

Rapid component-based development for Condor job submission



Jae Woo Lee

David W Braun

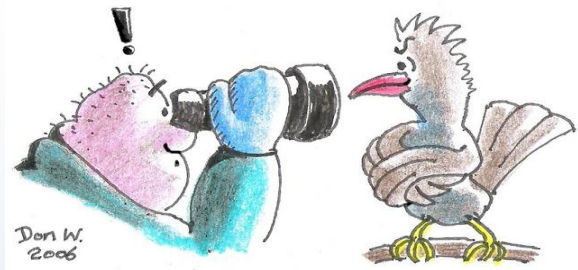
Carol X Song

Nov 16, 2008

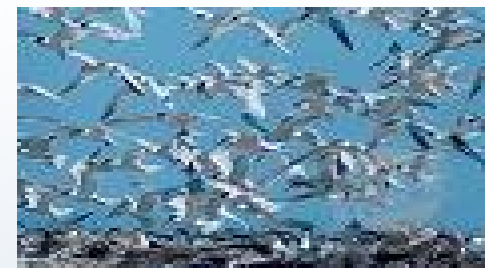
GCE08



Bird Identification Problem



5+ sites

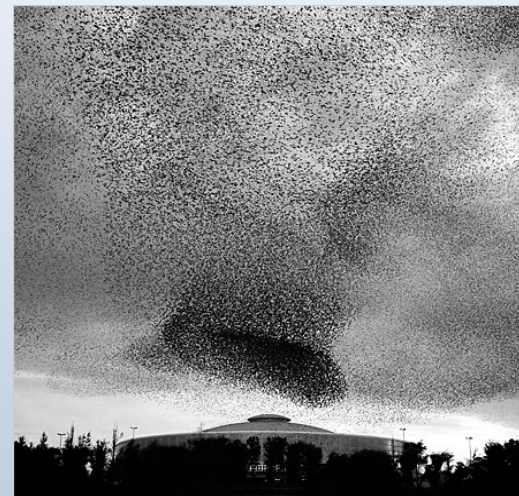
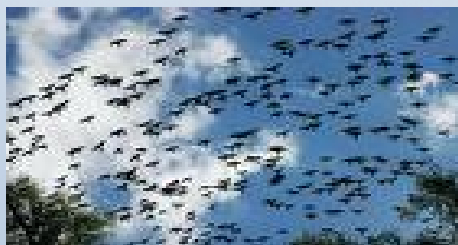


