



the globus alliance

[www.globus.org/cog](http://www.globus.org/cog)

# Java CoG Kit Workflow

Gregor von Laszewski

Argonne National Laboratory

University of Chicago

[gregor@mcs.anl.gov](mailto:gregor@mcs.anl.gov)

<http://www.cogkit.org>

Updated slides will be available on the CoG Kit web site



THE UNIVERSITY OF  
CHICAGO



## Funding sources & Acknowledgement

- The Java CoG Kit receives funding from the following sponsors
  - ◆ DOE MICS
  - ◆ NSF NMI
- Previous versions of the CoG Kit also received funding from
  - ◆ NCSA Alliance
- Please, contact [gregor@mcs.anl.gov](mailto:gregor@mcs.anl.gov) in case you like to work with us more closely.
- Acknowledgement:
  - ◆ CoG Team, Globus Team, Globus Alliance, many others as listed on [www.cogkit.org](http://www.cogkit.org)



the globus alliance

[www.globus.org/cog](http://www.globus.org/cog)

# Community

- Call on the community to help us with extending and improving the CoG Kit



# Outline

- Related to the Java CoG Kit ...
- What is the CoG Kit?
- History of workflow
- Concepts of Workflow
  - ◆ API
  - ◆ GridAnt/Karajan
  - ◆ GridShell
- Graphical Interfaces
- Conclusion



# Observation

- **Problem**

- ◆ Many application developers desire to program the Grid in familiar higher level frameworks that allow rapid prototyping.

- **Solution**

- ◆ We propose to reuse a variety of commodity tools, protocols, approaches, methodologies, while integrating Grid software based on the Globus Toolkit
  - Easier development of advanced Grid services
  - Easier and more rapid application development
  - Easier deployment of Grid services
  - Code reuse and use of component repositories
  - Use of Web services as part of the Grids
  - Widespread use of the Grid
  - Use of commodity technology is not limited to the client!



# What is the Java CoG Kit ?

- (Presentation earlier this week)
- Make Grid programming and use easier
- Includes Workflow abstractions:
  - ◆ Language based on XML/ant
  - ◆ Workflow engine called karajan (German conductor)
  - ◆ Workflow viewer and monitor
  - ◆ Portal components (to be completed)



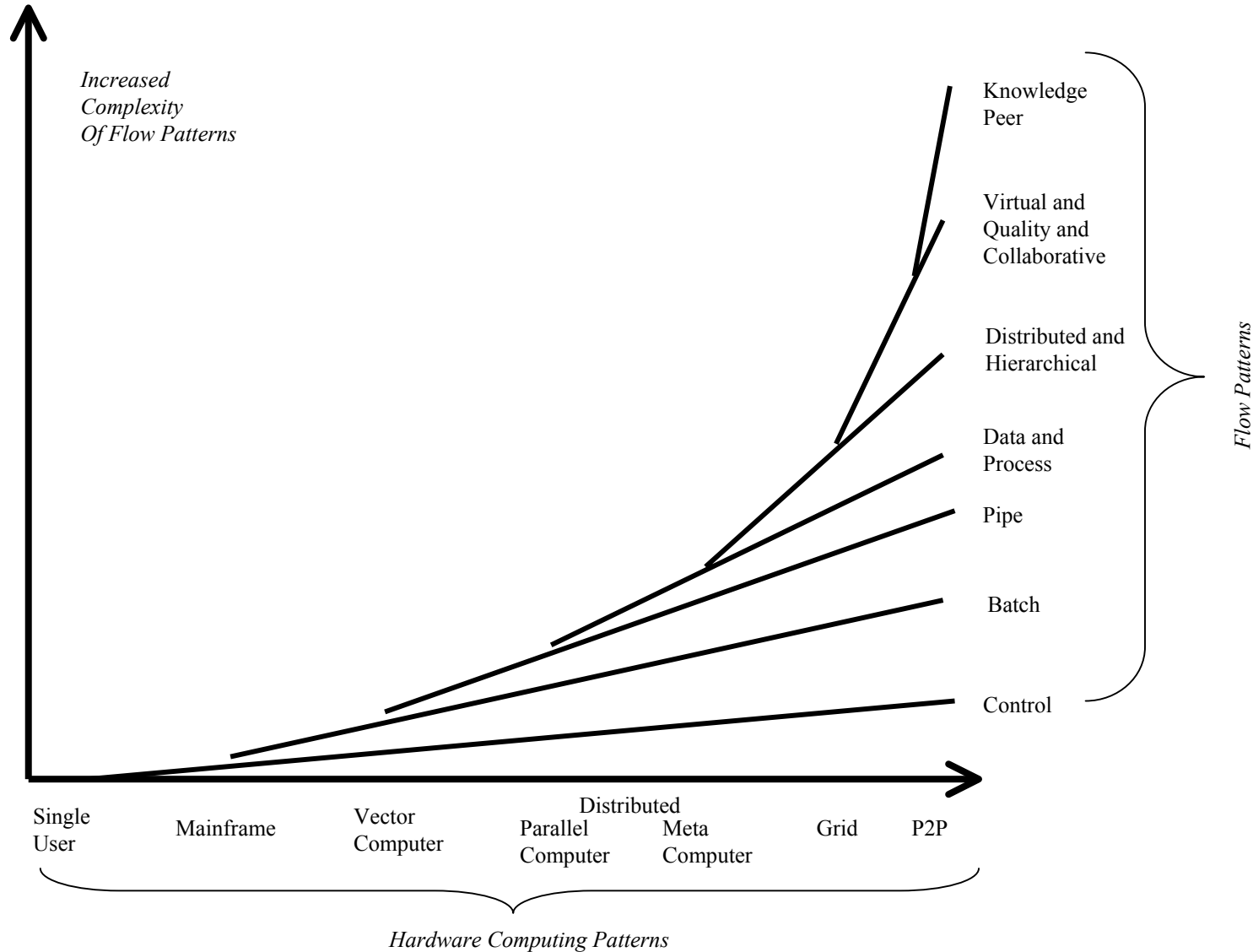
the globus alliance

[www.globus.org/cog](http://www.globus.org/cog)

Lets focus on workflow



# History: Flow Patterns



von Laszewski,  
et al.  
“Grid Middleware”

von Laszewski





# History of Workflow in CoG Kit

<b>1994</b>	<b>von Laszewski dataflow and job control Metacomputing system in tcl/tk</b>
<b>1996</b>	<b>von Laszewski data and workflow Metacomputing system in Java</b>
<b>1996</b>	<b>Start of the Globus Project</b>
<b>1997</b>	<b>a) Fault tolerant high throughput broker in Java for SnB application b) Less capable broker in sh as Globus does only use sh</b>
<b>1998</b>	<b>Improved GUI for the Java CoG Kit high throughput broker</b>
<b>1998</b>	<b>Development of Condor-G (C based)</b>
<b>...</b>	<b>A long period of inactivity as Globus toolkit is only C not Java</b>
<b>2000</b>	<b>Official start of the Java CoG Kit</b>
<b>2001</b>	<b>Reimplementation of parts of the features of the system from 1994 and 1997</b>
<b>2002</b>	<b>GSFL, Java CoG Kit jglobus is essential part of GT3 and in 2005 in GT4</b>
<b>2003</b>	<b>Java CoG Kit GridAnt &amp; Karajan, Workflow with GT2, GT3, SSH</b>
<b>2004</b>	<b>Java CoG Kit GridDesktop and GridShell, Workflow with GT2, GT3, SSH</b>
<b>2005</b>	<b>New integrated Release, Workflow with GT2, GT3, GT4, SSH, Condor</b>

