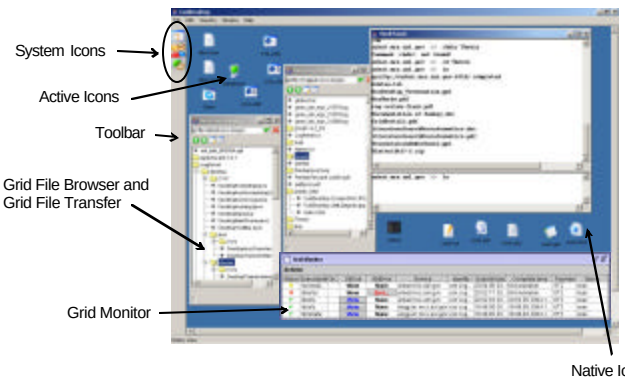
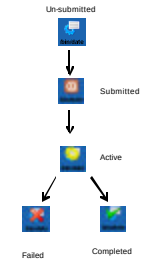
Easy to Use





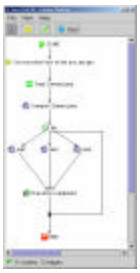
Active Icons change dynamically their appearance depending on the state of the calculation.

Java CoG Kit Grid Desktop

We are developing an advanced Grid Desktop that mimics the look of common graphical desktops such as KDE or Windows. It integrates with the native operating system and allows management and monitoring of tasks conducted on the Grid.

Monitoring the state of the application is conducted either through (a) a Grid monitor that lists and interacts with jobs in the Grid through a table or (b) active icons on the desktop that change their appearance based on the state the icon represents.

Besides simple form-based input, the Grid Desktop uses the drag-and-drop paradigm to initiate Grid calculations on Grid services.



```
<project>
  <include file="sysdefaults.xml"/>
  <set name="host" value=" hot.mcs.anl.gov" annotation="Set execution host to  hot.mcs.anl.gov"/>

  <!-- copy file -->
  <gridTransfersrcfile ="Climate.java"desthost="{host}"  provider="gridftp " annotation="Copy Climate.java"/>

  <!-- compile -->
  <gridExecute  host="{host}" provider="gt2" executable="/share/j2sdk1.4.2_05/bin/javac"
                arguments="Climate.java" annotation="Compile Climate.java"/>

  <!-- Run →
  <for name="i" from="1" to="10">
     <parallel>
        <gridExecute  host="{host}" provider="gt2"  executable="/share/j2sdk1.4.2_05/ bin/java"
                      arguments=" -cp . Climate {i}"  stdout="Climate.stdout{i}1" stderr="Climate.stderr{i}1" annotation="Job1"/>
        <gridExecute  host="{host}" provider="gt2 " executable="/share/j2sdk1.4.2_05/bin/java"
                      arguments=" -cp . Climate {i}"  stdout="Climate.stdout{i}2" stderr="Climate.stderr{i}2" annotation="Job2"/>
        <gridExecute  host="{host}" provider="gt2 " executable="/share/j2sdk1.4.2_05/bin/java"
                      arguments=" -cp . Climate {i}"  stdout="Climate.stdout{i}3" stderr="Climate.stderr{i}3" annotation="Job3"/>
     </parallel>

     <echo message="Execution completed for iteration {i}" annotation="Execution completed"/>
  </for>
</project>
```

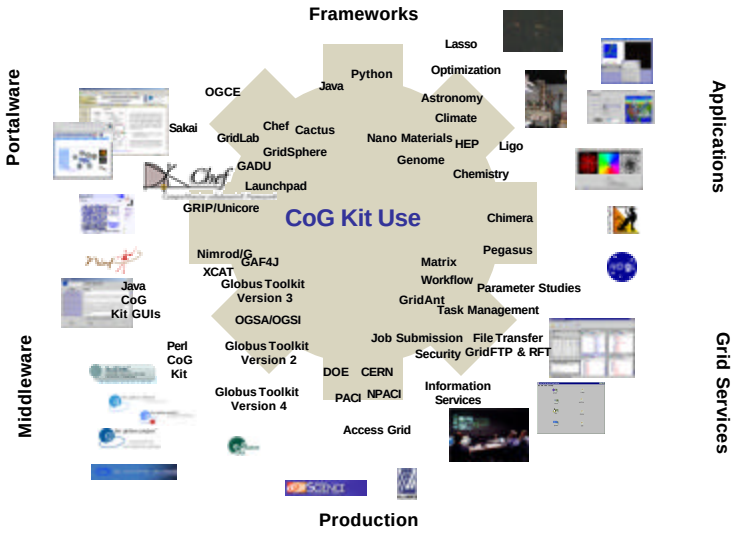
Java CoG Kit Workflow

The Java CoG Kit contains an XML-based language to specify workflows in a simple form. It contains a workflow engine called *karajan* that executes such workflows. The Java CoG Kit workflow viewer displays the state changes of the workflow interactively. Karajan's syntax is based on an extension to ant and is therefore also referred to as gridant.

OGCE Portal

The NSF NMI-sponsored Open Grid Computing Environment Consortium uses the Java CoG Kit to develop a portal framework for Girds. Using the Java CoG Kit enables easy access to Grid services that many portal users desire. We are developing portlets (which are reusable portal components) for file transfer, transfer monitoring, and workflow management.

For more information: http://www.ogce.org



CoG Kit Community

We maintain a community Web page on which users can announce their projects using CoG Kit technologies. At SC2003 more than 30 projects on the show floor used CoG Kits to access the Grid. At SC2004, four out of thirteen Grid related posters use the Java CoG Kit.

Please use the sticky notes to add applications using CoG Kits that are not mentioned on this poster. To add your project to the Web page, please visit

http://www.cogkit.org /

Invitation to Participate

For further information on this subject contact:
Gregor von Laszewski, Principal Investigator