8 Condor Pools 20K Slots

Machine Types

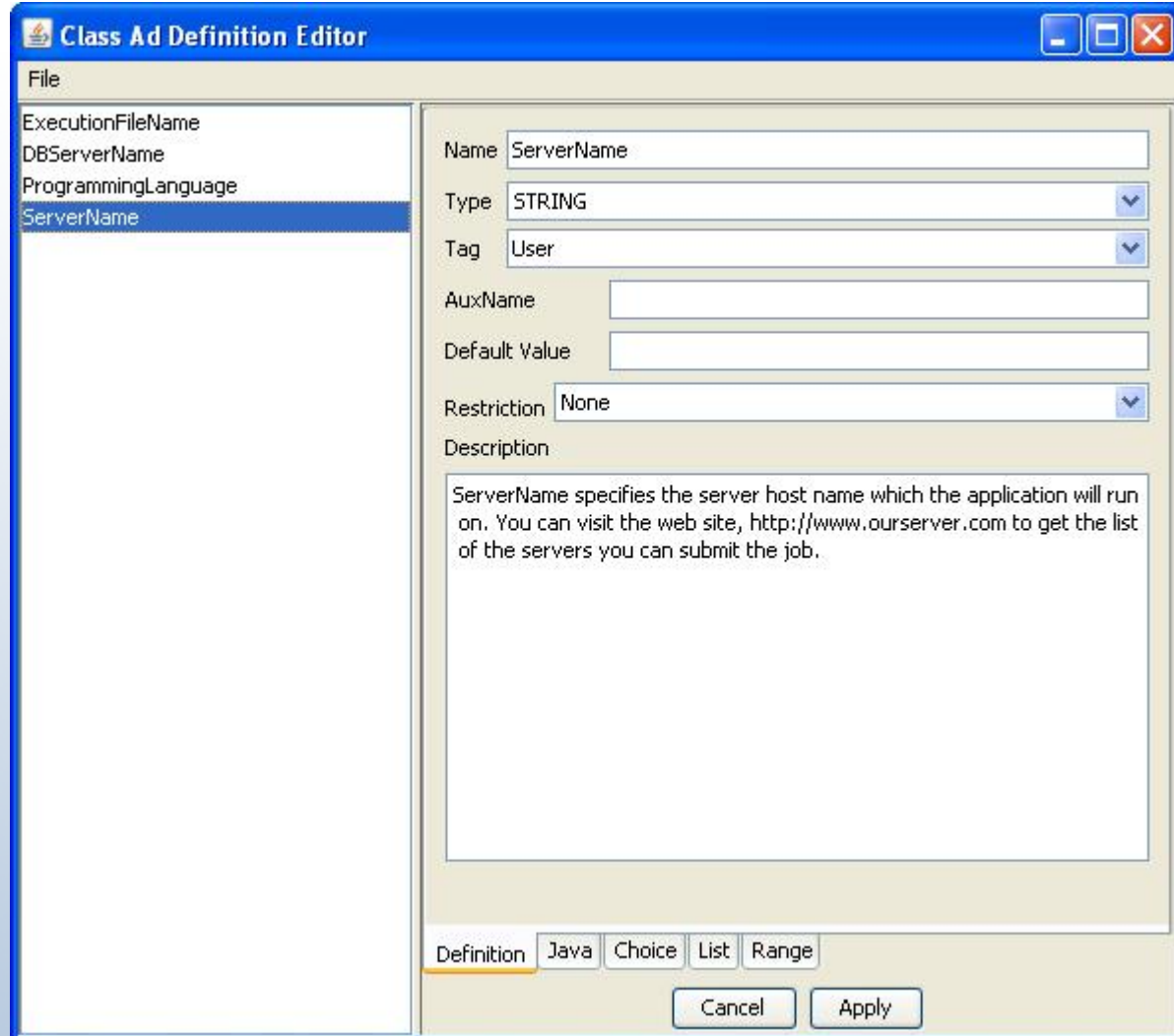
IA64/LINUX	4
INTEL/LINUX	1000
INTEL/WINNT51	5370
INTEL/WINNT60	2337
PPC/OSX	1
SUN4u/SOLARIS28	1
SUN4u/SOLARIS5.10	15
X86_64/LINUX	10779

Total 19507



A Little about Condor ClassAds

- ClassAds are units of information that are used to describe a condor batch submission.
- Name Value Pairs
 - Primitive and Expression value types
- How and where to find the definitions?
- How can we make this a little easier for an application developer?



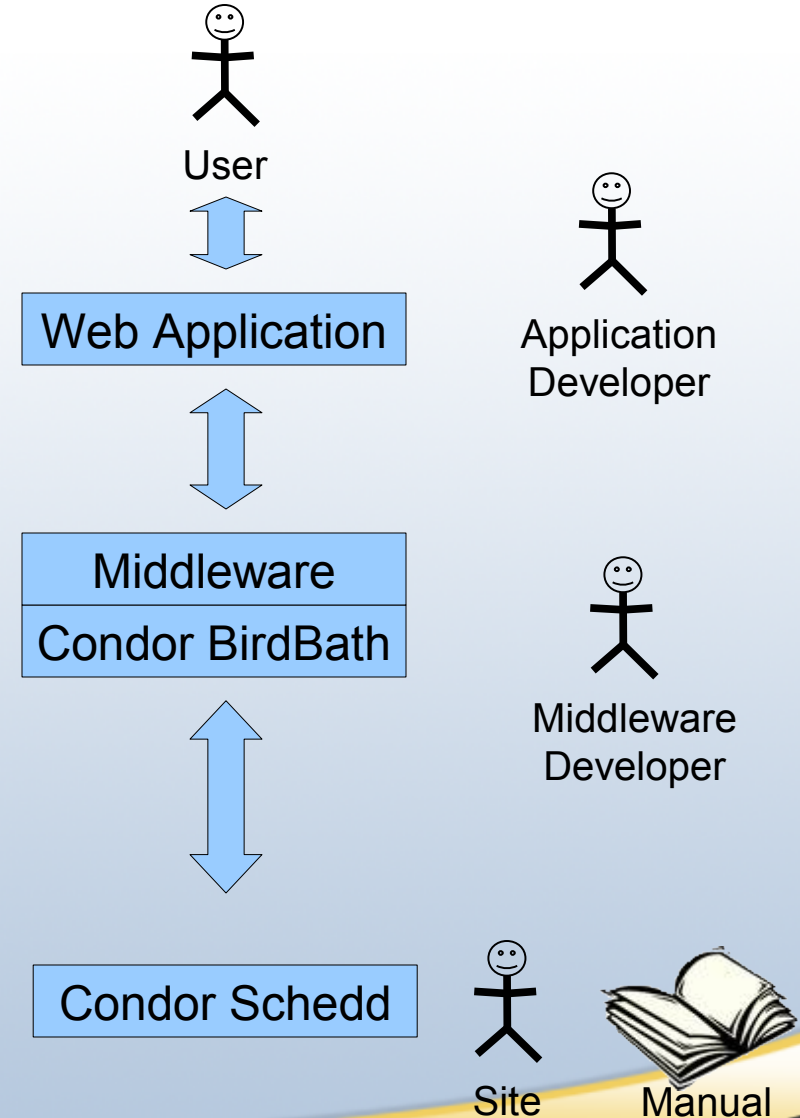
The screenshot shows the 'Class Ad Definition Editor' window. On the left, a list of ClassAd names includes 'ExecutionFileName', 'DBServerName', 'ProgrammingLanguage', and 'ServerName', with 'ServerName' selected. The right pane shows the configuration for 'ServerName':