**\* If a date is wrong please help us correcting it.**



the globus alliance

[www.globus.org/cog](http://www.globus.org/cog)

# Current status



# Concepts of Workflow (in Java CoG Kit)

- Three ways to do workflow
  - ◆ Workflow with CoG Kit abstractions (API)
  - ◆ Workflow with GridAnt/Karajan (XML)
  - ◆ Workflow with the GridShell (in future)
- Which to chose depends on your requirements



# Features

	<b>Abstraction API *</b>	<b>GridAnt/ Karajan</b>	<b>GridShell</b>
<b>Java API</b>	yes	undocumented	undocumented
<b>XML</b>	no	yes	no
<b>If, While, For</b>	implicit	yes	not yet
<b>Caching</b>	no	yes	no
<b>Checkpointing</b>	no	yes	no
<b>Logging</b>	no	not yet	yes
<b>Viewer</b>	no	yes	yes
<b>Batch</b>	no	yes	yes
<b>Uses Karajan</b>	no	yes	yes
<b>Uses Abstr. API</b>	yes	yes	yes

\* The features marked with \* can be implemented by the programmer



# Abstractions

- Hypothesis:
  - ◆ With rapidly changing technologies it may be beneficial to have an abstraction that can be assisting in this technical challenge.
- Solution:
  - ◆ CoG Kit abstractions are defined for precisely that reason.



the globus alliance

[www.globus.org/cog](http://www.globus.org/cog)

# Java CoG Kit Abstractions



# Java CoG Kit Abstractions

- We gave a talk earlier this week ...

```
TaskGraph tg = new TaskGraphImpl();
```

```
public void create () {  
    // define tasks
```

```
    ....
```

```
    /* Add the tasks to the TaskGraph */
```

```
    tg.add(task1);
```

```
    tg.add(task2);
```

```
    tg.add(task3);
```

```
    tg.add(task4);
```

```
    tg.addDependency(task1, task2);
```

```
    tg.addDependency(task1, task3);
```

```
    tg.addDependency(task2, task4);
```

```
    tg.addDependency(task3, task4);
```

```
}
```

```
public void submit() {
```

```
    TaskGraphHandler handler = new TaskGraphHandlerImpl();
```

```
    try {
```

```
        handler.submit(tg);
```

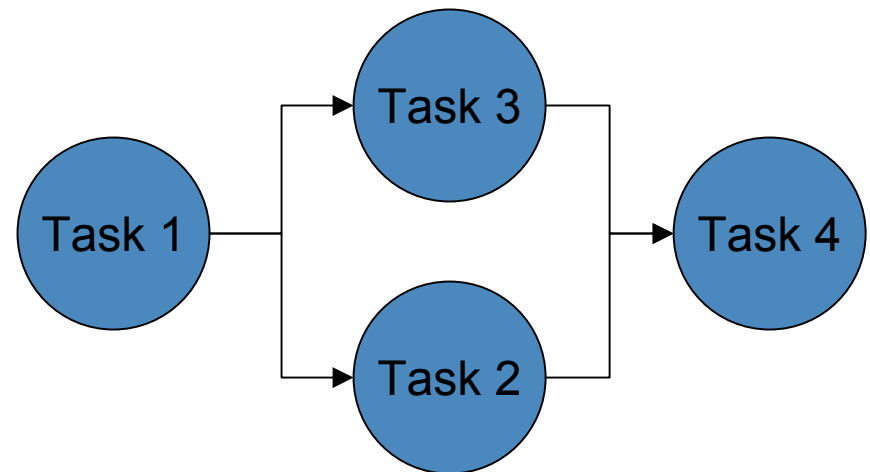
```
    } catch (Exception e) {
```

```
        logger.error("`` Some Error occured", e);
```

```
        System.exit(1);
```

```
    }
```

```
}
```





the globus alliance

[www.globus.org/cog](http://www.globus.org/cog)

# Java CoG Kit GridAnt/Karajan





# GridAnt

- One day ...
  - ◆ Why not use ant as workflow engine?
  - ◆ We already have a Grid interface through the Java CoG Kit ...
  - ◆ Prototype finished in less than a week
  - ◆ Worked great but ...
    - Scalability, some language constructs were missing
  - ◆ Advantages uses ant which is used by millions of people



# GridAnt/Karajan

- Goal:
  - ◆ improve what first version of GridAnt lacked and come up with a new version of GridAnt.
  - ◆ Be still able to call ant from GridAnt
  - ◆ Have more enhanced language features
  - ◆ Have an easy syntax
  
- ◆ At the time we started BEPL was not available/ready
- ◆ Be easier than BEPL



# Control structures

## Loops

```
<for name="iteration" from="1" to="4">  
  <echo message="Iteration {iteration}"/>  
</for>
```

```
<foreach name="iteration" in="one, two, three, four">  
  <echo message="Iteration {iteration}"/>  
</foreach>
```

```
<while>  
  <condition>  
    <false/>  
  </condition>  
  <echo message="You will never see this message"/>  
</while>
```

## Condition

```
<if>  
  <condition>  
    <condition1/>  
  </condition>  
  <then>  
    ...  
  </then>  
  ...  
  <condition>  
    <conditionN/>  
  </condition>  
  <elseif>  
    ...  
  </elseif>  
  <else>  
    ...  
  </else>  
</if>
```



# Variables

## assignment

```
<set name="var" value="1"/>
```

## Lists

```
<list:append>
```

```
  <argument value="1"/>
```

```
  <argument value="2"/>
```

```
  <argument value="3"/>
```

```
</list:append>
```

## Operators

```
<math:sum>
```

```
  <argument value="1"/>
```

```
  <argument value="2"/>
```

```
  <argument value="3"/>
```

```
</math:sum>
```

## Hashtable/map

```
<set name="map">
```

```
  <map>
```

```
    <map:entry key="name" value="John"/>
```

```
    <map:entry key="age" value="99"/>
```

```
  </map>
```

```
</set>
```

```
<echo message="Name: {1}, age: {2}">
```

```
  <map:get map="{map}" key="name"/>
```

```
  <map:get map="{map}" key="age"/>
```

```
</echo>
```



# Templates

## Definition

```
<templatedef name="sample">  
  <default name="arg1" value="default1"/>  
  <default name="arg2" value="default2"/>  
  <echo message="arg1 is {arg1}"/>  
  <echo message="arg2 is {arg2}"/>  
  <echo message="arg3 is {arg3}"/>  
</templatedef>
```

## Using

```
<template name="sample"  
  arg1="value1"  
  arg2="value2"  
  arg3="value3"/>  
\end{lstlisting}
```



# Recursion

## Recursion

```
<element name="fibonacci" arguments="n">
  <if>
    <math:le value1="{n}" value2="2"/>
    <then>
      <argument value="1"/>
    </then>
    <else>
      <math:sum>
        <fibonacci>
          <math:subtraction from="{n}" value="1"/>
        </fibonacci>
        <fibonacci>
          <math:subtraction from="{n}" value="2"/>
        </fibonacci>
      </math:sum>
    </else>
  </if>
</element>
```

