



Developing Techniques for Increasing Dependability of Service-Oriented Systems

Roman V. Ledyayev
National Aerospace University (Kharkiv, Ukraine)
June, 2008

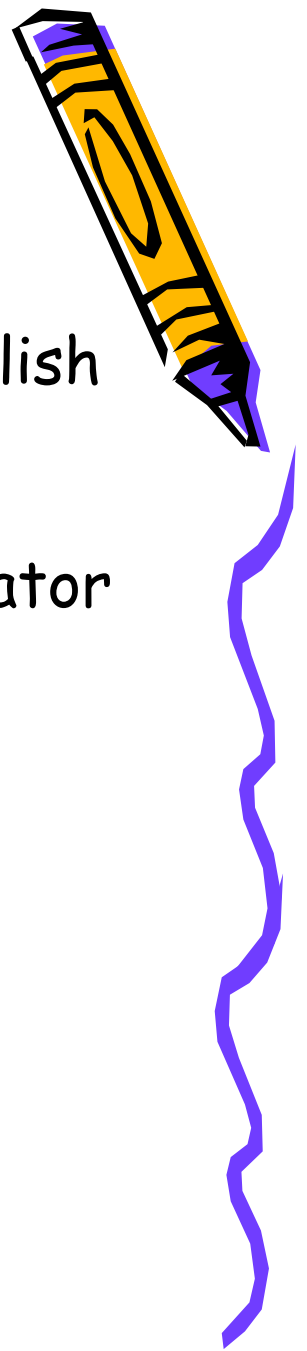
Table of contents



- Project goals
- Choice of tools
- Overview of the problem domain
- Defining the terms
- Patterns of dependability-oriented composition
- Dependability assessment based on user feedback
- Dependability assessment based on UDDI



Author's background

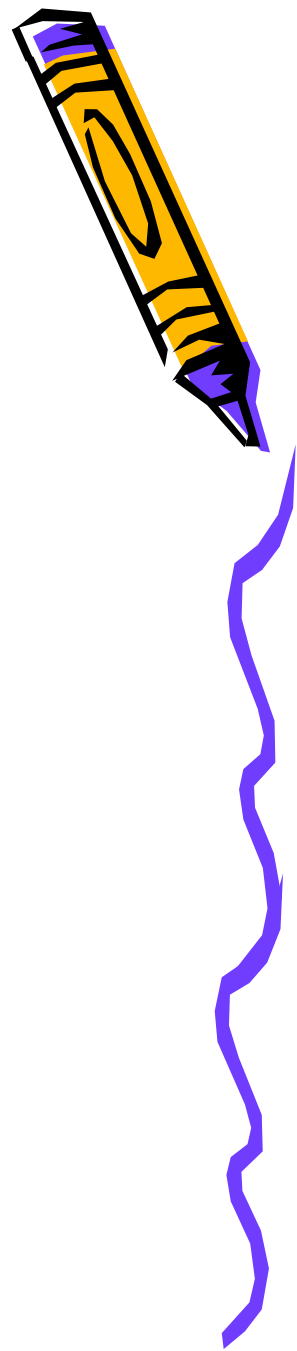


- Born in 1976 (Dimitrov, Ukraine)
- 1999 - graduated with M.A. in Linguistics (English and Literature major)
- 1999-2004 worked as high school, college teacher, camp counselor, private tutor, translator and technical writer
- 2008 - B.S. in Computer Engineering with distinction (Systems Programming major)
- 2008-2009 worked as Software Engineer
 - EchoStar Europe Corporation (Dish Network)
 - InnoWebTech LLC, Ukraine
- 2009 - master thesis intern at Uni.lu



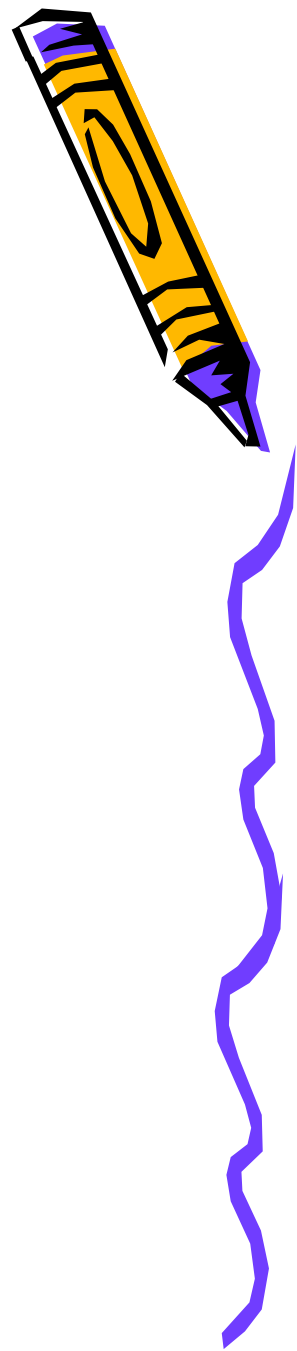
Project goals

- Develop UML Sequence Diagrams for the suggested models of dependable composition
- Develop a client enhanced with dependability assessment capabilities
- Research a technology of dependability assessment based on UDDI

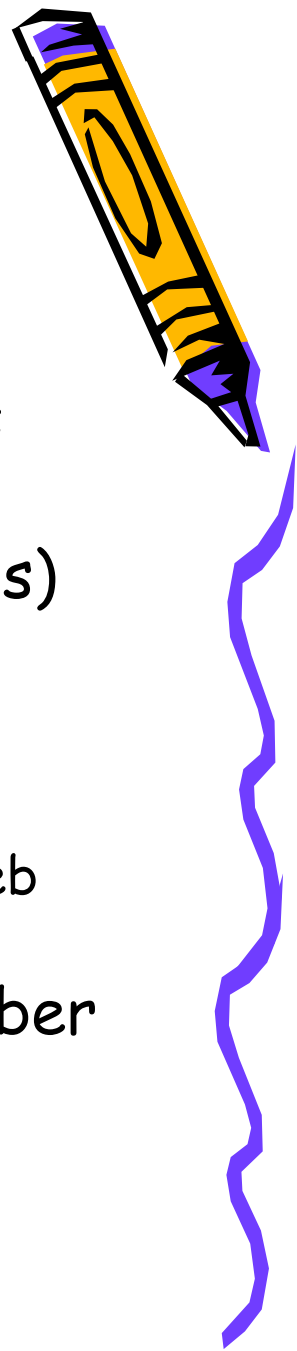


Choice of tools

- UML
- Web Services model
- Eclipse 3.3 Europa
- Apache Tomcat
- Apache Axis2
- jUDDI
- MySQL



