Performance Analysis of Grid workflows in K-WfGrid and ASKALON

Hong-Linh Truong and Thomas Fahringer

Distributed and Parallel Systems Group

Institute of Computer Science, University of Innsbruck

{truong,tf}@dps.uibk.ac.at

http://dps.uibk.ac.at/projects/pma

With contributions from:

Bartosz Balis, Marian Bubak, Peter Brunner, Schahram Dustdar, Jakub Dziwisz, Andreas Hoheisel, Tilman Linden, Radu Prodan, Francesco Nerieri, Kuba Rozkwitalski, Robert Samborski

Outline

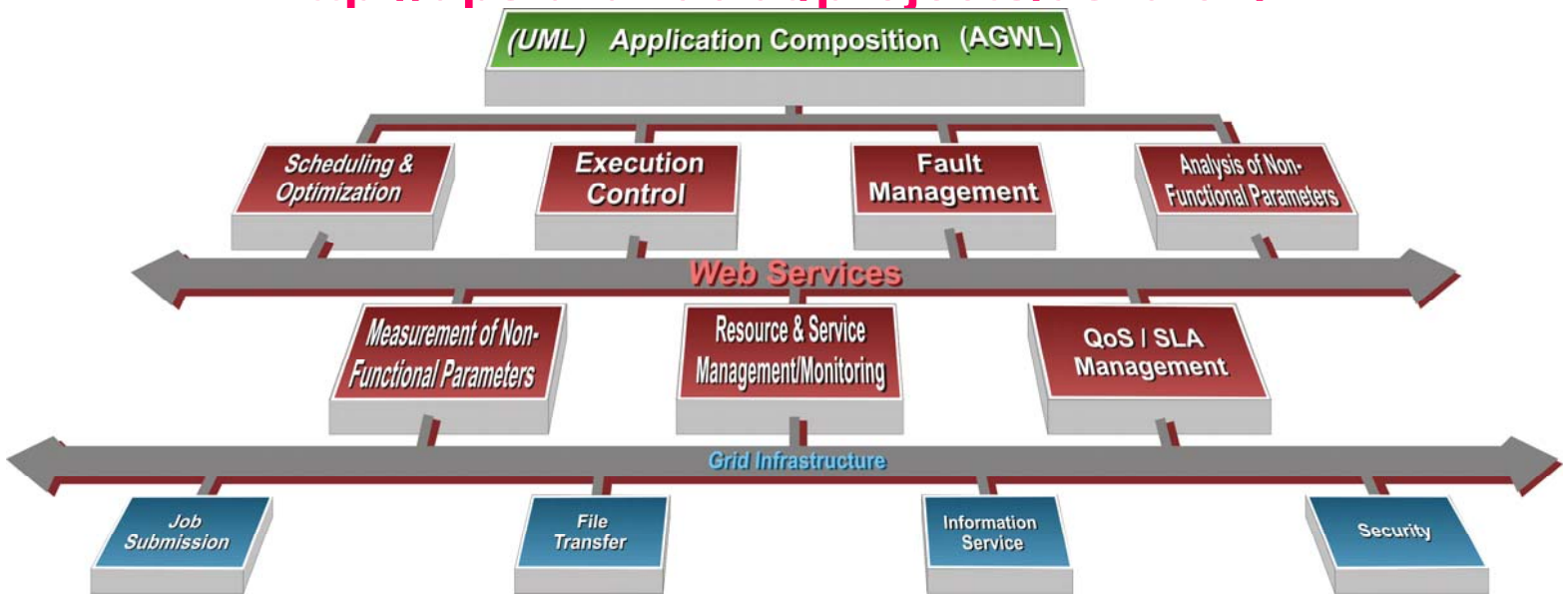
- Motivation
- Objectives and approach
- Performance metrics and ontologies
- Architecture of Grid monitoring and analysis
- Conclusions

Performance Instrumentation, Monitoring, and Analysis for the Grid

- Challenging task because of dynamic nature of the Grid and its applications
  - Combination of fine-grained and coarse-grained models
  - Multilingual applications
  - Heterogeneous and dynamic systems
- Not just performance but also dependability
- The focus of the talk
  - Our concepts, architecture, interfaces and integration

ASKALON Toolkit

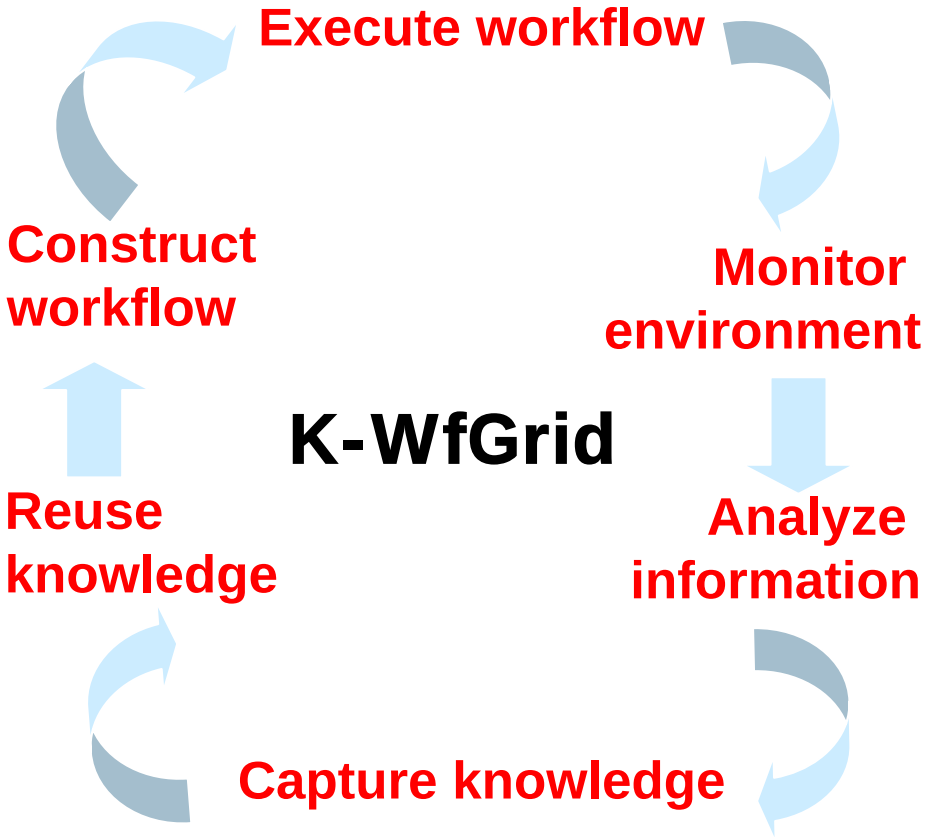
http://dps.uibk.ac.at/projects/askalon/



- Programming and execution environment for Grid workflows
  - Integrated various services for scheduling, executing, monitoring and analyzing Grid workflows
- Target applications
  - Workflows of scientific applications (C/Fortran)
    - Material science, flooding, astrophysics, movie rendering, etc.

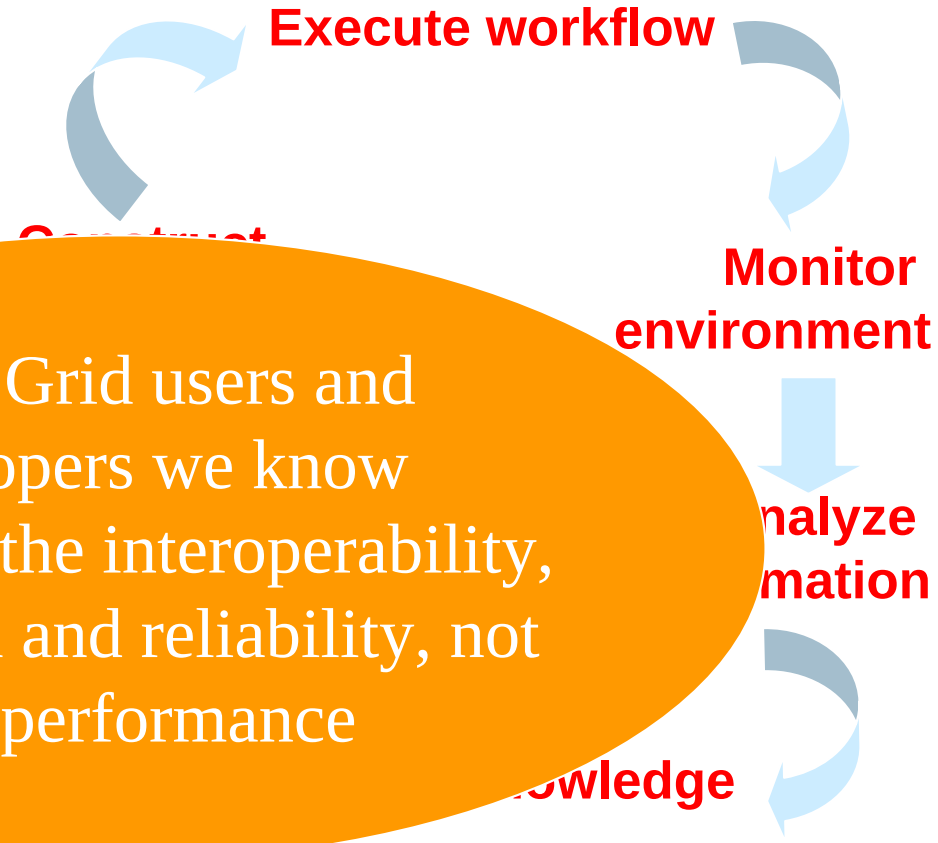
The K-WfGrid Project

- An FP6 EU project
  - www.kwfgrid.net
- Focuses
  - Automatic construction and reuse of workflows based on knowledge gathered through execution
- Target applications
  - Workflows of Grid/Web services
  - Grid/Web services may invoke legacy applications
  - Business (Coordinated Traffic Management, EPR) and scientific (e.g., Food simulation) applications



The K-WfGrid Project

- An FP6 EU project
  - www.kwfgrid.net
- Focuses
  - Automatic construction and reuse of workflows based on knowledge gathered through execution
- Target applications
  - Workflows of G
  - Grid/Web servi
    legacy applicati
  - Business (Coordi
    Management, EPR) a
    scientific (e.g., Food simul
    applications



