Realization Strategies for Rich Clients by Web Services

Master Thesis

Kingkarn Sriprasarn, 26.04.2005

RWTH Aachen
Lehrstuhl für Informatik V
Prof. Dr. Matthias Jarke
Institut für Sprach- und Kommunikationswissenschaft
Prof. Dr. Christian Stetter

Advisors:
Dr. Ralf Klamma
Dipl.-Inform. Mohamed Amine Chatti
Dipl.-Ing. Jose M. de la Rosa
Dipl.-Inform. Renate Strutz
Agenda

- Scenario
- Context of Thesis
- Objectives of the Thesis
- Software Quality Metrics
- Software Development
- Design and Implementation Issues
- Evaluation
- Conclusions & Outlook
Scenario

- CRM Prototype (existing „liNfo“ application)
- Transition from legacy systems to Internet-based systems
- Quality software with performance, productivity, and security
Context of Thesis

- Service-oriented architecture
- Web services engineering
  - Model, architecture and development methodologies
- Web user interface
  - Less performance
  - Rich features but fat clients
  - Security issue
- Quality software
  - Quality attributes
  - Measuring framework
Objectives of the Thesis

Implemented CRM Prototype

Quality Attributes
- productivity
- performance
- security

GQM plan

Metrics comparison

Evaluation for Achieving Quality CRM Web Services with Rich Clients

Web services
CRM
MVC
Rich Clients

J2EE with JSF
.NET
XTT

Pre-evaluation

Technology

J2EE with JSF
.NET
XTT

Evaluation for Achieving Quality CRM Web Services with Rich Clients
Goal-Question-Metrics (GQM)

A systematic approach involving three levels for defining a measurement model [Bas92]
Techniques and Steps for structuring GQM Plan

- Goal Template
- GQM abstraction sheet
- Structure of GQM plan

Five major aspects
- Object
- Purpose
- Quality focus
- Viewpoint
- Context

[DHL96]
GQM Plan using The Goal Template

- **Analyze** the design and development process [object]
- **for the purpose of** an evaluation [purpose]
- **with respect to** reusability, productivity [quality focus]
- **from the viewpoint of** the developer [viewpoint]
- **in the context of** quality CRM web services with rich clients [context]

- **Analyze** the final product [object]
- **for the purpose of** an evaluation [purpose]
- **with respect to** performance, security, user satisfaction [quality focus]
- **from the viewpoint of** the customer [viewpoint]
- **in the context of** quality CRM web services with rich clients [context]
## Questions and Metrics  
**according to process goal (1st goal)**

<table>
<thead>
<tr>
<th>Question</th>
<th>Q1</th>
<th>What is the design size and reusability?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Metrics</strong></td>
<td>M1</td>
<td>Number of components</td>
</tr>
<tr>
<td></td>
<td>M2</td>
<td>Number of widget (UI components)</td>
</tr>
<tr>
<td></td>
<td>M3</td>
<td>Amount of reused components</td>
</tr>
<tr>
<td></td>
<td>M4</td>
<td>Amount of reused UI components</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Question</th>
<th>Q2</th>
<th>What is the productivity of the final product?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Metrics</strong></td>
<td>M5</td>
<td>System size (source lines with comments)</td>
</tr>
<tr>
<td></td>
<td>M6</td>
<td>Number of satisfied requirements</td>
</tr>
<tr>
<td></td>
<td>M7</td>
<td>Staff time</td>
</tr>
<tr>
<td></td>
<td>M8</td>
<td>Line of Code/Man Month (LOC/MM)</td>
</tr>
</tbody>
</table>
# Questions and Metrics according to product goal (2\textsuperscript{nd} goal)

<table>
<thead>
<tr>
<th>Question</th>
<th>Metrics</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q3</td>
<td>M5</td>
<td>Cost of implementation (line of codes)</td>
</tr>
<tr>
<td></td>
<td>M7</td>
<td>Cost of implementation (staff time)</td>
</tr>
<tr>
<td></td>
<td>M9</td>
<td>Cost of software (development tools)</td>
</tr>
<tr>
<td>Q4</td>
<td>M10</td>
<td>Time to first byte (TTFB)</td>
</tr>
<tr>
<td></td>
<td>M11</td>
<td>Time to last byte (TTLB)</td>
</tr>
<tr>
<td></td>
<td>M12</td>
<td>Throughput (request per second)</td>
</tr>
<tr>
<td></td>
<td>M13</td>
<td>Time for searching and filtering data</td>
</tr>
<tr>
<td>Q5</td>
<td>M6</td>
<td>Number of satisfied requirements</td>
</tr>
<tr>
<td>Q6</td>
<td>M14</td>
<td>SSL implementation</td>
</tr>
</tbody>
</table>
JSF’s MVC Implementation