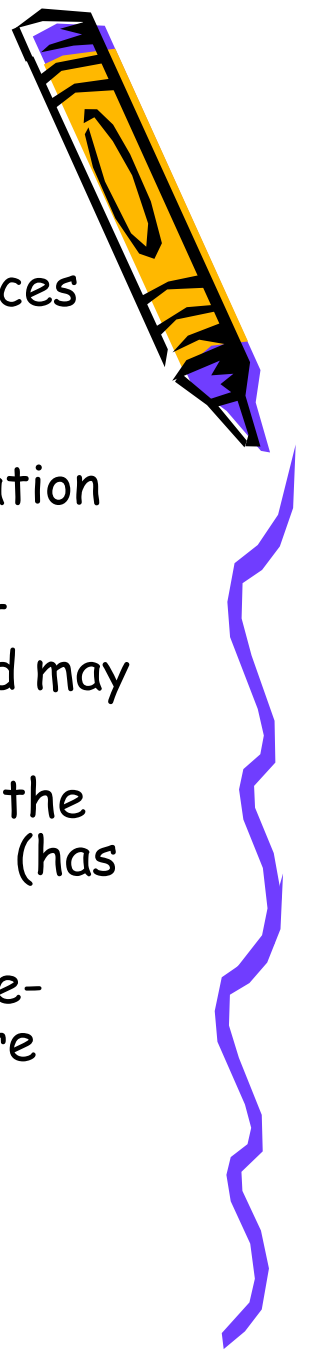
Overview of the problem domain



- High dependability of Software-Oriented Architecture (SOA) is crucial for a number of emerging and existing critical domains (telecommunication, Grid, e-science, e-business)
- One of the possible ways to improve this dependability is by employing
 - service redundancy
 - diversity represented by a number of component web services with identical or similar functionality
- To apply this approach we need to solve a number of problems



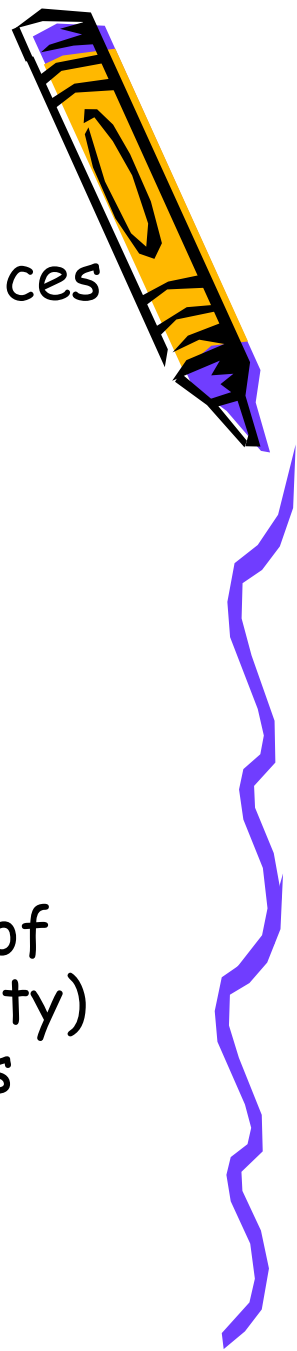
Web Services architecture



- The only known information about them is their interfaces
- Their quality is not completely known and they may not provide sufficient quality of service
- Often have bugs, do not fit enough, have poor specification and documentation
- Heterogeneous (might be developed following different standards, fault assumptions, different conventions and may use different technologies)
- Their construction and composition are complicated by the fact that the Internet is a poor communication medium (has low quality, not predictable)
- Ensuring and assessing dependability of complex service-oriented systems is complicated when these systems are dynamically built



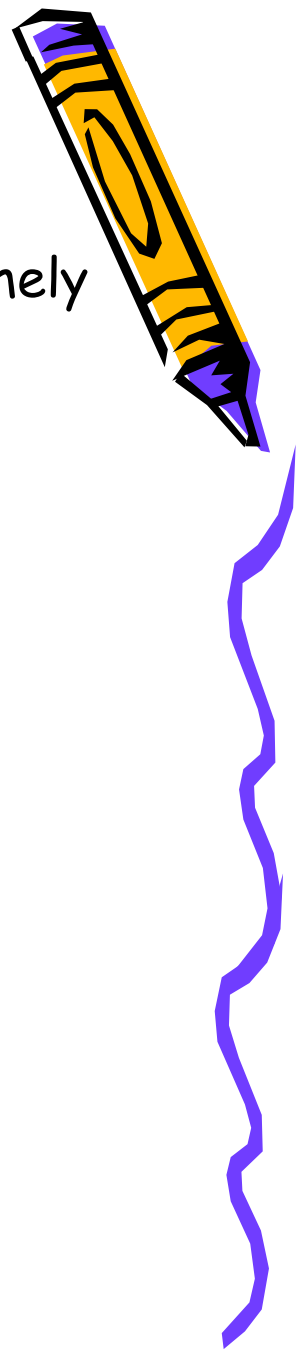
Web Services testing



- There are various tools for testing Web Services
 - Open source
 - SoapUI
 - TestMaker
 - WebInject
 - Grinder
 - JMeter
 - SOAPSonar Personal Edition
 - JUnit
 - Commercial
 - SoapUI Pro
 - SoapSonar
- However, they do not necessarily measure all of the suggested attributes (availability, reliability) and they do not provide means for storing this valuable information in a persistent manner