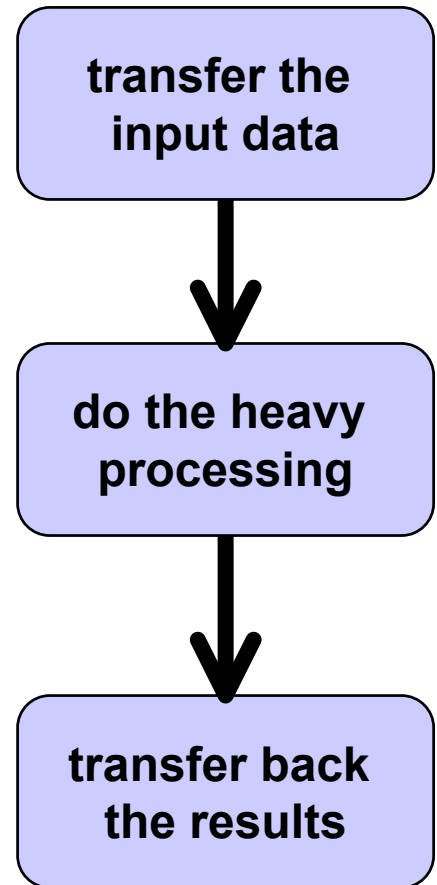


# A simple Grid job

```
<gridTransfer  
  srchost="localhost" srcdir="/tmp" srcfile="in"  
  desthost="{remote}" destdir="/tmp"/>
```

```
<gridExecute host="{remote}"  
  executable="/usr/bin/climate"  
  args="512 512"  
  stdin="/tmp/in"  
  stdout="/tmp/out"/>
```

```
<gridTransfer srchost="{remote}" srcdir="/tmp"  
  srcfile="out"  
  desthost="localhost" destdir="/tmp"/>
```





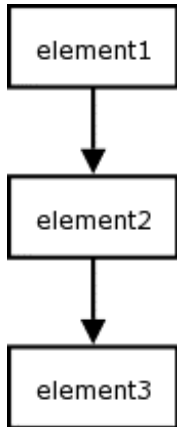
# Using Loops

```
<grid name="default">  
  <for name="index" from="1" to="20">  
    <host name="lg0n{index}.pts.uml.mov" cpus="1">  
      <gridExecute host="{host}"  
        executable="/usr/bin/climate"  
        args="512 512"  
        stdin="/tmp/in"  
        stdout="/tmp/out"/>  
    </host>  
  </for>  
</grid>
```

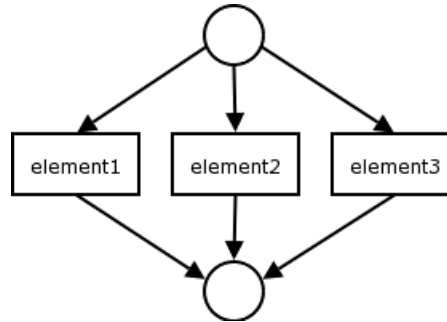




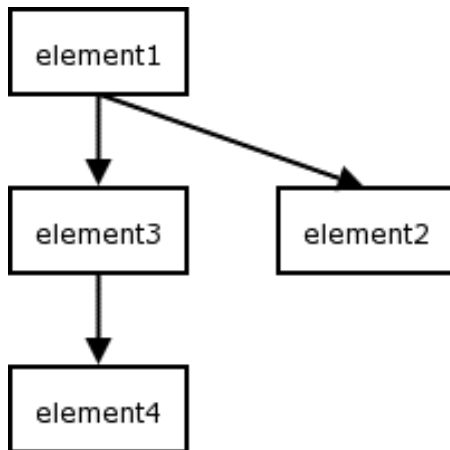
# Parallelism



```
<sequential>  
  <element1/>  
  <element2/>  
  <element3/>  
</sequential>
```



```
<parallel>  
  <element1/>  
  <element2/>  
  <element3/>  
</parallel>
```



```
<parallel>  
  <element1/>  
  <element2 sync="false"/>  
  <element3/>  
</parallel>
```



# Simple Types

## Arithmetic

math:sum ::= a + b

math:product ::= a \* b

math:subtraction ::= a - b

math:quotient ::= a/b

math:remainder

math:square ::= a\*2

math:sqrt

math:equals ::= a =b

math:gt ::= a > b

math:lt ::= a < b

math:ge ::= a >= b

math:le ::= a <= b

## Booleans

and, or, not, true, false

## Lists

list:append, list:prepend, list:concat,

list:first, list:last, list:butFirst, list:butLast,

list:size, list:isEmpty

## Map/Hashtable

map:put, map:entry, map:get, map:contains,

map:delete, map:size



# Error Handling

```
<onError match="(.*Expired credentials detected.*)"|(.*Proxy file.*not found.*)">
  <if>
    <condition>                                <!-- avoid recursive errors -->
      <math:equals value1="{errorcount}" value2="1"/>
    </condition>
    <then>
      <!-- pop up a proxy init window -->
      <echo message="Invalid GSI credentials detected. Executing proxy init...">
      <executeJava mainClass="org.globus.cog.karajan.util.ProxyInitWrapper"/>
      <!-- re-execute the element -->
      <echo message="Restarting failed element"/>
      <executeElement element="{element}"/>
    </then>
    <else>
      <!-- "Error count > 1" -->
      <!-- error produced by generateError are never intercepted -->
      <generateError message="{error}"/>
    </else>
  </if>
</onError>
```



# Java CoG Kit GridAnt Workflow

Overview  
Window

Workflow  
Details

Interactive  
Steering

The screenshot displays the Java CoG Kit Graph Viewer interface. The main window shows a workflow diagram with nodes: 'echo', 'setvar', 'javaBean', 'gridExecute', 'javaBean', 'gridTransfer', 'gridTransfer', 'javaBean', and 'gridExecute'. A 'Start' button is visible at the top left. An 'Overview' window is open on the left, showing a smaller version of the workflow. An 'Error' dialog box is open at the bottom right, displaying the message: 'The following error has occurred: GSSEException: Expired credentials... Location: gridExecute @ /home/mike/work/flow/templates.xml'. The dialog includes checkboxes for 'Apply to all errors of this type' and 'Apply to all errors for this element', and buttons for 'Continue', 'Ignore', and 'Restart'. The background shows a desktop environment with a terminal window and a file browser.



# XML GUI Forms

```
<form:form title="test" id="form" waitOn="IDOk">
```

```
<form:vbox>
```

```
<form:hbox>
```

```
<form:vbox>
```

```
<form:hbox>
```

```
<form:label text="First name: "/>
```

```
<form:textField id="IDFirst" columns="20"/>
```

```
</form:hbox>
```

```
<form:hbox>
```

```
<form:label text="Last name: "/>
```

```
<form:textField id="IDLast" columns="20"/>
```

```
</form:hbox>
```

```
</form:vbox>
```

```
<form:vbox>
```

```
<form:checkBox caption="Married"  
id="IDMarried" halign="0"/>
```

```
<form:HSeparator/>
```

```
<form:radioBox caption="Sex" id="IDSex">
```

```
<form:radioButton caption="Male"  
id="IDMale"/>
```

```
<form:radioButton caption="Female"  
id="IDFemale"/>
```

```
</form:radioBox>
```

```
</form:vbox>
```