- JSF in JSP page is responsible for View part
- JSF provides a servlet - FacesServlet as Controller part
- Web services are a partial of Model incorporating with JavaBeans
ASP.NET's MVC Implementation

- WebForms for View part (in ASPX page)
- Code Behind as Controller part
- Business Entities, DataAccess Components and Web service are the Model part

C# Code (.aspx.cs)

```csharp
public class ControlController
{
    public ActionResult Index()
    {
        return View();
    }
}
```

ASP.NET Page (.aspx)

- Realization Strategies for Rich Clients by Web Services
- 13 April 2005

Web Service (.asmx)
### Deployment of XTT

- Local deployment as an executable jar
- Browser deployment as an applet using the Java plug-in
- Java Web Start deployment

- Inconsistent to the J2EE and .NET prototypes
- No evaluation
Design & Implementation Issues

- **Pagination**
  - Simplest and common way to break up large amounts of data into more manageable chunks
  - Efficient data retrieval and paging is a trade-off between memory usage and response time
  - JSF provides **dataTable**, while ASP.NET provides **DataGrid**
  - Caching System might be concerned for real-world app

- **Filtering**
  - JDBC RowSet provides FilteredRowSet and WebRowSet
  - Offer property to cut down the number of rows
  - Do filtering without having to execute a query on the data source
  - Criteria is the high end and low end of the range
Design & Implementation Issues (Cont.)

- User-defined Data Types for Web Services and Interoperability
  - `ArrayList` and `String[]` return from Java Web Service
  - Performance comparable between Java Web Service return `String` and `Document`
  - `DataSet` return from .NET Web Service in `DiffGram` format
  - `DataSet` can be serialized/deserialized to XML and passed in a SOAP message

- Security for Web Services using SSL
  - SSL is transport-layer security, not message-layer security
  - Overhead is less, no need for extra memory to hold complete message; can process messages in parallel
Web Service Return Data & Interoperability

<table>
<thead>
<tr>
<th>Service</th>
<th>Method &amp; Signature</th>
</tr>
</thead>
<tbody>
<tr>
<td>CustomerService</td>
<td>String[] getCustomerAddress(String criteria)</td>
</tr>
<tr>
<td>SearchService</td>
<td>String getCustomerListAsString(String criteria)</td>
</tr>
</tbody>
</table>

- User-defined data type is more flexible, designed for interoperability.
- Though there may be equivalent data types, it is difficult to find interoperable data types (ArrayList in Java and ArrayList in .NET are not interoperable)

