Introduction

This course “Software Architecture with Java” will discuss the following topics:

Java servlets
Java Server Pages (JSP’s)
Java Beans
JDBC, connections to RDBMS and SQL
XML and XML translations

These tools will then be used to construct webapplications according to different models.

Literature:

Steelman & Murach,
Murach’s Java Servlets and JSP.
Mike Murach & Associates Inc, 2003

Software:

J2SE, Java Standard Edition, version 1.4.x
Tomcat, www.apache.org, we use version 4.1.18
MySQL, www.mysql.org

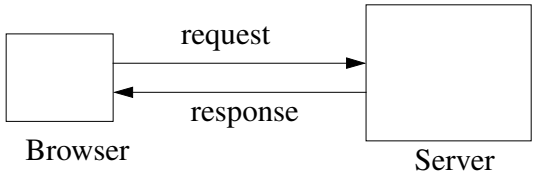
All of this is freeware and can be downloaded, (or installed from the CDROM that comes with the book)

What is a web application?

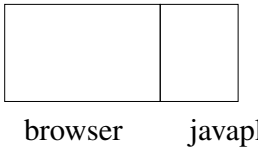
That is a set of web pages that are generated in response to a user request.

Examples:

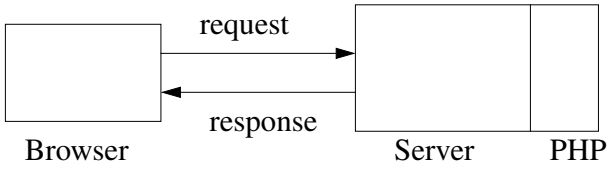
A browser that requests html-pages from a server. In the simplest case, these pages are static.



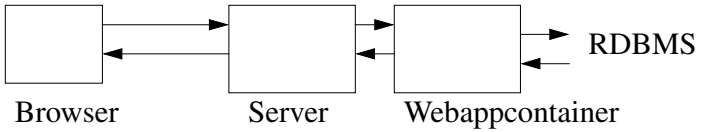
A browser that executes one or more Java applets. This is all done inside the browser (the client), therefore this is known as a “fat client”.



A browser that requests dynamic pages from a server. These can be generated by PHP or some similar system. This kind of system is called a “thin client” since the processing is done in the web server.



A browser that requests dynamic pages from a server. The server somehow forwards these requests to a web-container that generates and returns html-pages.



We are to discuss the last kind of systems.

Why do we need a special webcontainer?

An application may result in several request/response pairs. Since HTML is stateless (has no memory), we need something that can create resources, remember and identify connections and keep resources alive.

An example of a webcontainer is Jakarta Tomcat. It is not a complete J2EE container, it can therefore be called a servlet container.

There is a standard framework definition for Javabased webcontainers. This is known as J2EE, The Java2 Enterprise Edition. Servlets and JSP's are part of the J2EE specification. We will use version 2.3 of the Servlet spec. and 1.2 of the JSP specification.

Tomcat implements part of the J2EE specification but not all of it.

What is a servlet?

A servlet is a Java class that is derived from the HTTPServlet class.

It is executed on behalf of the webserver when requested from a user. The servlet receives the request, and produces html code that is returned via the response mechanism. Since the servlet is a program it can generate dynamic html code based on different condititions.

Since this is pure Java it is transportable and follows an official standard. It is multithreaded and only one instansiation is needed.

A simple servlet can look like

```
package serv;

import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;

public class Serv extends HttpServlet {

   public void doGet(HttpServletRequest request,
                     HttpServletResponse response)
      throws IOException, ServletException {
         response.setContentType("text/html");
        PrintWriter out =  response.getWriter();
        out.println("<H1> hello </H1>");
   }

   public void doPost(HttpServletRequest request,
              HttpServletResponse response)
      throws IOException, ServletException {
       doGet(request, response);
   }
}
```


This will receive a request and generate an HTML header-line as response to the browser.

To compile this you need some jar-fi les that are not part of the J2SE, but they are included in tomcat and in J2EE implementations.

To run it you need Tomcat or some other container. Tomcat can execute the servlet directly on reqeust from the browser, but the normal use of a servlet is as part of a deployed application.

A deployed application is a complete system with different components that are kept together with a description fi le.

What is a JSP?

Java Server Pages consists of HTML-code, Java code embedded in a script language (NOT Javascript!) and other JSP tags.

The purpose of a JSP is to generate HTML code.

Technically a JSP is compiled into a servlet by the web-container.

Since it is Java it is transportable, but you do not need to compile it, therefore it is easier to work with.

JSP is extentable, i. e. you can introduce new tags that extends the functionality of JSP.