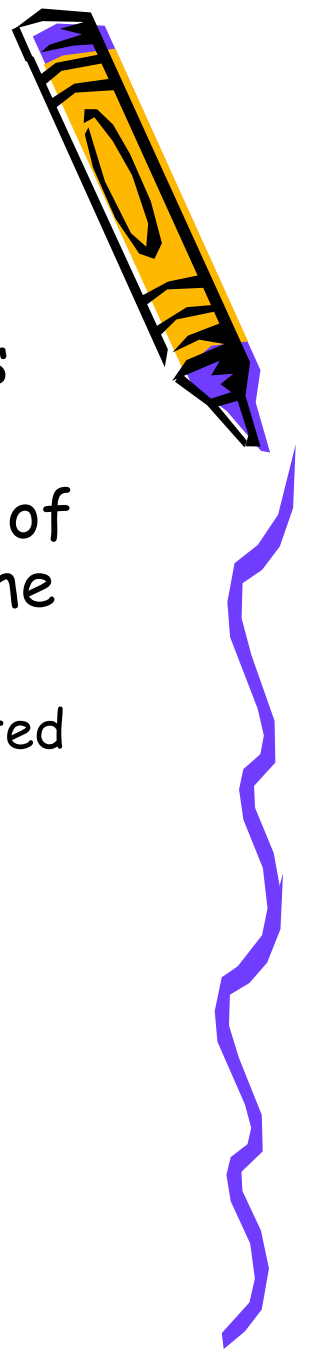
WS Dependability



- Dependability of a computing system is its ability to timely deliver services that can justifiably be trusted
- According to the definition we need to deal with the following dependability attributes (relevant to Web Services, easily measured during invocations)
 - Availability
 - Reliability
 - Response time (performance)
- There could be several other attributes
 - Authentication
 - Confidentiality
 - Non-repudiation
 - Service cost, etc



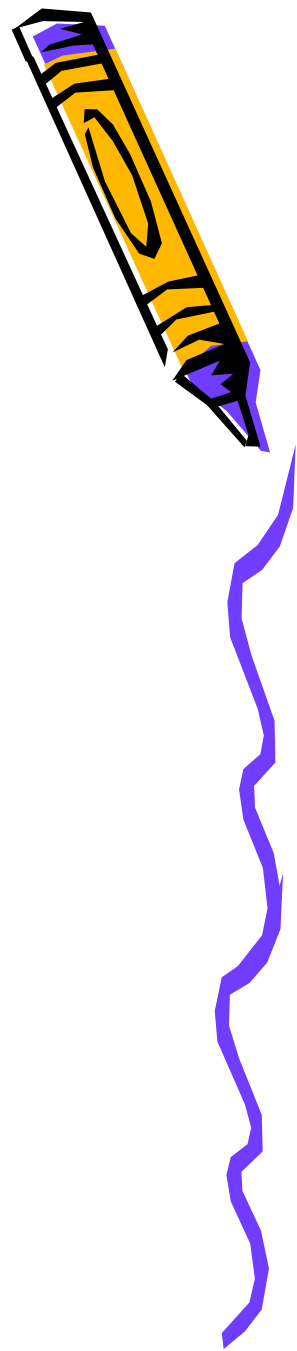
Defining the terms



- **Service availability** - the degree to which a service is operational and accessible when it is required for use
- **Service reliability** - can be measured in terms of probability of failure-free operation, mean time between failures (MTBF) or failure rate
 - assessment in regards to Web Services is complicated
 - not just dealing with the probability of exception
- **Service performance** (response time)
 - network delay time
 - connection waiting time
 - execution time



Patterns of dependability-oriented composition



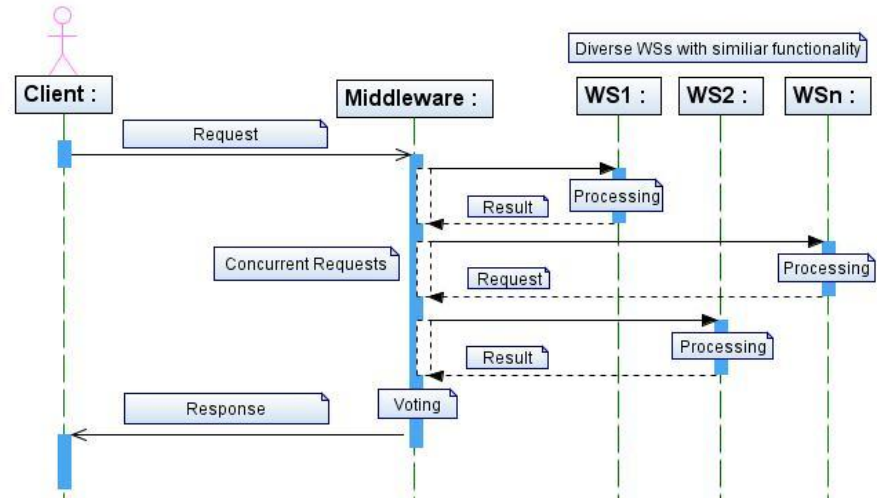
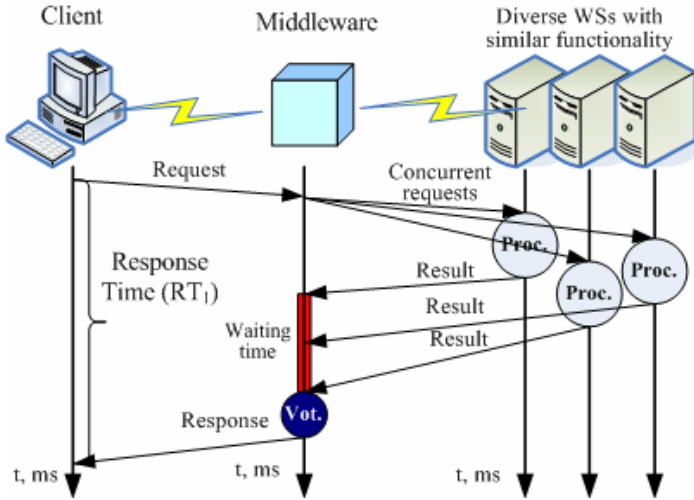
- Reliable concurrent execution pattern
- Fast concurrent execution pattern
- Adaptive concurrent execution pattern
- Sequential execution pattern