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <getCustomerAddressResponse>
      <soapenc:arrayType="xsd:string[65]"
xmlns:soapenc="http://schema.xmlsoap.org/soap/encoding">
        <Item>D_B23328856602619</Item>
        <Item>false</Item>
        <Item>7</Item>
        <Item>Suchbegriff 1111</Item>
        <Item>Boehringer Mannheim</Item>
        <Item>Name 1111</Item>
        <Item>Weiterer Name 1111</Item>
        ..............
      </soapenc:arrayType>
    </getCustomerAddressResponse>
    <soapenv:Body>
      <getCustomerListAsStringResponse xsi:type="xsd:string">
        <Results>
          <Row>
            <SA_ID>D_B283728856602619</SA_ID>
            <SA_Suchbegriff>Suchbegriff 1111</SA_Suchbegriff>
            <SA_Fon>012/1111</SA_Fon>
          </Row>
        </Results>
      </getCustomerListAsStringResponse>
    </soapenv:Body>
</soapenv:Envelope>
```
Web Service Return Data & Performance

FilterService

<table>
<thead>
<tr>
<th>Return Type</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>getCustomerListAsString(String criteria)</td>
</tr>
<tr>
<td>Document</td>
<td>getCustomerListAsDocument(String criteria)</td>
</tr>
</tbody>
</table>

- Document is the object that can represent XML document
- String return type can be interpreted easier in terms of programmability
- Document uses more overhead time for invoking

Using common data type String gives
- better performance and
- less sophisticated programming
### Realization Strategies for Rich Clients by Web Services

**April 2005**

<table>
<thead>
<tr>
<th>No.</th>
<th>Metrics</th>
<th>J2EE with JSF</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Number of components - classes</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>- methods</td>
<td>14</td>
</tr>
<tr>
<td>2</td>
<td>Number of widgets (UI components)</td>
<td>148</td>
</tr>
<tr>
<td>3</td>
<td>Amount of reused components</td>
<td>4 (33.33%)</td>
</tr>
<tr>
<td>4</td>
<td>Amount of reused UI components</td>
<td>44 (29.7%)</td>
</tr>
<tr>
<td>5</td>
<td>System size (source lines)</td>
<td>5,562</td>
</tr>
<tr>
<td>6</td>
<td>Number of satisfied requirements</td>
<td>all</td>
</tr>
<tr>
<td>7</td>
<td>Staff time (days)</td>
<td>72</td>
</tr>
<tr>
<td>8</td>
<td>Line of code/Manmonth</td>
<td>2,394.75</td>
</tr>
<tr>
<td>9</td>
<td>Cost of Software</td>
<td>-</td>
</tr>
<tr>
<td>10</td>
<td>Average Time to First Byte</td>
<td>164.80</td>
</tr>
<tr>
<td></td>
<td>- TTFB (msecs)</td>
<td>76.22</td>
</tr>
<tr>
<td>11</td>
<td>Average Time to Last Byte</td>
<td>165.50</td>
</tr>
<tr>
<td></td>
<td>- TTLB (msecs)</td>
<td>77.44</td>
</tr>
<tr>
<td>12</td>
<td>Throughput (requests per second)</td>
<td>6.00</td>
</tr>
<tr>
<td>13</td>
<td>Time for searching and filtering data (msecs)</td>
<td>3,297</td>
</tr>
<tr>
<td>14</td>
<td>SSL Implementation</td>
<td>yes</td>
</tr>
</tbody>
</table>

**Evaluation**

- **Different in design issue**
  - .NET: one class ties to one .aspx page
  - J2EE: more classes

- **Amount of reused components and UI**
  - J2EE is rather efficient than ASP.NET
  - Quite similar for reused UI components

- **System size shows obvious result**
  - amount is double in J2EE
  - Web forms brings less code than JSF

- **LOC/MM**
  - Developer skill
  - RAD Tools

- **Performance metrics**
  - .NET performs excellent execution.
  - ASP.NET is two fold faster than J2EE

- **Time to execute web service**
  - much better performance for Java WS

* It is optional for Visual Studio .NET. There are many open source software for developing .NET solutions such as Cassini as web server and #develop as IDE. 

April 2005

Realization Strategies for Rich Clients by Web Services
Conclusions

1. **Comparative study of the state of the art in SOA and Rich clients**
   - Survey model and design of application utilized SOA
   - Survey technology to implement Web services with Rich clients
   - Three approaches for developing Web services with Rich clients which are J2EE with JSF, ASP.NET, and XTT are studied

2. **Implementation CRM Web services prototype**
   - Specification and design CRM Web services using MVC design pattern
   - Rich user interface is realized

3. **Evaluation for achieving a quality software**
   - Study GQM approach to define the GQM plan
   - Applied GQM for extracting metrics from the specify three software quality attributes which are performance, productivity, and security
   - Measuring metrics from the prototype and making a comparison of them
Outlook

- **Methodologies and Implementation**
  - Apply other designs and architecture approaches such as Model Driven Architecture
  - Improve the rich client functionalities for web browsers
  - Use web services to build user interface such as using XML-based user interface language

- **Web Services Composition**
  - Quality of the composition of existing web services compared to the creation the new one web service

- **Web Services Interoperability**
  - Use WS-I (Web Service Interoperability Organization) guidelines to implement application

- **Evaluation**
  - Extending the evaluation framework by increasing more software quality attributes
Thank you