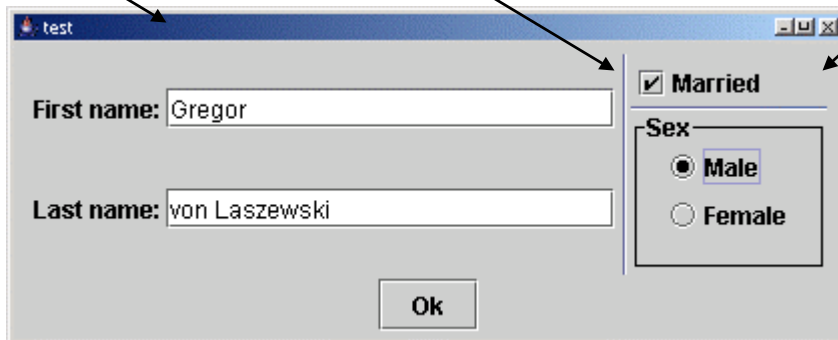
```
</form:hbox>
```

```
<form:button id="IDOk" caption="Ok"/>
```

```
</form:vbox>
```

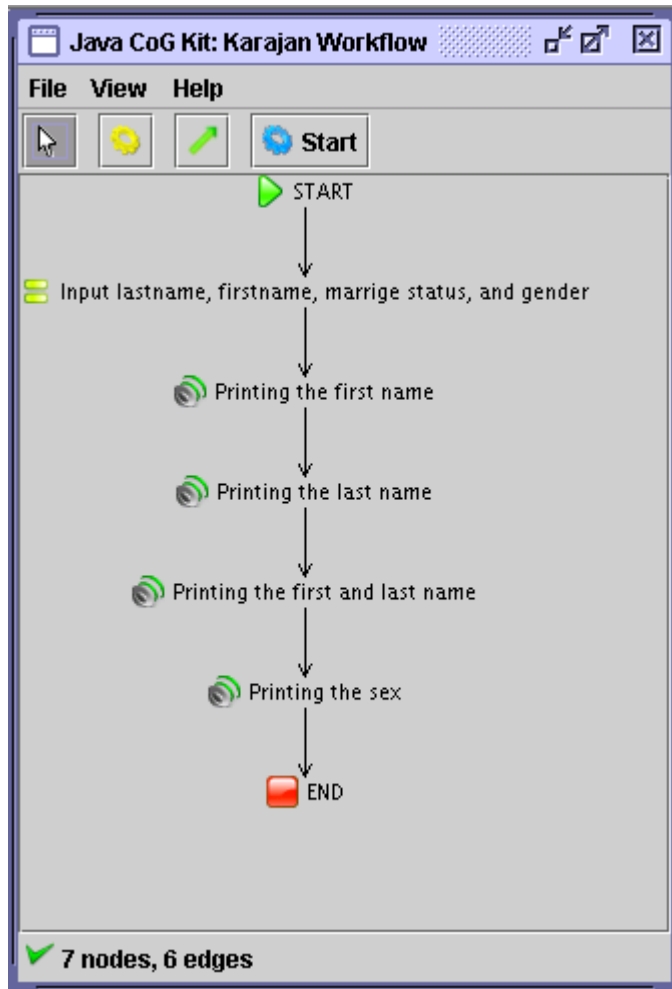
```
<form:VSeparator/>
```

```
</form:form>
```





# XML GUI Forms



test

First name:

Last name:

Married

Sex

Male

Female

Ok



the globus alliance

[www.globus.org/cog](http://www.globus.org/cog)

# Applications of Workflow

With the Java CoG Kit



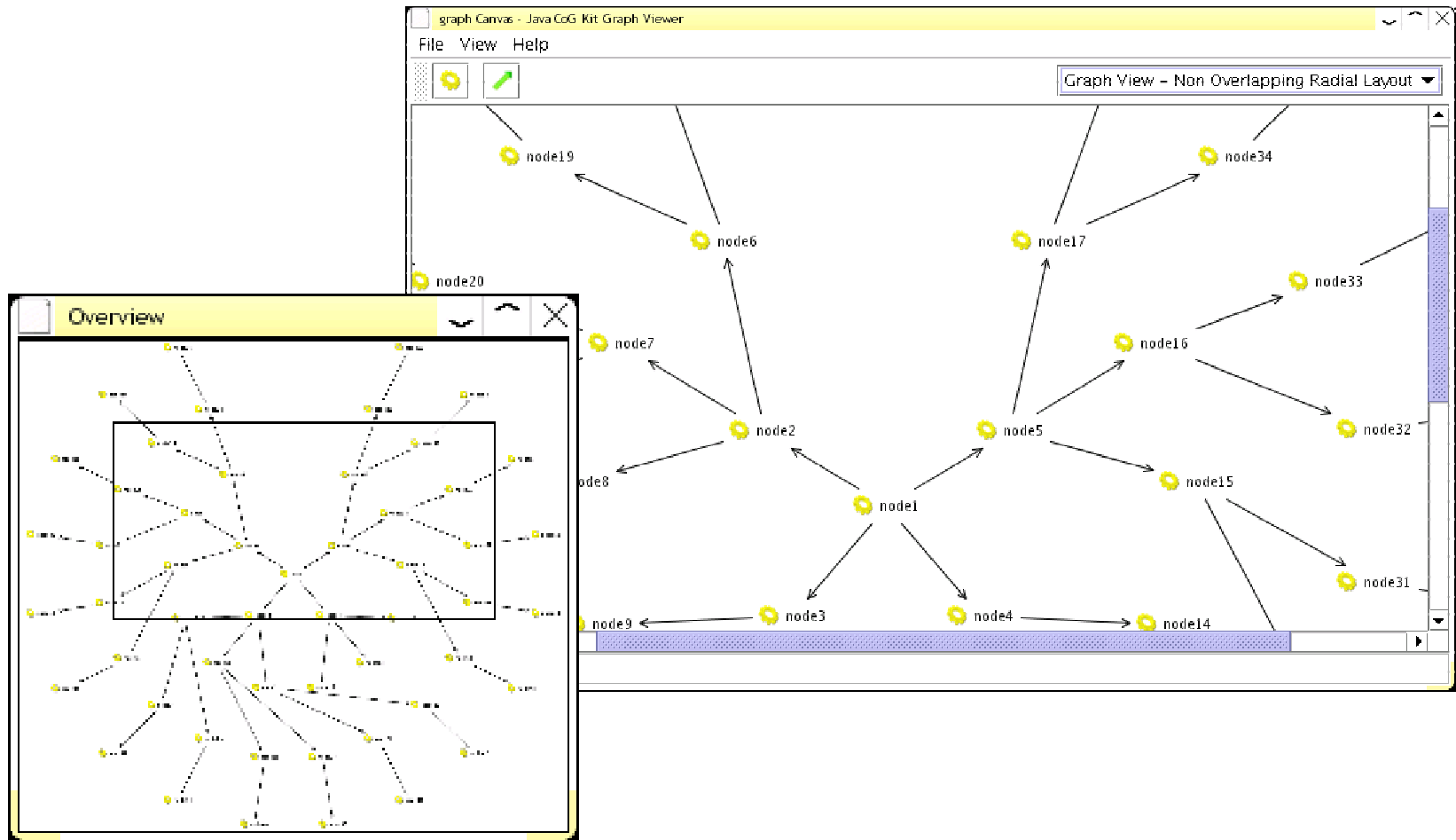
# Graph in trivial XML

- `<graph id="simple-graph" name="Hierarchical Graph">`
- `<node nodeid="node1" name="node1"/>`
- `<node nodeid="node2" name="node2"/>`
- `<node nodeid="node3" name="node3">`
- `<node nodeid="nodeA" name="nodeA"/>`
- `<node nodeid="nodeB" name="nodeB"/>`
- `<edge from="nodeA" to="nodeB"/>`
- `</node>`
- `<node nodeid="node4" name="node4"/>`
- `<node nodeid="node5" name="node5"/>`
- `<edge from="node1" to="node3"/>`
- `<edge from="node2" to="node3"/>`
- `<edge from="node3" to="node4"/>`
- `<edge from="node3" to="node5"/>`
- `</graph>`





