



PERFORMANCE ANALYSIS SERVICE USER MANUAL

WP3

Document Filename:	KWF-WP3-UIBK-v1.1-PASUserManual
Work package:	WP3
Partner(s):	UIBK, CYFRONET, FIRST
Lead Partner:	UIBK
Document classification:	PUBLIC

Abstract: This document is a user guide manual for performance analysis module (DIPAS) which is implemented in this K-WfGrid project. This report describes in detail steps for setting up and using DIPAS module.

Delivery Slip

	Name	Partner	Date	Signature
From	Hong-Linh Truong	UIBK	04/01/2006	
Verified by	Piotr Nowakowski	CYFRONET	07/01/2006	
Approved by	Steffen Unger	FIRST	10/01/2006	

Document Log

Version	Date	Summary of changes	Author
0.1	25/11/2005	Initial text, table of content	Peter Brunner, Francesco Nerieri, Robert Samborski, Hong-Linh Truong
0.2	30/11/2005	Change all sections	Hong-Linh Truong
0.3	1/12/2005	Update all sections	Hong-Linh Truong with input from Peter Brunner, Francesco Nerieri, Robert Samborski
0.4	04/01/2006	Revise all sections	Hong-Linh Truong
0.5	23/08/2006	Update and change all sections	Robert Samborski with input from Hong-Linh Truong, Peter Brunner
0.55	29/08/2006	Update and revise all sections	Hong-Linh Truong
0.6	31/08/2006	Revise all sections	Hong-Linh Truong
0.7	22/09/2006	Revise all sections, based on internal review comments	Hong-Linh Truong
1.0	07/10/2006	QA check	Piotr Nowakowski
1.1	07/02/2007	Update and Revise sections 3 and 4	Peter Brunner
	14/02/2007	Revise all sections	Hong-Linh Truong

CONTENTS

1	COPYRIGHT NOTICE.....	5
2	INTRODUCTION.....	6
2.1	ABBREVIATIONS AND ACRONYMS.....	7
2.2	REFERENCES AND SOURCE CODE.....	7
3	SOFTWARE USAGE.....	9
3.1	DEPLOYING DIPASPORTAL AND DIPASGATEWAY.....	10
3.1.1	<i>Prerequisites</i>	10
3.1.2	<i>Setup</i>	11
3.1.2.1	DIPASPortal.....	11
3.1.2.2	DIPASGateway.....	12
3.2	RUNNING DIPASGATEWAY.....	12
3.2.1	<i>Operating Requirements</i>	12
3.2.2	<i>Grid software infrastructure</i>	12
3.2.3	<i>Step-by-Step User Setup</i>	12
3.3	RUNNING THE PORTAL.....	13
3.3.1	<i>Operating Requirements</i>	13
3.3.2	<i>Local hardware requirements</i>	13
3.3.3	<i>Local software requirements</i>	13
3.3.4	<i>Grid infrastructure requirements</i>	14
3.3.5	<i>Step-by-Step User Setup</i>	14
3.4	BASIC OPERATION.....	16
3.4.1	<i>Infrastructure Monitoring</i>	16
3.4.1.1	Configure monitoring service plug-in.....	17
3.4.1.2	Create PDQS Request.....	19
3.4.1.3	Query/Subscribe Data.....	21
3.4.1.4	View Resulting Data.....	22
3.4.1.5	View Error Messages.....	24
3.4.2	<i>Workflow Monitoring and Analysis</i>	24
3.4.2.1	Configure DIPAS Gateway.....	24
3.4.2.2	Get existing Workflow IDs.....	25
3.4.2.3	Select a Workflow.....	25
3.4.2.4	Start Monitoring and Analysis.....	26
3.4.2.5	Conduct Online PMA.....	27
4	INTERFACE REFERENCE GUIDE.....	35
4.1	PORTAL LOGIN.....	35
4.2	INFRASTRUCTURE PMA.....	35
4.2.1	<i>Configuration</i>	36
4.2.2	<i>PDQS</i>	36
4.2.3	<i>Display Data</i>	37
4.2.4	<i>Error</i>	38
4.3	WORKFLOW PMA.....	38
4.3.1	<i>Menu File</i>	39
4.3.2	<i>Menu Monitoring</i>	39
4.3.3	<i>Menu View</i>	40
4.3.4	<i>Menu Analysis</i>	40
4.3.5	<i>Split Pane</i>	41
4.3.5.1	WF visualization.....	42
4.3.5.2	Status & Activity Display.....	43
4.3.6	<i>Popup GUI Elements</i>	44
4.3.6.1	Workflow Execution Finished Notification.....	44
4.3.6.2	Debug Window.....	45

4.3.6.3	Workflow Info.....	45
4.3.6.4	Execution Phases of Workflow Activities.....	45
4.3.6.5	Load Imbalance.....	45
4.3.6.6	Mean Execution Time.....	45
4.3.6.7	Distribution of Activity Instances.....	45
4.3.6.8	Workflow overhead.....	46
4.3.6.9	Performance Severity.....	46
4.3.6.10	Performance Problems.....	46
5	TROUBLESHOOTING Q&A	47
6	CONTACT INFORMATION AND CREDITS	48
7	THE GPL LICENSE AGREEMENT.....	49
8	REFERENCES.....	50

1 COPYRIGHT NOTICE

Copyright (c) 2004-2006 Distributed and Parallel Systems Group, University of Innsbruck. All rights reserved.

Use of this product is subject to the terms and licenses stated in the GPL license agreement. Please refer to Section 7 for details.

Java™ is a registered trademark of Sun. All rights reserved.

Applet™ is a registered trademark of Sun. All rights reserved.

Jena is copyrighted by Hewlett-Packard Development Company, LP.

JFreeChart is licensed under GPL, JFree software projects (www.jfree.org).

Gridsphere is copyrighted by Max-Planck-Gesellschaft/Max-Planck-Institut fuer Gravitationsphysik.

Castor is copyrighted by Intalio Inc., and others.

Tomcat, Xerces are licensed under the Apache License, The Apache Software Foundation.

This research is partly funded by the European Commission IST-2002-511385 Project “K-WfGrid”.

2 INTRODUCTION

DIPAS (Distributed Performance Analysis Service) controls the monitoring and instrumentation service, conducts the performance analysis of workflows at runtime and provides performance metrics proposed by the metric ontology to clients. This stable version is a collection of performance monitoring and analysis tools that can be accessed by the user through a web portal, supporting the user to conduct the performance monitoring and analysis of Grid infrastructure and workflows in a user friendly way with many options available. The tools collect the data dynamically from services running in distributed Grid sites, analyze performance and monitoring data, and visualize the performance results in the portal. Detailed technical background can be found in the DIPAS developer manual [DIPASDEV].

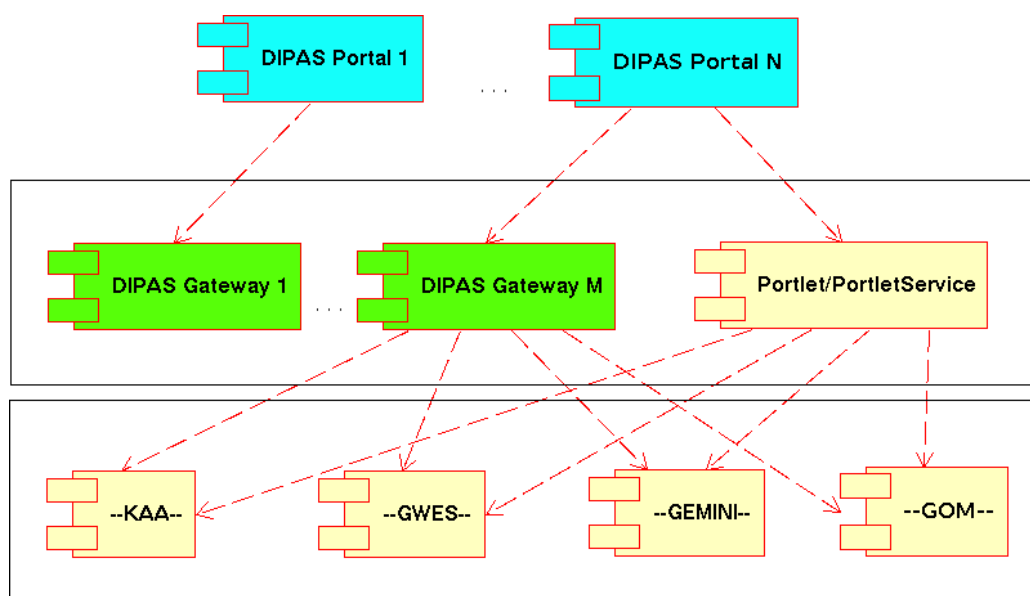


Fig. 2-1: Overview architecture of DIPAS.

Fig. 2-1 depicts the architecture of DIPAS that includes three main parts:

- *The DIPAS portal* provides a single place for the user to conduct the performance monitoring and analysis of Grid infrastructure and workflows. The content of DIPAS portal is provided by DIPAS portlet/portletservices. Moreover, the portal includes a Java applet implementing the main GUI of performance analysis and visualization of workflows. The applet visualizes the monitoring data of the workflows and partially analyzes the performance of the workflows. It also controls the DIPASGateways (see below) to perform the workflow overhead analysis and displays the resulting overhead analysis to the user. The Java applet is configured supporting Java Plug-in mechanism [JAVA-PLUGIN]. The portal is deployed into a web container based on Tomcat [TOMCAT].
- *The DIPAS Portlet/PortletServices:* they are implemented by using Gridsphere [GRIDSPHERE]. The portal interacts with portlets [JSR168] and portlet services which process user requests through web interfaces and generate contents displayed in the portal. Portlets/PortletServices are used to provide monitoring data of Grid infrastructure to the portal.

- *The DIPAS Gateway* is a GT4 WSRF [WSRF] service that acts as a mediator between the portal and various services (e.g., GOM, GEMINI, GWES) involved in the performance monitoring and analysis. Moreover, it implements the overhead analysis and search for performance problems.

2.1 ABBREVIATIONS AND ACRONYMS

Abbreviation	Description
DIPAS	Distributed Performance Analysis Service
GEMINI	Grid Monitoring and Instrumentation Service
GOM	Grid Organizational Memory
GWES	Grid Workflow Execution Service
KAA	Knowledge Assimilation Agent
PDQS	Performance Data Query and Subscription
URL	Uniform Resource Locator
WARL	Workflow Analysis Request Language
WF	Workflow
WfID	Workflow ID
WfMS	Workflow Management System
WSRF	Web Service Resource Framework

2.2 REFERENCES AND SOURCE CODE

The whole DIPAS source code consists of two parts:

- `dipasportal`: including DIPAS Portal and portlets/portletservices
- `dipasservice`: including DIPAS Gateway

The entire source code can be downloaded from K-WfGrid CVS. Fig. 2-2 presents the structure of source directory of `dipasportal`. There are four main directories:

- The `lib` directory contains all the third party libraries that we need to develop our portal. DIPASPortal is dependent on several third party libraries like Castor [CASTOR], JFreeChart [JFREECHART], the K-WfGrid DR package [DRUSER] [DRDEV], JUnit[JUNIT], SCALEA-G [SCALEAG] and other XML Apache libraries [XMLAPACHE].
- The `src` directory includes all Java source files developed in DIPASPortal.
- The `webapp` directory includes HTML and JSP files of DIPASPortal.
- The `test` directory includes JUnit test files

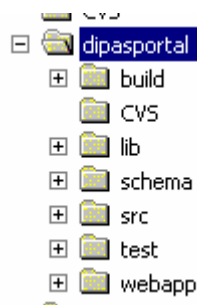


Fig. 2-2: Source code directory structure of dipasportal.

Fig. 2-3 presents the structure of source directory of dipasservice that includes 4 main directories:

- The `lib` directory includes the third party libraries. Important libraries here are Castor, Glassbox, K-WfGrid Gworkflowdl, GWUI, JFreechart [JFREECHART], JUnit, the K-WfGrid DR package, and the SCALEA-G libraries.
- The `samples` directory consists of XML samples.
- The `schema` contains WSDL files of Grid services of DIPASGateway.
- The `src` directory contains Java source files of DIPASGateway.
- The `test` directory includes JUnit test files

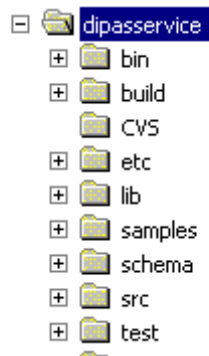


Fig. 2-3: Source code directory structure of DIPASGateway.

3 SOFTWARE USAGE

Monitoring and analyzing distributed workflows is a complex matter and requires many different techniques working hand-in-hand. This prototype supports the user to conduct the performance monitoring and analysis of Grid infrastructure and workflows through a single portal. This task is not simple but our goal is to make the usage of our tool easy and intuitive. The user has minimum effort to start: the user just opens a web browser, types in the URL of the portal, and logs into the portal. After the login the user can conduct the monitoring and analysis by selecting analysis functions available in the portal. The portal is implemented based on Gridsphere which is deployed in a Tomcat container. The content of the portal is provided by various portlets, like DIPASConfigService, DIPASPortalService, DIPASGateway and DIPASErrorService, and Java applets. Those portlets and services in return communicate with GOM, GEMINI, GWES and KAA. Communication among Grid services is hidden from the user and the user interface.

Please refer to the appropriate developer guide [DIPASDEV] for an in-depth technical description of the prototype and consult the use cases shown in Fig. 3-1 and Fig. 3-2.

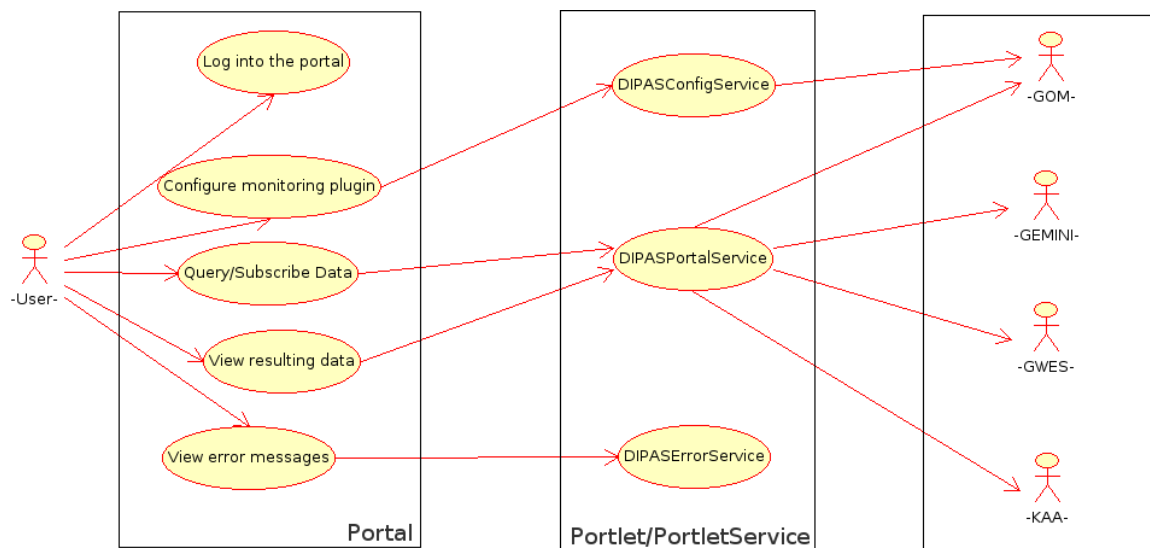


Fig. 3-1: Infrastructure monitoring.