An example of a JSP is:

```
<HTML>
<!--    Copyright (c) 1999 The Apache Software
        Foundation.  All rights reserved.-->

<HEAD><TITLE>
      Calendar: A JSP APPLICATION
</TITLE></HEAD>

<BODY BGCOLOR="white">
   <jsp:useBean id="table"scope="session"
    class="cal.T ableBean"/>
<%
      String time = request.getParameter (“  time');
%>
<FONT SIZE=5> Please add the following event:
<BR> <h3> Date <%= table.getDate() %>
<BR> Time <%= time %> </h3>
</FONT>
<FORM METHOD=POST ACTION=cal1.jsp>
<BR>
<BR> <INPUT NAME="date"TYPE=HIDDEN
VALUE="curr ent">
<BR> <INPUT NAME="time"TYPE=HIDDEN
         VALUE=<%= time %>
<BR> <h2> Description of the event <INPUT
      NAME="description"TYPE=TEXT SIZE=20>
</h2>
<BR> <INPUT TYPE=SUBMIT VALUE="submit">
</FORM></BODY></HTML>
```

What is a Bean?

A Bean is an ordinary Java class that adheres to the Bean rules.

Beans can be used for anything, but in a webapplication they are often used to perform datahandling.

An example:

```
package test;

public class TestBean  {
  String name;
  Integer age;

  public TestBean() {  }

  public String getName() {
    return name;
  }

  public void setName(String newName) {
    name = newName;
  }

  public Integer getAge() {
    return age;
  }

  public void setAge(Integer newAge) {
    age = newAge;
  }}
```


What is JDBC, RDBMS and SQL?

JDBC (Java DataBase Connectivity) is a serverbased, pure Java piece of software that allows Java classes to connect to a database and to perform queries and updates via SQL-statements. Each RDBMS (Relation Database managing System) requires a vendor specifi c JDBC driver.

SQL (Standard Query Language) is a language that allows you to maintain your database. You can store and retrieve data.

What is XML?

XML (Extensible Markup Language) is a markup language that is mainly used to store structured data. It is somewhat similar to HTML but there is no predefined tags. You can use any tags, but you must yourself assign meaning to the tags.

The benefit of XML is that it can be translated into other formats, i. e. HTML, PDF etc. Therefore it can be used as a general, transportable format.

XML can be translated into HTML by XSLT, Extensible Style Language Translations.

An example:

```
<?xml version="1.0" encoding="ISO8859-1"?>
<?xml-stylesheet type="text/xsl"
                                href="employees.xsl"?>

<employees>
  <employee empid="1">
   <name>Fredrik Ålund</name>
     <department depid="3">Services</department>
  </employee>

  <employee empid="2">
   <name>Helena Larsson</name>
     <department depid="3"> Services </department>
   </employee>
</employees>
```

Case sensitive, more strict than HTML. You must use closing tags for ALL elements.

How do you combine these into an application?

One way to organize this is the MVC model. This means Model, View and Control.

Model is the datahandling and business logic

View is the presentation, i. e. the HTML pages

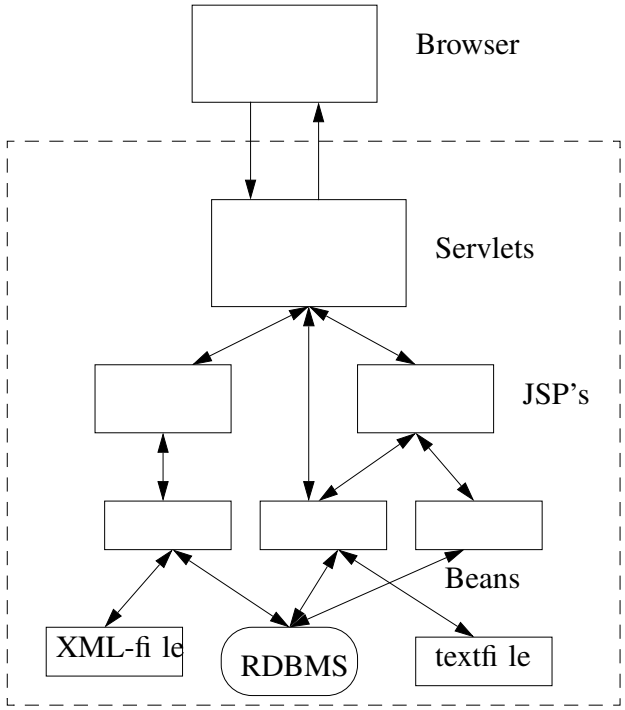
Control is the scheduler, the one that distributes the work.

XML-fi les are used to describe the application.

Beans are good to obtain and maintain data. They can connect to databases to get/store data. Since they are pure Java you have the full power of the language with all datatypes.

JSP's are used to produce HTML. In addition you can embed JAVA in a JSP. However this makes them more dif-fi cult to read and to maintain because you mix a number of languages. Therefore they are most often used to pro-duce presentation, i. e. HTML, of data produced by some beans.

Servlets are used as control. They receive the requests from the client, distributes the work to JSP's and beans and return the response. Usually you avoid generating HTML in the servlet because too much of that makes the servlet cluttered with strange HTML-strings hard to main-tain.



We have no special development tools available here but there are such tools:

Full J2EE tools

Oracle JDeveloper. Complete environment, arbitrary database. Free for personal use. High quality. Se www.oracle.com

Borland Jbuilder Enterprise. A 30-day trial version can be downloaded from www.borland.com

Eclips, Open Source but need plugins. www.eclipse.org

Just JSP and Servlets:

Sun Java Studio standard, www.sun.com

NetBeans, www.netbeans.org