# Workflow Layout





# Java CoG Kit GridAnt Workflow

The screenshot displays the Java CoG Kit Graph Viewer interface. The main window shows a workflow graph with nodes such as 'echo', 'setvar', 'javaBean', 'gridExecute', 'gridTransfer', and 'executeJava'. A yellow callout bubble labeled 'Workflow Details' points to the main graph. An 'Overview' window on the left provides a smaller view of the entire workflow, with a yellow callout bubble labeled 'Overview Window' pointing to it. At the bottom, an error dialog box is open, displaying the message: 'The following error has occurred: GSSEException: Expired credentials... Location: gridExecute @ /home/mike/work/flow/templates.xml'. The dialog includes checkboxes for 'Apply to all errors of this type' and 'Apply to all errors for this element', and buttons for 'Continue', 'Ignore', and 'Restart'. A yellow callout bubble labeled 'Interactive Steering' points to the error dialog. The interface also includes a 'Status' window at the top left and a 'Console' window at the bottom.

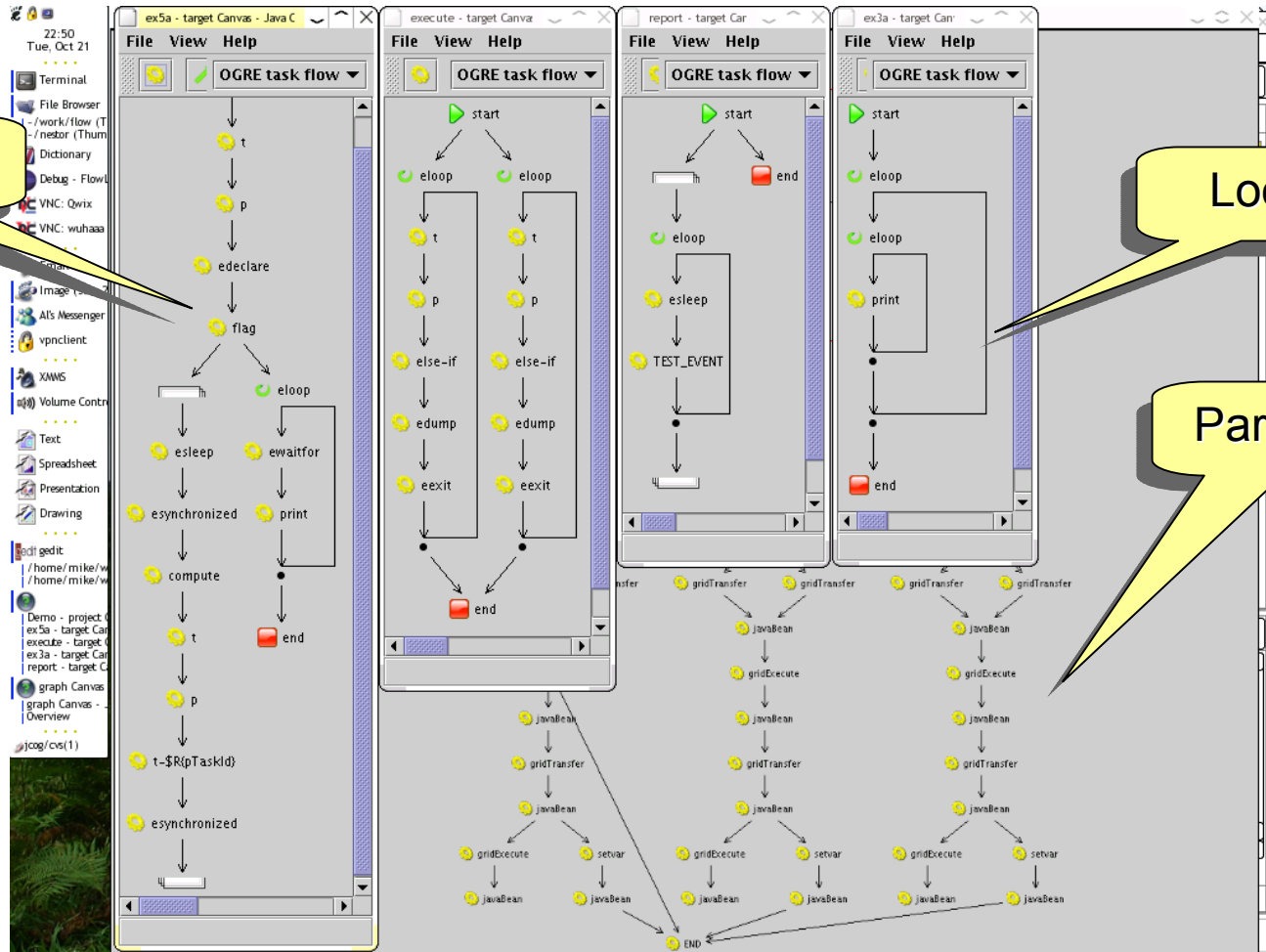
Overview Window

Workflow Details

Interactive Steering



# Java CoG Kit Workflow



Condition

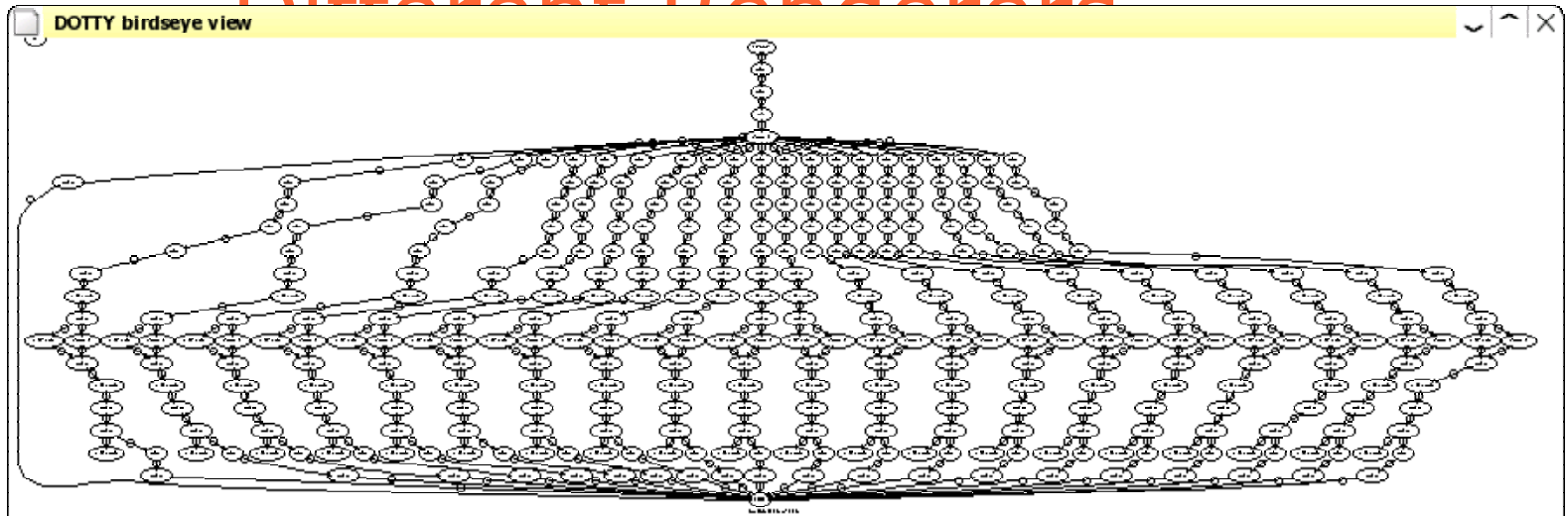
Loop

Parallelism

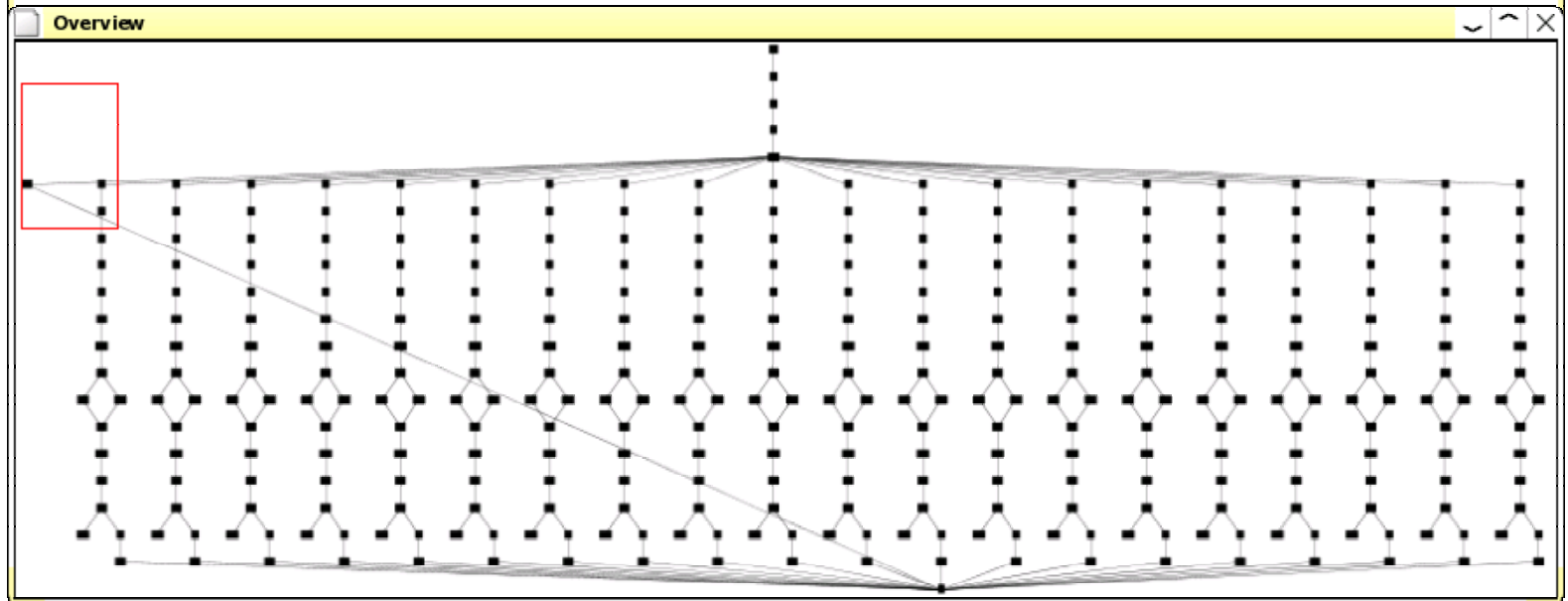


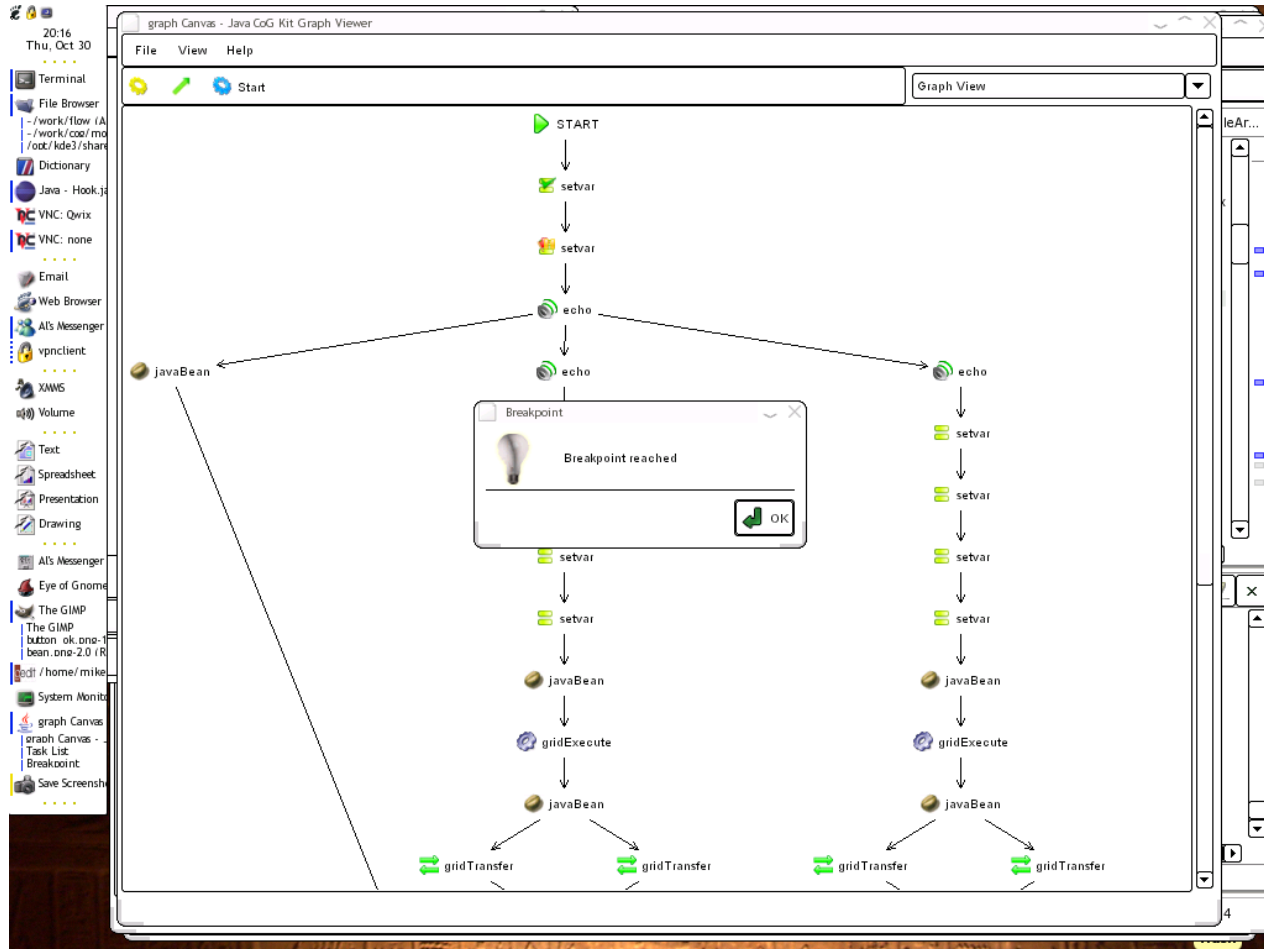
# Different Renderers

- Dot



- CoG







20:16  
Thu, Oct 30

graph Canvas - Java CoG Kit Graph Viewer

File View Help

Start

Graph View

START

setvar

setvar: editable, 0 sub-nodes

Properties

Edit

Remove Breakpoint

Resume execution

Expanded

Interactive

Keep aspect ratio

Show number of sub-nodes

javaBean

echo

setvar

setvar

setvar

setvar

setvar

setvar

setvar

javaBean

gridExecute

javaBean

gridTransfer

gridTransfer

gridTransfer

gridTransfer

4



# Fault tolerance

The screenshot shows the Java CoG Kit Graph Viewer interface. The main window displays a workflow graph with the following structure:

- Start (blue circle icon)
- setvar (green checkmark icon) - three parallel paths
- javaBean (globe icon) - one path
- gridExecute (globe with 'g' icon) - one path
- javaBean (globe icon) - one path

The status bar at the bottom indicates "167 nodes, 178 edges".