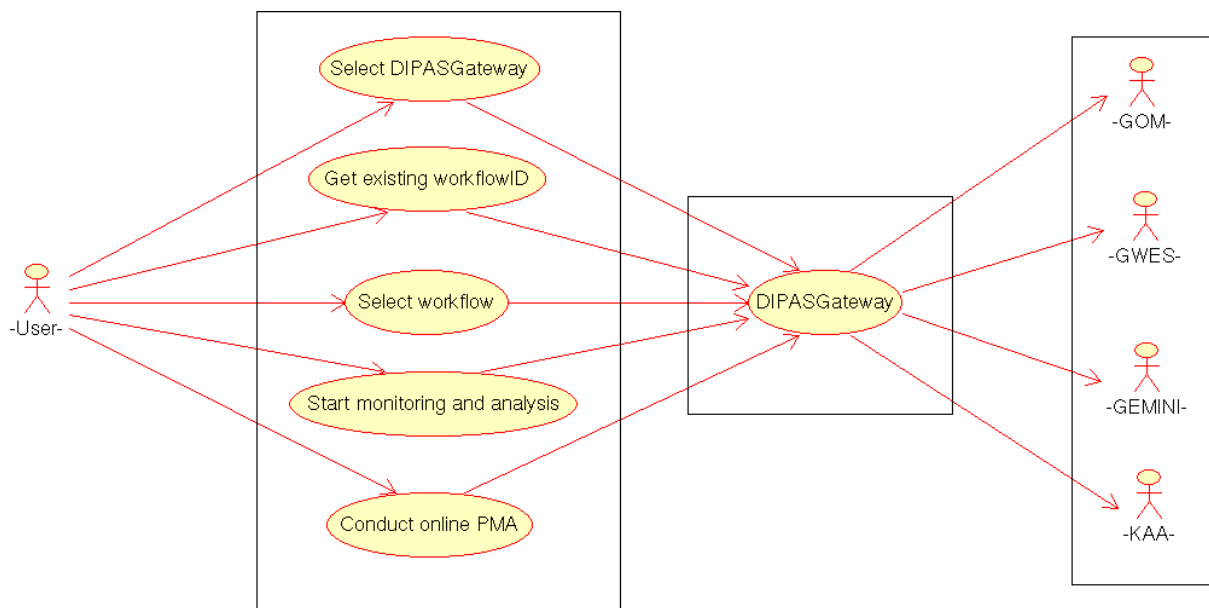


Fig. 3-2 Online performance monitoring and analysis of workflows

3.1 DEPLOYING DIPASPORTAL AND DIPASGATEWAY

It is not required to deploy DIPASPortal locally in order to use the full functionality of the program. Users who only want to use the services can skip this section entirely. Using the portal requires only a login to a web site hosting K-WfGrid or the DIPAS component.

Users who wish to run their own local copy of DIPASPortal must have Ant [ANT], Tomcat [TOMCAT] and Gridsphere [GRIDSPHERE] installed. The installation of this software is required before deploying the portal.

3.1.1 Prerequisites

You must have Ant, Tomcat and Gridsphere installed. To install the prerequisites refer to the installation document of corresponding software; examples are given below for your convenience:

- If not already installed, download ant version 1.6 or higher and install it.
- Download the jakarta-tomcat-5.5.16 archive and unpack it.
- Set CATALINA_HOME, e.g. in your .profile.local, to the directory you unpacked the archive, e.g.:

```
$export CATALINA_HOME=~/.jakarta-tomcat-5.5.16
```
- Add a user for gridsphere in \$CATALINA_HOME/conf/tomcat-users.xml, for example with values:

```
<user username="gridsphere" password="my_passwd"
roles="manager,admin" />
```

- Set the correct path to Java (1.5), e.g.:

```
$export JAVA_HOME=/usr/local/java/java-current
```

DIPAS has been developed and tested with Tomcat 5.5.16, Java 1.5, and Gridsphere 2.0.4/2.1.4. If you experience problems with other versions please use exactly the version stated here. If you experience problems with other components of K-WfGrid, check in the appropriate reference guides if all required Gridsphere components (e.g. gridporlets) have been downloaded, and choose Gridsphere with version 2.0.4 or 2.1.4.

- If you have a restriction on which ports you can use (or if some ports are not available behind a firewall), change the port in `$CATALINA_HOME/conf/server.xml`.
- Download Gridsphere from <http://www.gridsphere.org/> and install it.
- Check if `ANT_HOME` is set and points to the right installation of ANT (version 1.6 or higher). If you have problems when compiling Gridsphere with Java 1.5, compile it with Java 1.4.2, export `JAVA_HOME` to the correct installation, then run:

```
$ant install
```

The correct installation can be tested by starting the Tomcat server

```
$CATALINA_HOME/bin/startup.sh
```

Launch a web browser and go to URL: http://your_server:your_port/gridsphere

Example URL: <http://localhost:8080/gridsphere>

- Shutdown Tomcat before changing or updating configuration and restart afterwards.

3.1.2 Setup

3.1.2.1 DIPASPortal

- Download DIPAS or check out the DIPAS sources from CVS, e.g.:

```
$ cvs co kwfgrid/dipas
```

- Go to the directory `kwfgrid/dipas/dipasportal`
- Open the file `build.properties` and set in the value of the variable `gridsphere.home` to your Gridsphere installation (see prerequisites above).
- `JAVA_HOME` should point to JDK 1.5. To compile and deploy DIPASPortal, goto the `dipasportal` directory and execute the following commands:

```
$ ant compile
```

```
$ ant deploy
```

Note: "BUILD SUCCESSFUL" indicates a success, the project has been deployed into the directory `$CATALINA_HOME/webapps/` and `dipasportal/` subdirectories.

- Set `GLOBUS_LOCATION` to the directory where the WS-Core has been put, e.g.:

```
$ export GLOBUS_LOCATION=~/ws-core-4.0.2
```

- Deploy WS-Core into tomcat with:

```
$ cd $GLOBUS_LOCATION
```

```
$ ant -f share/globus_wsrf_common/tomcat/tomcat.xml  
deploySecureTomcat -Dtomcat.dir=<tomcat.dir>
```

- Copy the library `$GLOBUS_LOCATION/lib/xalan.jar` to `$CATALINA_HOME/common/endorsed/`

3.1.2.2 DIPASGateway

- Go to the directory of DIPASGateway, e.g.: `kwfgrid/dipas/dipasservice`
- Open the file `build.properties` and set in the value of the variable `ogsa.root` to your `$GLOBUS_LOCATION`.
- In the `dipasgateway` directory execute the following commands:

```
$ ant jar
$ ant gar
$ $GLOBUS_LOCATION/bin/globus-deploy-gar-
build/lib/dipas.gar
```

Note: "BUILD SUCCESSFUL" indicates a success, the project has been deployed into the directory `$GLOBUS_LOCATION` and subdirectories. Java libraries (JAR files) are copied from `dipasservice/lib` to `$GLOBUS_LOCATION/lib`, make sure you use the correct libraries.

3.2 RUNNING DIPASGATEWAY

3.2.1 Operating Requirements

DIPASGateway is a GT4-based service that must be deployed into GT4.0 container before it can be used.

3.2.2 Grid software infrastructure

DIPASGateway requires GT4 Java core services.

3.2.3 Step-by-Step User Setup

For compiling and installing DIPASGateway, refer to the `INSTALL` document in the source directory, e.g.: `$ more kwfgrid/dipas/dipasservice/INSTALL`

Before deploying DIPASGateway, information about GOM, GWES and optionally GEMINI has to be configured. For example, Fig. 3-3 presents parameters used to configuration information about GOM, GWES and GEMINI.

```
<parameter name="msPlugin" value="gemini" />
<parameter name="geminiURL" value="http://pc6163-
c703.uibk.ac.at:40065/wsrf/services/monitor/service/MonitoringService" /
>
<parameter name="gomURL" value="http://zeus72.cyf-
kr.edu.pl:8080/wsrf/services/gom/service/GOMService" />
<parameter name="gwesURL" value="http://pc6163-
c703.uibk.ac.at:40350/gwes/services/GWES" />
```

Fig. 3-3: Parameters used to configure information about GOM, GWES and GEMINI.

Currently DIPASGateway stores service configuration information in the file named `dipas-config.wsdd` (which is in the `dipasservice/` directory). Edit this file and change the values of the following parameters:

- `gomURL`: indicates the URL of GOM
- `gwesURL`: indicates the URL of GWES
- `geminiURL`: optionally, indicates the URL of GEMINI

After the deployment, DIPASGateway can be used by the portal.

3.3 RUNNING THE PORTAL

3.3.1 Operating Requirements

The user does not have to install any DIPAS software. All services provided can be accessed through a web browser. For workflow performance monitoring and analysis, the browser must be able to run Java applets.

3.3.2 Local hardware requirements

Because of the software written entirely in Java, it should be running independent of the platform. Therefore, any modern computers will suffice. Please compare your hardware to those given in Table 3-1. We have successfully tested the functionality of the stable version on the platforms given in Table 3-1.

Processor	MB RAM	Operating Systems
AMD Athlon XP 2600+	256	Windows XP Professional Version 2002
UltraSPARC Iii 650MHz	768	SunOS 5.9
Intel Pentium IV 1.8 GHz	512	Linux Fedora Core Release 3
Intel Pentium IV 1.8 GHz	512	Windows XP Professional Version 2002 SP 2

Table 3-1: Local hardware requirements for the portal.

We recommend using a system similar to or better than those shown in Table 3-1 with preferably 100 MB of free memory available (The Internet Explorer Task executing the applet loaded shows a usage of nearly 70 MB). There should be at least about 50 MB of free space on a file system for the required jars that the applet must download (approximately 27 MB at the time of this writing) before execution. Please check the availability of the resources before each start if your system is near those minimum requirements.

Further information is also available in chapter 5 (Q&A) and the developer manual [DIPASDEV].

3.3.3 Local software requirements

In order to use the full functionality of the portal the user must have the following components installed: Java runtime environment (Java 1.5) and a web browser which is capable of running Java applets.

We have successfully tested the functionality of the prototype with configurations shown in Table 3-2.

Operating Systems	Web Browser	Java Version
Windows XP	Mozilla Firefox Mozilla/5.0 (Windows; U; Windows NT 5.2; en-US; rv:1.7.12) Gecko/20050915	"1.5.0_02" Java(TM) 2 Runtime Environment, Standard Edition

	Firefox/1.0.7	
Windows XP	Internet Explorer 6.0	"1.5.0_04" Java(TM) 2 Runtime Environment, Standard Edition (build 1.5.0_04-b05) Java HotSpot(TM) Client VM (build 1.5.0_04-b05, mixed mode)
Linux	Firefox	"1.5.0_05" Java(TM) 2 Runtime Environment, Standard Edition (build 1.5.0_05-b05) Java HotSpot(TM) Client VM (build 1.5.0_05-b05, mixed mode, sharing)

Table 3-2: Local Software requirements.

3.3.4 Grid infrastructure requirements

There must be a working internet connection. Currently there are no further Grid infrastructure requirements for the user who is not running own services but only using the portal. Users running own components should refer to the appropriate developer guide (for example a valid proxy certificate might be required).

The portal, infrastructure monitoring and workflow monitoring and analysis are working fine with a 1 Mbps link; however before it can be used the workflow monitoring and analysis applet needs some time to load. A 100 Mbps link or better is highly recommended.

3.3.5 Step-by-Step User Setup

For compiling and installing DIPASPortal, refer to Section 3.1.2 and the INSTALL documents in the source directory. Also please check various README files within the directory structure of kwfgrid/dipas, e.g.:

- `$ more kwfgrid/dipas/dipasportal/INSTALL`
- `$ more kwfgrid/dipas/dipasportal/README`
- `$ more kwfgrid/dipas/dipasservice/INSTALL`
- `$ ls kwfgrid/dipas/docs`

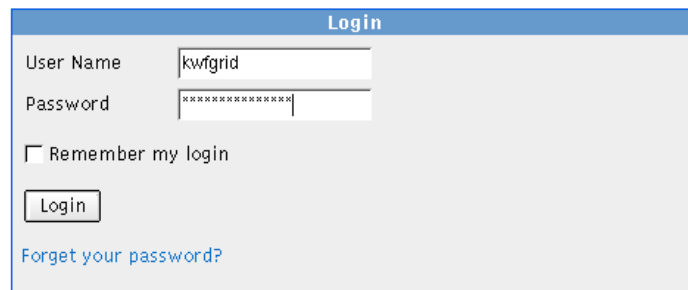
In later releases there may be the requirement of creating a security proxy, at the moment this step can be omitted.

Step 1: Open a web browser and connect to an appropriate URL.

Example: a Gridsphere portal is running on the server localhost on the port 40350.

Type in the URL: <http://localhost:40350/gridsphere>

Step 2: Log in to the portal (see Fig. 3-4 and Fig. 4-1) with your username and password.

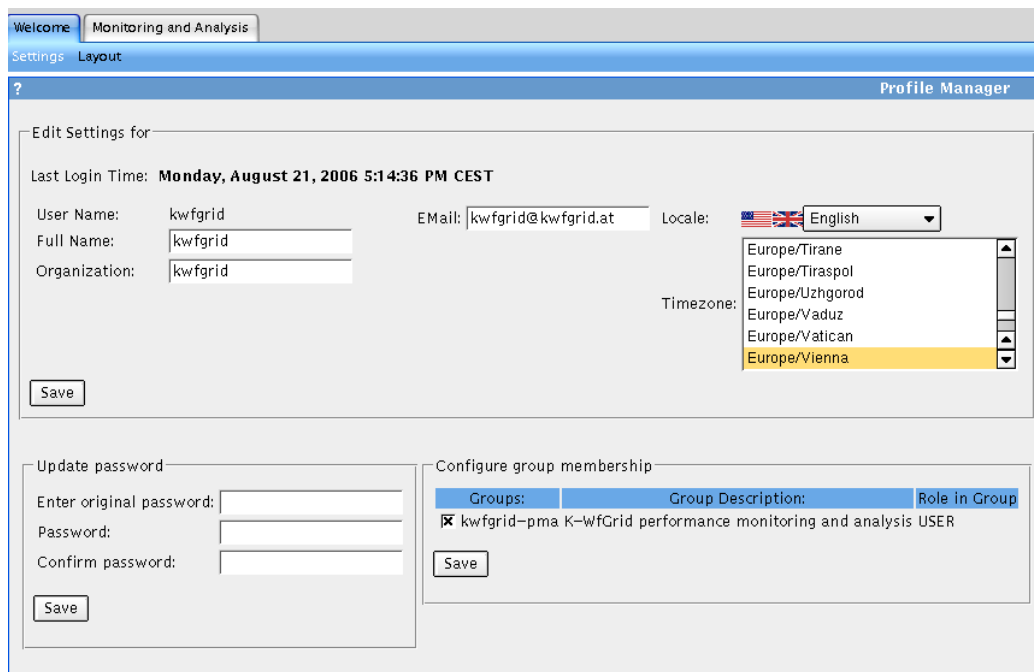


The screenshot shows a 'Login' window with the following fields and elements:

- User Name: kwfgrid
- Password: [masked with asterisks]
- Remember my login
- Login button
- [Forget your password?](#)

Fig. 3-4: Login portal.

Step 3: Configure membership; in the profile manager (see Fig. 3-5) make sure that the group “kwfgrid-pma” with the description “K-WfGrid performance monitoring and analysis...” is checked and then save your settings.



The screenshot shows the 'Profile Manager' interface with the following sections:

- Edit Settings for:** Last Login Time: Monday, August 21, 2006 5:14:36 PM CEST
- User Information:** User Name: kwfgrid, Full Name: kwfgrid, Organization: kwfgrid, Email: kwfgrid@kwfgrid.at
- Locale:** English
- Timezone:** Europe/Tirane, Europe/Tiraspol, Europe/Uzhgorod, Europe/Vaduz, Europe/Vatican, Europe/Vienna (selected)
- Update password:** Fields for original password, password, and confirm password.
- Configure group membership:** A table with columns: Groups, Group Description, Role in Group. The entry 'kwfgrid-pma K-WfGrid performance monitoring and analysis USER' is checked.