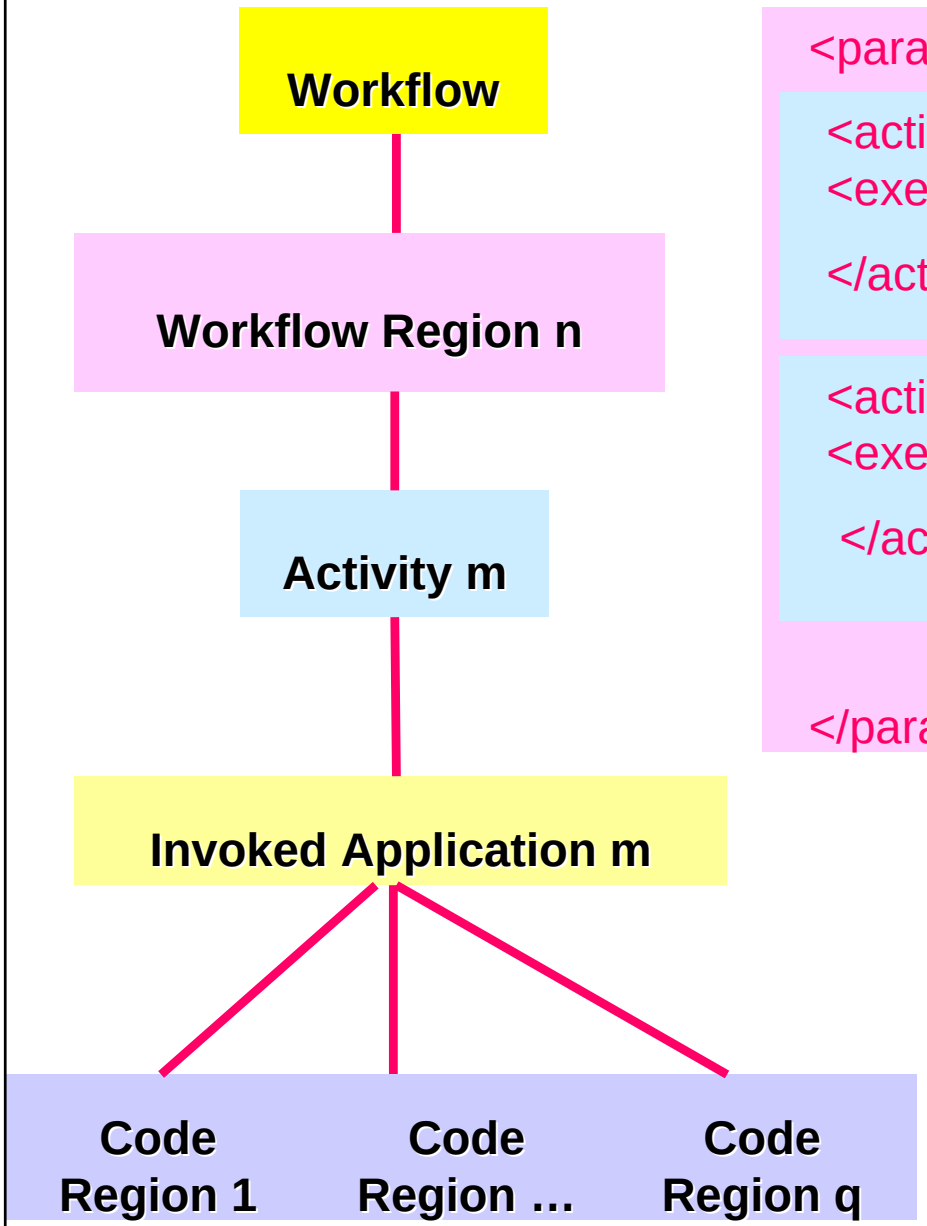
Many Grid users and developers we know emphasize the interoperability, integration and reliability, not just performance

K-Wf Grid

Objectives, Requirements, Approaches

- Performance monitoring and analysis for Grid workflows of Web/Grid services and multilingual components
- Performance and dependability metrics
- Monitoring and performance analysis
  - Service-oriented distributed architecture and peer-to-peer model
  - Unified monitoring and performance analysis system covering infrastructure and applications
  - Standardized data representations for monitoring data and events
  - Adaptive and generic sensors, distributed analysis, performance bottleneck search

Hierarchical View of Grid Workflows

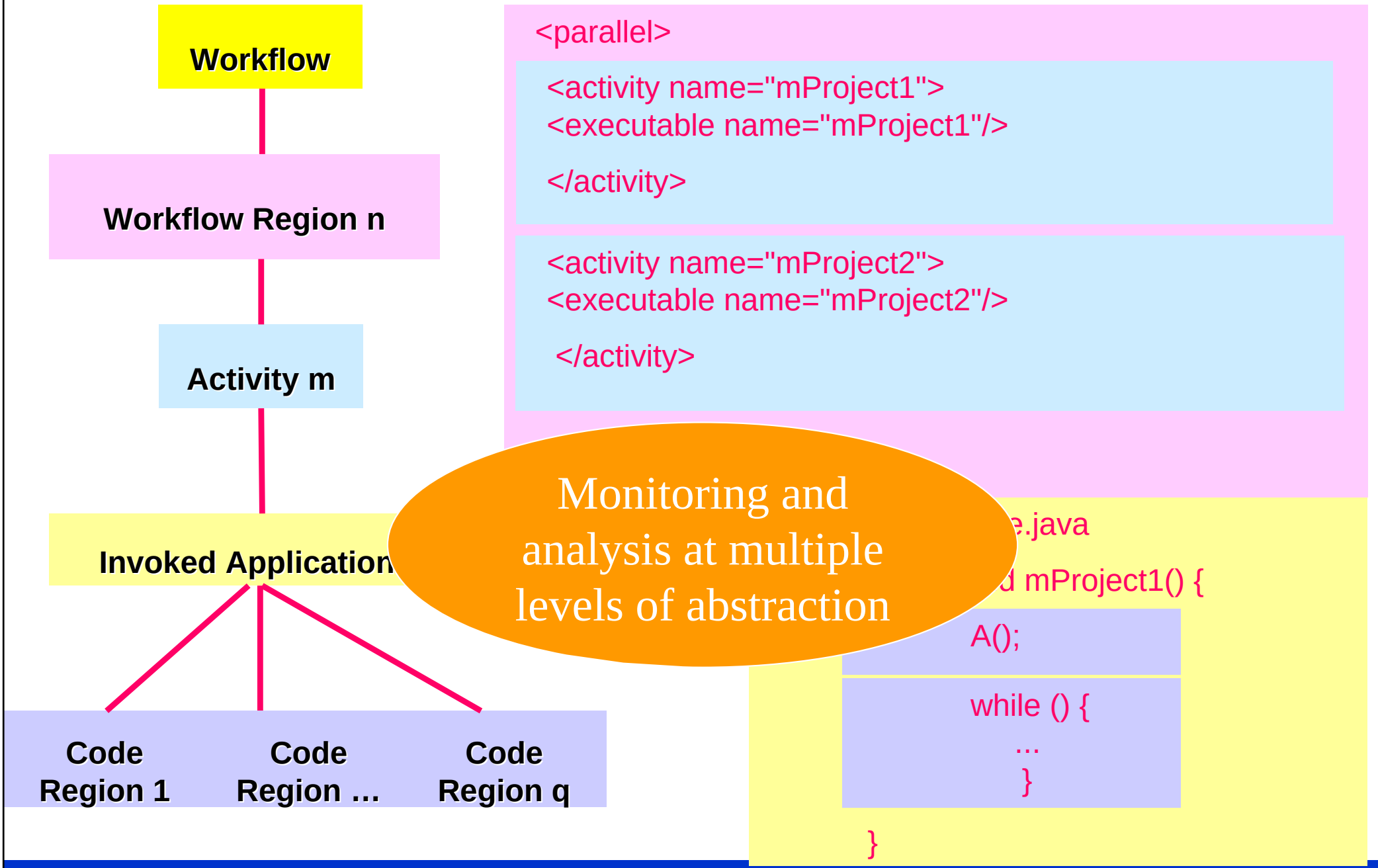



```
<parallel>
  <activity name="mProject1">
  <executable name="mProject1"/>
  </activity>
  <activity name="mProject2">
  <executable name="mProject2"/>
  </activity>
</parallel>
```

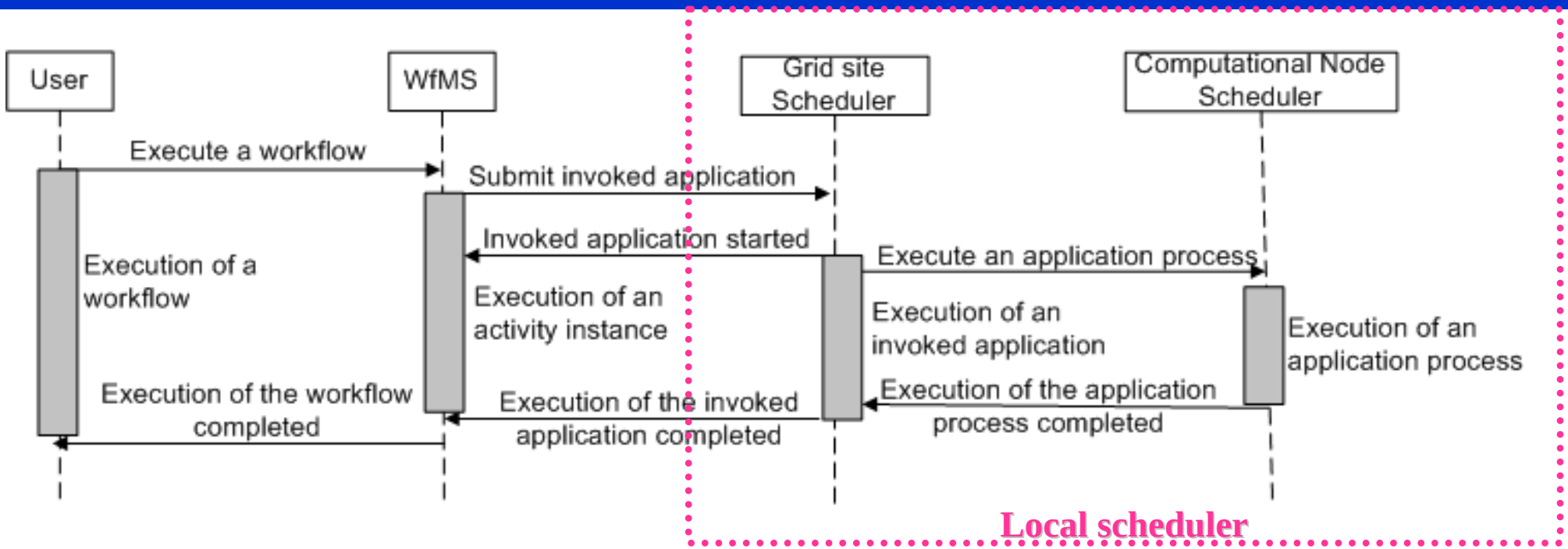
mProject1Service.java

```
public void mProject1() {
    A();
    while () {
    ...
    }
}
```

