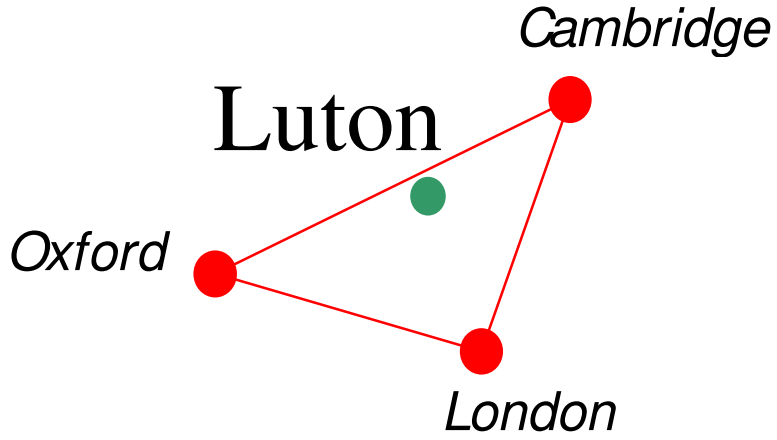
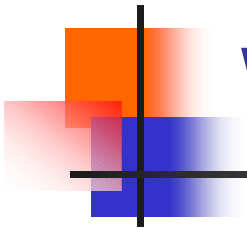
JICC9 – January 31, 2005

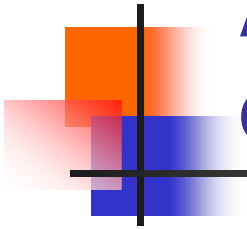
- A gentle transition from Java programming to Web Services using XML-RPC
- by Marc Conrad & Tim French
- University of Luton





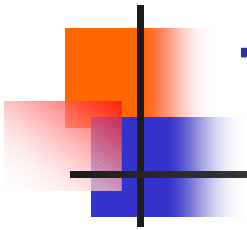
Web Services

- Important role in e-commerce and other distributed applications.
- Protocols are XML-based (eg. SOAP – Simple Object Access Protocol).
- Influenced by W3C, Apache, SUN, …
- Should be integrated in *any* Computer Science Curriculum (somehow).



At JICC 7, Draganova identified the central topics in teaching Web Services

- Discovering a location of a service provider
- Discovering provided services
- Providing a way of communication with a particular service
- Providing a way to execute available functions
- Providing a standard messaging
- Providing a way of data representation



The solution

- We present a teaching unit that
  - addresses all these topics
  - is based on Java
  - has no other prerequisites
  - covers two weeks
  - available for free (open source)
    - (from http://perisic.com/xmlrpc)
  - relevant in practical applications

What is XML-RPC?

Slide from the teaching material

- It's a spec and a set of implementations that allow software running on disparate operating systems, running in different environments to make procedure calls over the Internet
- It's remote procedure calling using HTTP as the transport and XML as the encoding. XML-RPC is designed to be as simple as possible, while allowing complex data structures to be transmitted, processed and returned.

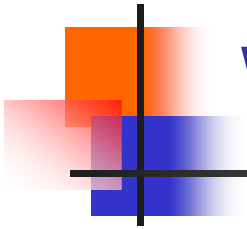
Example: "Hello World" Client

```
import java.util.*;
import org.apache.xmlrpc.*;
public class HelloClient {
    public static void main (String [] args) {
    try {
            XmlRpcClient server = new XmlRpcClient("http://localhost/RPC2");
            Object result = server.execute("sample.hello", new Vector() );
            System.out.println("The result is: "+result.toString());
    } catch (Exception exception) {
            System.err.println("JavaClient: " + exception);
            }
    }
}
```

Example "Hello World" Server

```
import org.apache.xmlrpc.*;
public class HelloServer {
    public String hello() {
            return "Hello from Marc Conrad";
            }
    public static void main (String [] args) {
    try {
            WebServer server = new WebServer(80);
            server.addHandler("sample", new HelloServer());
            server.start();
    } catch (Exception exception) {
            System.err.println("JavaServer: " + exception);
            }
    }
}
```

Student Activities in the "Hello World" Example



- Run Client and Server
- Change the server, so that it displays a different text, e.g. your name.
- Use the command *ipconfig* to get your ip number. Give your ip number to a friend, and run the server. Ask your friend to access your server.
- Change the client by replacing "localhost" with a valid ip number.
- Add the line uk.co.wilson.xml.MinML.xmlinfo = true; to make the xml visible.