Fig. 3-5: Profile manager.

After saving your settings you will be able to select the Monitoring and Analysis tab (see Fig. 3-6) in the portal. Note that tabs other than Monitoring and Analysis depend on which components of K-WfGrid have been deployed at contacted site.



Fig. 3-6: Monitoring and analysis tab.

3.4 BASIC OPERATION

There are two aspects in using the K-WfGrid monitoring and analysis services:

- Monitoring and analysis of Grid infrastructure: the user can select available Grid resources and examine performance behaviour of the selected resources, for example, system loads, machine information, availability, etc.
- Monitoring and analysis of Grid workflows: the user can conduct the online performance analysis of Grid workflows. The user can select existing workflows (completed or being executed by GWES) and examine the performance analysis of the selected workflows.

Each time after logging into Gridsphere and selecting the monitoring and analysis tab (see Fig. 3-6), several options (see Fig. 3-7) will be available for use, such as infrastructure monitoring (tab Infrastructure PMA) or workflow monitoring and analysis (tab Workflow PMA).



Fig. 3-7: Sub tabs of performance monitoring and analysis portal.

During the performance monitoring and analysis every instance of the portal interacts with DIPAS Gateways and other DIPAS distributed services (see Fig. 2-1). The interaction is hidden from the user. For more detailed information please consult the developer guide. In the following, we detail steps in the monitoring and analysis.

3.4.1 Infrastructure Monitoring

Infrastructure Monitoring can be conducted in the Infrastructure PMA tab. Fig. 3-8 gives an overview of the components within this tab. The components are MonitoringService Configuration, Data Query and Subscription, Error Messages and Display Data. For each component a help information can be displayed by clicking on the question mark in the upper left corner (marked with a red circle in Fig. 3-8).

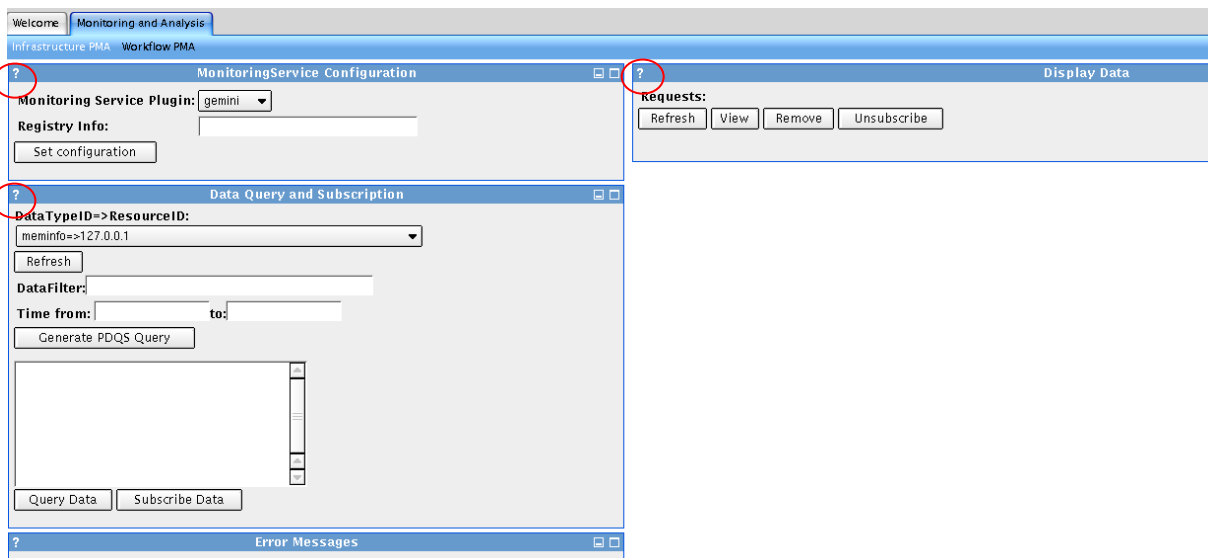


Fig. 3-8: Components of Infrastructure PMA.

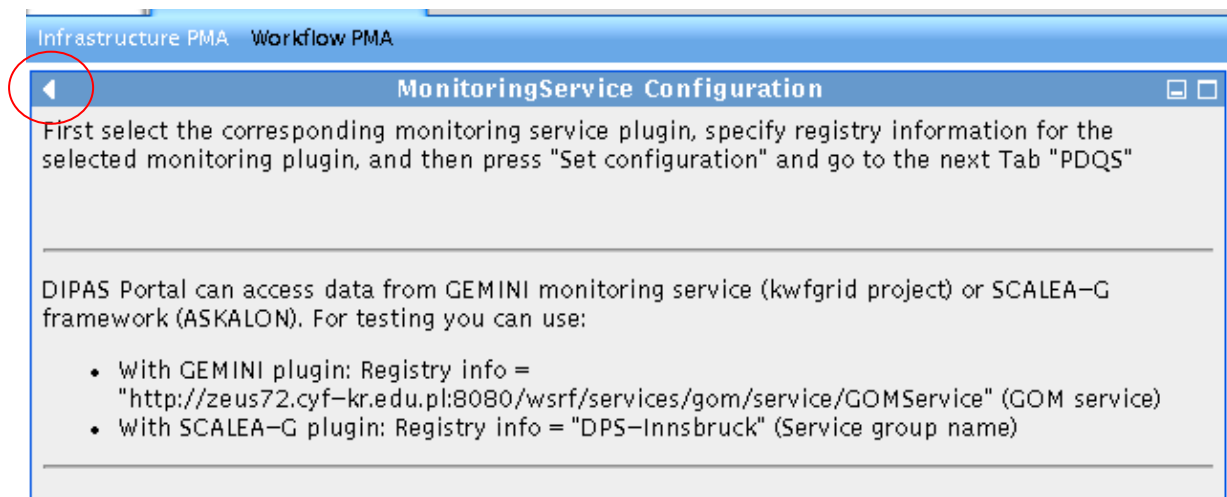


Fig. 3-9: Help for Configuring Monitoring Service.

An example of online help is shown in Fig. 3-9. To return to the component window the arrow left symbol (surrounded by a red circle in Fig. 3-9) must be clicked.

3.4.1.1 Configure monitoring service plug-in

In the `MonitoringService Configuration` area (see Fig. 3-8) the user can choose either the SCALEA-G or the GEMINI plug-in (see Fig. 3-10).

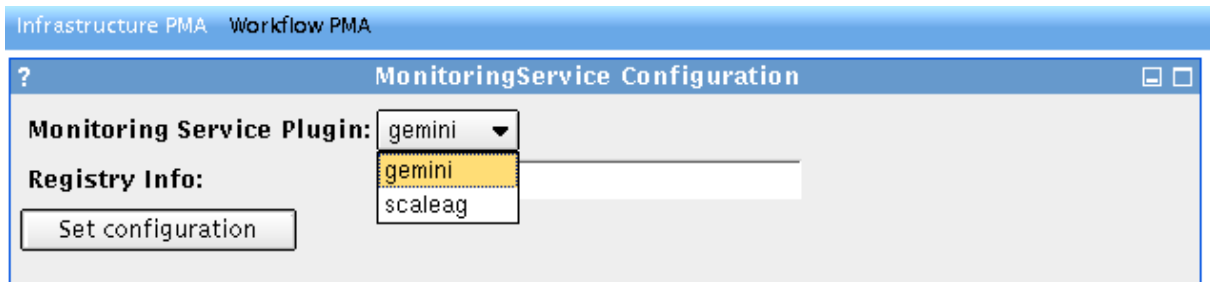


Fig. 3-10: Choose service plug-in.



Fig. 3-11: GEMINI configuration.

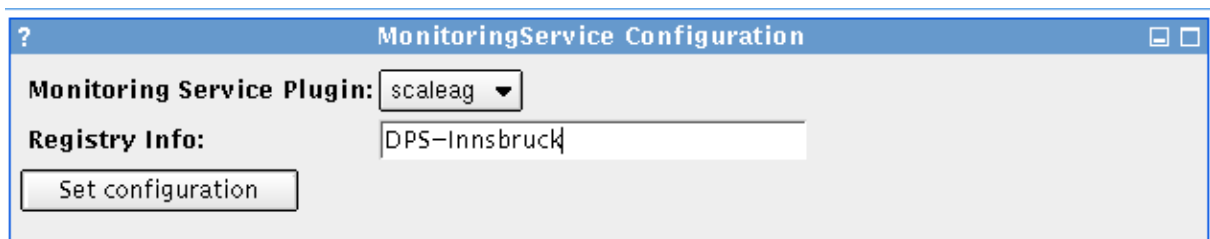


Fig. 3-12: SCALEA-G configuration.

After selecting the plug-in and setting the registry information (see Registry Info in Fig. 3-11 and Fig. 3-12, in doubt use the information proposed in help, see Fig. 3-9) press the Set configuration button, then proceed with the creation of a PDQS request.

This step is normally done only at the first time the user starts to use the monitoring and analysis of Grid infrastructure.

3.4.1.2 Create PDQS Request

The screenshot shows a software interface titled "Data Query and Subscription". At the top, there is a dropdown menu labeled "DataTypeID=>ResourceID:" with the value "meminfo=>127.0.0.1" selected. Below the dropdown is a "Refresh" button. Underneath is a "DataFilter:" text input field. Below that are "Time from:" and "to:" text input fields. A "Generate PDQS Query" button is located below the time fields. At the bottom of the window are two buttons: "Query Data" and "Subscribe Data". A large empty text area is located between the "Generate PDQS Query" button and the bottom buttons.

Fig. 3-13: Data Query and Subscription.

In the Data Query and Subscription component (see Fig. 3-13: Data Query and Subscription.) depending on the previously selected plug-in a request can be generated. First the desired data type ID and the resource ID must be selected (see Fig. 3-14). The available IDs should be listed in a combo box below the **DataTypeID=>ResourceID** label; to update the view please press the Refresh button.

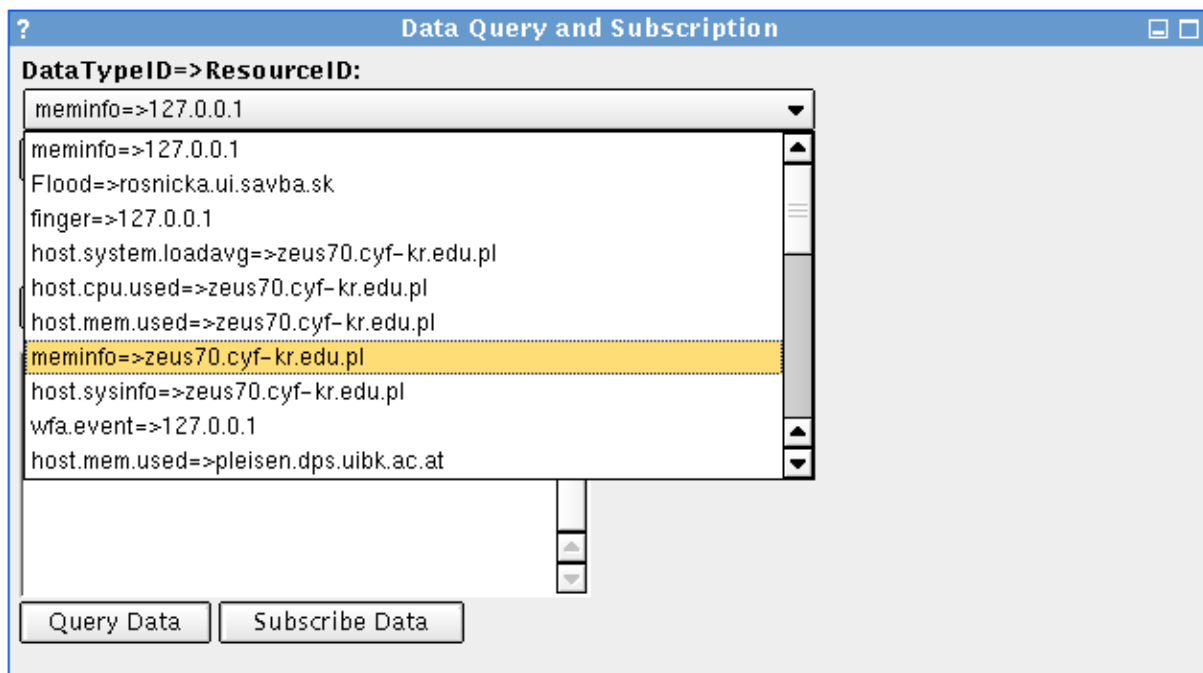


Fig. 3-14: Select resources to be monitored.

When editing a PDQS request, the user can specify a filter and/or type in the time boundaries for a data subscription. After that the user just clicks the Generate PDQS Query button and then the correct PDQS request will automatically be generated and displayed (see Fig. 3-15 and Fig. 3-16 for examples).

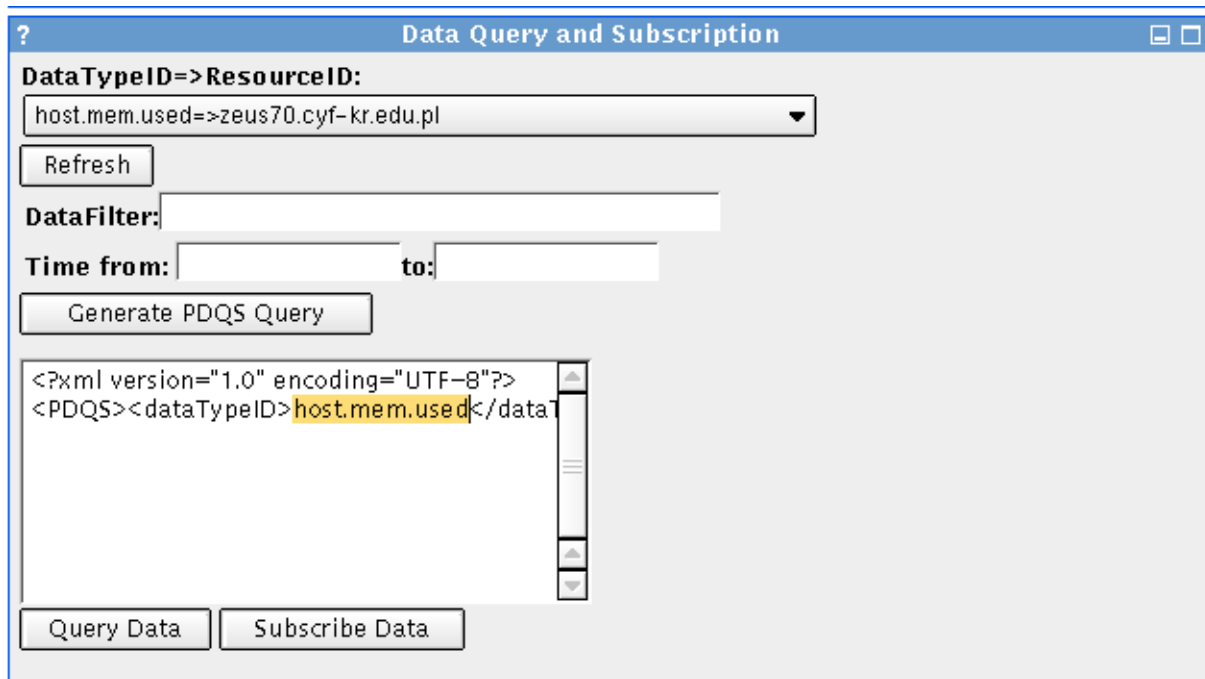


Fig. 3-15: Request for host memory used (GEMINI).