Hierarchical View of Grid Workflows



Workflow Execution Model (Simplified)



- Workflow execution
  - Spanning multiple Grid sites
  - Highly inter-organizational, inter-related and dynamic
- Multiple levels of job scheduling
  - At workflow execution engine (part of WfMS)
  - At Grid sites

Performance Metrics of Grid Workflows

- Interesting performance metrics associated with multiple levels of abstraction
  - Metrics can be used in workflow composition, for comparing different invoked applications of a single activity, adaptive scheduling, etc.
- Five levels of abstraction
  - Code region, Invoked application, Activity ,Workflow Region, Workflow
  - Topdown PMA: from a higher level to a lower one

Monitoring and Measuring Performance Metrics

- Performance monitoring and analysis tools
  - Operate at multiple levels
  - Correlate performance metrics from multiple levels
- Middleware and application instrumentation
  - Instrument execution engine
    - Execution engine can be distributed or centralized
  - Instrument applications
    - Distributed, spanning multiple Grid sites
- Challenging problems: the complexity of performance tools and data
  - Integrate multiple performance monitoring tools executed on multiple Grid sites
  - Integrate performance data produced by various tools

We need common concepts for performance data associated with Grid workflows

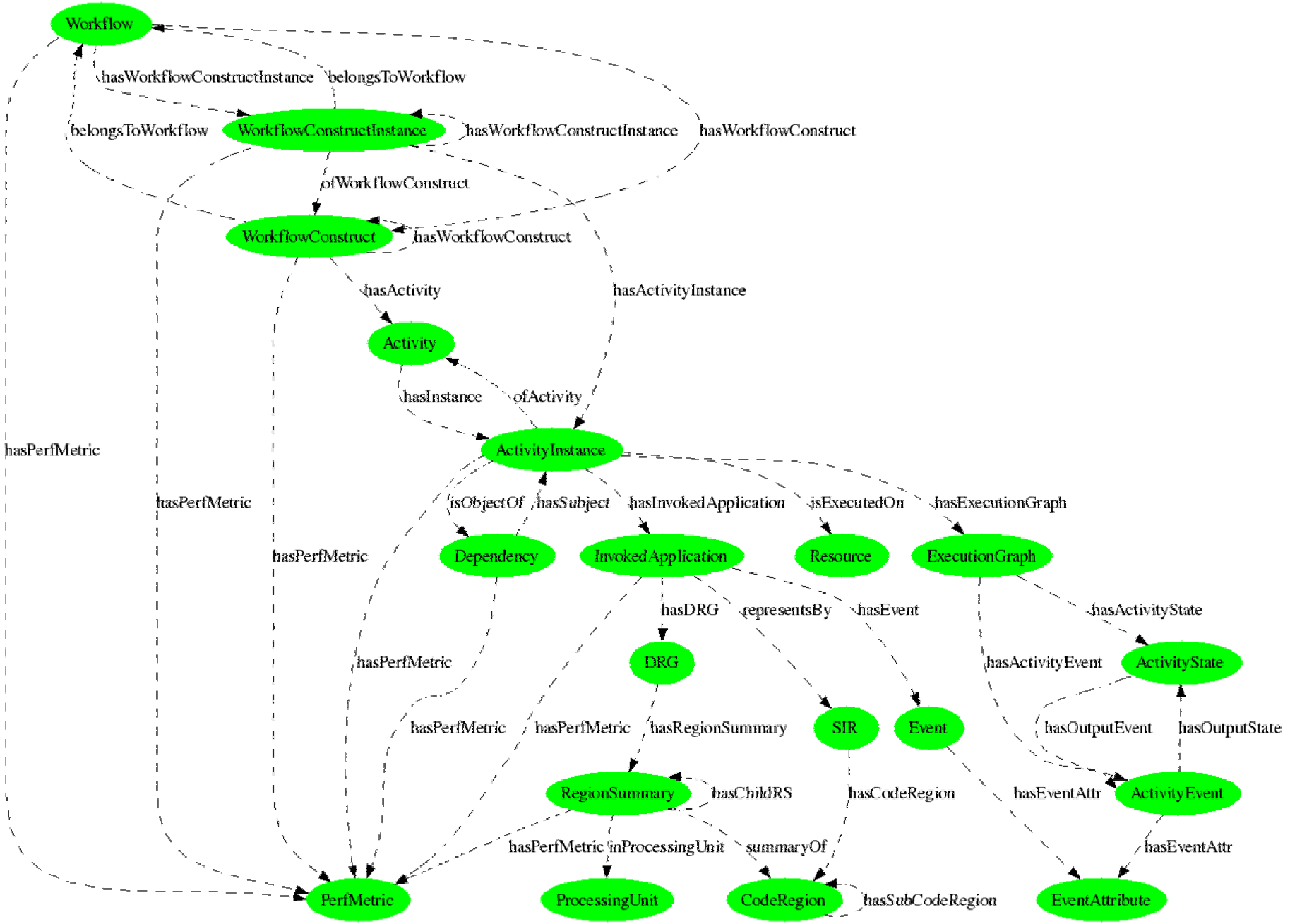
Utilizing Ontologies for Describing Performance Data of Grid workflows

- Metrics ontology
  - Specifies which performance metrics a tool can provide
  - Simplifies the access to performance metrics provided by various tools
- Performance data integration
  - Performance data integration based on common concepts.
  - High-level search and retrieval of performance data
- Knowledge base performance data of Grid workflows
  - Utilized by high-level tools such as schedulers, workflow composition tools, etc.
  - Used to re(discover) workflow patterns, interactions in workflows, to check correct execution, etc.
- Distributed performance analysis
  - Performance analysis requests can be built based on ontologies

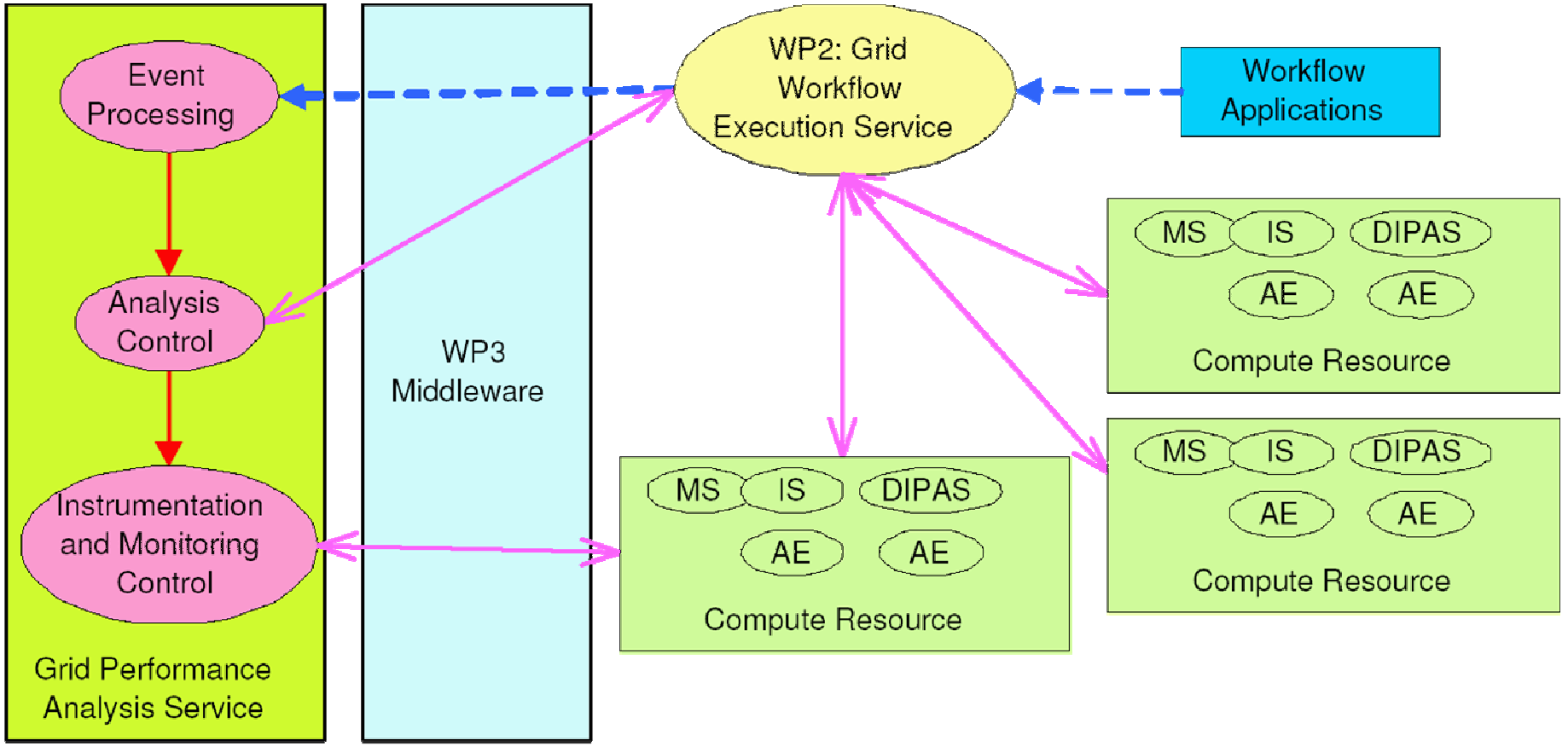
Workflow Performance Ontology

- WfPerfOnto (Ontology describing Performance data of Grid Workflows)
  - Specifies performance metrics
  - Basic concepts
    - Concepts reflect the hierarchical structure of a workflow
    - Static and dynamic workflow performance data
  - Relationships
    - Static and dynamic relationships among concepts