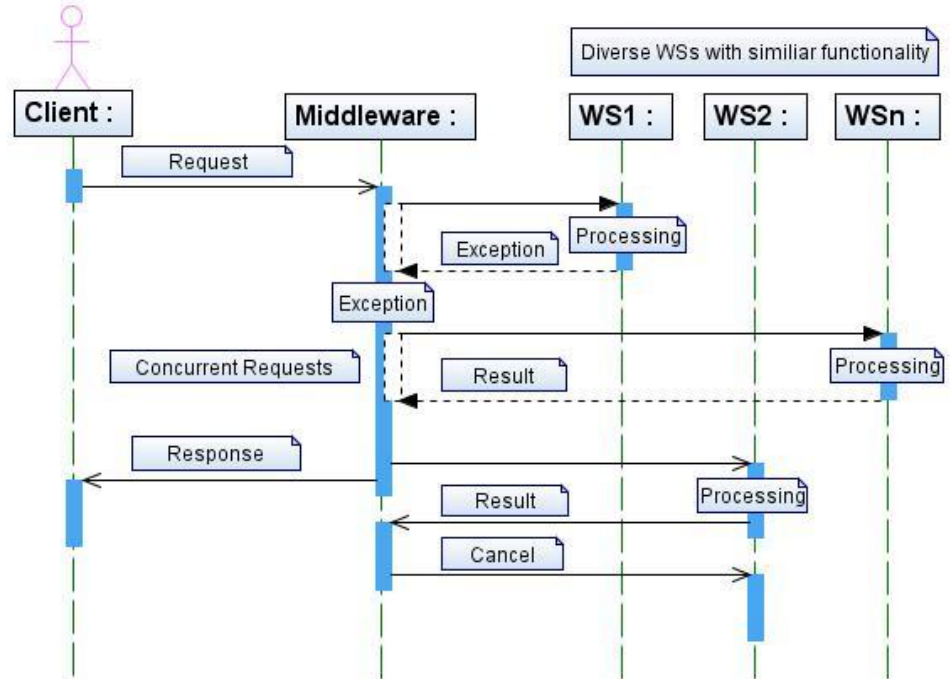
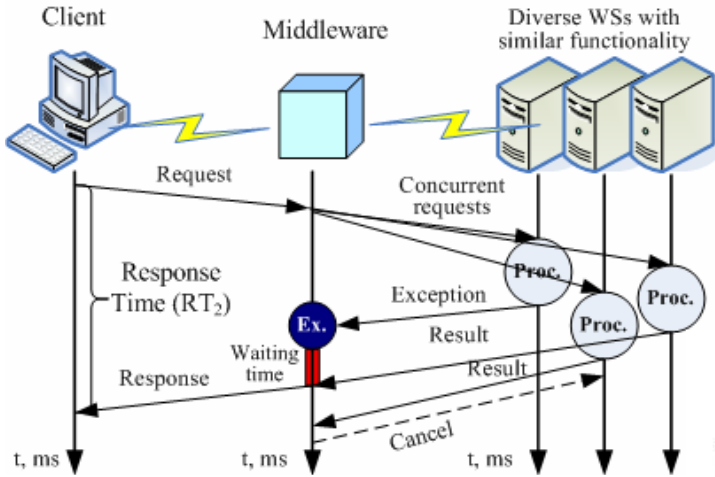
UML Sequence Diagrams



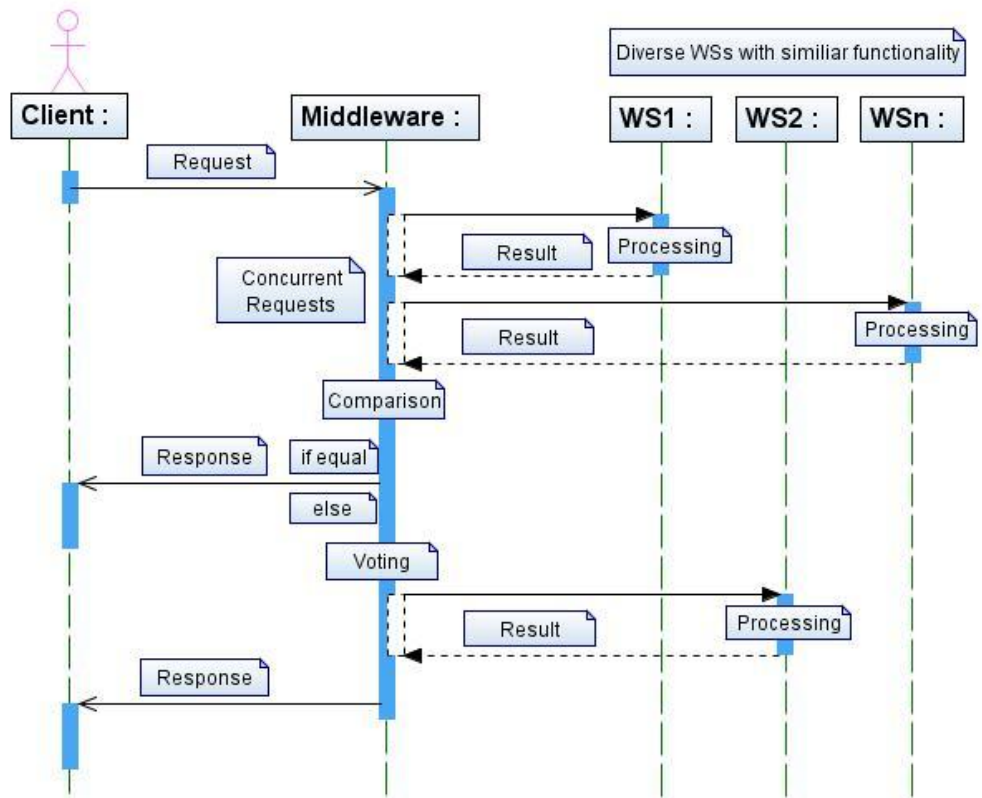
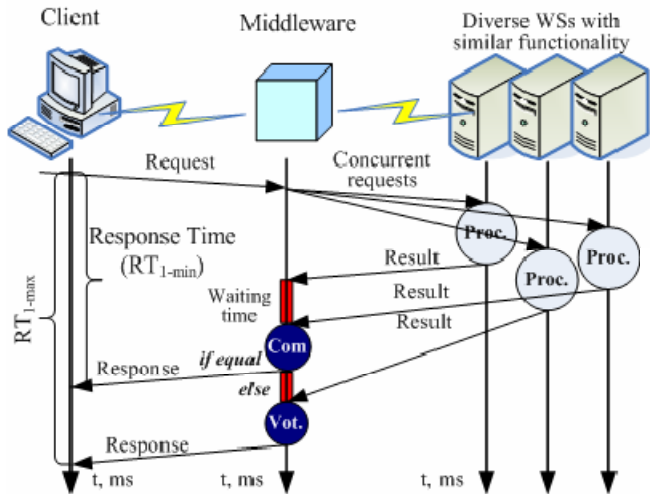
Reliable Concurrent Execution pattern