An "Error" dialog box is open in the foreground, displaying the following information:

- Error details:** @SSEException: Expired credentials detected
- Location:** gridExecute @ /home/mike/work/recovered/workflow/templates.xml, line: 28
- Actions:**
  - Abort
  - Ignore
  - Restart ; times:
- Options:**
  - Apply to all errors of this type
  - Apply to all errors for this element
-



the globus alliance

[www.globus.org/cog](http://www.globus.org/cog)

# Integration





# CoG Kit Desktop (in progress)

The screenshot displays the CoG Kit Desktop interface. On the left, a 'Directory Browser' window shows a file tree for 'gsiftp://arbat.mcs.anl.gov', including folders like 'cogDevel' and 'listener', and various Java files. In the center, a 'Java CoG Kit: Karajan Workflow' editor shows a vertical flowchart with nodes: 'START', 'Input lastname, firstname, marrige status, and gender', 'Printing the first name', 'Printing the last name', 'Printing the first and last name', 'Printing the sex', and 'END'. A status bar at the bottom of the workflow editor indicates '7 nodes, 6 edges'. On the right, a 'Shell Panel' window shows a command prompt with some text. The desktop background is blue and contains several icons, including 'New Icon', 'Arbatt.mcs', 'GT3-JOB', 'Class', and 'GT2-JOB'. A 'Status area...' is visible at the bottom left of the desktop window.

Job Icons

Machine Icons

File Transfer GUI

Native Icons

Grid Shell

Grid Log

GridAnt/Karajan



# Portlets: OGCE.org

Jakarta Jetspeed Portal: My Home

File Edit View Web Go Bookmarks Tabs Help

http://localhost:8080/jetspeed/portal/media-type/html/user/turbine/page/default.psmi?events=100

Welcome **Tommy Turbine**

My Pages: My Home

Customize: [HTML](#) [WML](#)  
[Edit account: turbine](#)  
[Logout](#)

the globus alliance  
www.globus.org/cog

CoG Workflow

Java CoG Kit: Workflow

Grid Identity: /DC=org/DC=doegrids/OU=People/CN=Mihael Hategan 336881

New Refresh

Workflows:

Id	Workflow Name	Status	Actions
0	Nano materials	Stopped	Edit Delete Start
54	test	Stopped	Edit Delete Start
55	test2	Completed	Edit Delete Start

xportlets : ProxyManager

(default proxy) /DC=org/DC=doegrids/OU=People/CN=Mihael Hategan 336881

Get New Proxy

Jakarta Jetspeed Portal: My Home

File Edit View Web Go Bookmarks Tabs Help