WfPerfOnto


Workflow
hasWorkflowConstructInstance
belongsToWorkflow
belongsToWorkflow
WorkflowConstructInstance
hasWorkflowConstructInstance
hasWorkflowConstruct
ofWorkflowConstruct
WorkflowConstruct
hasWorkflowConstruct
hasActivity
hasActivityInstance
Activity
hasInstance
ofActivity
hasPerfMetric
ActivityInstance
hasPerfMetric
isObjectOf
hasSubject
hasInvokedApplication
isExecutedOn
hasExecutionGraph
hasPerfMetric
Dependency
InvokedApplication
Resource
ExecutionGraph
hasDRG
representsBy
hasEvent
hasActivityState
hasPerfMetric
DRG
hasActivityEvent
ActivityState
hasPerfMetric
hasPerfMetric
hasRegionSummary
SIR
Event
hasOutputEvent
hasOutputState
RegionSummary
hasChildRS
hasCodeRegion
hasEventAttr
ActivityEvent
hasPerfMetric
inProcessingUnit
summaryOf
hasEventAttr
PerfMetric
ProcessingUnit
CodeRegion
hasSubCodeRegion
EventAttribute

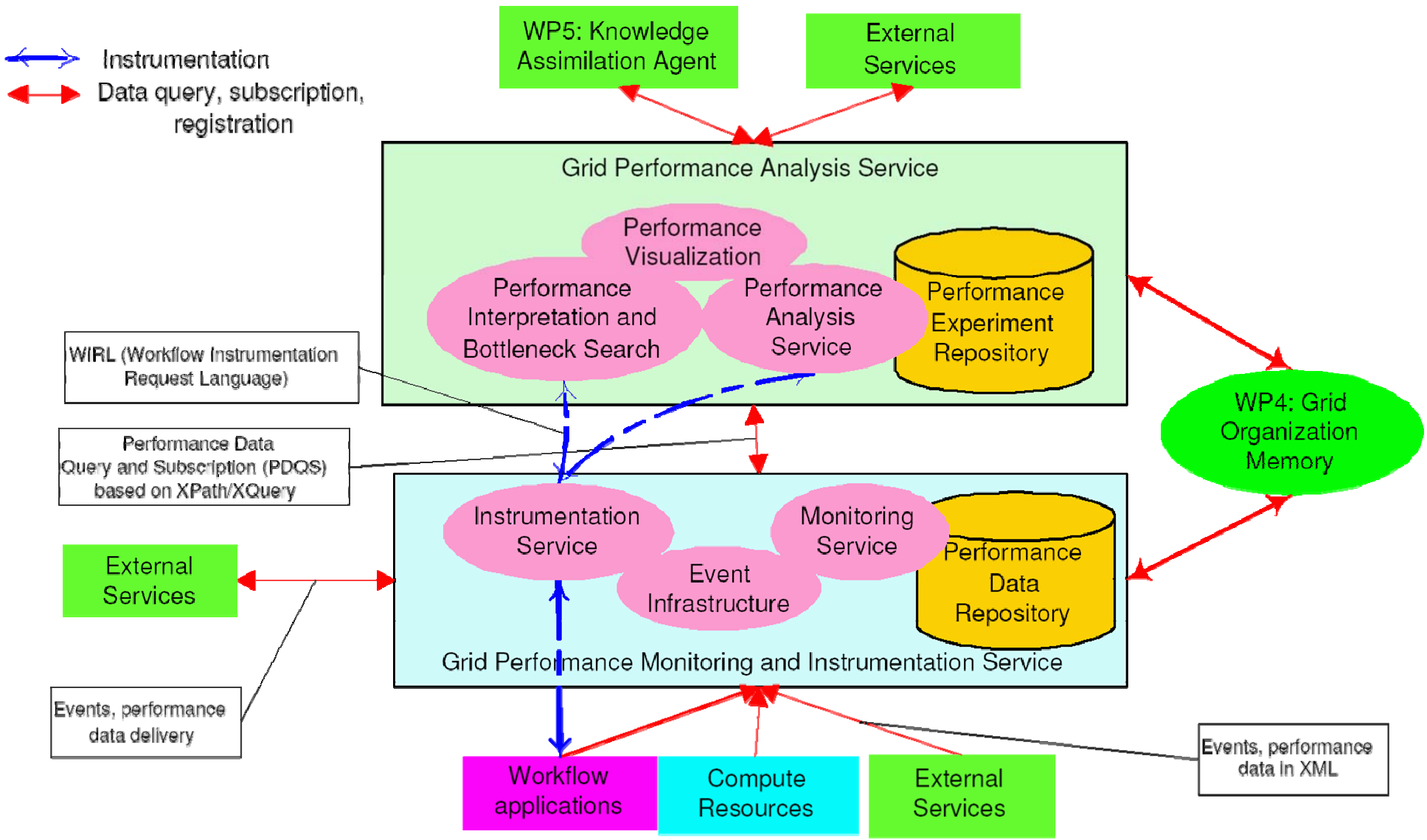
Monitoring and Analysis Scenario



Components for Grid PMA

- Three main parts of a unified performance monitoring, instrumentation and analysis system
  - Monitoring and instrumentation services
  - Performance analysis services
  - Performance service interfaces and data representations
    - We must reuse existing tools and techniques as much as possible
- Integration model
  - Loosely coupled: among Grid sites/organizations
    - Utilizing SOA for performance tools
  - Tightly coupled: within services deployed in a single Grid site/organization
    - Interfacing to existing (parallel) performance tools

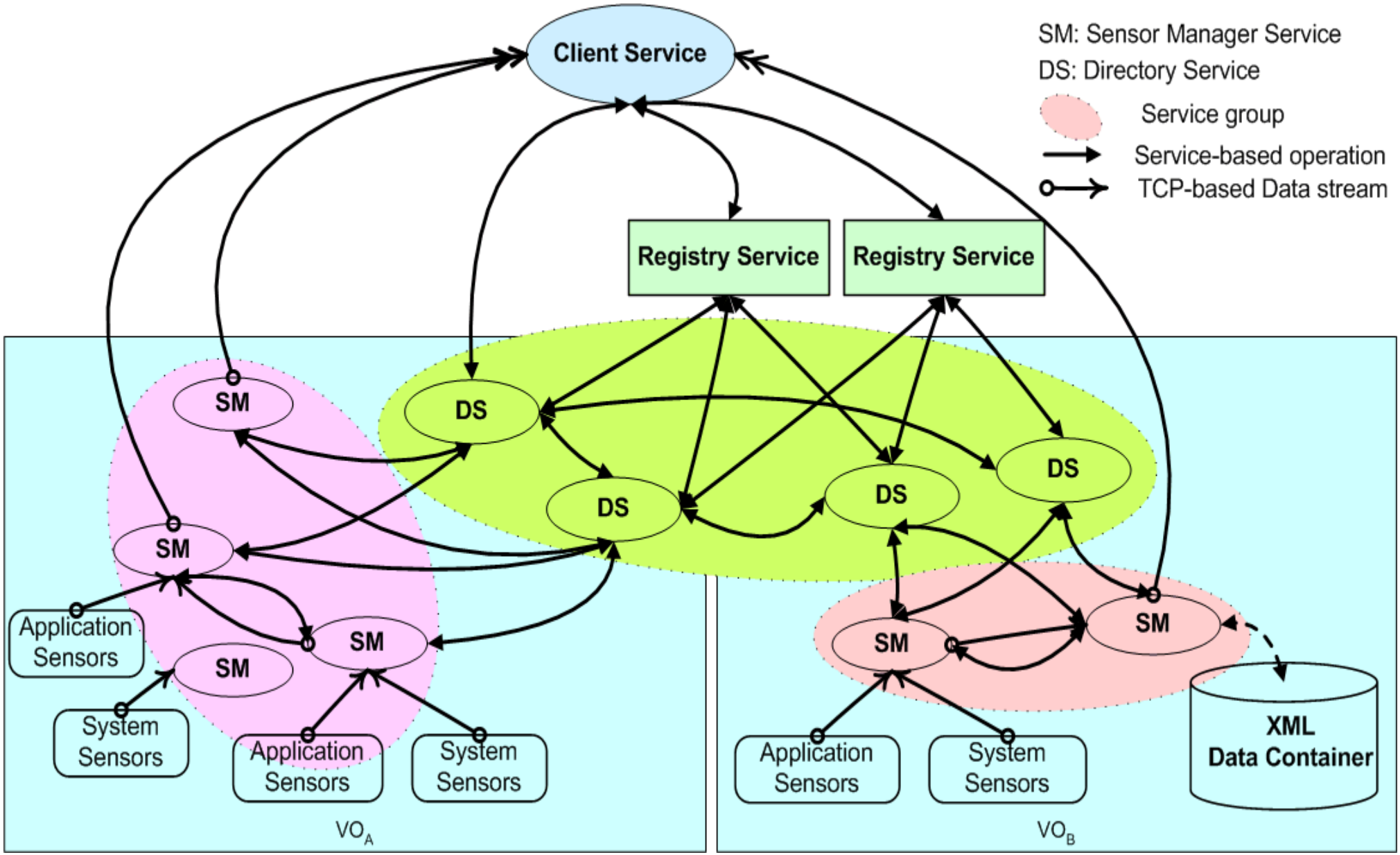
K-WfGrid Monitoring and Analysis Architecture