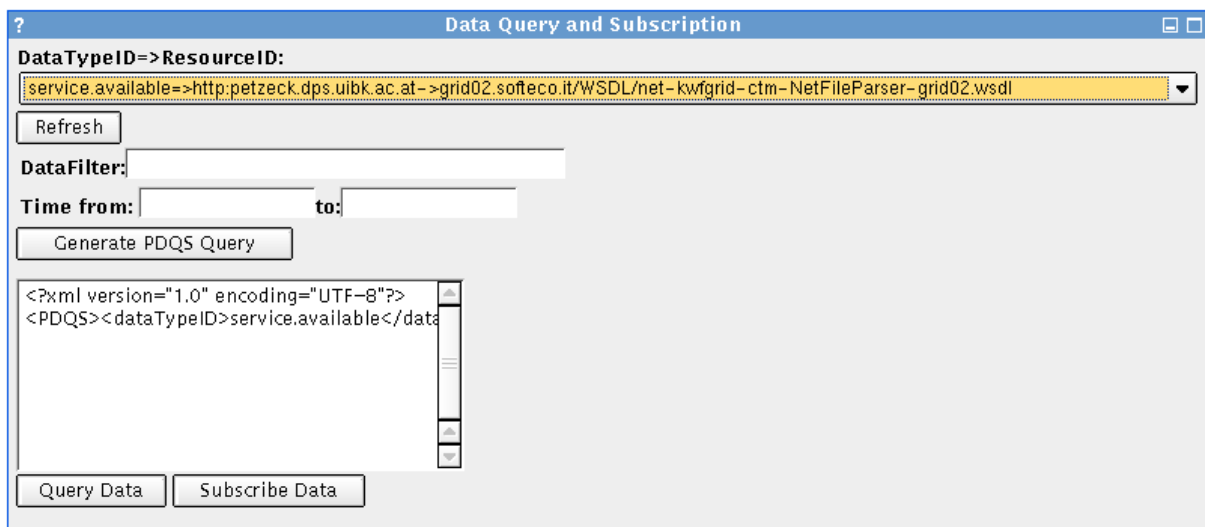


Fig. 3-16: Request for service availability (SCALEA-G).

3.4.1.3 Query/Subscribe Data

After the query has been generated, the XML-based PDQS request can be sent by either pressing the Query Data or the Subscribe Data button (see figures above). Some data types can be only queried. For subscriptions setting the time interval is recommended. The DataFilter is optional.

3.4.1.4 View Resulting Data

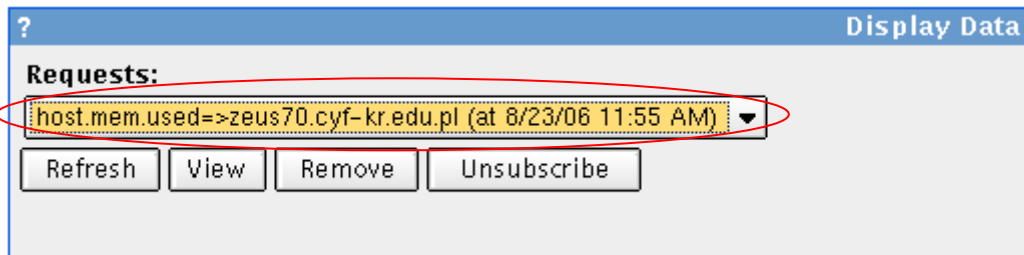


Fig. 3-17: Display Data component.

In the Display Data component (see Fig. 3-17 and Fig. 3-8) the results of the queries are displayed. After conducting a query, as soon as the web browser has stopped loading the results listed in a combo box below the Requests label can be viewed. If the result of the query is not in the list (the combo box with results is surrounded by a red ellipse in Fig. 3-17), press the Refresh button. Select a result ID to display and click the View button. The resulting data will be displayed in a table (Fig. 3-19) or graphical visualization (e.g., for dynamic and time series data, see Fig. 3-18). The representation is chosen automatically depending on the type of monitoring data.

If the data was not entirely transferred or if the data is not being displayed properly the user can force a reload by clicking on the View data button again.

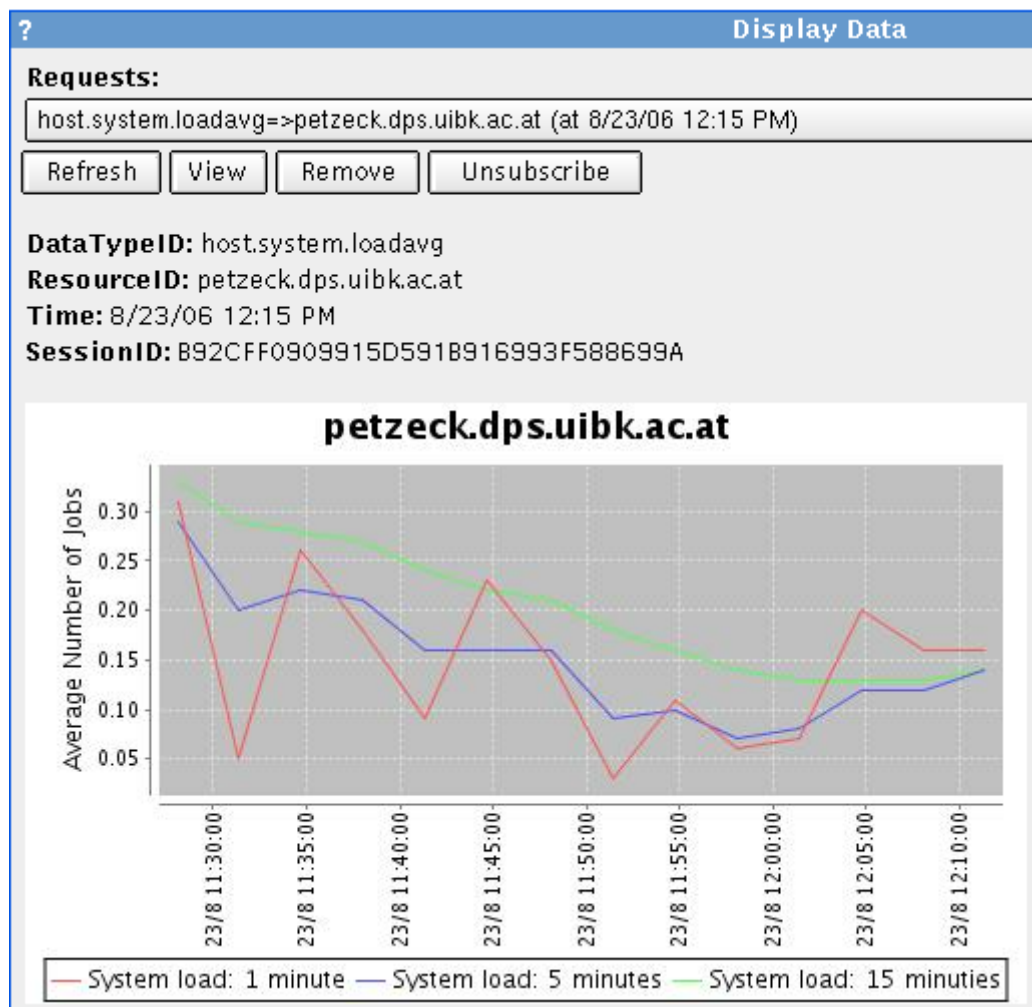


Fig. 3-18: Visualize monitoring data in dynamic mode.

Display Data

Requests: `service.available=>http:petzeck.dps.uibk.ac.at->grid02.softeco.it/WSDL/net-kwfggrid-ctm-NetFileParser-grid02.wsdl (at 8/23/06 12:05 PM)`

DataTypeID: service.available
ResourceID: http:petzeck.dps.uibk.ac.at->grid02.softeco.it/WSDL/net-kwfggrid-ctm-NetFileParser-grid02.wsdl
Time: 8/23/06 12:05 PM
SessionID: B92CFF0909915D591B916993F588699A

Status	Time
UP	Wed Aug 23 11:31:58 CEST 2006
UP	Wed Aug 23 11:39:51 CEST 2006
UP	Wed Aug 23 11:55:53 CEST 2006
UP	Wed Aug 23 11:47:52 CEST 2006
UP	Wed Aug 23 12:03:54 CEST 2006

Fig. 3-19: Visualize monitoring data in static mode.

3.4.1.5 View Error Messages

The user can view information about errors occurred in the Error Messages Component (see Fig. 3-8, bottom left component).

3.4.2 Workflow Monitoring and Analysis

In K-WfGrid, the user submits workflows to GWES, e.g., by using GWES client or portal. The workflows will be executed by GWES. Such workflows will be monitored, and then their performance can be analyzed by using the workflow monitoring and analysis portal. The monitoring and analysis portal supports the user to analyze completed and running workflows.

Clicking on the tab Workflow PMA (see Fig. 3-7) starts the process of loading the applet for WPMA. The applet has been configured using the Java plug-in mechanism. The loading of the necessary jar files may take some time, depending on the speed of the internet connection.

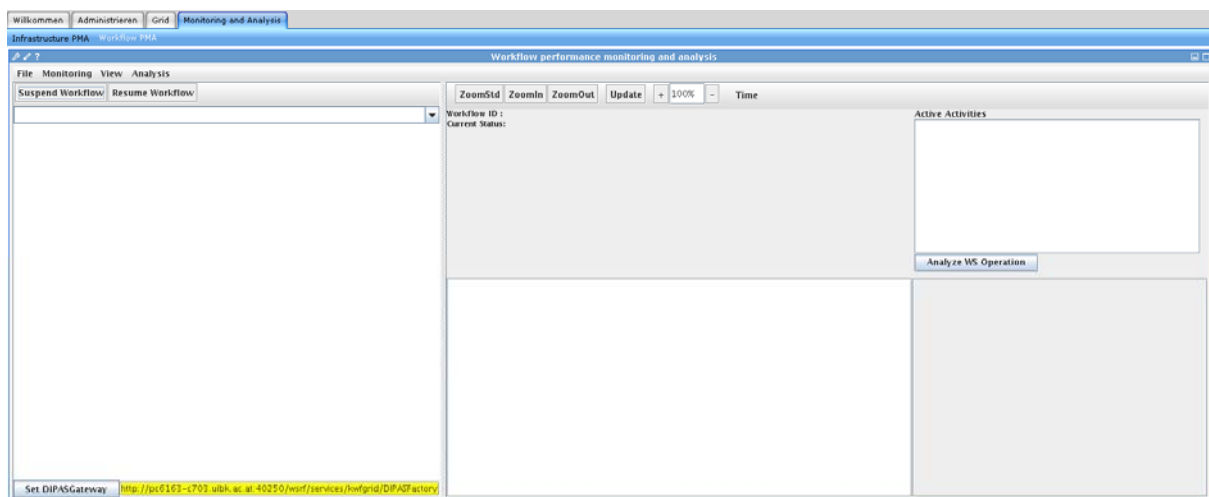


Fig. 3-20: Applet supporting workflow performance monitoring and analysis

When the applet finished loading the user should see the main GUI as depicted in Fig. 3-20. This step is normally done only at the first time the user starts to do the performance analysis of workflows.

Note: You can activate a log window in the File→Debug menu. For every action you will receive feedback in the log text field of the debug window.

3.4.2.1 Configure DIPAS Gateway

The bottom line (see Fig. 3-21) of the applet shows the URL of the current DIPASFactory (a factory that creates an instance of DIPASGateway when requested).

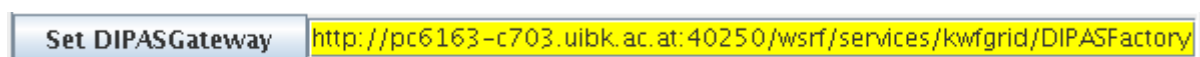


Fig. 3-21: Gateway URL

To set a new URL click into the yellow line and type in the new address then select Set DIPASGateway.

This step is normally done only at the first time the user starts the performance analysis of workflows.

Note: The log information in the optional feedback window area informs (if activated) that the new address has been recognized. However, the portal does not force the factory to produce a new service; this will happen on demand only if information from the service is needed. This means that setting DIPASGateway does not conduct an immediate availability test.

3.4.2.2 Get existing Workflow IDs

The user can start the performance analysis of workflows by obtaining existing workflows: Click "Get Workflow ID" from the Workflow menu (see Fig. 3-122) to retrieve a list of existing workflows.

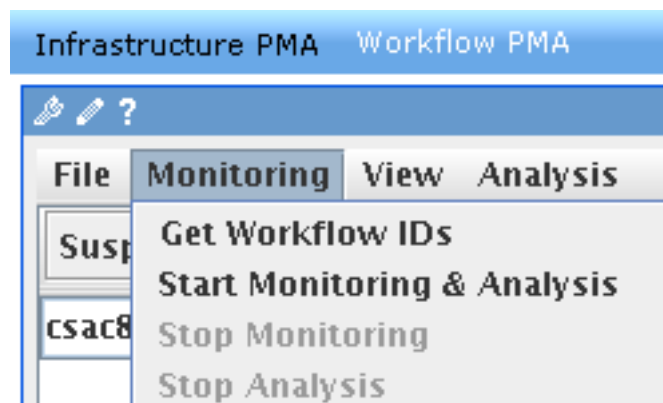


Fig. 3-22: Monitoring menu items.

The applet will show IDs of all existing workflows which can be running or already finished. The output in the log area will look like area (if you have chosen to activate the feedback window):

Adding workflow:=truong-03331c50-4537-11da-953e-e3f268ddc24d

The WfIDs are available in the initially empty (see Fig. 3-20) combo box below the menu bar.

3.4.2.3 Select a Workflow

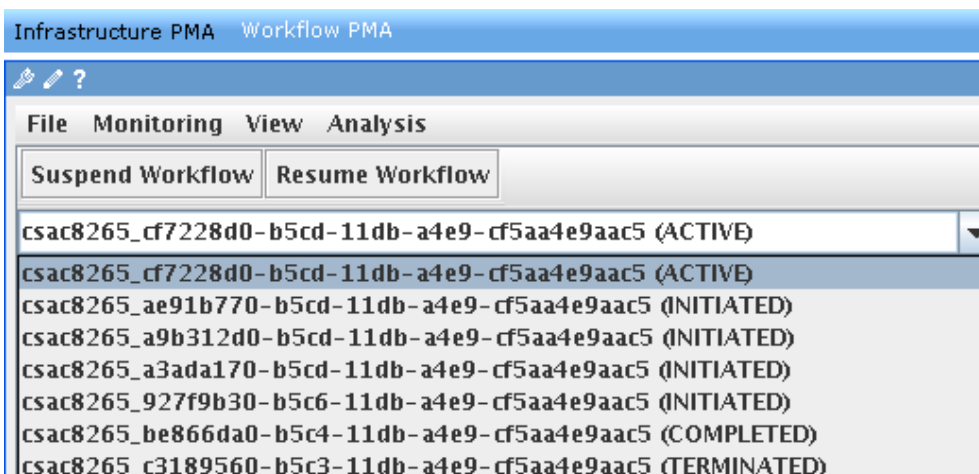


Fig. 3-23: Select an existing WfID.

The user can either select a Workflow ID from the list of available WfIDs or type an ID manually into the combo box; see Fig. 3-23. If an ID is provided manually the user must confirm his input by hitting the <RETURN> key on the keyboard. The ID will be added to the list. Please note that duplicate names and empty strings will be not added.

3.4.2.4 Start Monitoring and Analysis

After selecting a WfID the user can start the monitoring and analysis of workflows by clicking “Start monitoring and Analysis” from the Workflow menu, see Fig. 3-22.