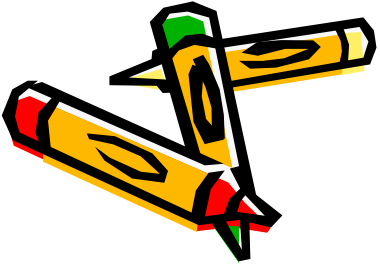
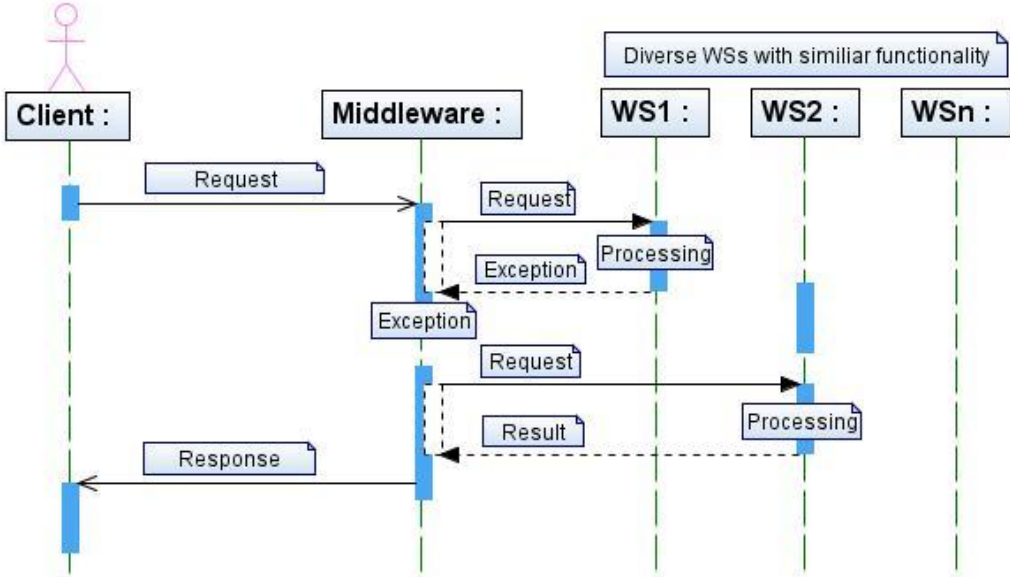
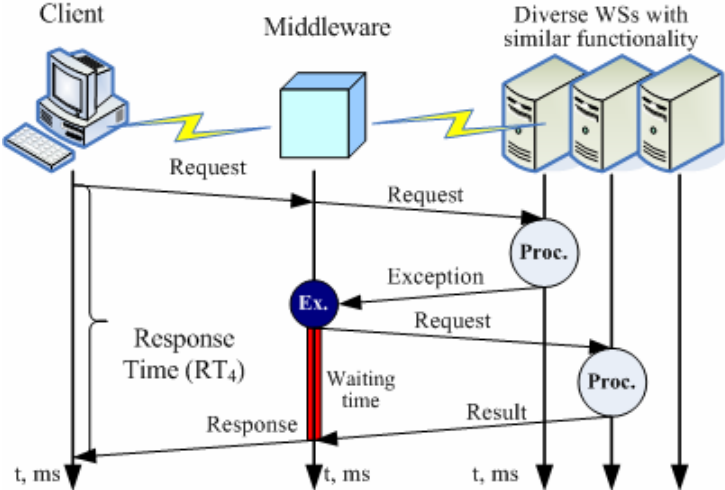
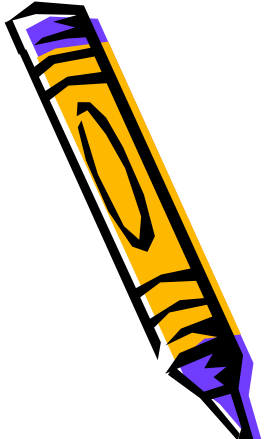
Fast Concurrent Execution pattern