Self-Managing Sensor-Based Middleware

- Integrating diverse types of sensors into a single system
  - Event-driven and demand-driven sensors for system and applications monitoring, rule-based monitoring
- Self-managing services
  - Service-based operations and TCP-based stream data delivery
  - Peer-to-peer Grid services for the monitoring middleware
- Query and subscription of monitoring data
  - Data query and subscription
  - Group-based data query and subscription, and notification

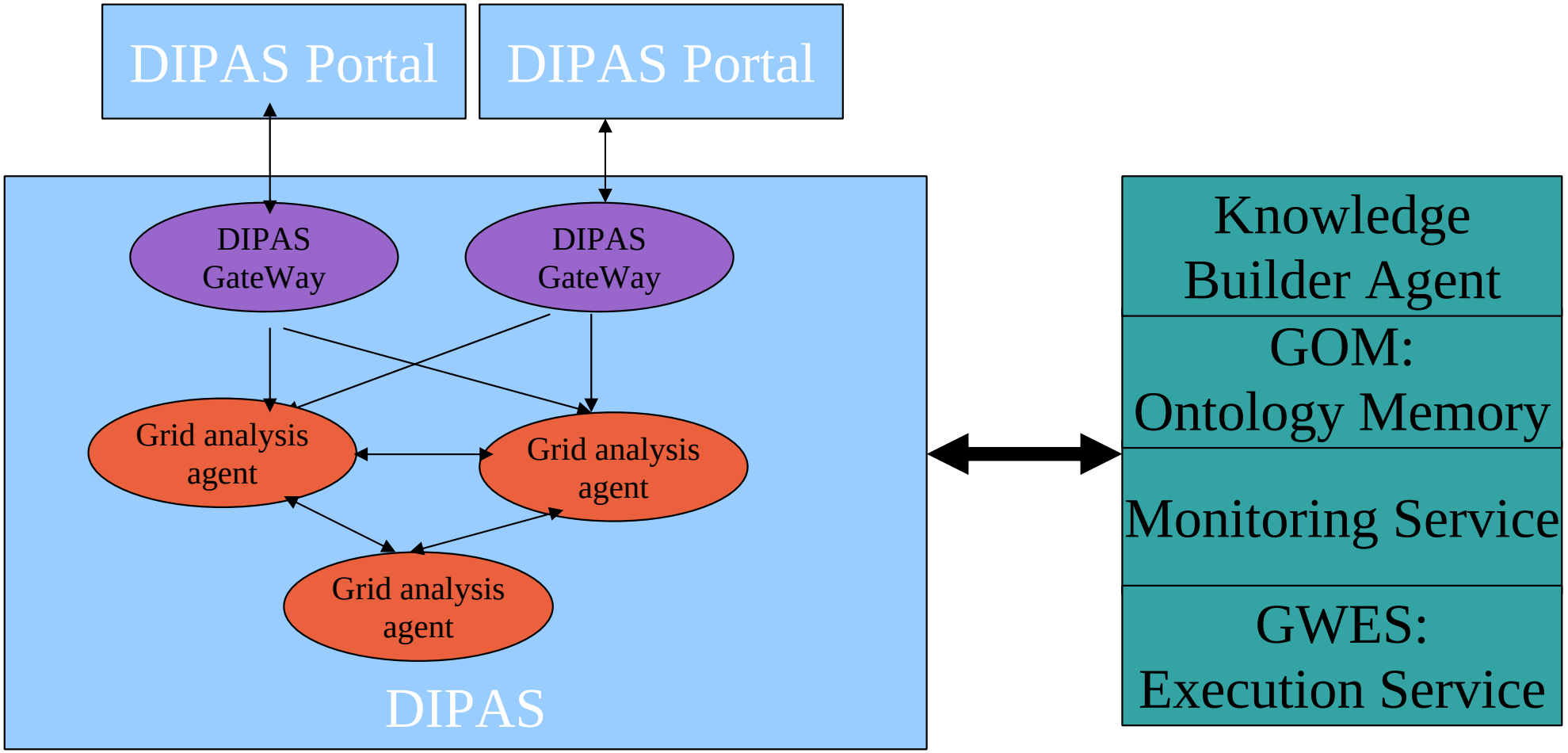
Self-Managing Sensor Based Monitoring Middleware



Workflow Instrumentation

- Issues to be addressed
  - Multiple levels of instrumentation
  - Instrumentation of multilingual applications (C/Java/Fortran)
  - Must be dynamic (enabled) instrumentation
- ASKALON and K-WfGrid approach
  - Utilizing existing instrumentation techniques
    - OCM-G (Roland Wismueller and Marian Bubak)
      - dynamic enabled instrumentation C/Fortran
    - Dyninst (Barton Miller, Jeff Hollingsworth)
      - for binary code generated from C/Fortran
    - Source/dynamic instrumentation of Java (from Java 1.5)
  - Using APART standardized intermediate representation (SIR)
  - XML-based request for controlling instrumentation

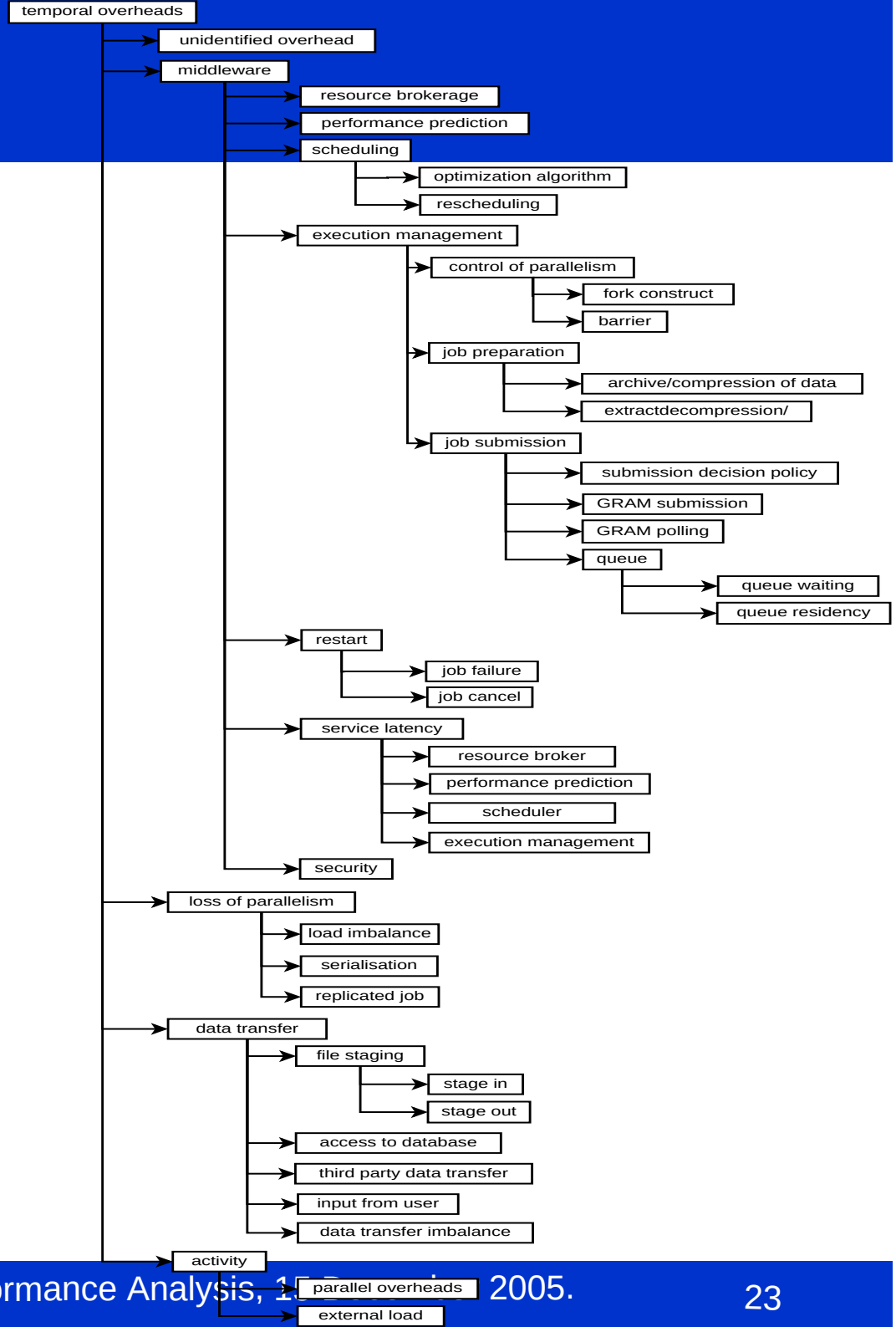
DIPAS: Distributed Performance Analysis



- Grid analysis agent accepts WARL and returns performance metrics described in XML under a tree of metrics
  - WARL (workflow analysis request langauge): based on concepts and properties in WfPerfOnto

