

# TOWARDS WORKFLOW SHARING AND REUSE IN THE ASKALON GRID ENVIRONMENT

Jun Qin

Max Berger, Thomas Fahringer

Distributed and Parallel Systems group  
Institute of Computer Science  
University of Innsbruck, Austria

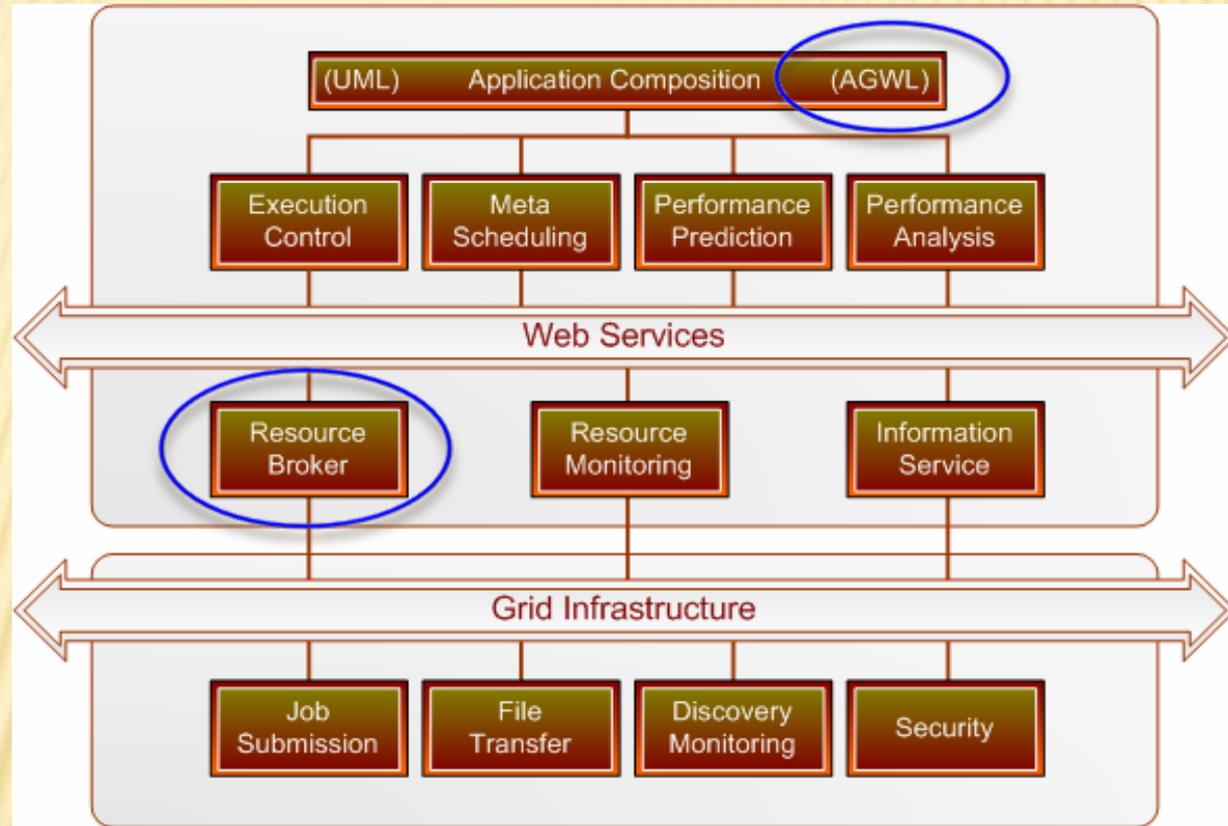
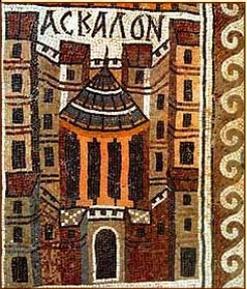
*Tuesday, October 14, 2008*

# OUTLINE

- × Introduction
  - + Need of workflow sharing and reuse
- × Workflow sharing and reuse in ASKALON
  - + ASKALON
  - + Workflow sharing through AWHE
  - + Workflow component, sub-workflow and entire workflow reuse
    - × Infinite sub-workflow invocation detection
- × Case studies
  - + Meteorology: MeteoAG
  - + Material Science: Wien2K
- × Summary