- Name: ServerName
- Type: STRING
- Tag: User
- AuxName: (empty)
- Default Value: (empty)
- Restriction: None
- Description: ServerName specifies the server host name which the application will run on. You can visit the web site, <http://www.ourserver.com> to get the list of the servers you can submit the job.

At the bottom, there are tabs for 'Definition', 'Java', 'Choice', 'List', and 'Range', and 'Cancel' and 'Apply' buttons.

The Use Case

- Provide JSF components that will allow the application developer to visually layout a condor submit web page.
- Based on a definition of a classad allow an application developer to configure the behavior of that component.



ClassAd Definitions

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<classad_defs>
  <classad_def type="STRING" name="ExecutionFileName">
    <description>ExecutionFileName is the application executable binary file name which you want
      to run on the remote machine.
    </description>
  </classad_def>

  <classad_def type="STRING" name="DBServerName">
    <description>
      You can specify Database Server Name which will be used when the application need to access
      the database.
    </description>
  </classad_def>

  <classad_def type="STRING" name="ProgrammingLanguage">
    <description>
      ProgrammingLanguage specify the lanaguage which the program is written in. Currently C, C++,
      and Java are supported.</description>
  </classad_def>

  <classad_def type="STRING" name="ServerName">
    <description>
      ServerName specifies the server host name which the application will run on. You can visit
      the web site, http://www.ourserver.com to get the list of the servers you can submit the
      job.
    </description>
  </classad_def>
</classad_defs>
```



ClassAd Transformation Stack

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
  Document      : transform.xml
  Created on    : 07/16/2008
  Author       : Jae-Woo Lee
  Description:
    Purpose of the document follows.
-->
<transform_factory>
  <transform src="universe" target="JobUniverse"
transformer="edu.purdue.rcac.condor.classad.transform.RenameTransform"
nextTransformer="edu.purdue.rcac.condor.classad.transform.TranslateTransform"/>

  <transform src="standard" target="1" relevantClassAdName="JobUniverse"
transformer="edu.purdue.rcac.condor.classad.transform.TranslateTransform" />

  <transform src="vanilla" target="5" relevantClassAdName="JobUniverse"
transformer="edu.purdue.rcac.condor.classad.transform.TranslateTransform" />

  <transform src="scheduler" target="7" relevantClassAdName="JobUniverse"
transformer="edu.purdue.rcac.condor.classad.transform.TranslateTransform" />

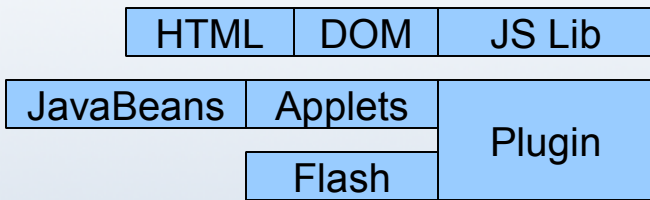
  <transform src="MPI" target="8" relevantClassAdName="JobUniverse"
transformer="edu.purdue.rcac.condor.classad.transform.TranslateTransform" />

  <transform src="grid" target="9" relevantClassAdName="JobUniverse"
</transform_factory>
```