Workflow Overhead Classification

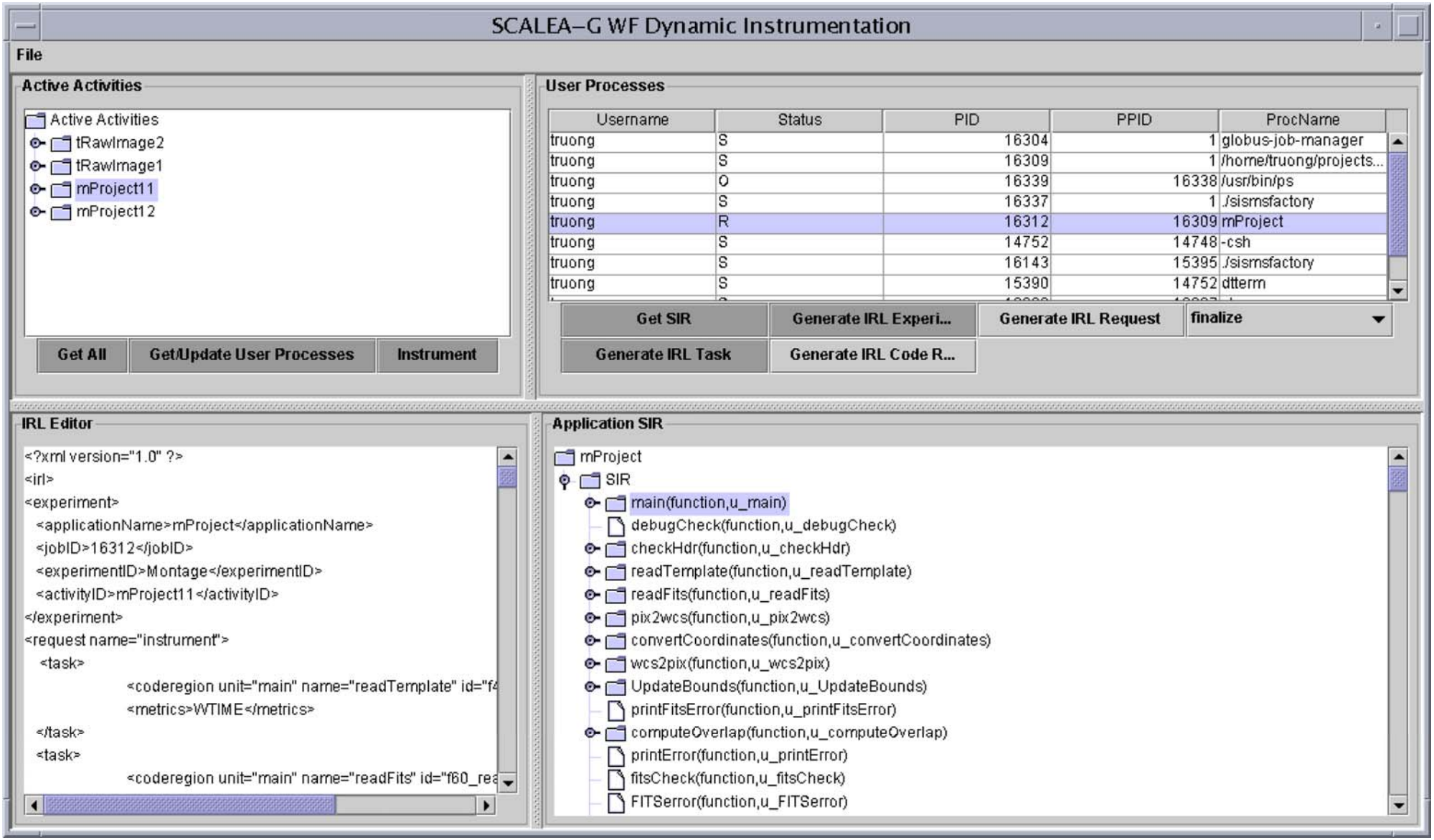
- Middleware
  - Scheduler
  - Resource Manager
  - Execution management
    - Control of parallelism
    - Loss of parallelism
- Loss of parallelism
  - Load imbalance
  - Serialization
  - Replicated job
- Data transfer
- Activity
  - Parallel processing overheads
  - External load



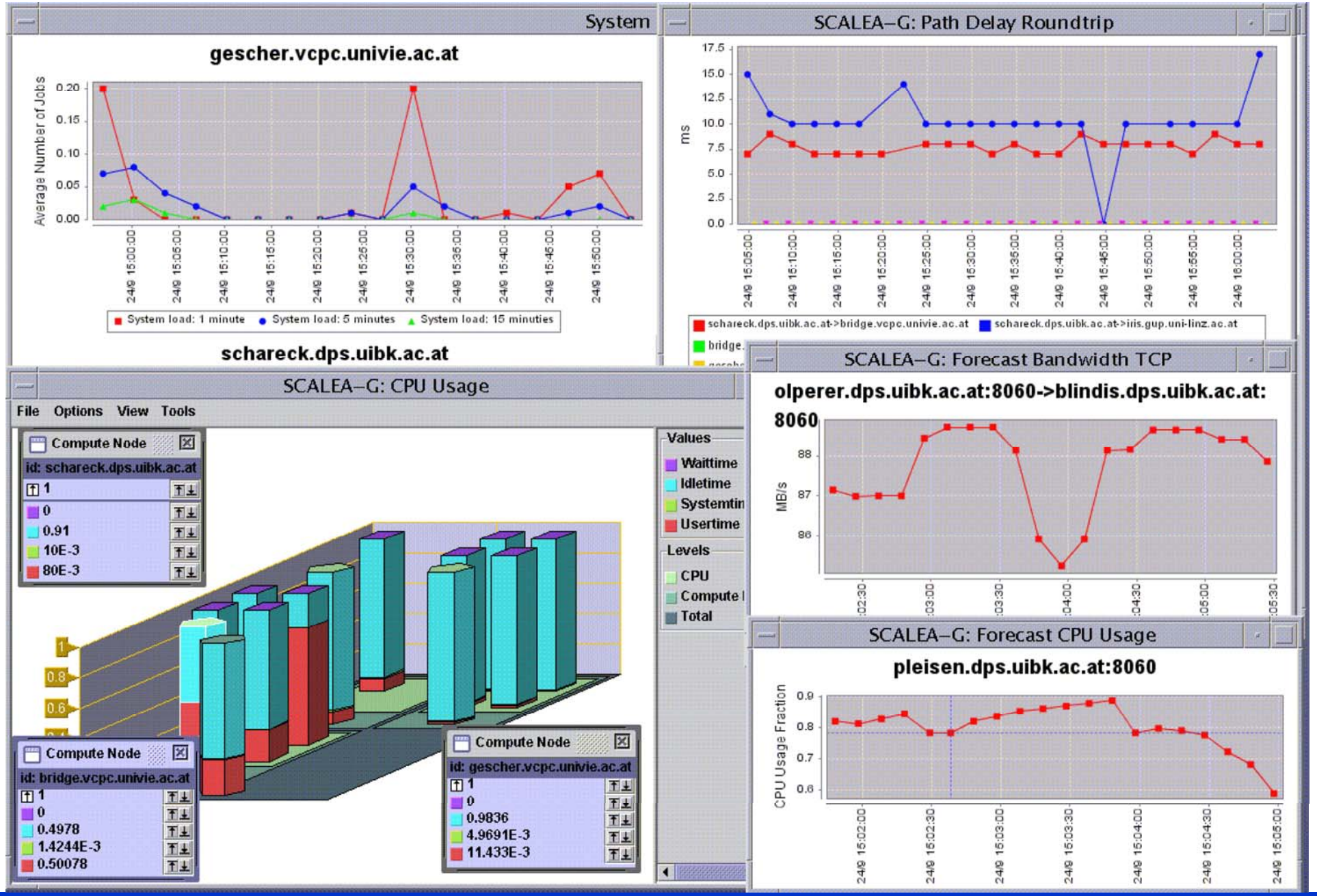
Current status of the implementation

- SOA-based
  - Monitoring, Instrumentation and Analysis services are GT4 based
  - XML-based for performance data representations and requests
- Monitoring and Instrumentation Services
  - gSOAP, GSI-based dynamic instrumentation service
  - Java-based dynamic instrumentation
  - OCM-G
  - But we need to integrate them into a single framework for workflows and it is a non trivial task
- Analysis services
  - GT4-based with distributed components
  - Simple language for workflow analysis request (WARL), designed based on WfPerfOnto
  - Metrics are described in XML