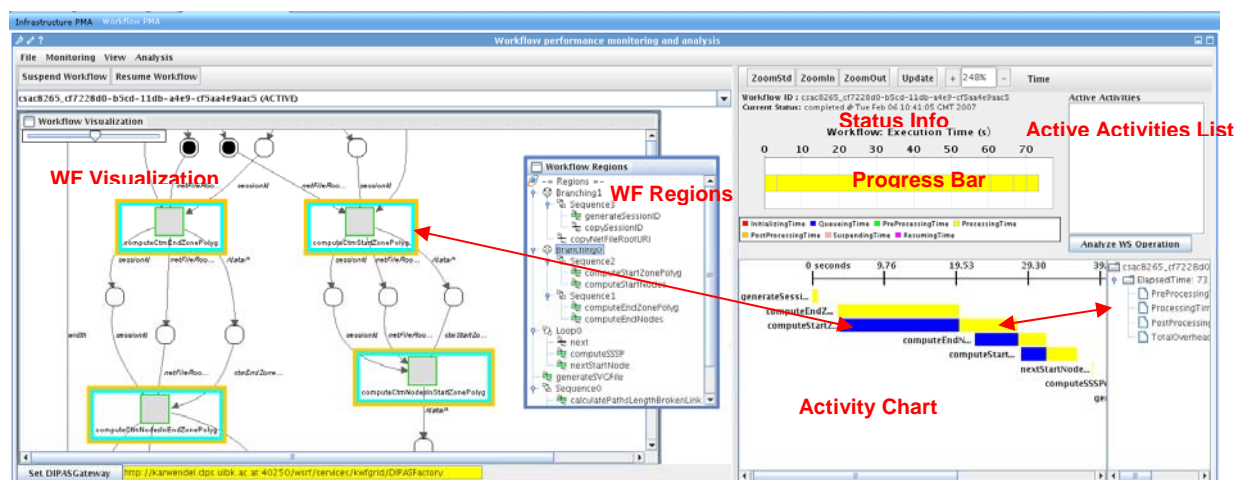


Fig. 3-24: Example of a workflow being monitored and analyzed.

The applet polls for data with the selected ID and displays it as long as the workflow is running, see Fig. 3-24 for an example of a running workflow. Please note, that the red arrows in the example and the bold red labels naming each component are not part of the GUI they have been supplied only for this example.

Below the selected WfID in the left part of the split pane there is the **WF visualization**. Each activity is surrounded by a colored border. The color corresponds to the last received status of the activity as shown in the chart in the middle (on the right side of the split pane) the color changes during the execution of the workflow according to the status of the activity.

On the right side of the Workflow Visualization window we have the **Workflow Regions** pane. This panel shows to the user the structured view of workflows that includes workflow regions like *Branching*, *Loop* or *Sequence*, making clear how activities are mapped together in the workflow.

The **Activity Chart** shows the execution status of activities during execution time. Clicking on an activity instance, information for this activity will be displayed in the window to the right of the activity chart.

Above the activity chart is the overall **Status Information** of the WF. The current status and the **Progress Bar** will be updated after each poll to DIPAS. The progress bar is divided into parts colored according to the sum of time spent during one of the lifecycles of a WF. In Fig. 3-24 the entire execution time consists of many different activities as shown in the legend below the progress bar.

To the right of the activity chart is the **Active Activities List** which shows the names and IDs of active activities.

3.4.2.5 Conduct Online PMA

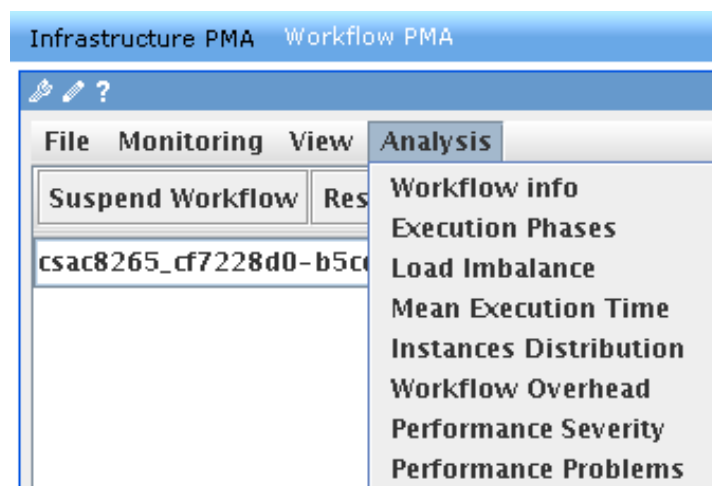


Fig. 3-25: Analysis menu items.

During WF execution and after the execution has been finished the user can conduct analysis of each activity or for the entire workflow online. When the WF execution ended it is possible to collect data about the whole workflow by sending a request to DIPAS which will send the separate requests to several distributed agents.

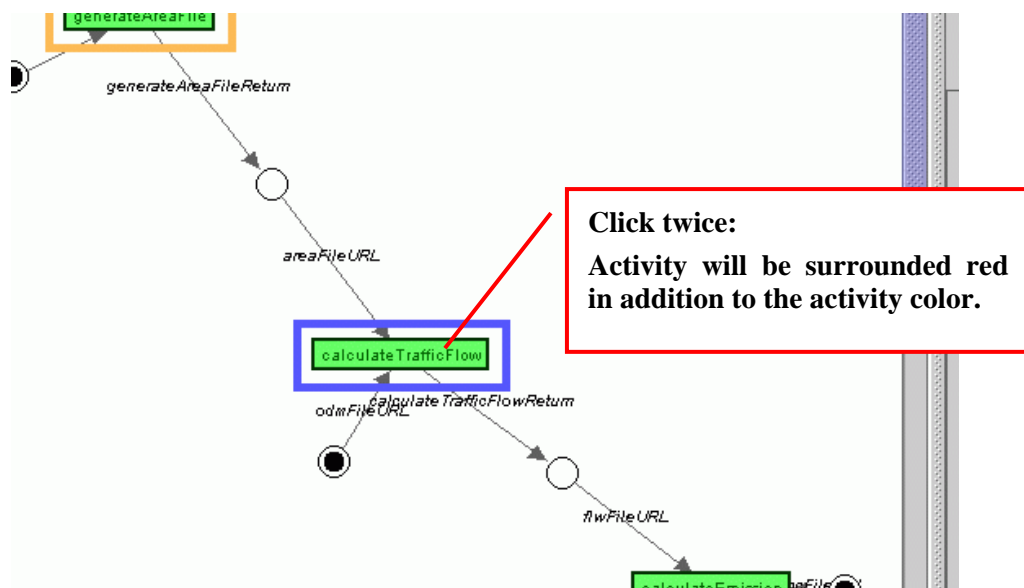


Fig. 3-26: Select an activity.

For an individual activity analysis, the user must choose the activity in the workflow graph (see Fig. 3-26) before selecting the appropriate item from Analysis menu (see Fig. 3-25).

The user can choose to get static information of all the workflows that currently are in GWES selecting the item Workflow Info. The user gets a panel showing useful information for the selected workflow as shown in Fig. 3-27.

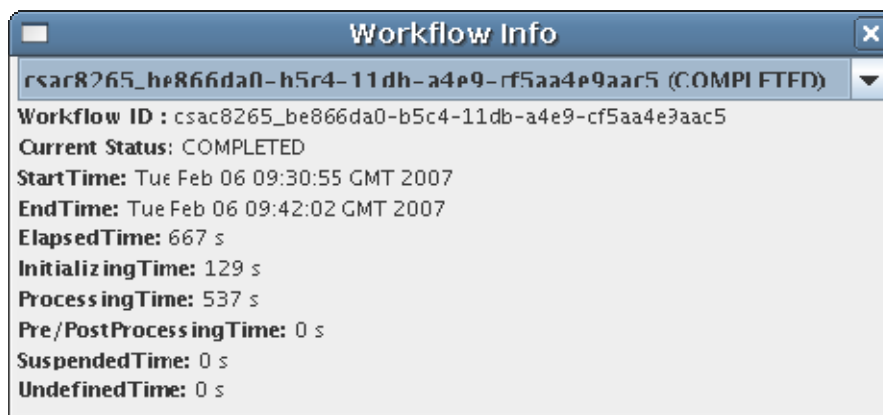


Fig. 3-27: Workflow Info.

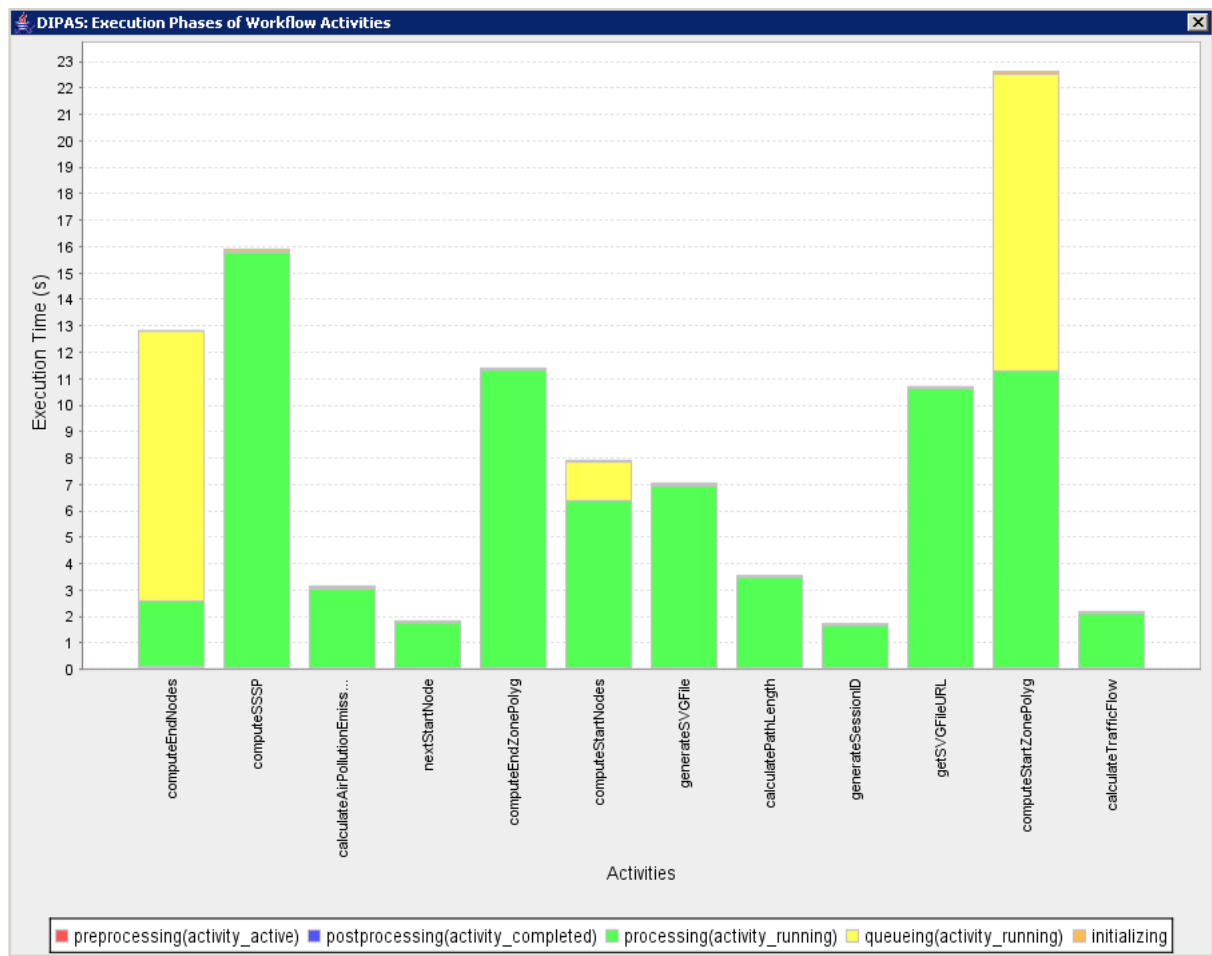


Fig. 3-28: Summary of execution phases.

Choosing Execution Phases shows the execution phases of all activities for the entire workflow. Each bar has many sections; each section indicates time spent in an execution phase of an activity, as shown in Fig. 3-28.

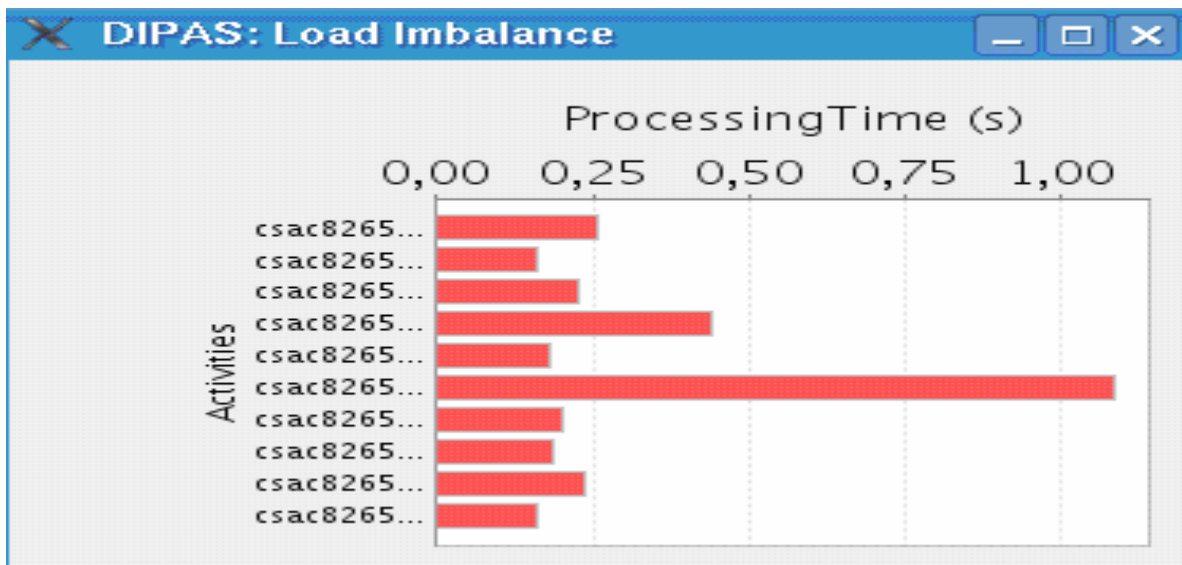


Fig. 3-29: Example of load imbalance.

Choosing Load Imbalance shows a comparison of processing time of several activities (see Fig. 3-29).

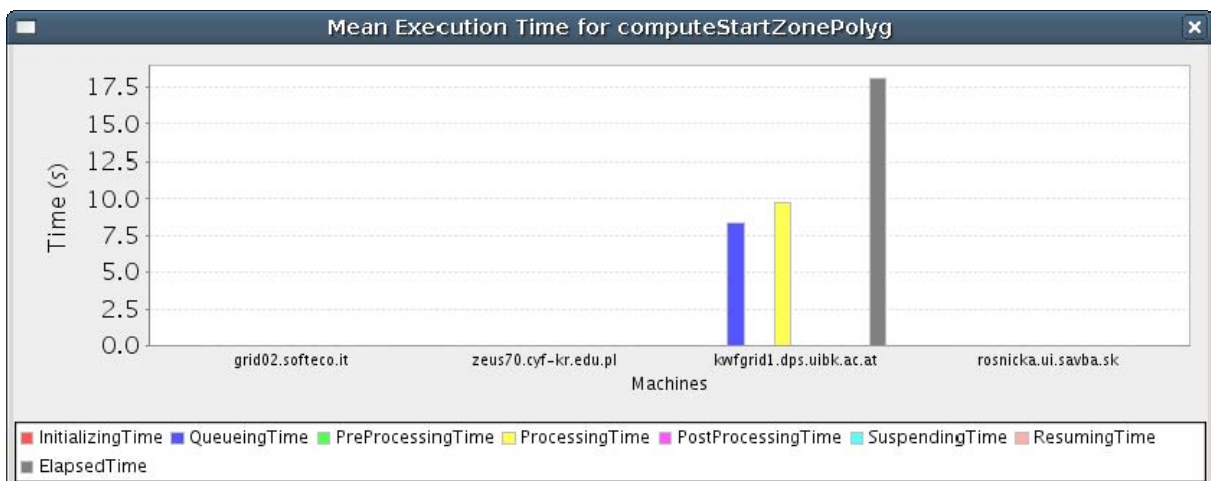


Fig. 3-30: Mean Execution Time example.