# NEED OF WF SHARING & REUSE

- × More and more Grid workflow application are built
  - + Built Grid workflows from scratch is still challenging
  - + Reuse established workflow
    - × Reduces workflow authoring time
    - × Improves the quality of workflows
- × Example scenarios
  - + a domain scientist in one research group wants to run a workflow constructed by another group, without knowing how to construct Grid workflows
  - + a domain scientist wants to invoke a workflow multiple times within his own workflow
    - × based on a for loop



- ✘ ASKALON: invisible Grid
  - + AGWL
  - + GLARE, part of resource manager

# ASKALON (CONT.)

- ✘ Abstract Grid Workflow Language (AGWL)
  - + XML-based language
  - + High level of abstraction
    - ✘ Activity: described by activity types
    - ✘ Activity Type: abstract description of a group of activity deployments
    - ✘ Activity Deployment: concrete implementations of computational entities deployed in the Grid
- ✘ GLARE
  - + a distributed framework for dynamic registration, automatic deployment, and on-demand provision of workflow activities
  - + provide mapping from activity types to activity deployments
- ✘ Workflow component reuse
  - + Activity types are abstract, can be reused

# AWHE

- ✘ ASKALON Workflow Hosting Environment
- ✘ AWHE saves workflows and their metadata into database
  - + domain, author, version, etc.
- ✘ AWHE is integrated into ASKALON and provides functionalities for users to:
  - + publish workflows directly from the workflow composition tool.
  - + associate workflows with metadata
    - ✘ Automatically generate next version
  - + search workflows by their metadata
  - + show views of workflow graphs, including graphs of its sub-workflows.
  - + run workflows with a single click.
  - + mark a workflow as deprecated for discouraging their use because a better alternative exists.

# AWHE - SCREENSHOT

ASKALONE Workflow Hosting Environment

name	version	owner	created	domain
MeteoAG	1.0.1	jerry	11/30/07 7:09 PM	Meteorology
MeteoAG	1.0.2	jerry	11/30/07 7:10 PM	Meteorology
SubWorkflowDemo	1.0.0	jerry	11/30/07 7:40 PM	testing
SubWorkflowDemo	1.1.0	jerry	1/5/08 10:25 AM	testing
SubWorkflowDemo	1.2.0	jerry	1/6/08 9:24 PM	testing
Wien2K	1.0.0	jerry	11/30/07 7:13 PM	Material Science
Wien2K	1.1.0	jerry	11/30/07 7:10 PM	Material Science

Model | ACWL | General |  
main | subA | subB

1

4

3

Search workflows in ASKALONE WHE:  
 Match all of the following     Match any of the following

Domain	begins with	test	+	-
Name	ends with	demo	+	-
Version	equals	1.1.0	+	-
Author	contains	jerry	+	-

Save this Search As: \_\_\_\_\_

Cancel    Search

Search    Refresh

# WORKFLOW REUSE

- × Workflow Component Reuse
  - + each activity has attribute “type”
    - × type=“an activity type in GLARE”
  - + activity type is abstract, therefore can be reused
- × Sub-Workflow/Workflow Reuse
  - + Extend the semantics of the attribute “type”:
    - × Refer to Activity Type: atrp://domain/activitytype
    - × Refer to Sub-Workflow: subw://subworkflow
    - × Refer to Workflow in AWHE: awhe://domain/workflow/version

# EXAMPLE OF SUB-WORKFLOW REUSE

```
1 <agwl name="w">
2   <subWorkflow name="subW1" .../>
3   <workflowInput .../>
4   <workflowBody>
5     <activity name="a" type="subw://subW1">
6       <dataIns .../> <dataOuts .../>
7     </activity>
8   </workflowBody>
9   <workflowOutput .../>
10 </agwl>
```