Example: Dynamic Instrumentation



Snapshot: Online System Monitoring

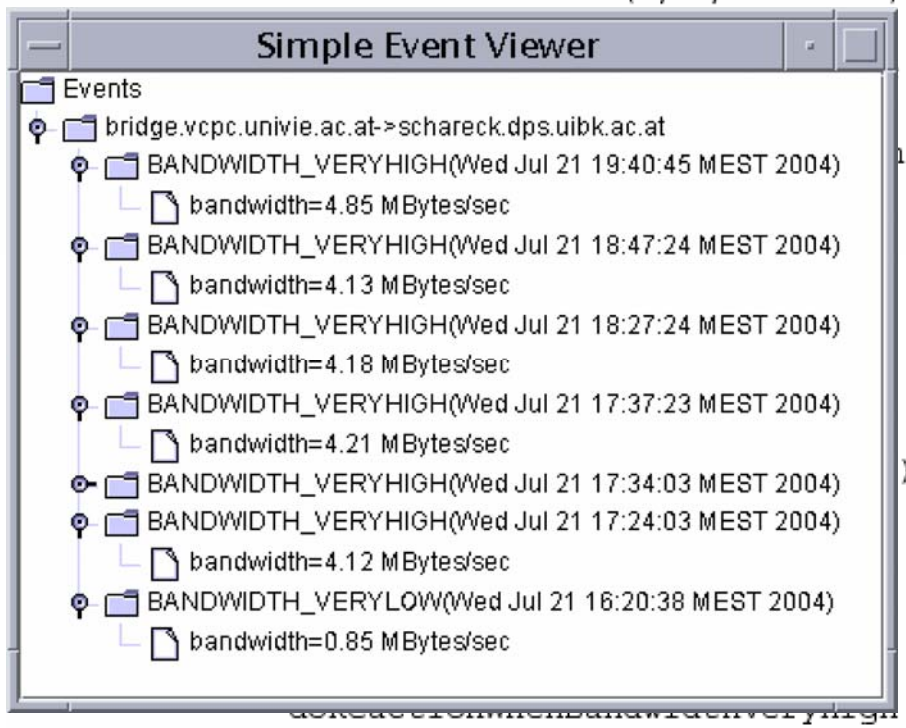


Rule-based Monitoring

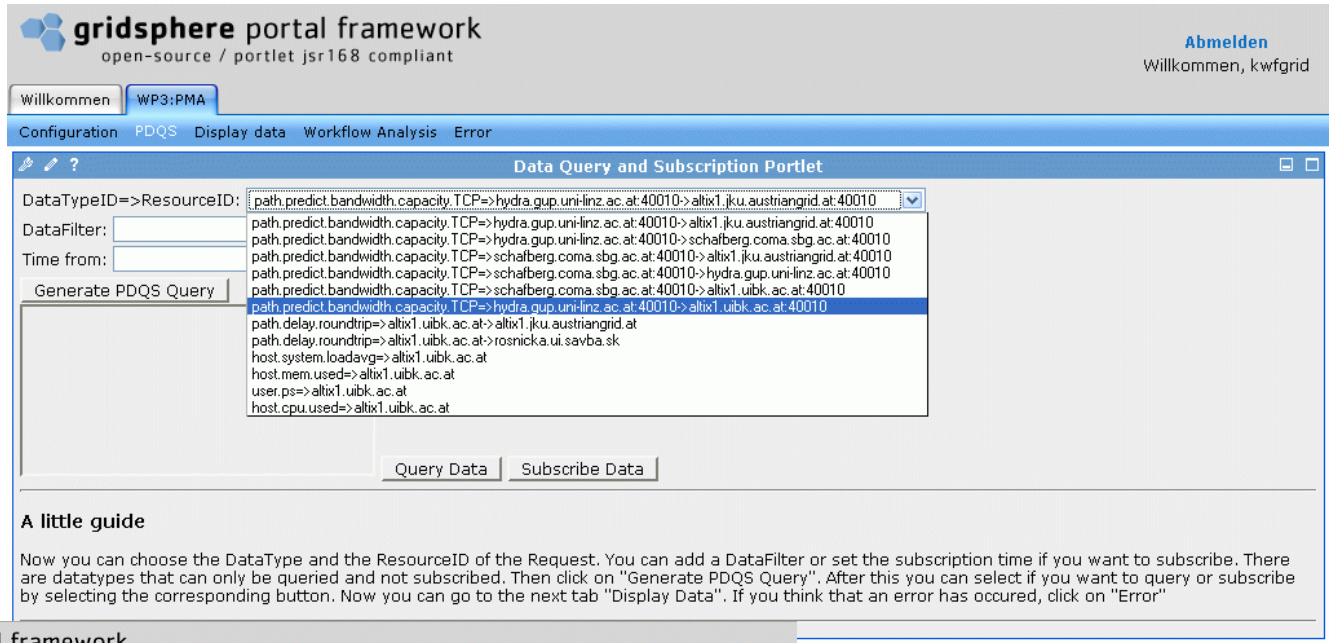
- Sensors use rules to analyze monitoring data
- Rules are based on IBM ABLE toolkit

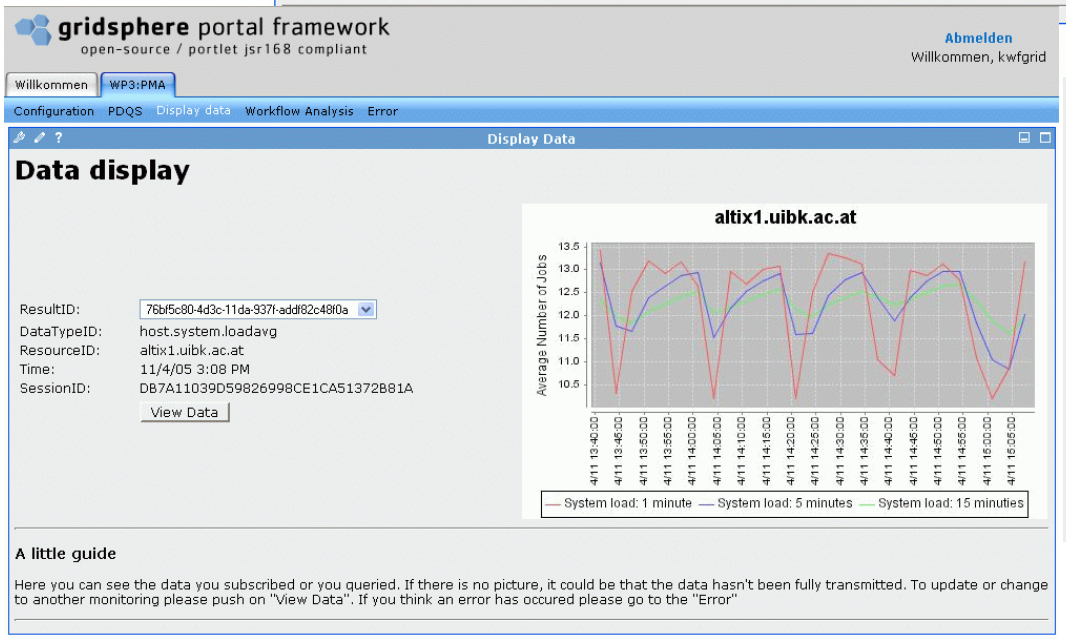
- Example:
  - A network path in the Austrian Grid, bandwidth < 5MB/s (based on Iperf)
  - Define a fuzzy variable with VERY LOW, LOW, MEDUM, HIGH, VERY HIGH
  - Fuzzy rules: Events send when bandwidth VERY LOW or VERY HIGH

```
Fuzzy bandwidth= new Fuzzy( 0,5) {
      Shoulder VERYLOW = new Shoulder(0, 1, ARL.Left);
                                                          nt);
                                                          );
                                                          ();
                  }
R_OTHER:      doNormalReaction();
```