Example 2: Chat

```
import javax.swing.*;
public class PrimitiveChat {

// The procedure:
 public String printText(String str) {
    System.out.println( "Received: "+str);
    return "ok";
    }
  public static void main (String [] args) {
              String input;
              PrimitiveChat pc = new PrimitiveChat();
              do {
                 input = JOptionPane.showInputDialog(
                                        "Enter your message");
              if( input != null ) {
         pc.printText(input);
                 }
    } while ( input != null );
  System.exit(0);
  }
}
```

- Activity 2: Make the procedure call a *remote* procedure call

```
public class Example {
 public int sum(int a, int b) {
    return a+b;
    }
 public static void main (String [] args) {
  Example eg = new Example();
  eg.sum(13,17);
 }
}
```

Local procedure call

Slide from the teaching material

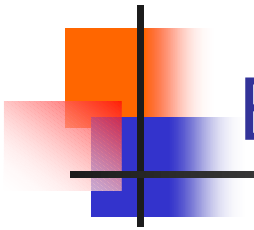
Remote procedure call (RPC)

Client

```
[…]
eg.sum(13,17);
[…]
```

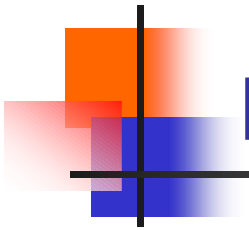
Server

```
[…]
public int sum(int a, int b) {
    return a+b;
    }
[…]
```



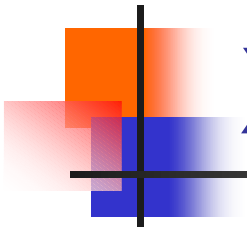
Back to Draganova (1)

- Discovering a location of a service provider
  - Activity 1: move from “localhost” to IP-address.
- Discovering provided services
  - Activity 2: What is the name of the remove procedure?
- Providing a way of communication with a particular service
  - Activity 1: Examine XML code that is sent; identify the name of the method and parameters in the XML (human readable..).



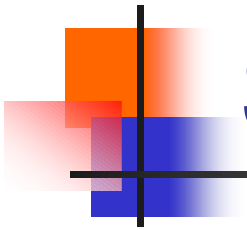
Back to Draganova (2)

- Providing a way to execute available functions
  - Activity 2: Encounter similarities between local and remote procedure calls.
- Providing a standard messaging
  - Activity 2: Implementation of a simple “Chat”-program
- Providing a way of data representation
  - Lecture slides: the XML-RPC specification is covered in *full,* i.e. data representations for int, string, arrays, etc. are all covered.



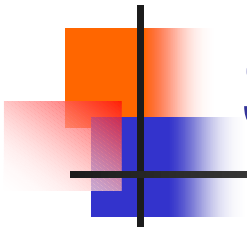
XML-RPC and SOAP

- Wikipedia (2005): "*[XML-RPC] was first created by Dave Winer of UserLand Software in 1995 with Microsoft. However Microsoft considered it too simple, and started adding functionality. After several rounds of this, the standard was no longer so simple, and became what is now SOAP.*"
- *So, XML-RPC is*
  - *an excellent starting point for understanding the principles behind SOAP*
  - *much more lightweight*



Student Experience

- Taught in 2003 (100 students) and 2004 (160 students)
- Feedback positive (anecdotal evidence)
- Natural transition from individualistic work to team working (Activity 1), enhancing soft skills
- Encouragement to go beyond the tasks of the assignment (Activity 2)



Summary

- Teaching material covers two weeks
- Introduces web services on a basic level
- Does not require any prerequisites except Java programming knowledge
- Focusing on central topics in the context of web services
- Based on Open Source standards (Apache)
- All material available at http://perisic.com/xmlrpc (free).