Tutorial on Workflows

Eva Sciacca

eva.sciacca@oact.inaf.it

VisIVO Team – Team Leader Ugo Becciani

INAF- Astrophysical Observatory of Catania
Outline

- Background
  - Workflow Lifecycle
- Workflow Submission: Basic Steps
- Hands-On Session
- Supplementary Material
Outline

- **Background**
  - Workflow Lifecycle
- **Workflow Submission: Basic Steps**
- **Hands-On Session**
- **Supplementary Material**
Concrete workflow defines the semantic of the workflow execution. Algorithms describe job interiors and may be defined by binaries, by service calls or by references to other workflows. Resourse references may define the places where the jobs run and/or the way to find these places.

Repository stores developed items in a compressed form.

Application is a tested semantically defined Concrete workflow together with the definitions of its eventual embedded Workflows. Only input files, command line arguments and the destination of the submission may be left to the end user to define.

Project is a Concrete Workflow together with the definitions of each referenced classes.

Workflow part is either a Graph, a Concrete Workflow, a Workflow Instance or a Template.

Inputs define the input files elaborated by the Jobs. Inputs may be extended by job running conditions and by multiplication factor indicating a set of file to elaborate in subsequent “PS” Job submissions.
User Activities

- **Graph**
  - Jobs, Edges, Ports

- **Concrete Workflow**
  - Algorithms, Resource references, Inputs

- **Template**
  - Constraints, Comments, Form Generators

- **Workflow Instance**
  - Running state, Outputs

- **Repository Item**
  - Application OR Project OR Workflow part (G,T,CW)

Legend:
- **a** and **b** as argument
- **c** is created using **a** and **b** as argument
1. Create and edit a Graph of a workflow

2. Create the WF, and define the semantics, file association and destination by Configure

3. Submit the Concrete Workflow to observe its state and fetch its result

4. For reusability Template can be made from a Workflow by fixing some of its features

5. Template can be used as an alternative way to define a Concrete Workflow

6. A new CW can be defined by matching a Graph and a CW

7. Tested WF can be exported to end user
Outline

- Background
  - Workflow Lifecycle
- Workflow Submission: Basic Steps
- Hands-On Session
- Supplementary Material
Workflow Submission: Basic Steps

- Create Graph (or reuse an existing one)
- Create Concrete Workflow from Graph
- Setup Concrete Workflow (ports, jobs)
- Submit and check the running Workflow
Graph Editor - Graph Creation

There are two ways to create a Workflow Graph (WfG):

- Opening the Graph Editor (with the proper button) in the portlet Workflow/Graph
- Clone an existing one (Saving the actual Graph with a new name)
Graph Editor - Graph Creation

New Job
New Port
Right Click
Port Property
Select Output
Port name can be changed
Comment can be inserted
Close by OK
Press mouse...
, and drag to Input Port
New Port
New Job
Creating the Concrete Workflow

Create Concrete Workflow from Graph, from template, or from another workflow

- give a name to the Concrete Workflow
- give notes
Creating the Concrete Workflow

Create a WF inheriting parameters fixed in the selected Template

Create an “empty” WF using only a Graph selectable from list

Create a WF copying the parameters of an existing Workflow selectable from list

Confirmation button

Name of the new Workflow to be configured

Free text filed for the creator of the new Concrete Workflow
Setup Concrete Workflow

- Setup all the job properties
  - Execution model: Workflow, Service, Binary
  - Type: gLITE, ARC, GEMLCA, GT-4, GT-2, Local
  - Computing resource
  - Type of binary: SEQ, MPI, Java
  - Number of MPI Nodes
  - Executable
  - Additional parameters
- Setup port properties
Workflow Submission

A workflow can be submitted by the following ways:

1) Interactively started by the user hitting the button **Submit** belonging to the given concrete workflow on the portlet *Workflow/Concrete*.

2) Started by a –*crontab* like - predefined time schedule. The corresponding timetable can be set in the portlet *Workflow/Timing*

3) Started by an external event. The corresponding event to be waited for and the name of the Workflow can be defined in the portlet *Workflow/Remoting*
Workflow Submission

Step 1
The workflow is selected by button “Submit”

Step 2
The submission can be confirmed or refused after the optional filling of a free description field identifying Workflow Instance for the user.
1. If all state counters are 0 then there is no Instance of the given Workflow.
2. In Column “Error” the number instances being in states “Error” and “Aborted” are summed.
3. Instances in state “Suspended” are displayed according their preceding states.
Upload Overview

The user can upload a previously downloaded Concrete Workflow from the Client Machine to the Portal Server. To avoid name collisions the user has the possibility to rename:

- the workflow
- the graph of the workflow
- the eventual Template belonging to the workflow

The operation collaborates with the Upload operations of Portlet Workflow/Storage accepting the same way of encoding of Workflows.
**Upload Implementation**

**Step 1**
Select the compressed file in the client machine containing the requested Workflow.

**Step 2** (option)
Check the kind of name(s) you want to redefine.

**Step 3** (If Step 2 performed):
Enter the new name(s) which will not collide.

**Step 4**
Confirm the operation.
Storage - Overview

A Concrete Workflow and its eventual instances can be downloaded from the Portal Server to the client Machine.