Online Infrastructure Monitoring

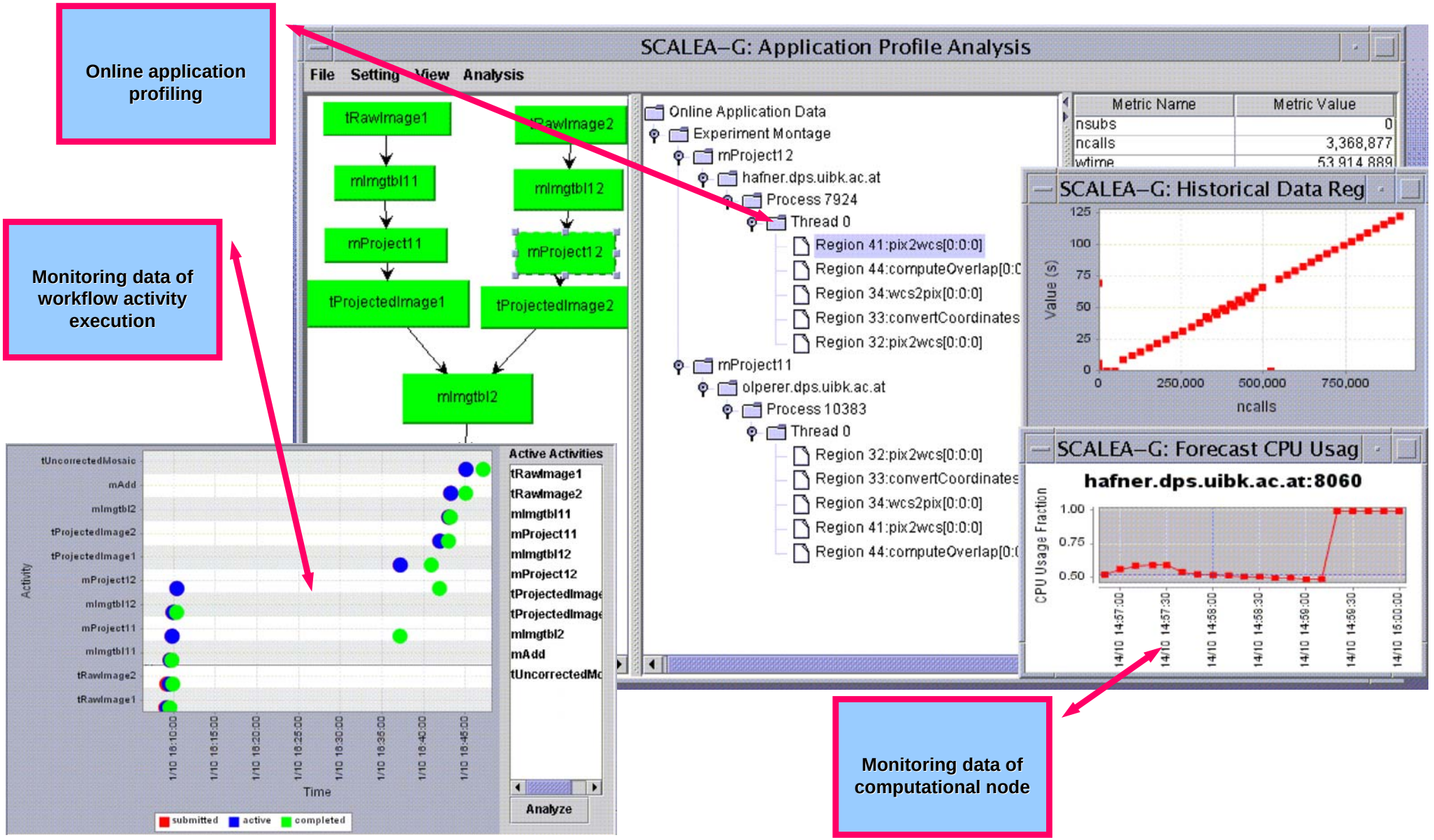




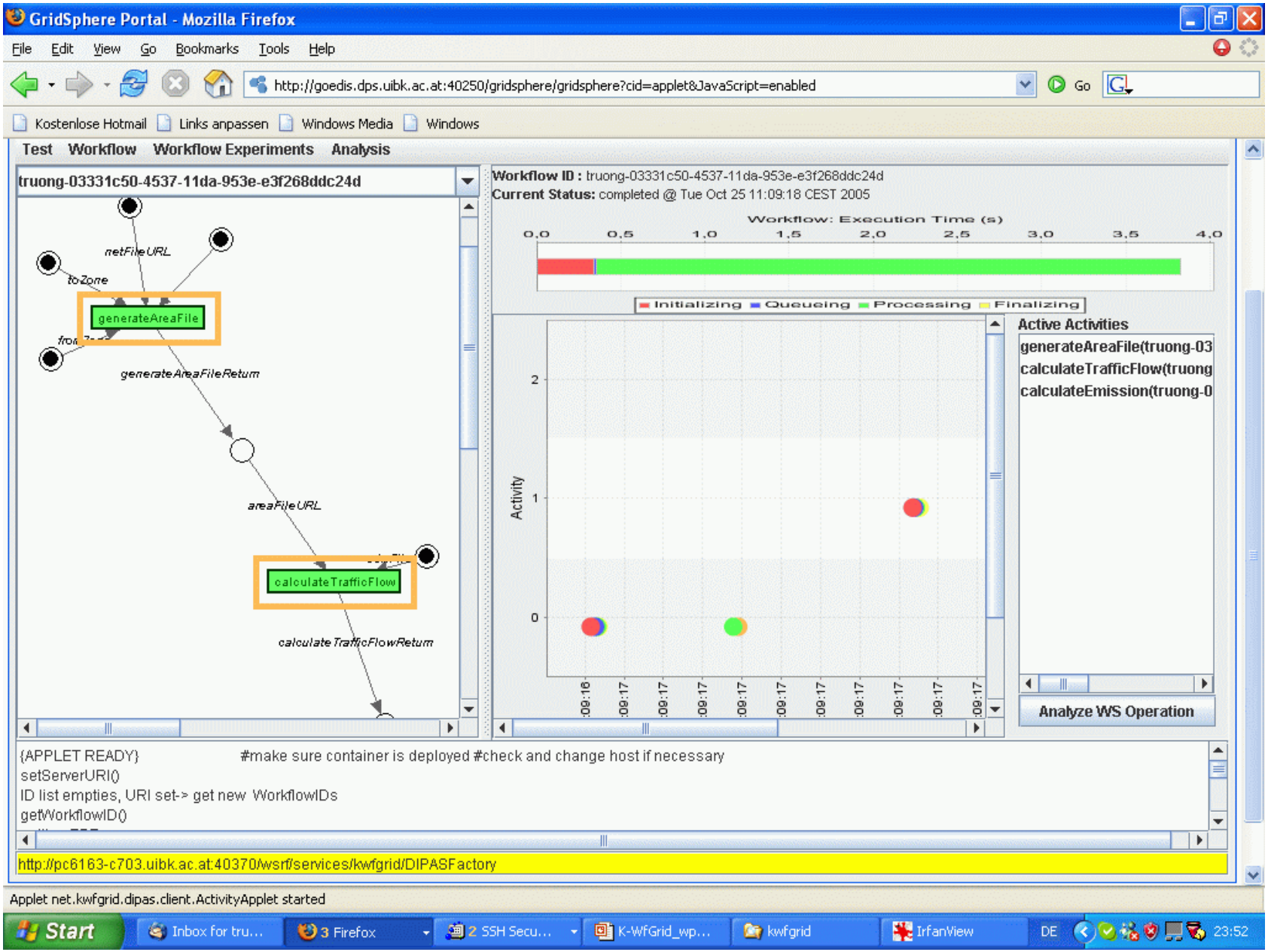
Display static Data

Hostname	zeus70.cyf-kr.edu.pl
Resource id	zeus70.cyf-kr.edu.pl
Addresses:	149.156.9.110
System model	null
Physical memory	2020.7 MB
Virtual memory	2651.3 MB
Number of CPUs	2
CPU type	x86
CPU speed	2800 MHz
OS name	Linux
OS version	2.4.27-2-686-smp
Harddisk Size	75 GB

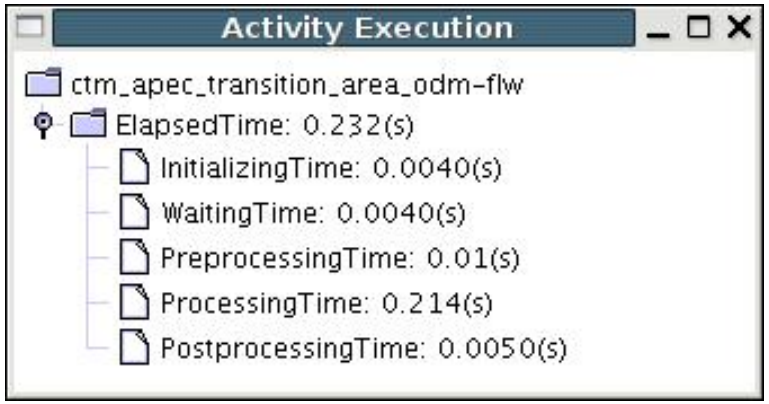
DAG-based workflow monitoring and analysis

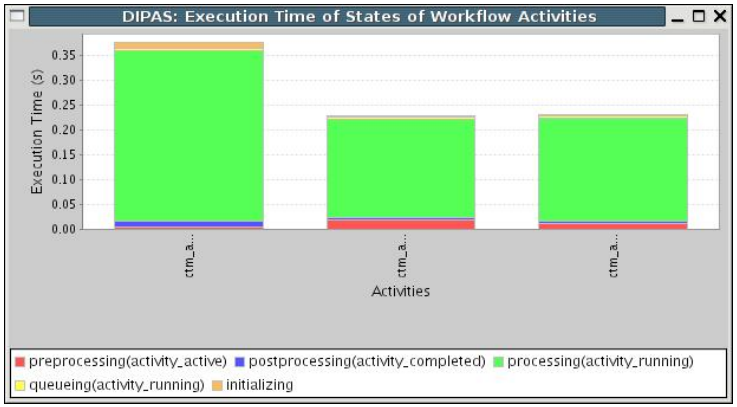


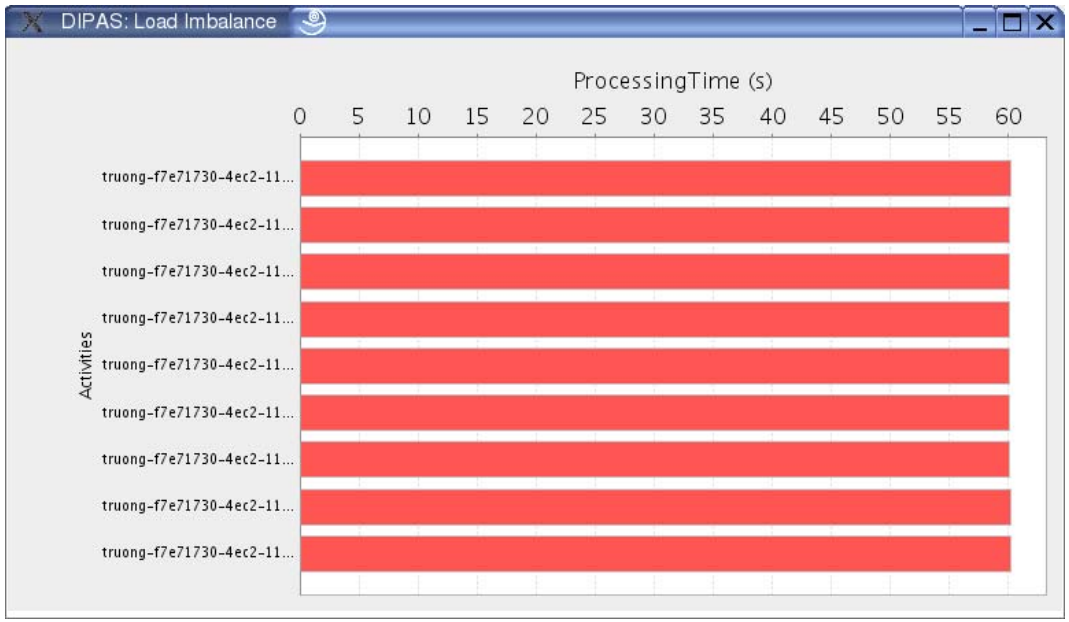
K-WfGrid: Online Workflow Monitoring and Analysis

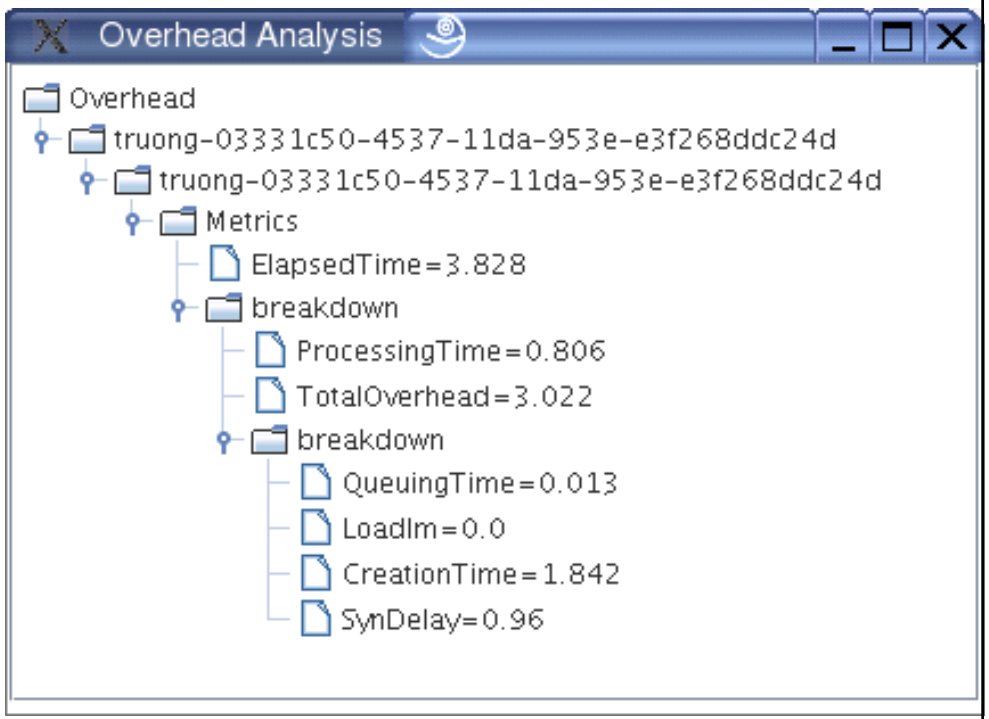


Online Workflow Analysis









Conclusions

- The architecture of monitoring and analysis service must tackle the dynamics and the diversity of the Grid
  - Service-oriented, peer-to-peer model, adaptive sensors
- Integration and reuse are important issues
  - Loosely coupled and tightly coupled
  - Do not neglect data representations and service interfaces
- Performance metrics and ontology for Grid workflows
  - What performance metrics are important and how to measure them
  - Common and generic concepts and relationships monitored and alayzed
- Given well-defined service interfaces, data representation, performance metrics and ontology
  - Simplify the integration among components
  - Towards automatic, intelligent and distributed performance performance analysis

UIBK perf. work: http://dps.uibk.ac.at/projects/pma/

Papers: http://dps.uibk.ac.at/index.pl/publications