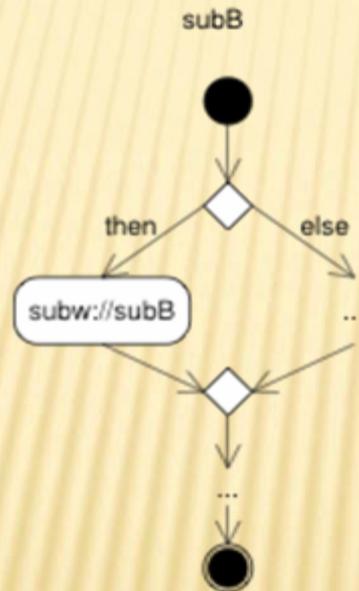
sub-workflow “subW1” is defined at line 1  
activity “a” invokes subW1 at line 5

# DETECTION OF INFINITE RECURSION

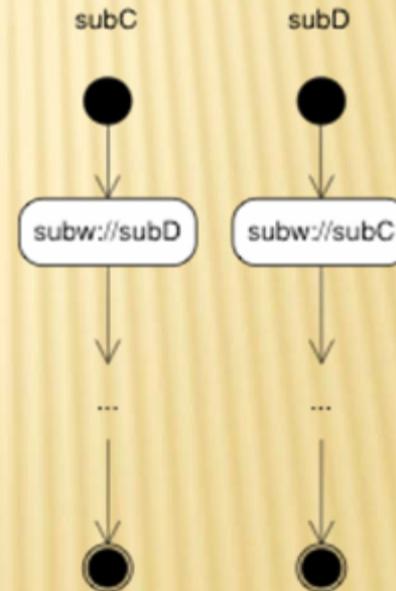
- ✘ Not well designed recursive sub-workflow invocations can provoke infinite recursion



(a)



(b)



(c)

# DETECTION OF INFINITE RECURSION

- ✘ Incorrect recursive sub-workflow invocations are detected by traversing the workflow models
  - + while the workflows is being built
  - + before submitting workflows for execution on Grid resources.
- ✘ The detailed algorithm will be included in the final version of the paper

# ENTIRE WORKFLOW REUSE

- ✘ Workflows published in AWHE can be reused
- ✘ Generation of workflow template
  - + because workflows in AWHE may have input data or output location specified
- ✘ Two mechanisms of code generation
  - + Embed the AGWL code of the reused workflow into the new workflow
  - + Late binding: Use the references of the reused workflows in the new workflow

# WORKFLOW COMPONENT, SUB-WORKFLOW AND WORKFLOW REUSE

ASKALON Grid Applications Development Environment - Untitled

File Edit View Insert Format Workflow Resources Tools Security Help

SubWorkflows | Activity Types | Workflows

Activity Types

- Tutorial
  - tutorial:helloWorld
  - sleep
  - echoDate
  - tutorial:sleep
  - helloWorld
  - tutorial:echoDate
- Femilab
- grasil
- chemistry
- Rendering
- Imaging
- Astrophysics
- Hydrology
- unknown

drag & drop

drag & drop

plugins

(sub-)workflow diagrams

Tree View

- SubWorkflowDemo
  - main
    - Control Flow
      - ControlFlow\_1
      - ControlFlow\_2
      - ControlFlow\_3
      - ControlFlow\_4
    - Data Flow
    - InitialNode
    - FinalNode
    - Activity\_1
    - Activity\_2
    - Activity\_3
  - subA
    - Control Flow
      - ControlFlow\_1
      - ControlFlow\_2
    - Data Flow
    - InitialNode\_1
    - FinalNode\_1
    - Activity\_4
  - subB