For a selected activity, menu item Mean Execution Time shows how much time spent in Grid sites with respect to execution phases (see Fig. 3-30).

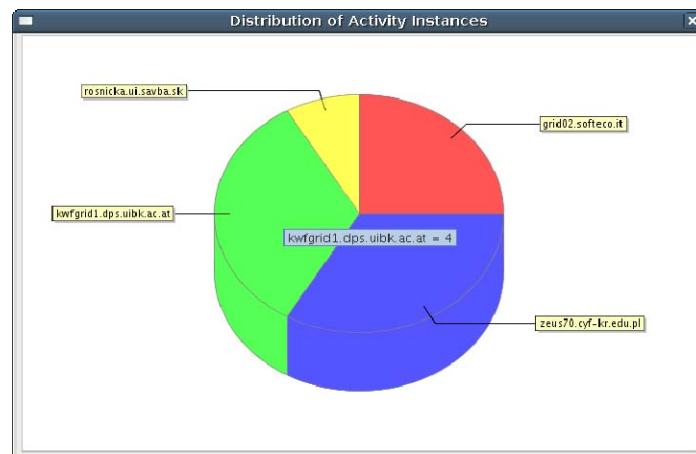


Fig. 3-31: Example of the distribution of activity instances.

For a selected activity, menu item Instances Distribution displays a pie diagram showing the number of instances of the selected activity per Grid site, as shown in Fig. 3-31.

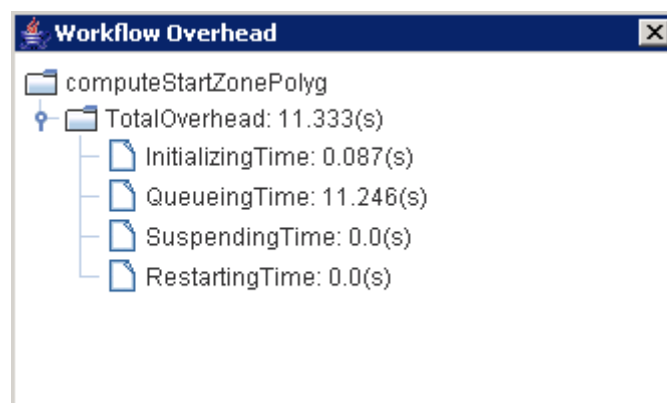


Fig. 3-32: Workflow overhead.

For a selected activity menu item Workflow Overhead displays a tree with overhead times, broken into subcategories, as shown in Fig. 3-32.

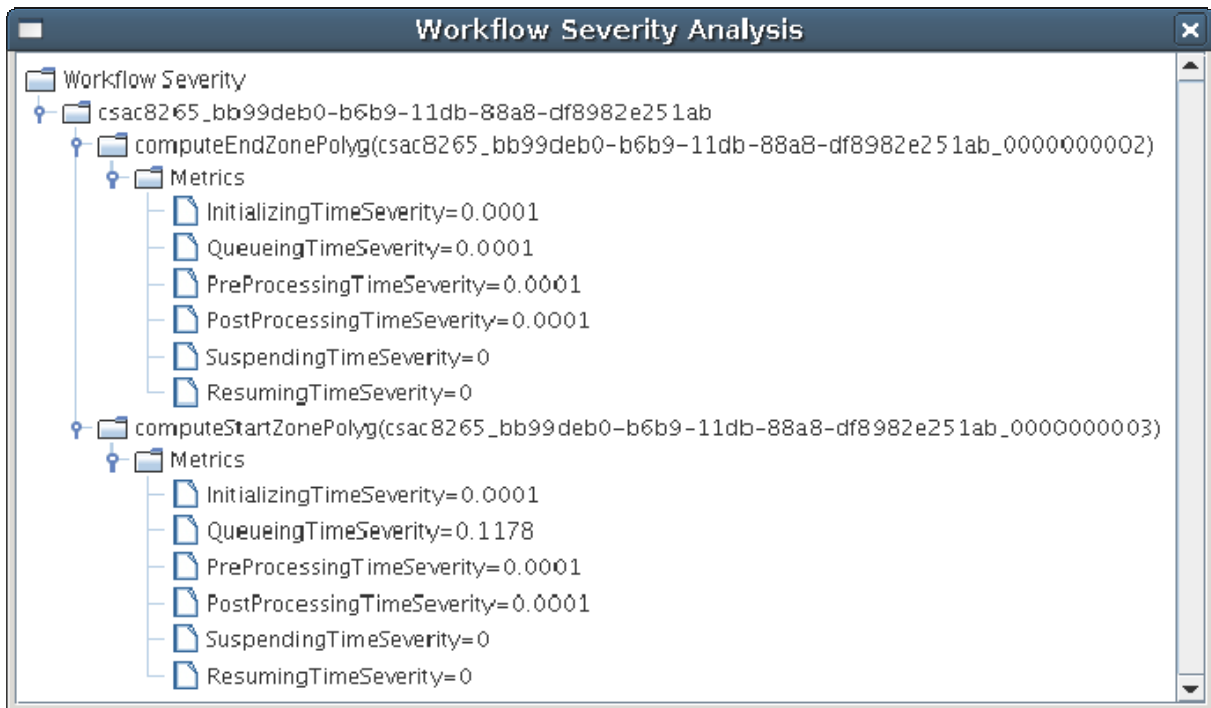


Fig. 3-33: Severity example.

For a selected activity menu item Performance Severity shows computed time severity metrics of the selected activity in a tree, broken into subcategories (see Fig. 3-33). These metrics indicate how severe the impact of a certain waiting time is on the workflow.

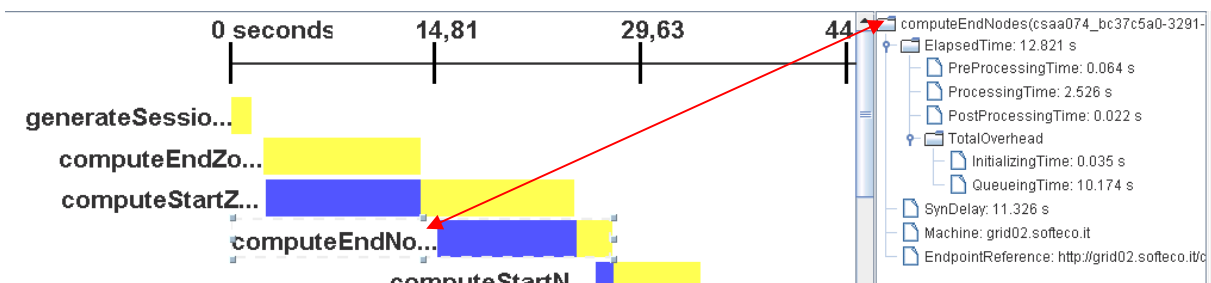


Fig. 3-34: Activity details.

Selecting an activity in the activity chart (see Fig. 3-24 and Fig. 3-34) displays detailed activity information for this activity on the right side of the activity chart.

The user can also select an activity or a region that he/she wants to analyze through the Workflow Regions panel you can see in Fig. 3-35.

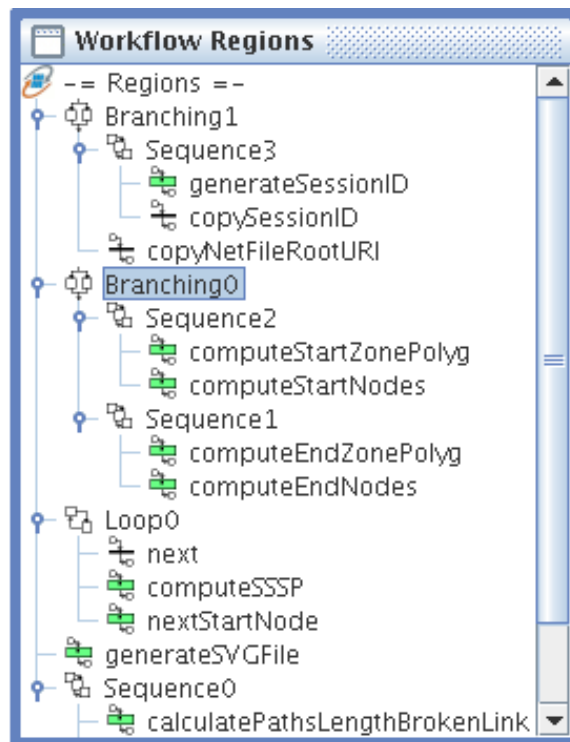


Fig. 3-35: Workflow Regions.

In the Workflow Regions window, the activity or region that the user selects will be marked. If the user wants to analyze performance problems of any specific activity, region, or the whole workflow, he has to select the menu item Performance Problems. A panel will appear (see Fig. 3-36) where the user has to specify the performance problems which could occur during the run of the workflow.

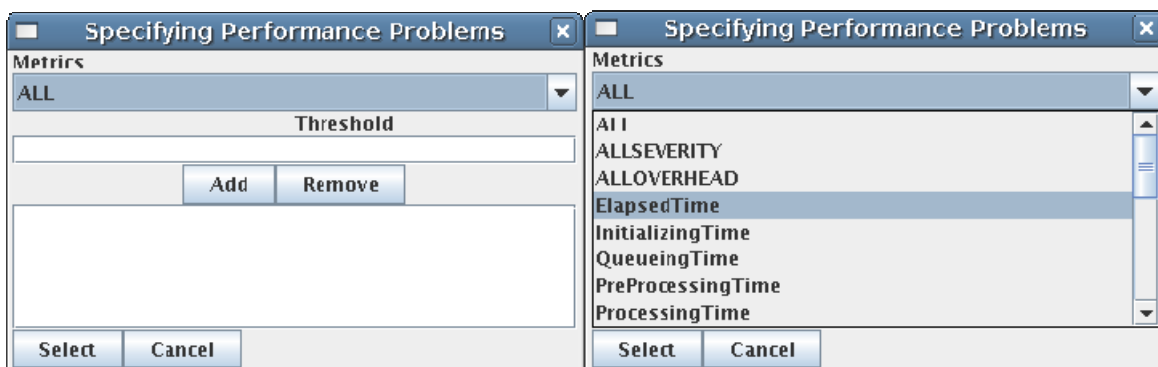


Fig. 3-36: Specifying Performance Problems.

The user can choose various different metrics in different forms such as time, measured in seconds, and timestamp. For example he can specify to be informed if the selected activity has an ElapsedTime of more than 10 seconds; or if the user wants to know if the StartTime of the workflow was after a specific timestamp, he can add this performance problem selecting StartTime from the Metrics box, select the date and click on Add. At the end the user clicks on the Select button.

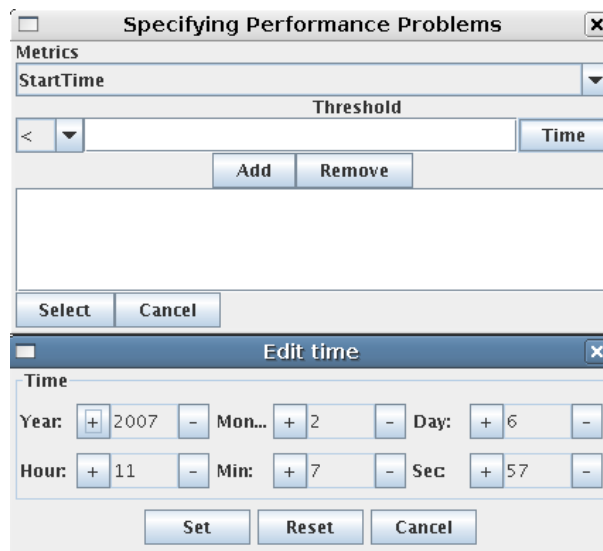


Fig. 3-37: Specifying date for Performance Problems.

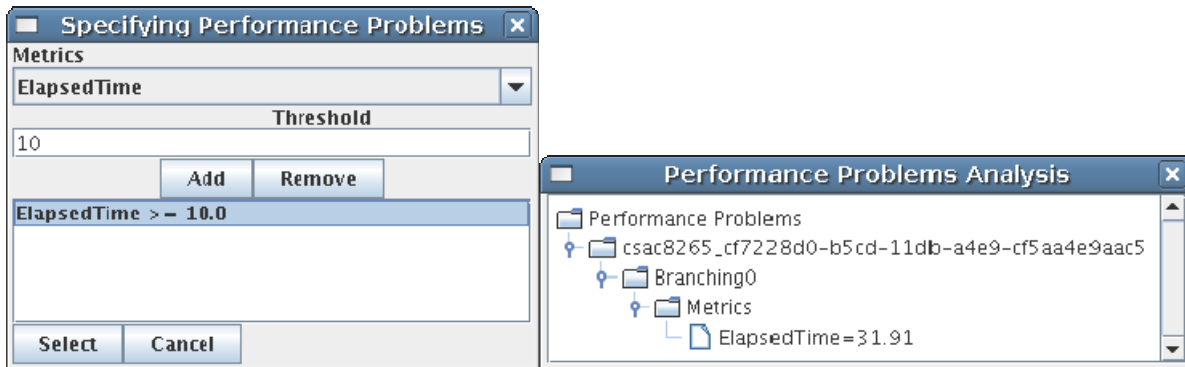


Fig. 3-38: Result for Performance Problems.

If there was a problem met the specified performance condition, the user will be immediately informed of this problem in the Performance Problems Analysis window (see Fig. 3-38).

4 INTERFACE REFERENCE GUIDE

4.1 PORTAL LOGIN

The screenshot shows the GridSphere portal login interface. At the top, there is a navigation bar with the GridSphere logo and the text 'gridsphere portal framework' and 'open-source / portlet jsr168 compliant'. A language dropdown menu is set to 'English'. Below the navigation bar, there is a 'Home' link. The main content area is titled 'Welcome to GridSphere 2.0.4' and contains a message: 'Thank you for downloading and installing the GridSphere portal. Below you will find links to documents relating to GridSphere installation, administration and portlet development.' This is followed by a list of links: 'GridSphere User's Guide (HTML) (PDF)', 'GridSphere Portal Administrator's Guide (HTML) (PDF)', 'GridSphere Portlet Reference Guide (HTML) (PDF)', 'GridSphere Tag Library User's Guide (HTML) (PDF)', 'GridSphere Frequently Asked Questions (HTML) (PDF)', and 'GridSphere JavaDoc API (HTML)'. Below this, there is a section for mailing lists: 'Please join the mailing lists for more involvement:' followed by 'Developers List', 'Users List', and 'CVS List'. At the bottom, there is a link to 'GridSphere (Jira) bugtracker'. On the right side, there is a 'Login' form with fields for 'User Name' and 'Password', a 'Remember my login' checkbox, a 'Login' button, and a 'Forgot your password?' link. In the bottom right corner, there is a 'powered by gridsphere' logo.

Fig. 4-1: Login portal.

