:853)
at com.ibm.ws.webcontainer.servlet.StrictServletInstance.doService(StrictServletInstance.java:110)
at com.ibm.ws.webcontainer.servlet.StrictLifecycleServlet._service(StrictLifecycleServlet.java:174)
at com.ibm.ws.webcontainer.servlet.IdleServletState.service(StrictLifecycleServlet.java:313)
at com.ibm.ws.webcontainer.servlet.StrictLifecycleServlet.service(StrictLifecycleServlet.java:116)
at com.ibm.ws.webcontainer.servlet.ServletInstance.service(ServletInstance.java:283)
at com.ibm.ws.webcontainer.servlet.ValidServletReferenceState.dispatch(ValidServletReferenceState.java:42)
at com.ibm.ws.webcontainer.servlet.ServletInstanceReference.dispatch(ServletInstanceReference.java:40)
at
com.ibm.ws.webcontainer.webapp.WebAppRequestDispatcher.handleWebAppDispatch(WebAppRequestDispatcher.java:948)
at com.ibm.ws.webcontainer.webapp.WebAppRequestDispatcher.dispatch(WebAppRequestDispatcher.java:530)
at com.ibm.ws.webcontainer.webapp.WebAppRequestDispatcher.forward(WebAppRequestDispatcher.java:176)
at com.ibm.ws.webcontainer.srt.WebAppInvoker.doForward(WebAppInvoker.java:79)
at com.ibm.ws.webcontainer.srt.WebAppInvoker.handleInvocationHook(WebAppInvoker.java:201)
at com.ibm.ws.webcontainer.cache.invocation.CachedInvocation.handleInvocation(CachedInvocation.java:71)
at com.ibm.ws.webcontainer.srp.ServletRequestProcessor.dispatchByURI(ServletRequestProcessor.java:182)
at com.ibm.ws.webcontainer.oselistener.OSEListenerDispatcher.service(OSEListener.java:334)
at com.ibm.ws.webcontainer.http.HttpConnection.handleRequest(HttpConnection.java:56)
at com.ibm.ws.http.HttpConnection.readAndHandleRequest(HttpConnection.java:610)
at com.ibm.ws.http.HttpConnection.run(HttpConnection.java:431)
at com.ibm.ws.util.ThreadPool$Worker.run(ThreadPool.java:Compiled Code))
]]>

java.lang.NullPointerException
at com.ibm.ws.webservices.engine.xmlsoap.builders.SOAPFaultBuilder.createFault(SOAPFaultBuilder.java:227)
at com.ibm.ws.webservices.engine.xmlsoap.builders.SOAPFaultBuilder.endElement(SOAPFaultBuilder.java:155)

```

```
at com.ibm.ws.webservices.engine.events.P2DConverter.endElement(P2DConverter.java:405)
at org.apache.xerces.parsers.AbstractSAXParser.endElement(AbstractSAXParser.java:569)
at org.apache.xerces.impl.XMLNamespaceBinder.handleEndElement(XMLNamespaceBinder.java:853)
at org.apache.xerces.impl.XMLNamespaceBinder.endElement(XMLNamespaceBinder.java:643)
at org.apache.xerces.impl.dtd.XMLDTDValidator.handleEndElement(XMLDTDValidator.java:3003)
at org.apache.xerces.impl.dtd.XMLDTDValidator.endElement(XMLDTDValidator.java:931)
at
org.apache.xerces.impl.XMLDocumentFragmentScannerImpl.handleEndElement(XMLDocumentFragmentScannerImpl.java:1147)
at
org.apache.xerces.impl.XMLDocumentFragmentScannerImpl.scanEndElement(XMLDocumentFragmentScannerImpl.java:988)
at
org.apache.xerces.impl.XMLDocumentFragmentScannerImpl$FragmentContentDispatcher.dispatch(XMLDocumentFragmentScannerImpl.java:1448)
at
org.apache.xerces.impl.XMLDocumentFragmentScannerImpl.scanDocument(XMLDocumentFragmentScannerImpl.java:333)
at org.apache.xerces.parsers.StandardParserConfiguration.parse(StandardParserConfiguration.java:525)
at org.apache.xerces.parsers.StandardParserConfiguration.parse(StandardParserConfiguration.java:581)
at org.apache.xerces.parsers.XMLParser.parse(XMLParser.java:147)
at org.apache.xerces.parsers.AbstractSAXParser.parse(AbstractSAXParser.java:1158)
at javax.xml.parsers.SAXParser.parse(Unknown Source)
at com.ibm.ws.webservices.engine.encoding.DeserializationContextImpl.parse(DeserializationContextImpl.java:256)
at com.ibm.ws.webservices.engine.SOAPPart.getAsSOAPEnvelope(SOAPPart.java:698)
at com.ibm.ws.webservices.engine.Message.getSOAPEnvelope(Message.java:440)
at com.ibm.ws.webservices.engine.client.Connection.invokeEngine(Connection.java:700)
at com.ibm.ws.webservices.engine.client.Connection.invoke(Connection.java:604)
at com.ibm.ws.webservices.engine.client.Connection.invoke(Connection.java:434)
at com.ibm.ws.webservices.engine.client.Call.invoke(Call.java:1381)
at ai.loony.xai12.testwsibm.InvoceWS.invoce(InvoceWS.java:127)
at ai.loony.xai12.testwsibm.InvoceWS.invoceByVector(InvoceWS.java:77)
at testwsibm.Main.main(Main.java:44)
```