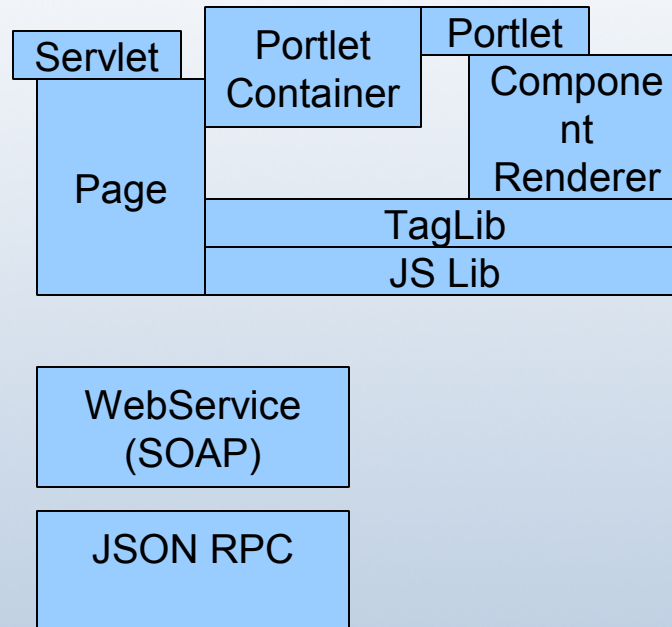


Web Components Overview

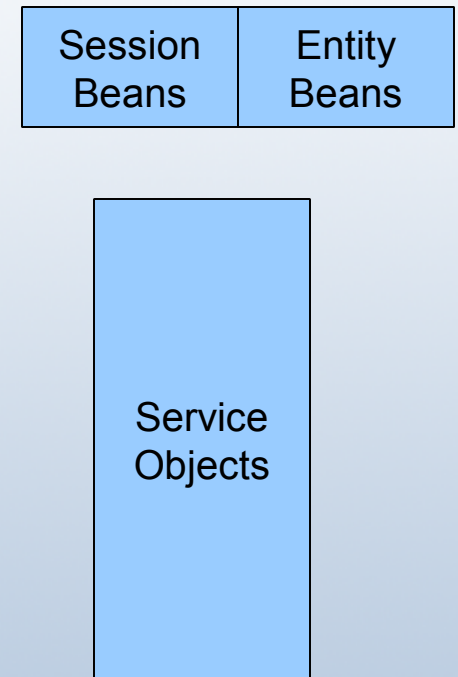
Client Browser



Server UI

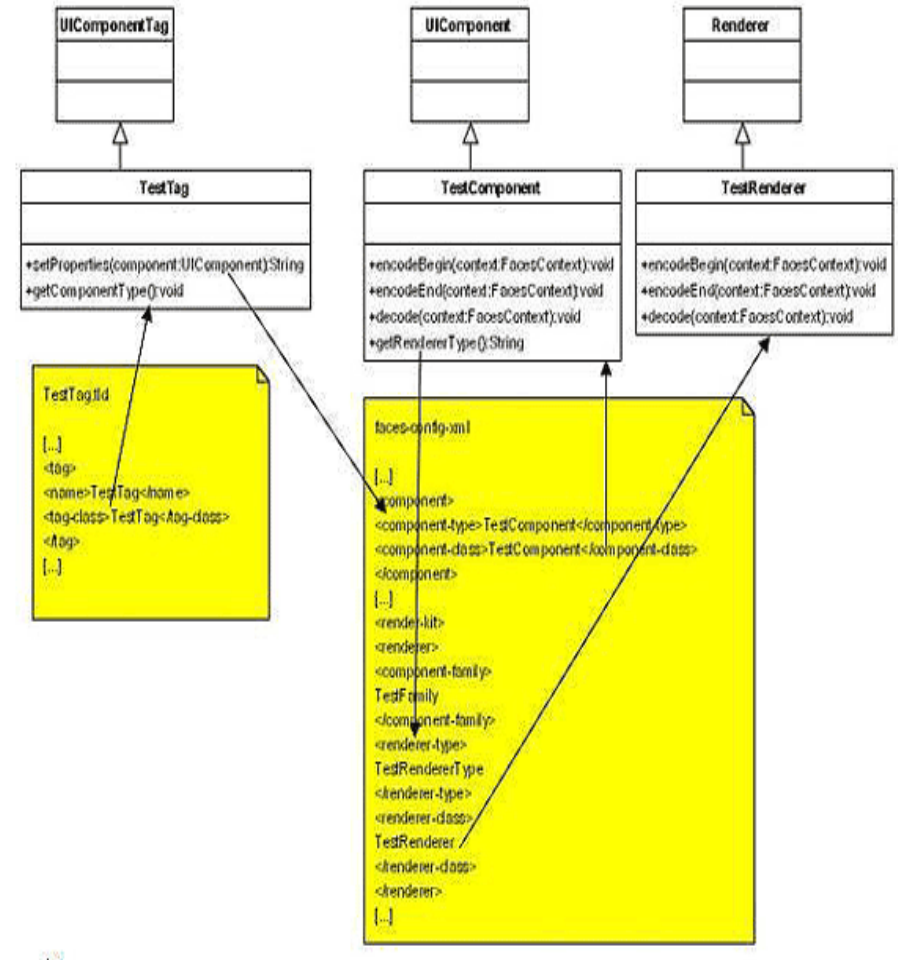


Server Business