UML / AGWL

Activity | DataIn Ports | DataOut Ports | Properties | Constraints

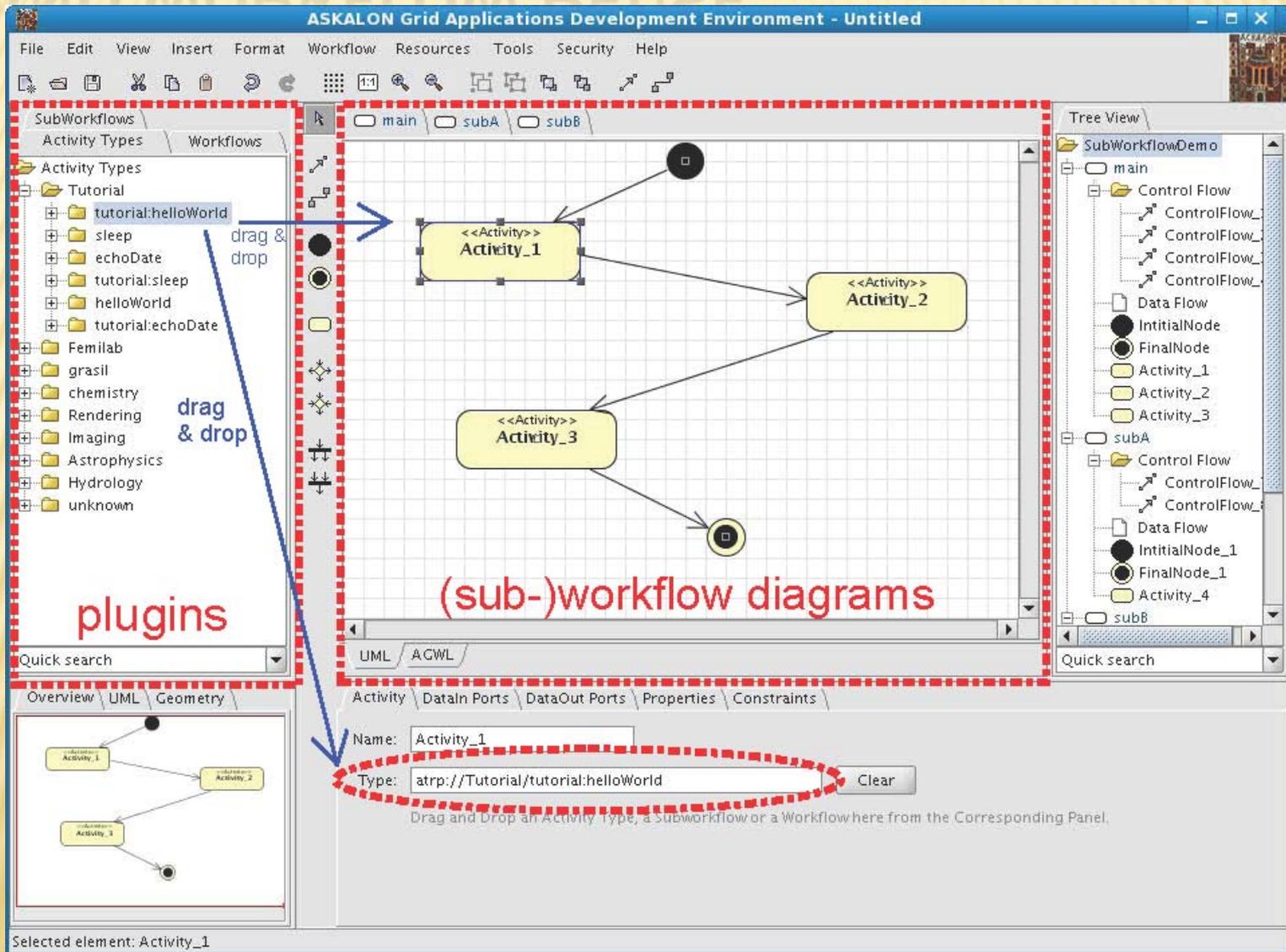
Name: Activity\_1

Type: atp://Tutorial/tutorial:helloWorld

Clear

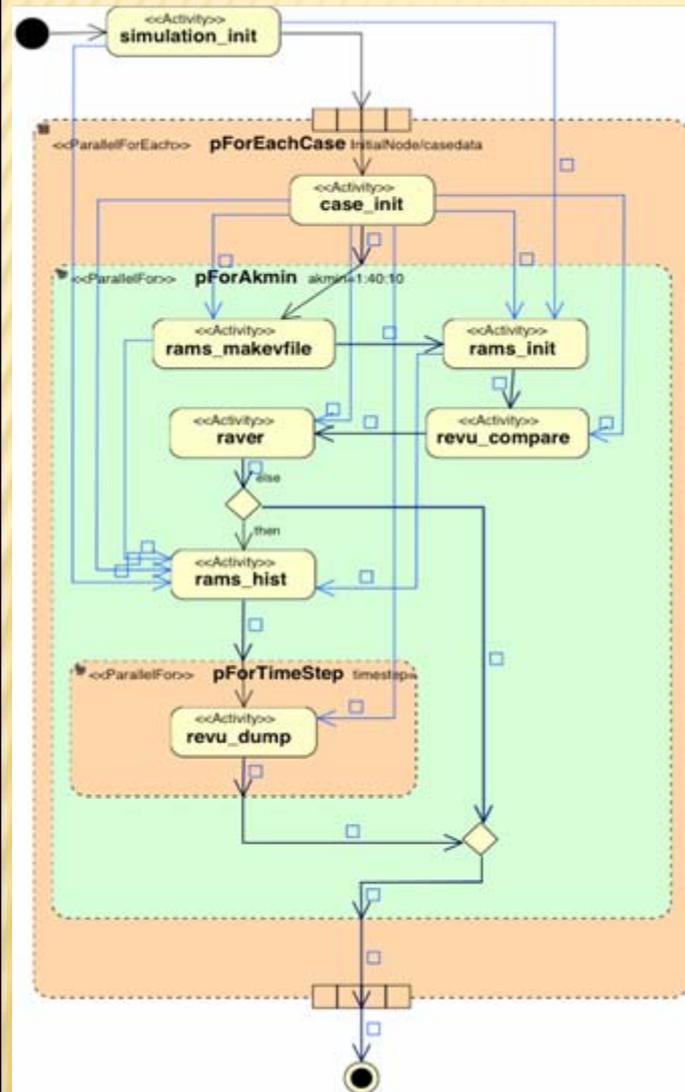
Drag and Drop an Activity Type, a Subworkflow or a Workflow here from the Corresponding Panel.

Selected element: Activity\_1

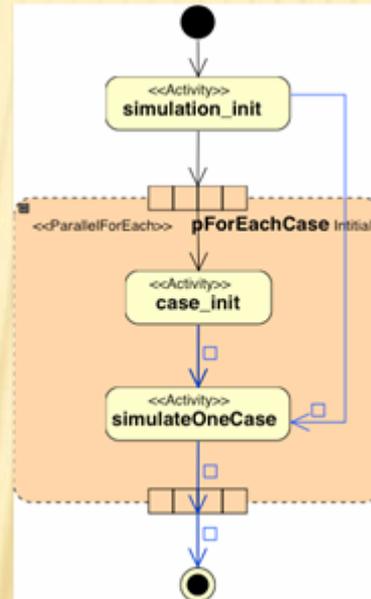


The screenshot displays the ASKALON Grid Applications Development Environment. The main workspace shows a workflow diagram with three activity nodes: Activity\_1, Activity\_2, and Activity\_3. Activity\_1 is connected to Activity\_2, and Activity\_2 is connected to Activity\_3. The diagram is surrounded by a red dashed box. On the left, a 'SubWorkflows' panel shows a tree view of activity types, with 'tutorial:helloWorld' selected. A blue arrow points from this selection to Activity\_1 in the diagram. Below the diagram, a 'Name' field contains 'Activity\_1' and a 'Type' field contains 'atp://Tutorial/tutorial:helloWorld'. A red dashed oval highlights the 'Type' field. At the bottom left, a 'Selected element: Activity\_1' label is visible.