http://localhost:8080/jetspeed/portal/media-type/html/user/turbine/page/default.psmi

Welcome **Tommy Turbine**

My Pages: My Home

Customize: [HTML](#) [WML](#)  
[Edit account: turbine](#)  
[Logout](#)

the globus alliance  
www.globus.org/cog

CoG Workflow

Java CoG Kit: Workflow

Grid Identity: /DC=org/DC=doegrids/OU=People/CN=Mihael Hategan 336881

Back Update

Auto update (0 to disable): 10

Show only grid tasks

Submit and update

xportlets : ProxyManager

(default proxy) /DC=org/DC=doegrids/OU=People/CN=Mihael Hategan 336881

Get New Proxy

Java CoG Kit Graph Viewer

```
graph TD; START --> G1[gridExecute]; G1 --> G2[gridExecute]; G1 --> G3[gridExecute]; G1 --> G4[gridExecute]; G2 --> G5[gridExecute]; G3 --> G5; G4 --> G5; G5 --> G6[gridExecute]; G6 --> G7[gridExecute]; G7 --> END[END];
```



the globus alliance

[www.globus.org/cog](http://www.globus.org/cog)

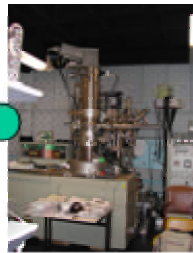
# Nanomaterials Application

With the Java CoG Kit

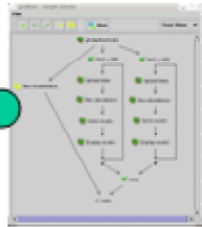


1)

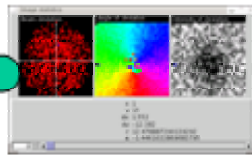
monitoring



2)



3)



Computational steering



Asynchronous processes

Data Acquisition

Data Analysis

Result Display

Default Backup

aquisition

Optional Backup

analysis

Optional Backup

results

When results are good

Computational steering

Collaborators  
With secure access





# Nanomaterials

The screenshot displays the Java CoG Kit Graph Viewer interface. At the top, there are several panels: 'Image statistics' with a 'Status' bar, and four data columns: 'Beam deviation', 'Angle of deviation', 'Intensity of deviation', and 'Angle and intensity'. Below these is a 'Kill task' button. The main area is a 'Graph View' window showing a workflow graph with nodes like 'javaBean', 'gridTransfer', 'gridExecute', and 'gridTransfer' connected by arrows. A 'Zoom Area' callout points to a specific part of the graph. At the bottom, an 'Overview Window' shows a grid of small icons representing the entire workflow, with a red box highlighting the area shown in the 'Graph View' window. A 'Zoom Window' callout points to the 'Graph View' window itself. The interface also includes a 'File View Help' menu, a 'Start' button, and a search bar at the bottom.