The Gridsphere portal framework login window (see Fig. 4-1) provides the following functionalities:

- Upper right Combobox: changes language settings
- User Name Textfield: inputs login name
- Password Textfield: inputs password
- Checkbox Remember my login: remembers data for later logins
- Button Login: starts login procedure with your data

4.2 INFRASTRUCTURE PMA

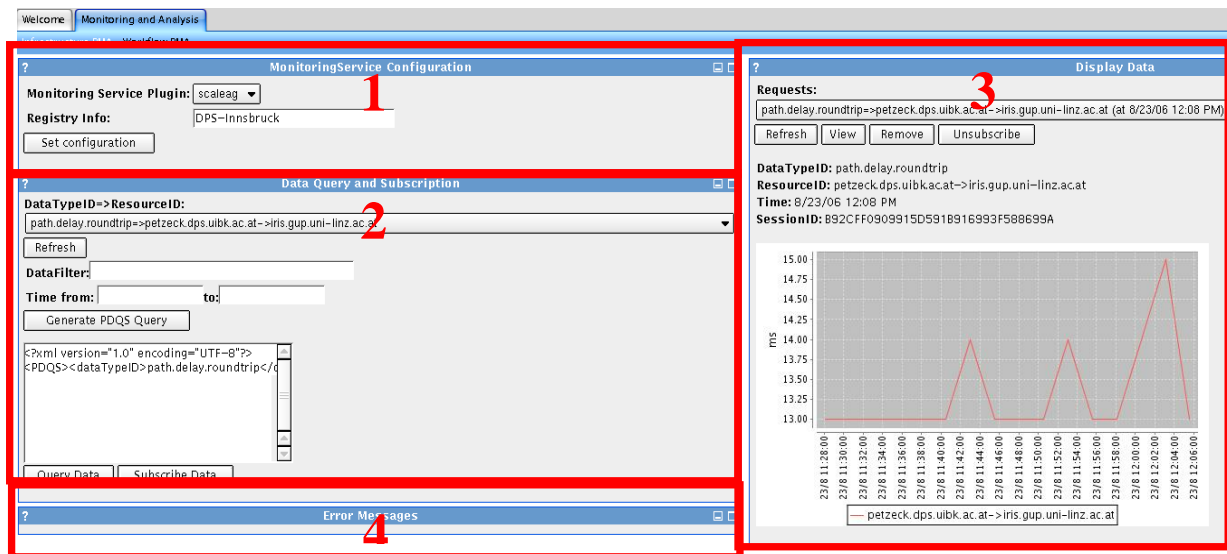


Fig. 4-2: Infrastructure PMA Details

The Infrastructure PMA tab in Monitoring and Analysis has four main components:

- Monitoring Service Configuration (see Fig. 3-10 and Fig. 4-2).
- Data Query and Subscription (see Fig. 3-13 and Fig. 4-2).
- Display Data (see Fig. 3-17 and Fig. 4-2).
- Error Messages (see Fig. 4-2).

4.2.1 Configuration

The Monitoring Service Configuration (see Fig. 4-2) component in Monitoring and Analysis provides the following functionalities:

- **ComboBox** Monitoring Service Plugin: selects monitoring plug-in, e.g. GEMINI or SCALEA-G (see Fig. 3-10)
- **TextField** Registry Info: specifies information about registry service (see Fig. 3-12).
- **Button** Set Configuration: configures the portal with input data provided in the above steps (selection in the combo box and value in the textfield).
- **Icon ?**: displays help information (see Fig. 3-8 and Fig. 3-9).

4.2.2 PDQS

The Data Query and Subscription (see Fig. 3-13 and Fig. 4-2) component allows the user to generate a PDQS request [DRDEV],[DRUSER], query and subscribe data. **ComboBox** Data TypeID=>ResourceID: selects data types associated with monitored resources. Depending on the monitoring plug-in and configuration set different selections like bandwidth capacity or host memory used are available.

It provides the following functionality:

- **Textfield** Data Filter: types in the data filter to be applied to monitoring data

- *Textfield* Time from: specifies start time of the subscription time
- *Textfield* to: specifies end time of subscription time
- *Button* Generate PDQS Query: generates a PDQS request.
- *TextArea* below Generate PDQS Query: the generated PDQS XML request will be displayed here after pressing the Generate PDQS Query Button
- *Button* Query Data: queries monitoring data with the given XML request
- *Button* Subscribe Data: subscribes monitoring data with the given XML request
- *Icon* ?: displays help information (see Fig. 3-8 and Fig. 3-9).

4.2.3 Display Data

In the *Display Data* (see Fig. 3-17 and Fig. 4-2) component results of the PDQS queries are being visualized. This component can be divided into a lower and an upper part (Fig. 4-3 above and below the separating red line respectively). The upper part provides an interface for actions, the lower for viewing data and meta information about the data.

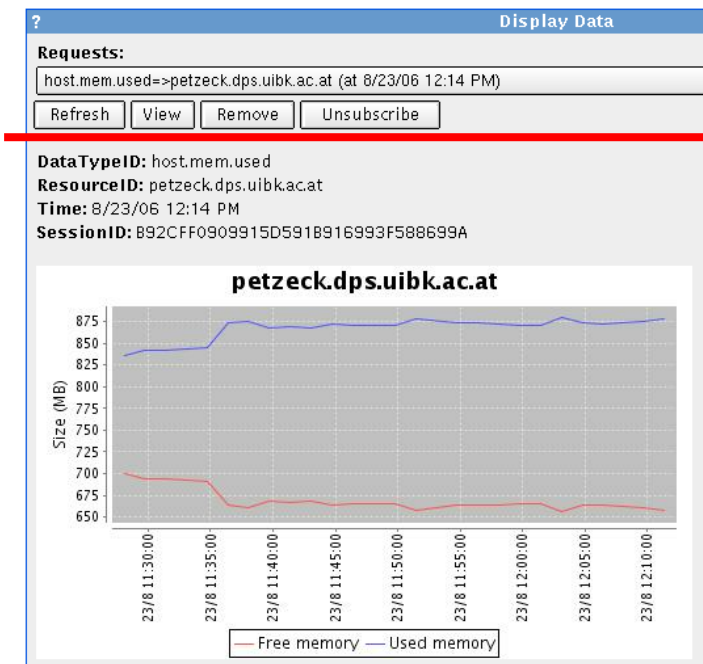


Fig. 4-3: Data Display.

The upper part (in Fig. 4-3 above the red line) provides the following functionalities:

- *ComboBox*: (see Fig. 4-3 below label **Requests:** and Fig. 3-14) selects one of the retrieved results, may be empty or non-existent if there are no results for display.
- *Button* Refresh: updates the combo box with results.
- *Button* View: Displays visualization in the lower part; in case the data has not been transmitted entirely pressing this button will reload the visualization.
- *Button* Remove: removes results from the combo box.

- *Button Unsubscribe*: if the data was not queried but subscribed, this button cancels the subscription.
- *Icon ?*: displays help information (see Fig. 3-8 and Fig. 3-9).

The lower part (in Fig. 4-3 below the red line) provides the following functionalities:

- *Label Data Type ID*: shows the data type used to generate the result visualization.
- *Label Resource ID*: the resource ID of the data.
- *Label Time*: the time when the result has been received in the portal.
- *Label Session ID*: the identifier of the displayed result.
- *Data Visualization*: either a graphic (see Fig. 3-18) or a table (see Fig. 3-19) expressing the selected result.

4.2.4 Error

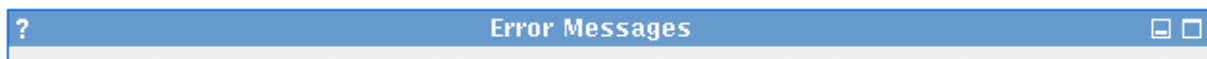


Fig. 4-4: View error messages.

The Error Messages component (see Fig. 4-4 and Fig. 4-2) displays information about the last error that occurred.

4.3 WORKFLOW PMA

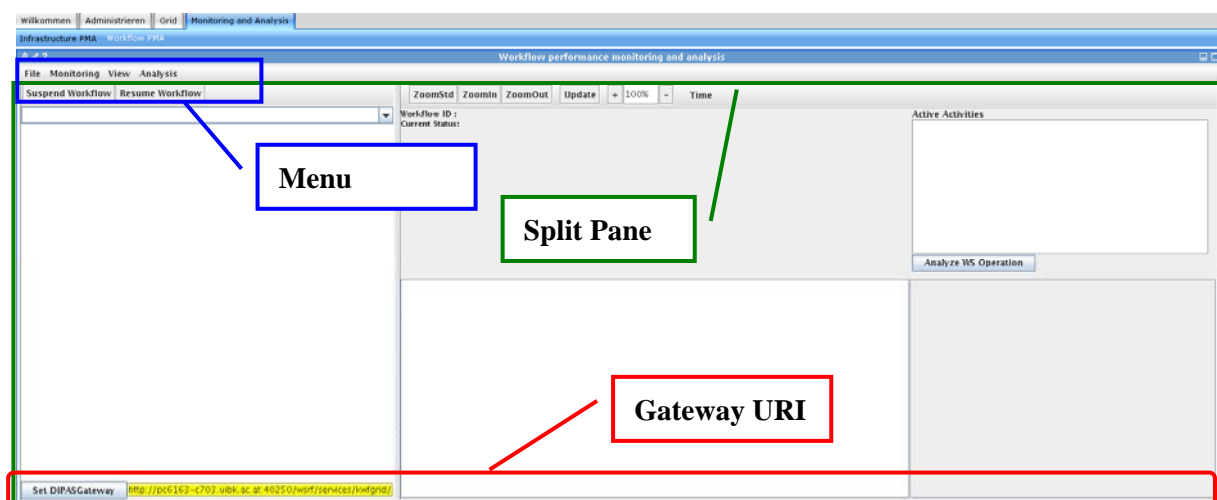


Fig. 4-5: Workflow PMA components.

The Workflow Performance Monitoring and Analysis (WFPMA) tab (see Fig. 4-5) loads an applet that allows visualizing, monitoring and analyzing workflows. It consists of the following components (including further elements):

- *Menu bar*: contains menus that provide further functionality
 - *Menu File*
 - *Menu Monitoring*
 - *Menu View*
 - *Menu Analysis*
- A *split pane* graphical display area: contains elements that will be active as soon as workflow monitoring has been started, the elements in the split pane provide further functionality, they are explained further below. The split pane allows arranging the distribution of space between WF visualization and activities.
- *Text field Gateway URI* with URL of DIPAS Gateway: editable, shows the current DIPAS gateway (see Fig. 3-21).

4.3.1 Menu File

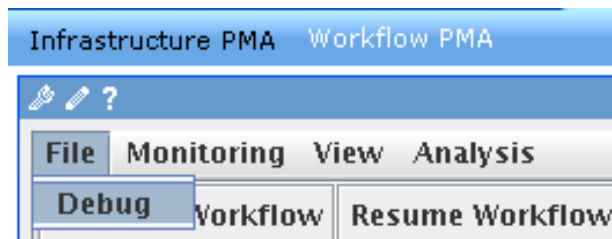


Fig. 4-6: File menu.

Menu File (see Fig. 4-6) includes the following menu items and functionalities:

- *Item Debug*: launches a log window with feedback.

4.3.2 Menu Monitoring

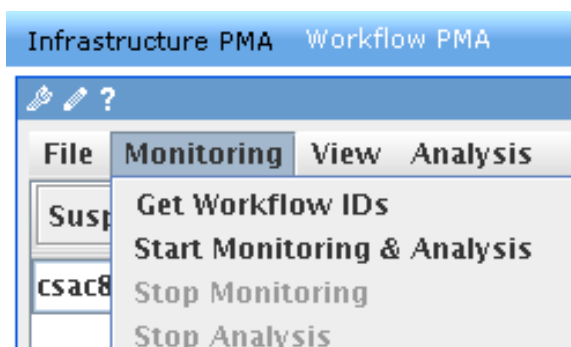


Fig. 4-7: Monitoring menu.

Menu Monitoring (see Fig. 4-7) includes the following menu items and functionalities:

- *Item* Get Workflow ID: get current WF IDs from DIPAS and put them into the ID list (see Fig. 3-23)
- *Item* Start monitoring and analysis: visualize the WF with the WfID selected from the list of WfIDs.
- *Item* Stop monitoring: stop the visualization.
- *Item* Stop analysis: stop the analysis.

4.3.3 Menu View

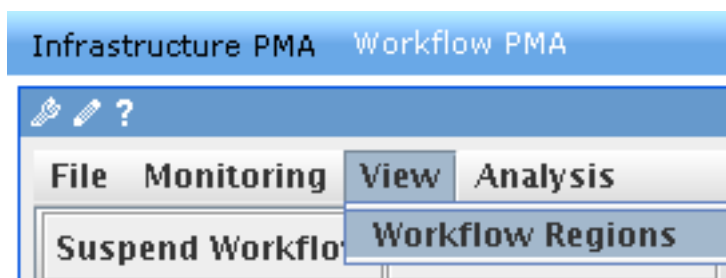


Fig. 4-8: View menu.

Menu View (see Fig. 4-9) includes the following menu items and functionality:

- *Item* Workflow Regions: displays the Workflow Regions window (see Fig. 3-35).

4.3.4 Menu Analysis

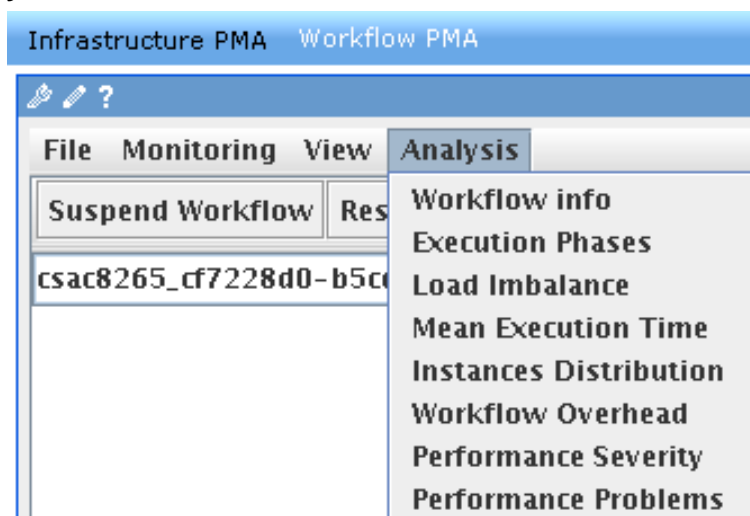


Fig. 4-9: Analysis menu.

Menu Analysis (see Fig. 4-9) includes the following menu items and functionality:

- *Item Workflow Info*: gets and visualizes static workflow information.
- *Item Execution Phases*: gets and visualizes state execution time data in a popup window.
- *Item Load Imbalance*: visualizes the load balance in a popup window.
- *Item Mean Execution Time*: shows execution time split into categories in a popup window.
- *Item Instances Distribution*: shows the distribution of an activity on several machines.
- *Item Workflow Overhead*: shows a selection of metrics in a tree for overhead computation.
- *Item Performance Severity*: shows performance severity of selected activities/instances
- *Item Performance Problems*: specifies constraints for performance problems

4.3.5 Split Pane

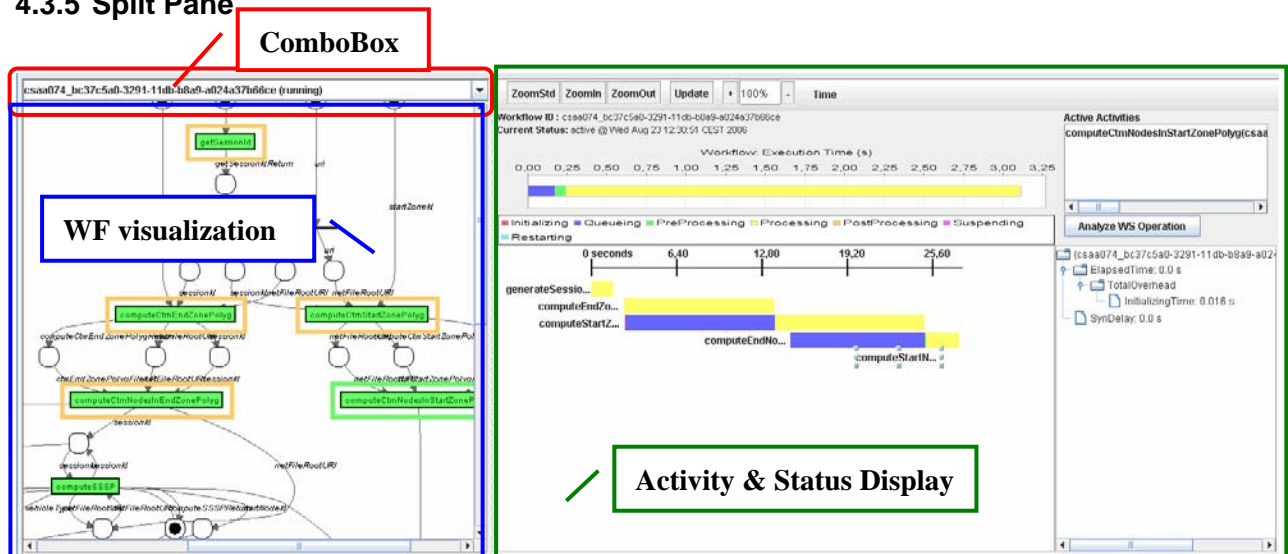


Fig. 4-10: Split pane overview.