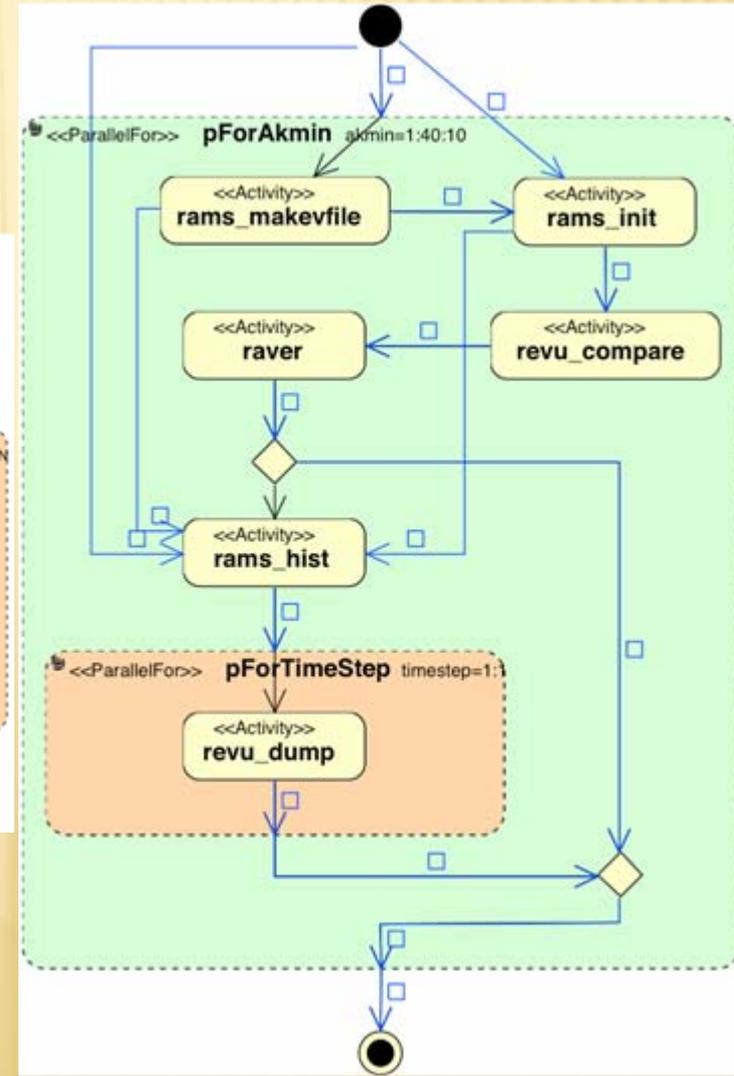
# CASE STUDY - METEOAG



single workflow

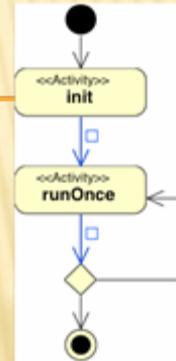


main workflow

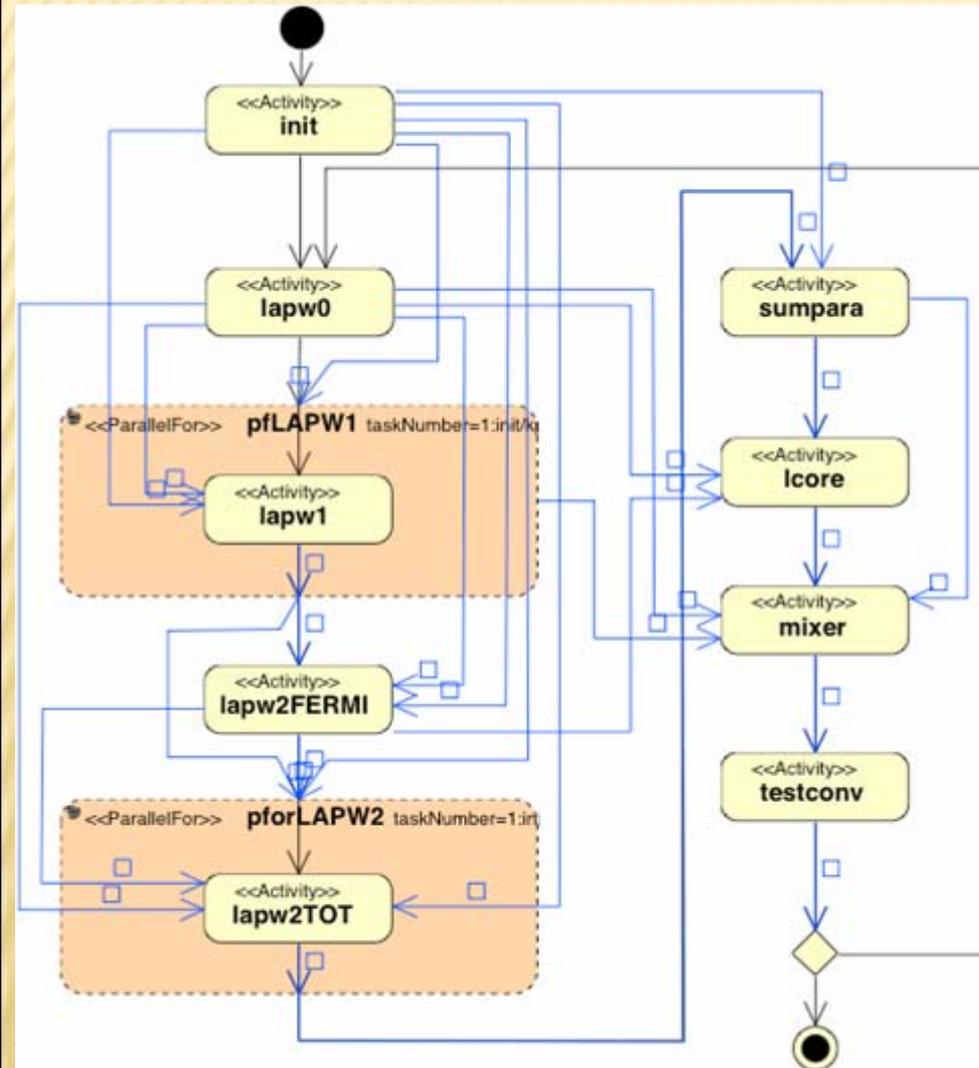


sub-workflow

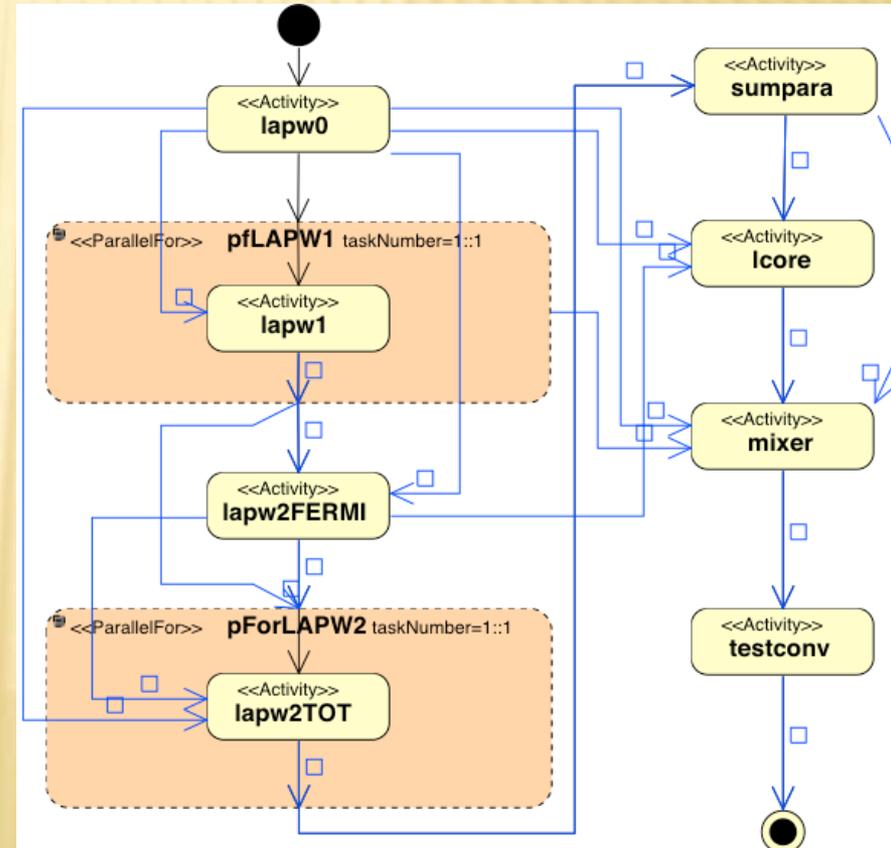
# CASE STUDY - WIEN2K



main workflow



single workflow



sub-workflow

# SUMMARY

- ✘ A framework for workflow sharing and reuse in the ASKALON Grid environment
  - + AWHE
  - + Extension of AGWL
    - ✘ simple and consistent way for reusing workflows, supporting workflow components, sub-workflows, and entire workflows
  - + Detection of incorrect recursive sub-workflow invocation
- ✘ The implementation and two case studies from the meteorology domain and the material science domain are presented

# THANK YOU

- × For more information:
  - + **ASKALON**: <http://www.askalon.org>
  - + **AGWL**: <http://www.askalon.org/agwl>

Thank You