Application Window

Zoom Window

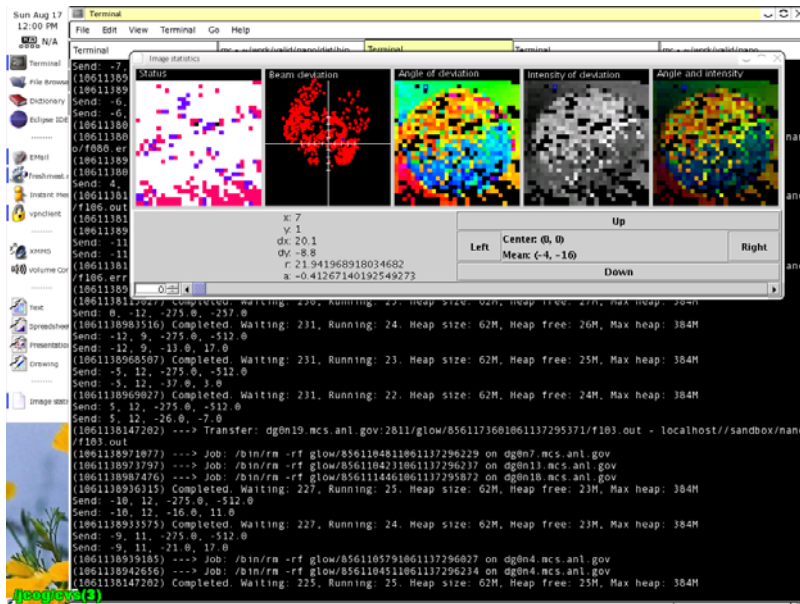
Overview Window

Zoom Area



# Nanomaterials in Adhoc Fashion

- 900 images taken in novel apparatus
- 6 hours analysis in 45 minutes
- Injection of new filters during runtime
- Step toward Adhoc Grid

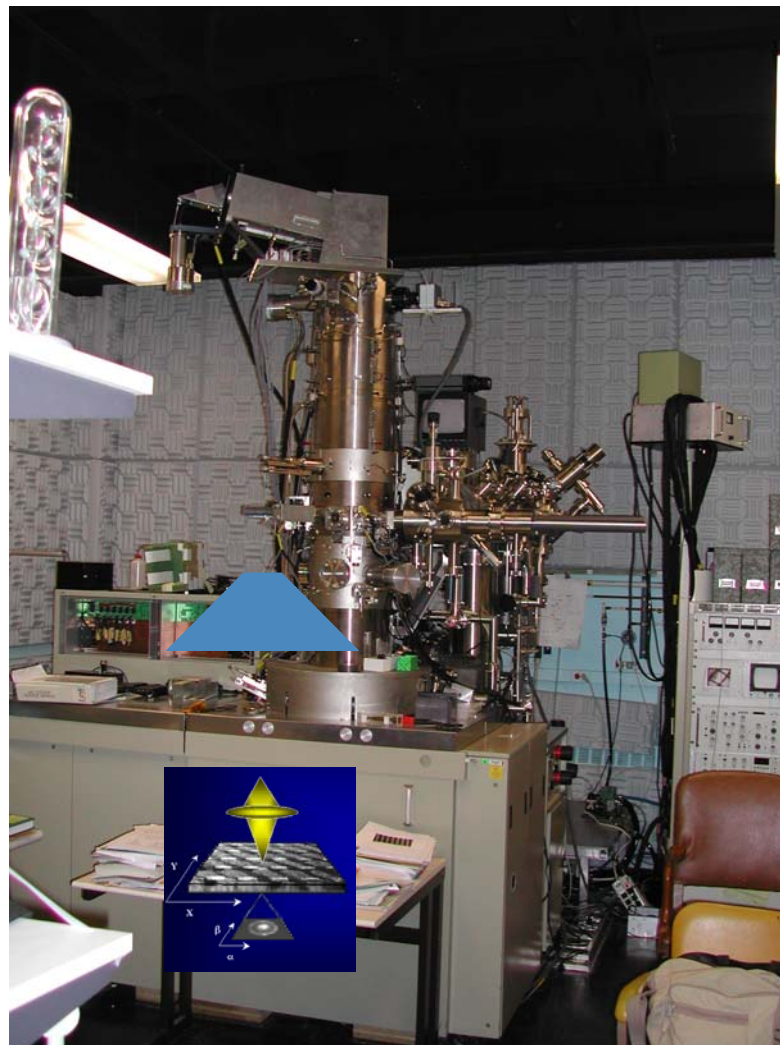
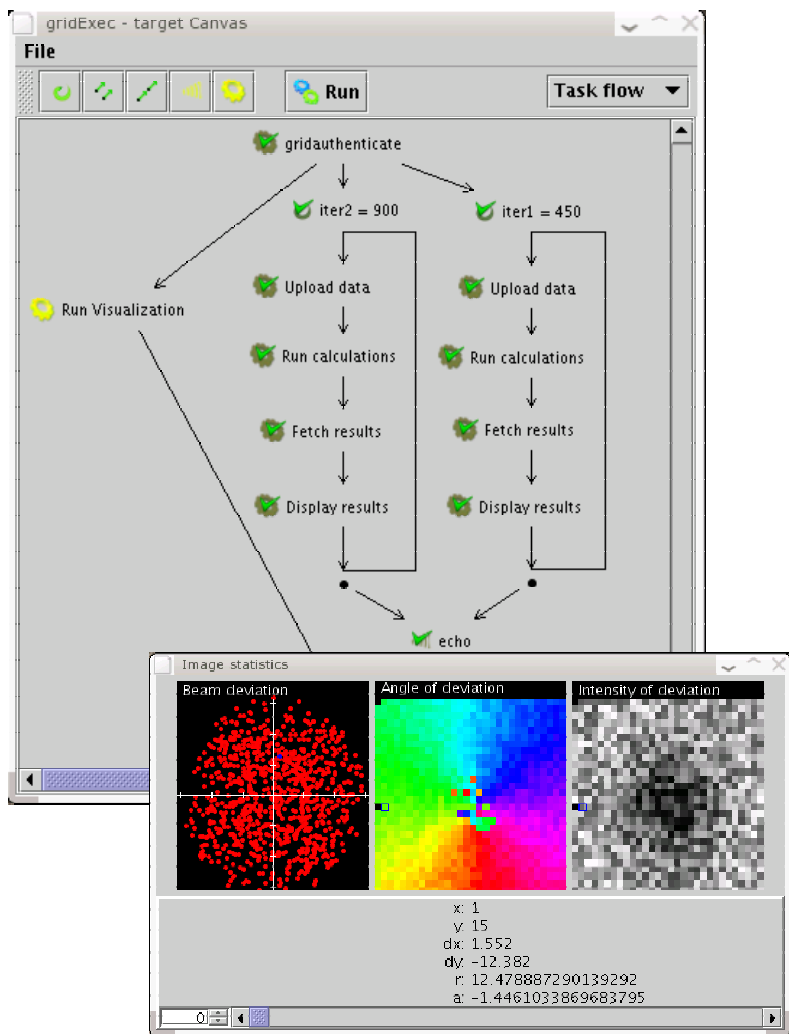




the globus alliance

www.globus.org/cog

# Instrument





# Contributing

- You can contribute
- We have a module concept allowing components to be integrated in the distribution easily





## Conclusion

- Workflow is possible in Java CoG Kit
- Allows integration of other commodity technology
- Open source
- Expandable
- Support of current and future Grid middleware
- We can customize!



# Conclusion

- Programming with CoG workflows is simple
- We envision multiple programming models in CoG
- We envision multiple backend services
- We can support multiple protocols
  
- We like to engage the community
- Contributions:
  - ◆ CA management, Unicore provider, gsissh
  - ◆ These contributions are being integrated.
  
- We did not talk about the GridShell which is a topic by its own.