Adaptive Concurrent Execution pattern



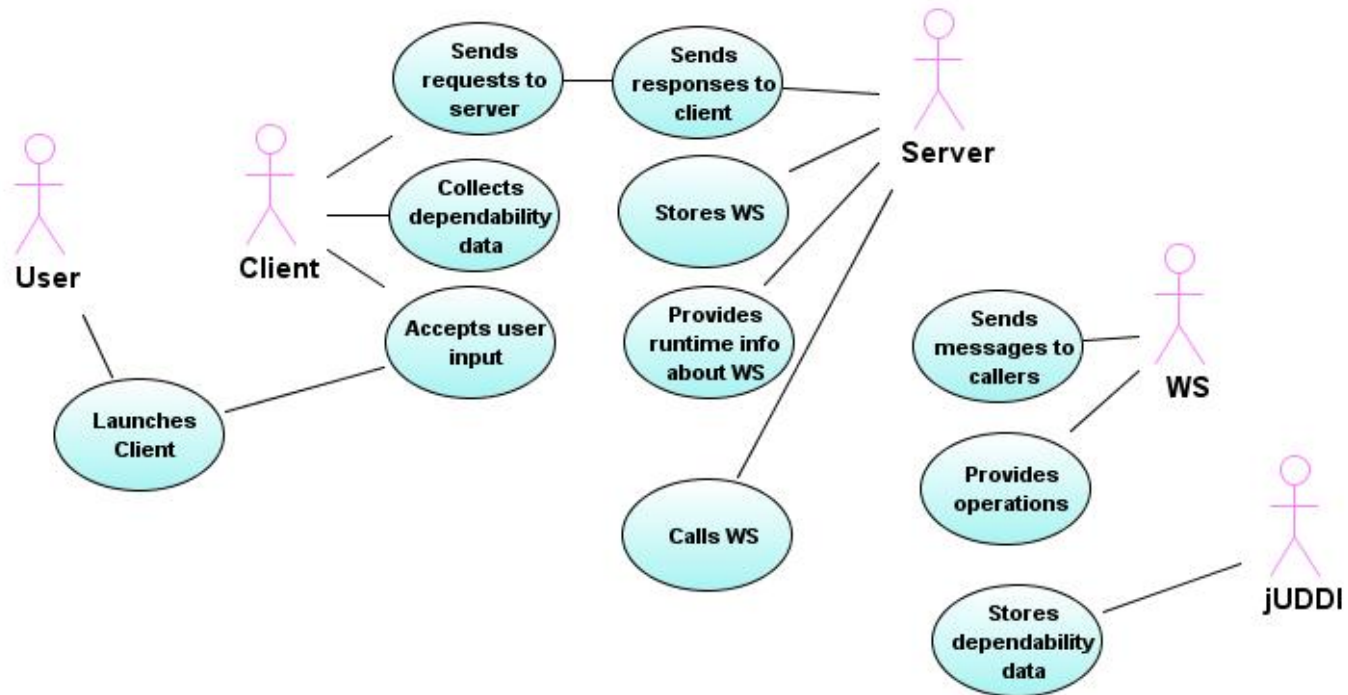
Sequential Execution pattern



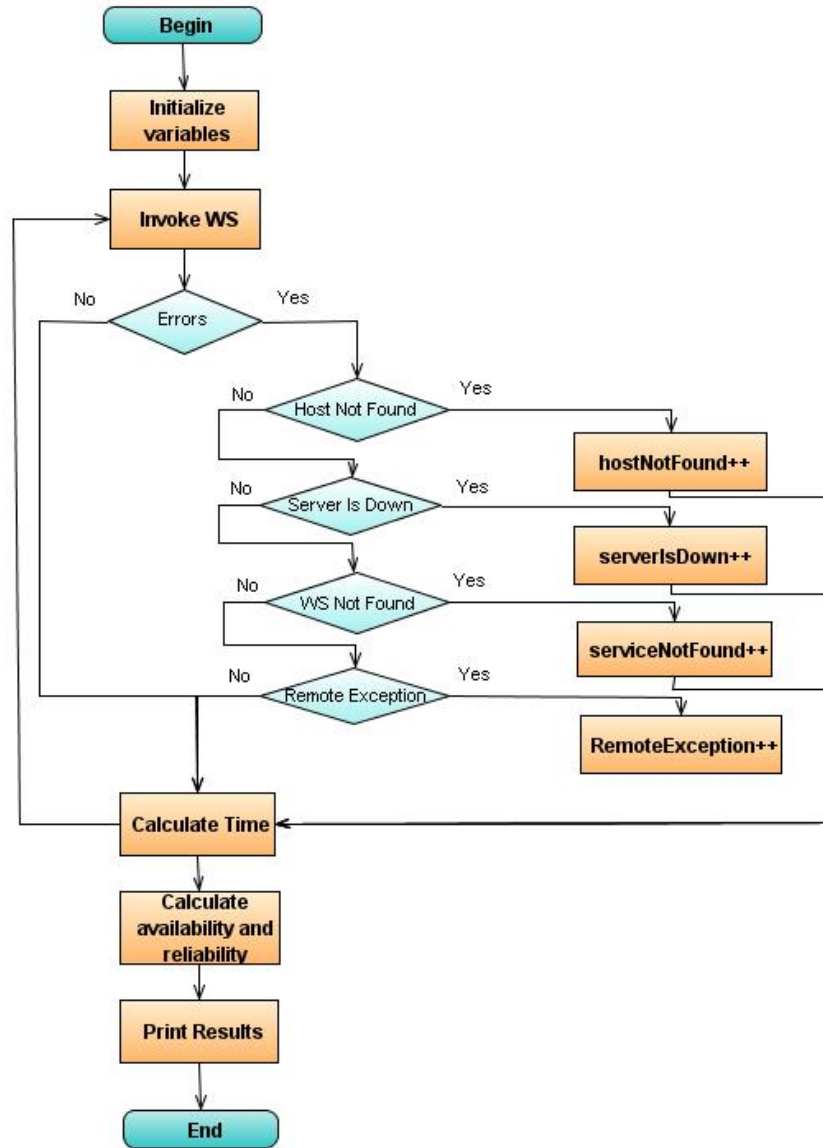
Dependability assessment based on user feedback



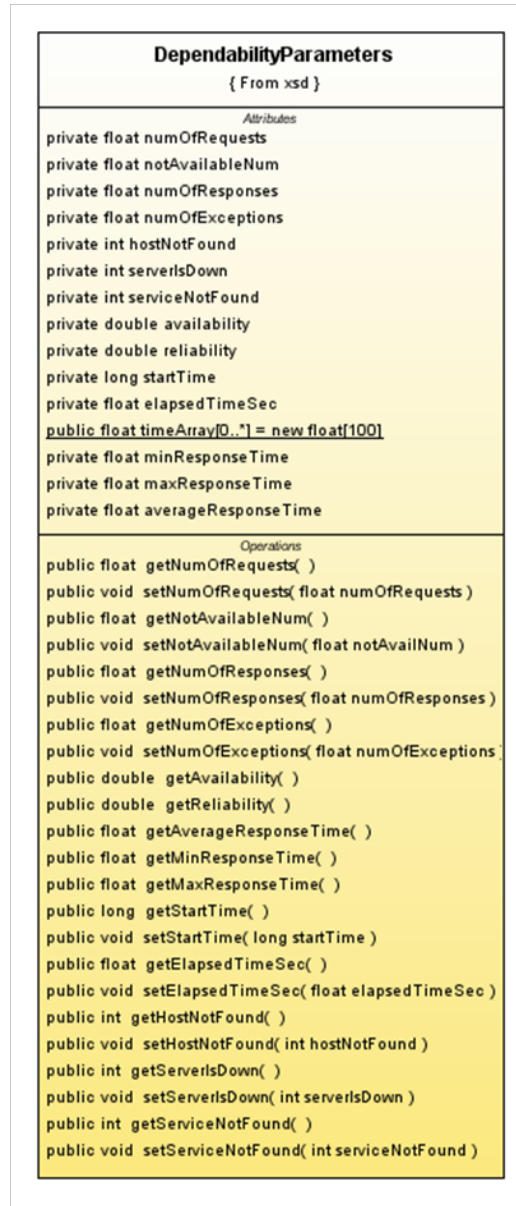
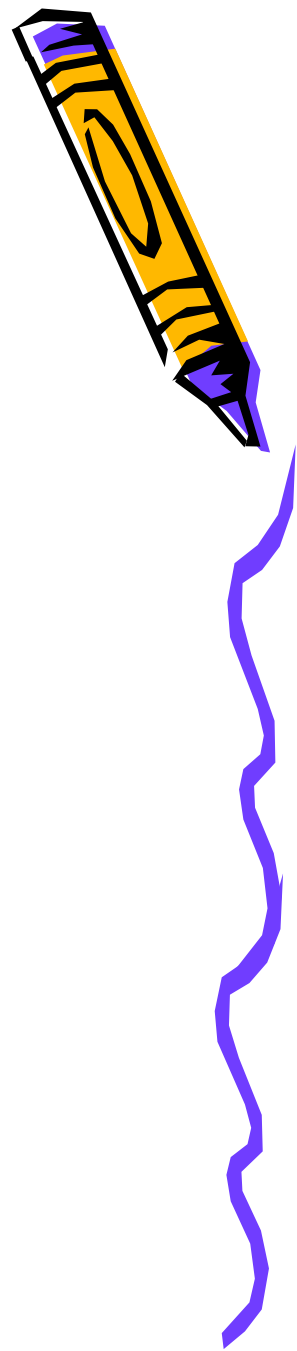
- Use Case Diagram