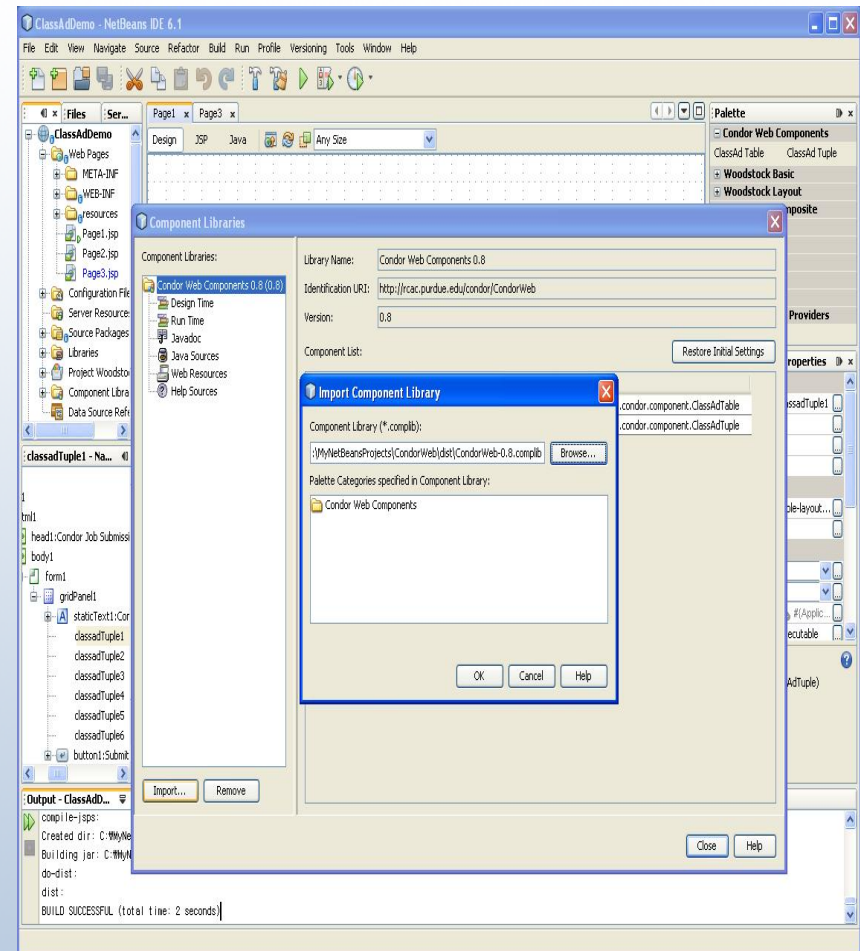
Component Based Development

- Tag Libraries
- Java Server Pages
- Java Server Faces
- Java Beans
- Portlets
- Java Script Libraries
- Containers