The split pane contains the following components (see Fig. 4-10) and functionality:

- *WfID ComboBox*: lists all received WfIDs, allows selection and manual input of an ID (see Fig. 3-23).
 - Clicking on the arrow down symbol: shows a scrollable list of all retrieved and typed WfIDs.
 - Scrolling through the list and selecting an entry: selects the WfID that will be used upon the next start and monitoring of a WF.

- Typing characters in the white area: edits an existing name or creates a new WfID, the user can add this name to the list by hitting <RETURN>, duplicate names and empty strings will not be added to the list
- *Scroll Pane* WF visualization: shows a dynamic graphic of the monitored WF and has the following functionalities:
 - Scrolling: shows another part of the WF, large WFs will be impossible to capture entirely on the screen.
 - Resizing: A slider in the left upper corner of the WF allows resizing, an example is given further below.
 - Dragging WF objects: rearranges the items in the visualization.
 - Activities of the WF are surrounded by a color corresponding to the status of the activity (an example is given further below).
 - Clicking on an activity: twice clicking on an activity selects that activity, deselecting the previous. Selected activities can be further monitored, see Fig. 3-26.
- *Activity & Status Display*: lists all active activities, shows a chart with time and status of activities as well as the progress bar and the current status display.
- *Dragging the separator*: rearranges the sizes of the left and right sides in that the pane is split

4.3.5.1 WF visualization

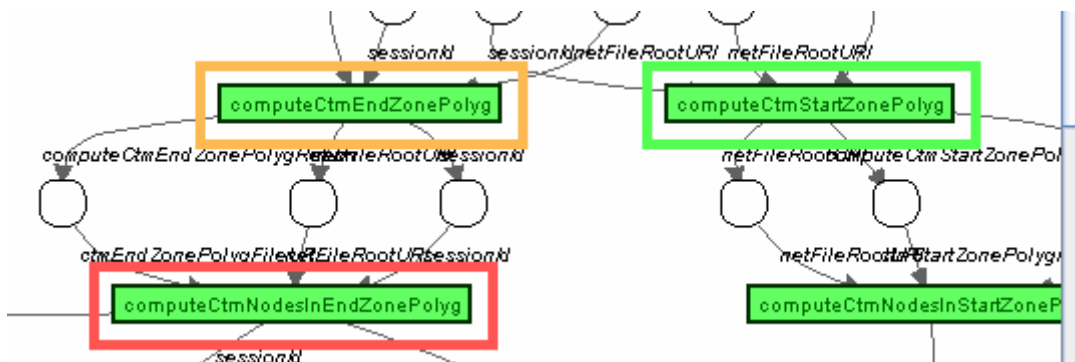


Fig. 4-11: Activity status example.

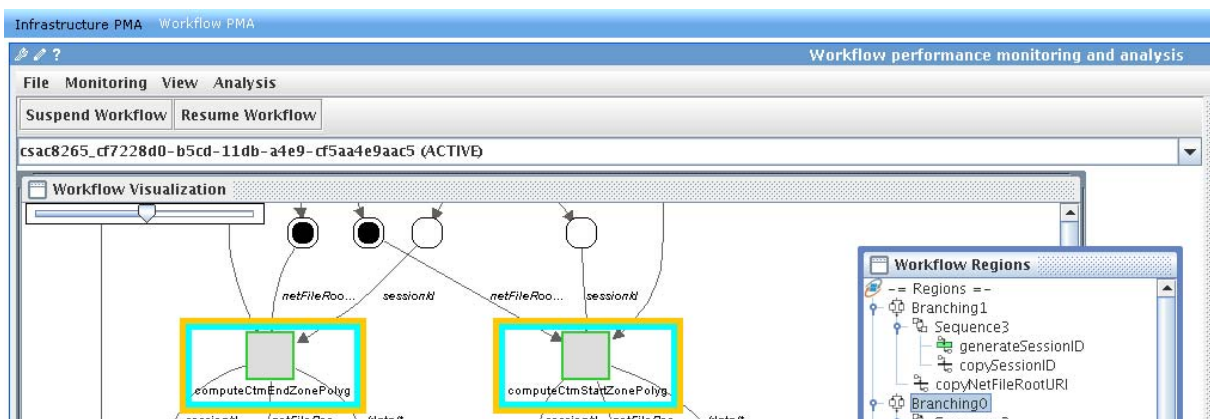


Fig. 4-12: Workflow with slider.

The WF visualization scroll pane (see Fig. 4-10 and Fig. 3-24) shows a visualization of the WF that is being monitored. The main functionalities are listed in the section about the *split pane* above; two of them are shown in more detail here:

- Resizing: The WF can be resized using a slider in the upper left corner of the WF, as shown in Fig. 4-12.
- Fig. 4-11 shows four activities with different status: the activity surrounded by a brown border (`computeCTMEndZonePolyg`) is finished, the activity surrounded by a red border (`computeCtmNodesInEndZonePolyg`) is being pre processed, the activity surrounded by the green border (`computeCtmStartZonePolyg`) is being executed and the activity that is not surrounded (`computeCtmNodesInStartPolyg`) has not been started yet.

4.3.5.2 Status & Activity Display

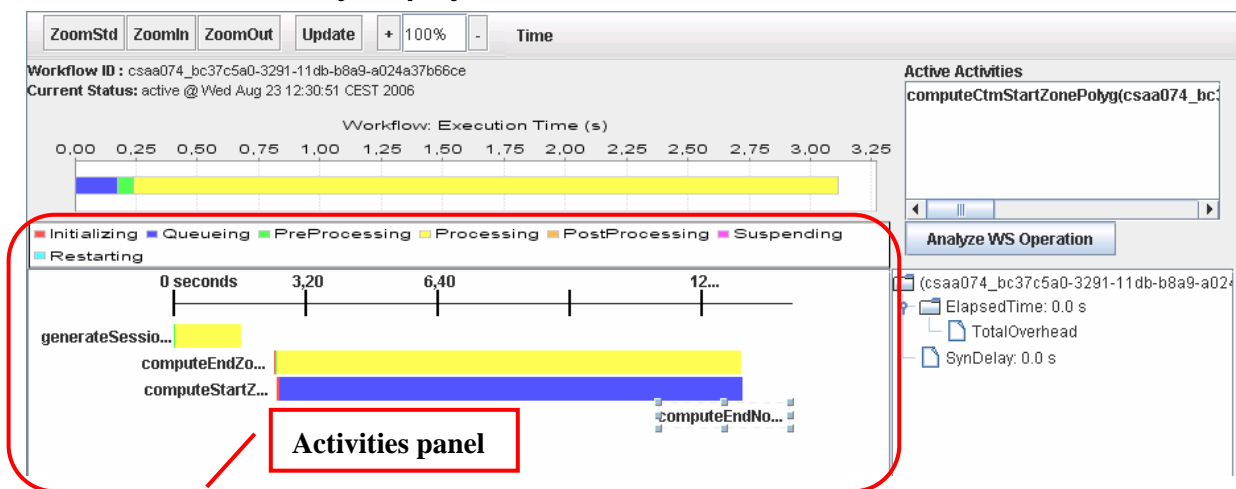


Fig. 4-13: Status and activity display.

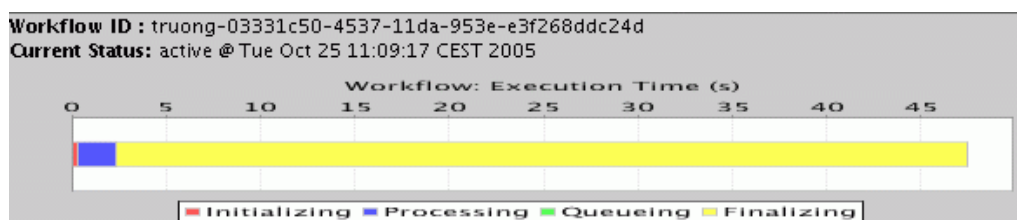


Fig. 4-14: Progress bar and current status.

The display (see Fig. 4-13) contains several elements with the following functionalities:

- A *progress bar* (Fig. 4-14) that is being updated periodically based on data polled from DIPAS.
 - Right Click on the Progress Bar: allows individual configuration of the visualization
- A *status area* above the *progress bar* (Fig. 4-14) the ID of the currently monitored WF and its current overall status followed by the time and date of last update.
- *Tool bar* above the *status area*:

- ZoomStd: sets the zoom in the activities panel to 100%
- ZoomIn: enlarges items in the activities panel by the factor 1.2.
- ZoomOut: shrinks items in the activities panel by the factor 1.2.
- Update: Redraws the activities in the activities panel.
- +, -, % *field*: changes the time scale in the activities panel.
- *Panel Activities* (marked in Fig. 4-14): All activities of the workflow are shown here with a time scale above. The color matches the color given in the legend above the scale.
 - Clicking on an activity displays information to the right of the activities panel.
- *Button Analyze WS Operation*: action not implemented yet.
- *List Active Activities*: Activities that are active within the WF are shown here.

4.3.6 Popup GUI Elements

Some of the GUI Elements are not always visible. They pop up when some events occur or when the user requests them.

4.3.6.1 Workflow Execution Finished Notification

The message: “The workflow execution has been finished!” (see Fig. 4-15) pops up automatically to notify the user that the execution of the currently monitored WF ended. This is useful since the user might wonder why no more updates in the GUI can be noticed. Also after the execution the workflow overhead can be computed. There is no further functionality to this notification window except closing.

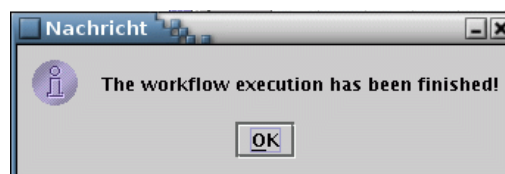


Fig. 4-15: WF Finished notification.

4.3.6.2 Debug Window



Fig. 4-16: Debug window.

The logs window (see Fig. 4-16) provides some additional information to the user. It gives insight into the internal mechanism behind the GUI displaying incoming XML data and showing errors. The log is not editable.

4.3.6.3 Workflow Info

See Fig. 3-27. Selecting the workflow ID in the Combobox will change the information of the selected workflow.

4.3.6.4 Execution Phases of Workflow Activities

See Fig. 3-28. Right clicking on the graphic allows individual configuration of the visualization.

4.3.6.5 Load Imbalance

See Fig. 3-29. Right clicking on the graphic allows individual configuration of the visualization.

4.3.6.6 Mean Execution Time

See Fig. 3-30. Right clicking on the graphic allows individual configuration of the visualization.

4.3.6.7 Distribution of Activity Instances

See Fig. 3-31. Right clicking on the graphic allows individual configuration of the visualization.

4.3.6.8 Workflow overhead

See Fig. 3-32. Clicking on the tree nodes breaks up metrics showing or hiding details.

4.3.6.9 Performance Severity

See Fig. 3-33 and **Error! Reference source not found.** Clicking on the tree nodes breaks up metrics showing or hiding details.

4.3.6.10 Performance Problems

See Fig. 3-36, Fig. 3-37 and Fig. 3-38. We can select different Performance Problems which will be analyzed during runtime. The user will be informed if a performance problem occurs.

5 TROUBLESHOOTING Q&A

Q: Why does the applet use so much memory?

A: The current prototype version has been developed under the aspects of good visualization and functionality. Both aspects cannot be satisfied without extensively using the hardware resources. The memory usage must be seen in the right context. The portal is not meant as a background job running automatically but as a powerful tool that expects a lot of user interaction. To make sure that there are no real usage problems it has been tested on machines that are not state of the art but very common standard computers. We are however always looking for ways to optimize the performance and any improvements will be implemented in future releases.

Q: Why does the applet use so much disk space?

A: When developing the applet we took the decision to use stable and proved libraries in order to achieve good visualization results. Applets that need those libraries must download them first. The advantage for the user is that there is no software installation required and that using the portal is platform independent. Compared to other locally installed software the amount of disk space used (about 27 MB) is reasonable. As with memory usage we are looking for ways to optimize the download.

Q: The applet is running, why do I get no data?

A: First make sure that the DIPAS factory URL is set correctly then set server URI in the Workflow menu (check the setup section, how to configure the DIPAS gateway). If there is still no data coming, check if the Globus container with the portlets and services is running. Check if the services providing the data are running.

Q: The portal seems to be working, why is there no data from GEMINI or SCALEA-G?

A: There might be a problem with security certificates on the site where you access the DIPAS portal. Try to use an alternative address and ask the organization providing the portal to check the certificates.

6 CONTACT INFORMATION AND CREDITS

DIPAS framework is developed at:

Distributed and Parallel Systems Group,
Institute of Computer Science, University of Innsbruck
Technikerstrasse 21A, A-6020, Innsbruck, Austria

The developers include:

- Peter Brunner
- Robert Samborski
- Vlad Nae
- Hong-Linh Truong (team leader)

With contributions from:

- Francesco Nerieri, Simon Ostermann

Visit <http://www.dps.uibk.ac.at/projects/pma> for further information about our work on performance monitoring and analysis. Further comments, suggestions, and bug reporting, contact Hong-Linh Truong at truong@dps.uibk.ac.at.

7 THE GPL LICENSE AGREEMENT

Copyright (c) 2004-2006 Distributed and Parallel Systems Group, University of Innsbruck. All rights reserved.

This software includes voluntary contributions made to K-WfGrid. For more information on K-WfGrid, please see <http://www.kwfgrid.eu>.

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

8 REFERENCES

- [APPLET] Sun Developer Network, <http://java.sun.com/applets/>
- [CASTOR] <http://www.castor.org/>
- [DIPASDEV] K-WfGrid Consortium, "K-WfGrid Performance Analysis Service - Developer Manual", August, 2006.
- [DRDEV] K-WfGrid Consortium, "K-WfGrid Performance Service Interfaces and Data Representation -Development Manual", August, 2006
- [DRUSER] K-WfGrid Consortium, "K-WfGrid Performance Service Interfaces and Data Representation - User Manual", August, 2005
- [JFREECHART] JFreeChart, <http://www.jfree.org/jfreechart/>.
- [GLOBUS] Globus Toolkit. <http://www-unix.globus.org/toolkit/>
- [GRIDSPHERE] The Gridsphere Portal Framework.
<http://www.gridisphere.org/gridsphere/gridsphere>
- [JAVA PLUG-IN] Sun Developer Network,
<http://java.sun.com/j2se/1.4.2/docs/guide/plugin/index.html>
- [JFREECHART] JfreeChart, <http://www.jfree.org/jfreechart/>
- [JSR168] JSR 168: Portlet Specification. <http://www.jcp.org/en/jsr/detail?id=168>
- [JUNIT] <http://www.junit.org/>
- [SCALEA-G] Hong-Linh Truong and Thomas Fahringer. "SCALEA-G: a Unified Monitoring and Performance Analysis System for the Grid", Scientific Programming, 12(4):225-237, IOS Press, 2004.
- [WSRF] Ian Foster et al., "Modeling Stateful Resources with Web Services", Globus Alliance, Argonne National Laboratory, IBM, USC ISI, Hewlett-Packard, Jan, 2004. <http://www.globus.org/wsrif/ModelingState.pdf>.
- [TOMCAT] The Apache Software Foundation, <http://tomcat.apache.org/>
- [XERCES] <http://xerces.apache.org/>
- [XMLAPACHE] <http://xml.apache.org/>