Client Execution Algorithm



DependabilityParameters Class Diagram



Errors Diagnostics



- Three particular types of errors were diagnosed:
 - Host Not Found (Axis2 "No route to host: connect")
 - Server Is Down (Axis2 "Connection refused: connect")
 - Service Not Found (Axis 2 "The service cannot be found for the endpoint reference (EPR)")



Dependability Characteristics Assessed



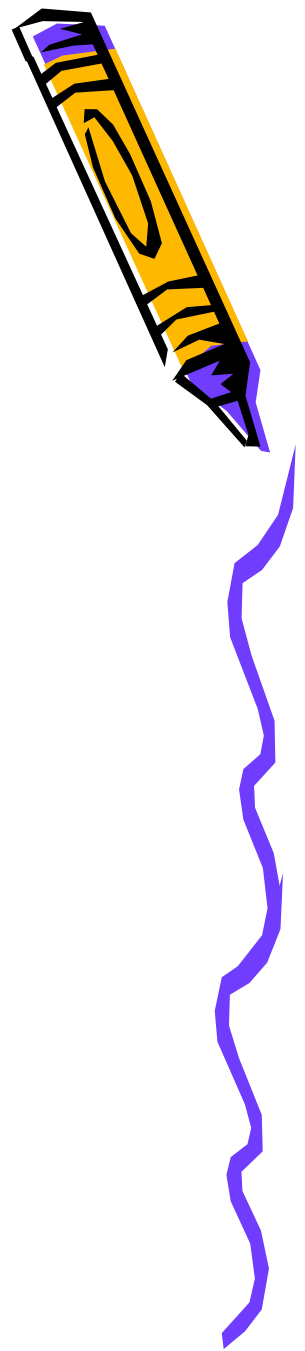
- Total number of requests made to the service
- Total number of responses received from the service (successful invocations)
- Number of times the host was not found
- Number of times the server was down (Tomcat was not running)
- Number of times the service was not found (was moved or deleted)
- Total number of times the WS was unavailable for various reasons
- Number of Remote Exceptions
- Average response time (seconds)
- Minimum response time (seconds)
- Maximum response time (seconds)
- WS availability (value in the range from 0.0 to 1.0)
- WS reliability (value in the range from 0.0 to 1.0)



Availability and Reliability calculation

$$\text{availability} = \frac{\text{numOfRequests} - \text{notAvailableNum}}{\text{numOfRequests}}$$

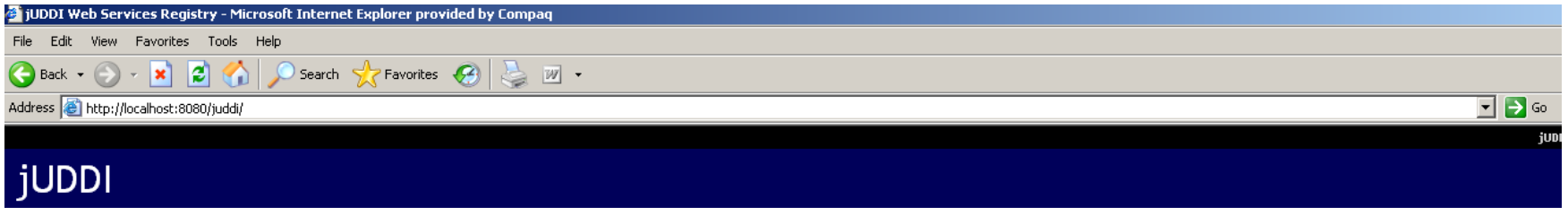
$$\text{reliability} = \frac{\text{numOfResponses}}{\text{numOfRequests} - \text{notAvailableNum}}$$



Dependability assessment based on UDDI



- jUDDI flavor of UDDI was chosen
- Is yet to be implemented



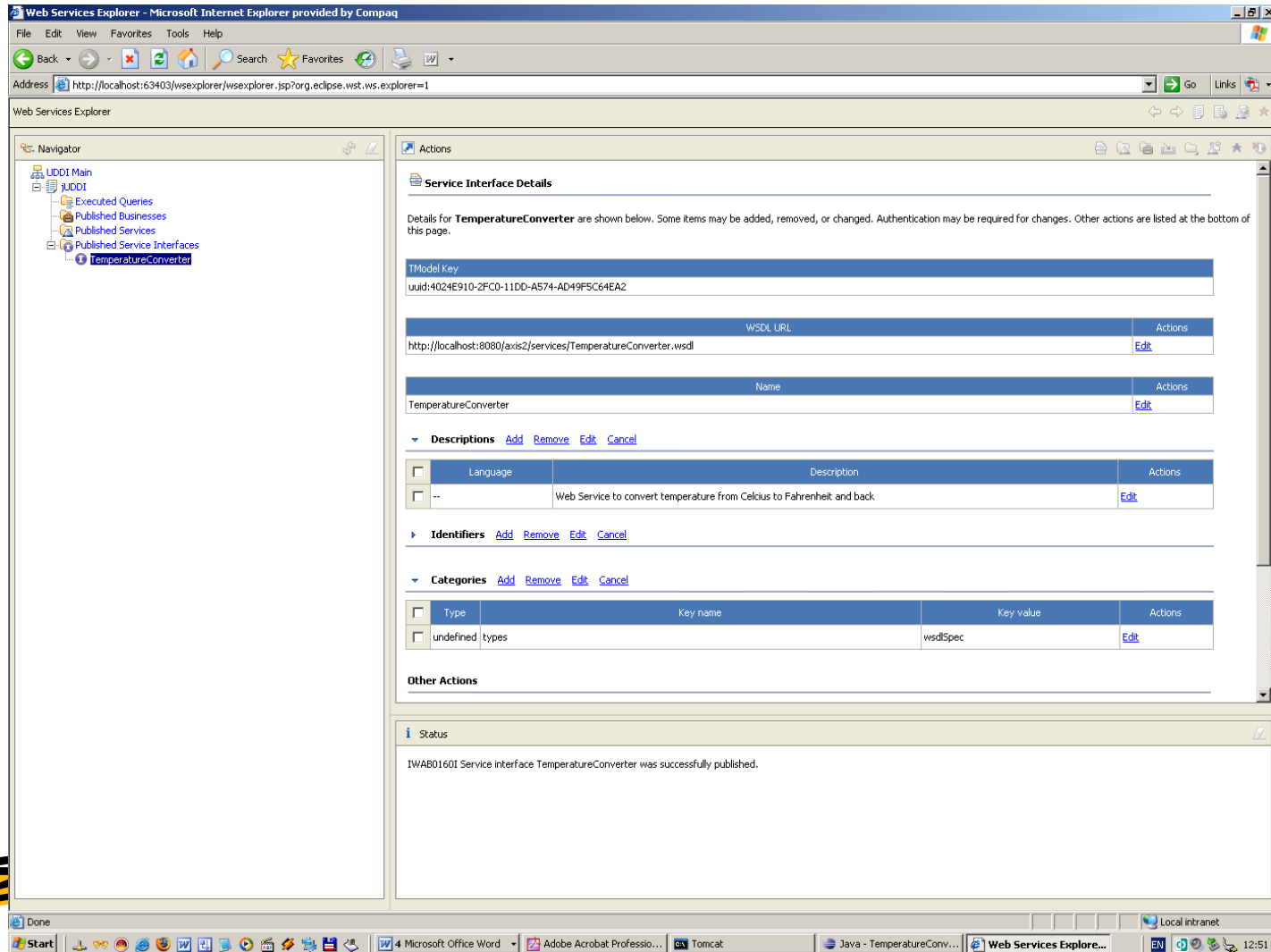
Welcome to jUDDI!