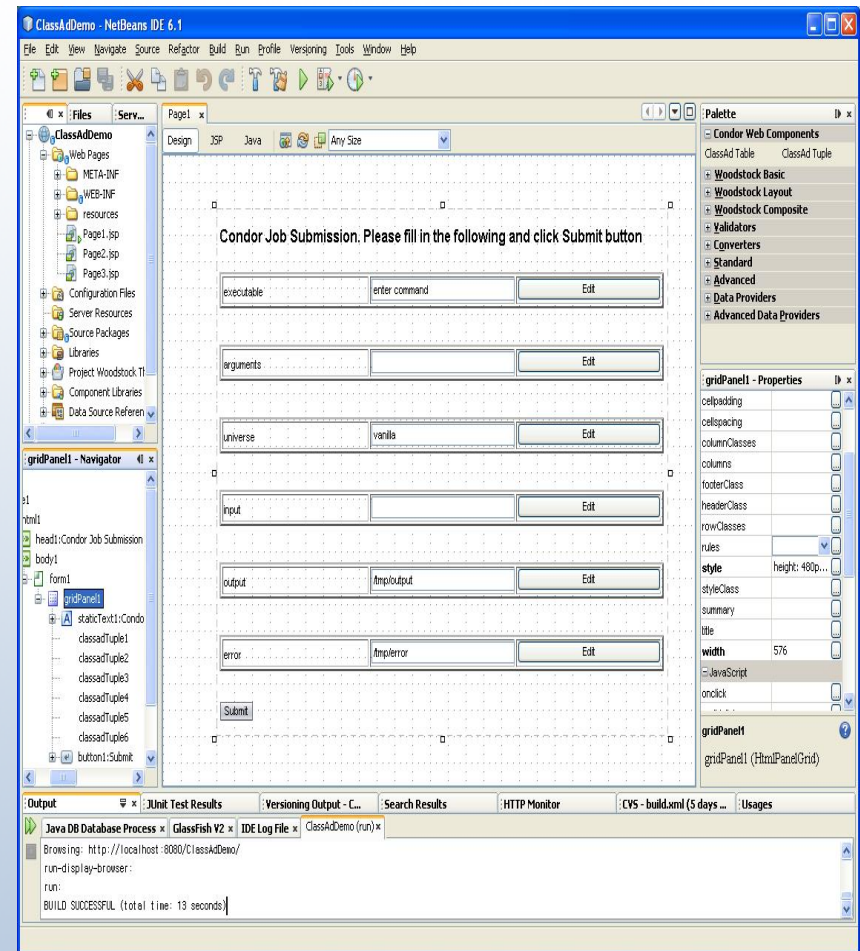
Netbeans Visual Web IDE

- Integrated Development Environment (IDE)
 - Ex: Netbeans or Eclipse
- Visual Layout
- Component Library Installation
- Design Time Support
- Runtime Support



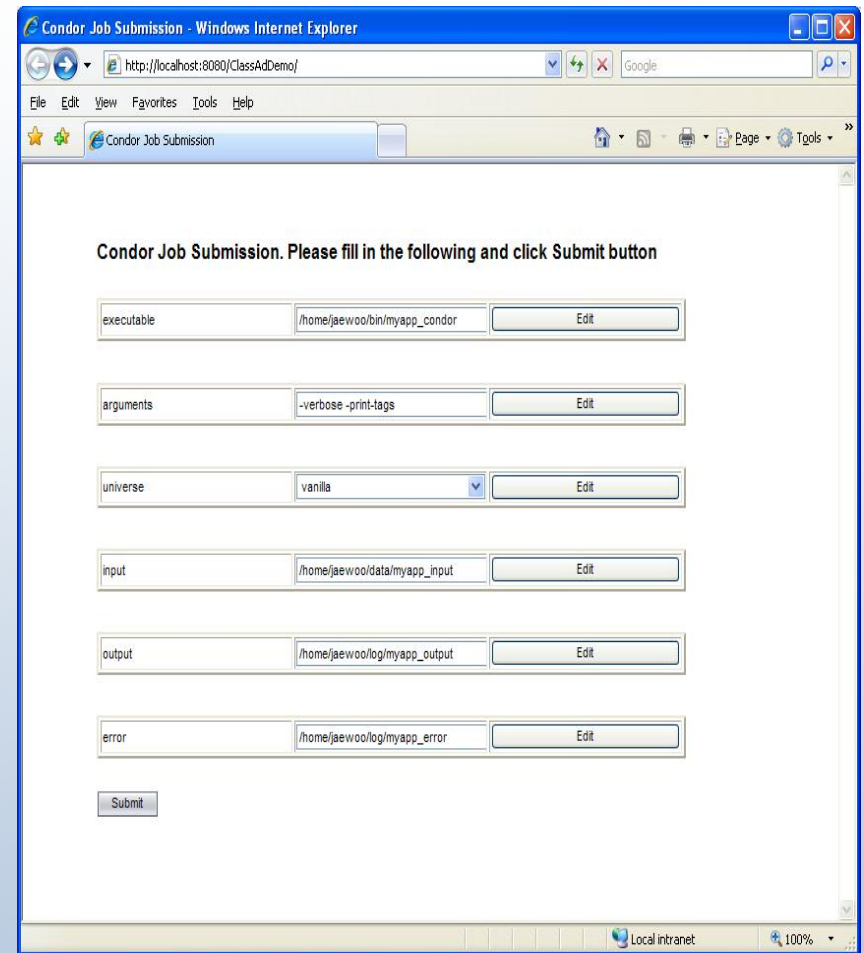
Netbeans Visual Web IDE

- Visual Layout
 - Makes HTML layout fun!
- Property Editors
- Preview
- Configuration Wizards



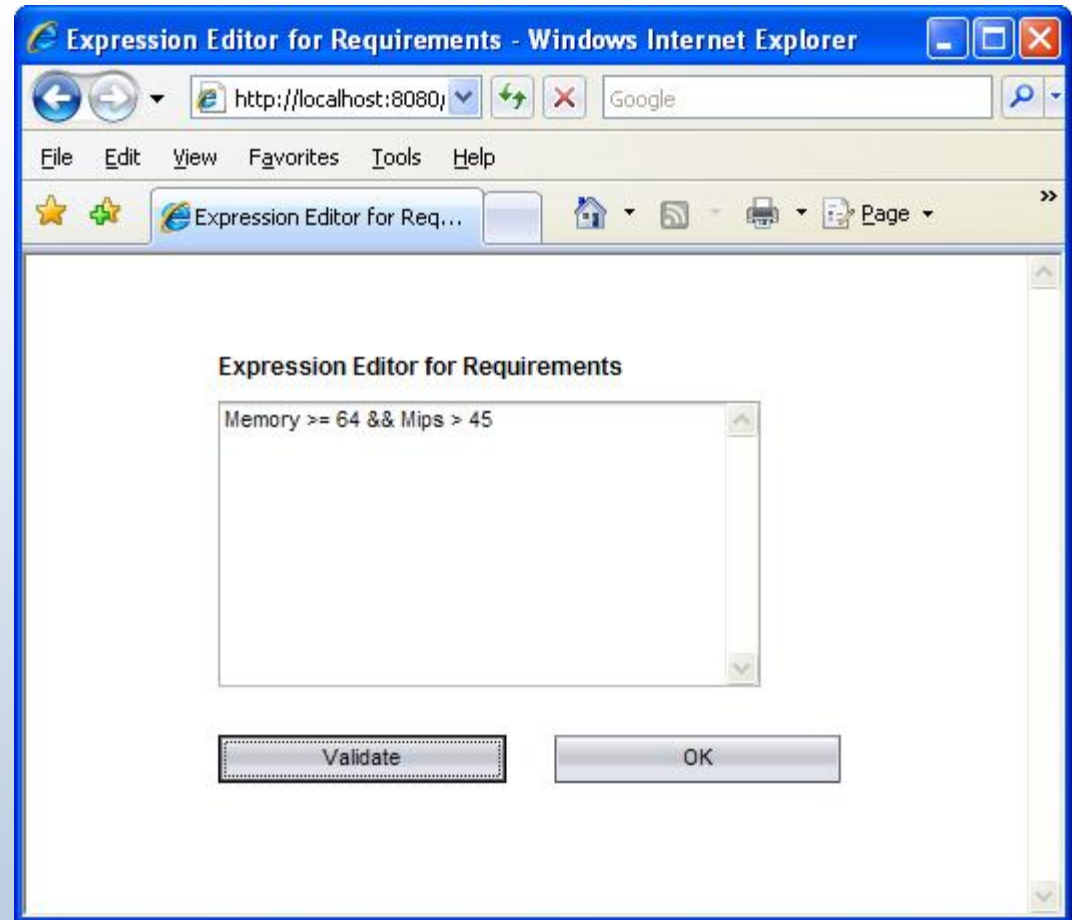
Netbeans Visual Web IDE

- Runtime
- Style Sheet support
- Future
 - AJAX support
 - Validation



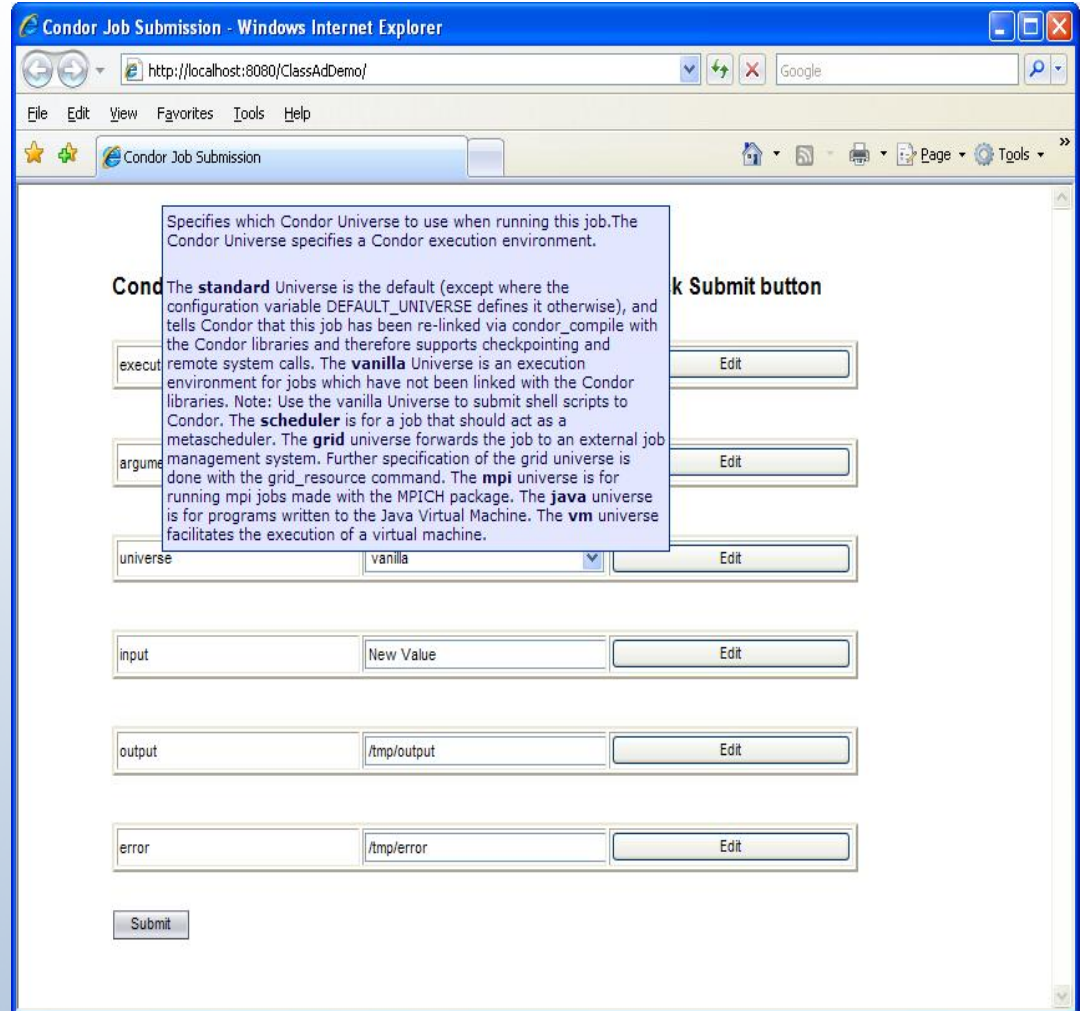
Expression Editor

- Validation
- Future
 - Evaluation
 - Construction
 - Suggestions



Additional Features

- Tooltips
- CSS styled
- Action and ValueChanged Events are supported.



The screenshot shows a web browser window titled "Condor Job Submission - Windows Internet Explorer". The address bar shows "http://localhost:8080/ClassAdDemo/". The browser's menu bar includes "File", "Edit", "View", "Favorites", "Tools", and "Help". The page content is a form for submitting a Condor job. A tooltip is displayed over the "universe" dropdown menu, which currently shows "vanilla". The tooltip text reads: "Specifies which Condor Universe to use when running this job. The Condor Universe specifies a Condor execution environment. The **standard** Universe is the default (except where the configuration variable DEFAULT_UNIVERSE defines it otherwise), and tells Condor that this job has been re-linked via condor_compile with the Condor libraries and therefore supports checkpointing and remote system calls. The **vanilla** Universe is an execution environment for jobs which have not been linked with the Condor libraries. Note: Use the vanilla Universe to submit shell scripts to Condor. The **scheduler** is for a job that should act as a metascheduler. The **grid** universe forwards the job to an external job management system. Further specification of the grid universe is done with the grid_resource command. The **mpi** universe is for running mpi jobs made with the MPICH package. The **java** universe is for programs written to the Java Virtual Machine. The **vm** universe facilitates the execution of a virtual machine." The form includes fields for "execut", "argume", "universe", "input", "output", and "error", each with an "Edit" button. A "Submit" button is at the bottom.

Conclusion

- Finer grain component development
- Integration of both system and information
 - Manual, Site, Middleware, Application, User
- Visual Layout is fast! (for developers)
- Future
 - Expression Editor
 - Expression Evaluation
 - Repository of ClassAd Definitions
 - Used in Condor Web Submission Portal
- Moving code to either SourceForge or Google Code stay tuned.