- [Validate](#) the local installation's configuration
- Have a look at the new [jUDDI Console \(Beta\)](#)
- [Visit](#) the Apache-jUDDI Home Page

If the jUDDI validation page returns some kind of error, it is invariably a configuration issue.



WS successfully published to jUDDI page



The screenshot displays the Web Services Explorer interface in Microsoft Internet Explorer. The browser's address bar shows the URL: `http://localhost:63403/wsexplorer/wsexplorer.jsp?org.eclipse.wst.ws.explorer=1`. The main content area is titled "Service Interface Details" and provides information for the "TemperatureConverter" service.

Service Interface Details

Details for **TemperatureConverter** are shown below. Some items may be added, removed, or changed. Authentication may be required for changes. Other actions are listed at the bottom of this page.

TModel Key

uuid:4024E910-2FC0-11DD-A574-AD49F5C64EA2

WSDL URL	Actions
<code>http://localhost:8080/axis2/services/TemperatureConverter.wsdl</code>	Edit

Name	Actions
TemperatureConverter	Edit

Descriptions [Add](#) [Remove](#) [Edit](#) [Cancel](#)

Language	Description	Actions
--	Web Service to convert temperature from Celcius to Fahrenheit and back.	Edit

Identifiers [Add](#) [Remove](#) [Edit](#) [Cancel](#)

Categories [Add](#) [Remove](#) [Edit](#) [Cancel](#)

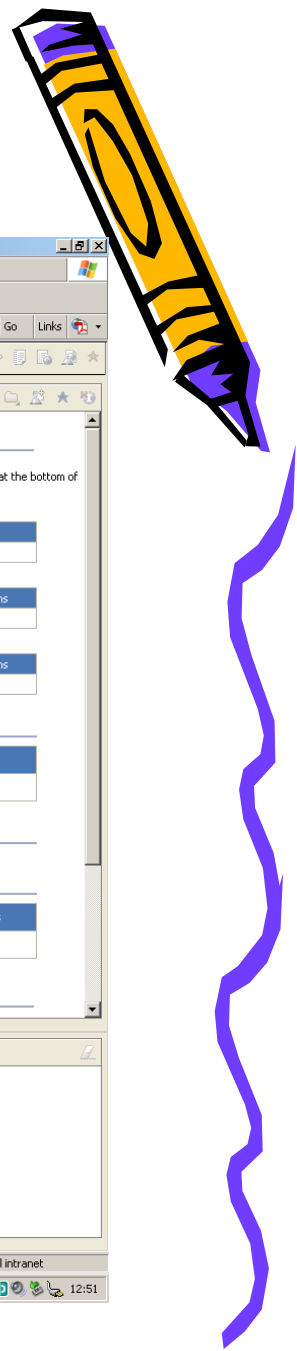
Type	Key name	Key value	Actions
<input type="checkbox"/>	undefined types	wsdlSpec	Edit

Other Actions

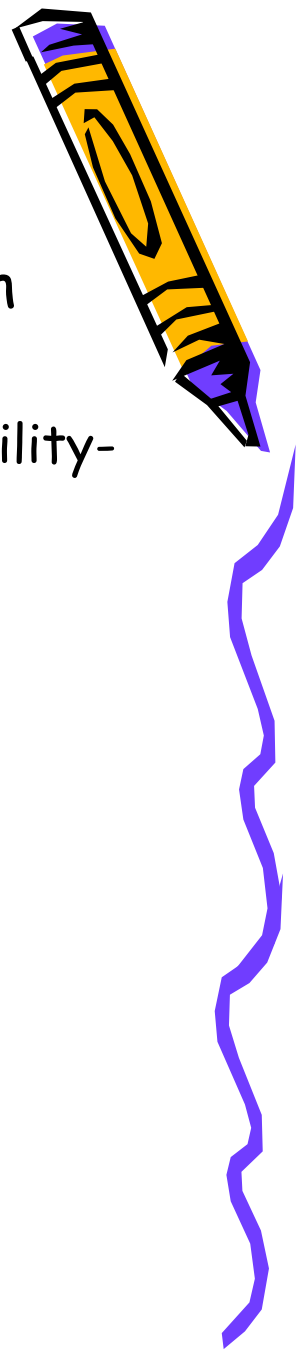
Status

IWAB0160I Service interface TemperatureConverter was successfully published.

The taskbar at the bottom shows the Start button, system tray, and several open applications: Microsoft Office Word, Adobe Acrobat Professional, Tomcat, Java - TemperatureConv..., and Web Services Explorer. The system clock indicates 12:51 on the local intranet.



Conclusion



- This paper touched upon three major research and development areas
 - UML sequence diagrams for the proposed dependability-oriented composition patterns
 - Web Service client with enhanced dependability collecting capabilities
 - Dependability assessment based on jUDDI
- Further research and development
 - enhancing UDDI with dependability capabilities
 - more rigorous classification of the registered WS's
 - supporting a search for alternative services





Thank you for your kind
attention!!!