- The storage can serve
  - a subsequent Upload operation,
  - recording the work done, or
  - access to the results of the calculations.
- The actual file transfer is prepared by compressing the needed data
- The Instances and the Outputs of Instances can be downloaded separately.
- The download of Instances includes
  - the download of the generator Concrete Workflow
  - the outputs
  - the messages resulting from the eventual user jobs and the messages resulting from the runs.

Note: In the case of Instances all produced output is booked – and can be downloaded. However – at present - only the actual Input is stored, therefore – if the user has changed the input between two successful Workflow Submissions it is not automatically assured, that the user can fetch the input of the former run.
Storage - Implementation

Information about the quota of the user allotted storage capacity in the Portal server

Columns of individual instances, please note, that outputs can be downloaded separately

To access instance
Outline

- Background
  - Workflow Lifecycle
- Workflow Submission: Basic Steps
- Hands-On Session
- Supplementary Material
Hands On Session

- Open VisIVO Gateway Home Page at: http://visivo.oact.inaf.it:8080

- Login with the user name ad password.

- Download the needed files: https://dl.dropboxusercontent.com/u/45733492/VisIVO_Demo_workflows.zip
Hands On Session

- Navigate to «Advanced Tools» and then Graph editor
Hands On Session

Example 1) FirstWF
Hands On Session

- The **FirstWF** is the first very simple workflow.
- It consists of one job script:
  
  ```sh
  #!/bin/sh
  echo "Hello World" > out
  ```
  
- Creating an output file named «out».

Note: input and output files of job executables MUST be named as the I/O Job ports.
Hands On Session

- Create a graph clicking on «Graph Editor» and save as FirstWF
Hands On Session

- Create the Concrete from «Create Concrete»
- Add name «FirstConcrete»
- Click «OK»
Hands On Session

- Submit the concrete workflow from «Run Concrete»
- Click on «Configure».
Hands On Session

- Click on the job yellow box
- Set Resource Type: local - Grid: DCI-Bridge-host
- Set executable file from: VisIVODemo_workflows/FirstWF/FirstJob/execute.bin
- Save it
Hands On Session

- Click on «Submit»
- Click on «Details» to inspect the submission.
Hands On Session

- Download the result.
- Click «View finished»
Hands On Session

Example 2) PSWF
Hands On Session

• The **PSWF** is a simple example of «Parameter Sweep» workflow.
• The **PS** applications are typically used when the same processing is executed on many different input sets.
• A **Generator** job produces more than one output data elements associated to output port(s). The code is expected to produce files with the naming convention `<prefix name>_<index>`, where `<index>`-s must be subsequent integers starting from zero.
• A **Collector** job is used to collect several files at one (or more) distinguished („collector”) input port(s) and then process them collectively within a single step.
Hands On Session

- Create a graph clicking on «Graph Editor» and save as PSWF

![Graph Diagram]
Hands On Session

- Create the Concrete from «Create Concrete»
- Add name «PSWF»
- Click «OK»
Hands On Session

- Configure the concrete workflow from «Run Concrete»
- Click on «Configure».
Hands On Session

- Click on the job yellow box
- Set Resource Type: local - Grid: DCI-Bridge-host
- Set executable files from: VisIVODemo_workflows/PSWF/.../execute.bin
- Save it
Hands On Session

- Set output port of Generator job as «Generator»
- Set input port of Collector job to wait all inputs
- Save it
Hands On Session

- Click on «Submit»
- Click on «Details» to inspect the submission.
Hands On Session

- Download the result.
- Click «View finished»
Hands On Session

Example 3) VisIVODemo
Hands On Session

- The VisIVODemo is a more complex workflow.
- It consists of 4 job scripts employing VisIVO tools
- Create several output files preforming some statistics and visualization of the input dataset.

Note: input and output files of job executables MUST be named as the I/O Job ports.
Hands On Session

- Create a new graph clicking on «Graph Editor» and save:

  0: inputFile
  1: paramFile
  2: VBT.bin
  3: VBT.bin.head

  0: VBT.bin
  1: VBT.bin.head
  2: statistic
  3: histogram
  4: paramFile

  0: VBT.bin.head
  1: histogram
  2: hist.zip

  0: VBT.bin
  1: VBT.bin.head
  2: paramFile
  3: image0.png
  4: image1.png
  5: image2.png
  6: image3.png
Hands On Session

- Create the Concrete from «Create Concrete»
- Add name «VisIVOConcrete»
- Click «OK»
Hands On Session

- Submit the concrete workflow from «Run Concrete»
- Click on «Configure»
- Set Resource Type: local - Grid: DCI-Bridge-host
- Use executable files and input files from: VisIVODemo_workflows/VisIVODemo
Hands On Session

- Click on «Submit»
- Click on «Details» to inspect the submission.
Hands On Session

- Click on «View finished» to download the results.
Hands On Session

- **Export** the workflow to the local repository
- From «Concrete» click «Export» button then select Local Repository. Export as application and add a description.
Hands On Session

- **Import** the workflow from the local repository
- From «Import» select Local repository
Hands On Session

- The local repository allows the workflow sharing among the gateway users.
- The SHIWA repository allows sharing among different gateways and among different workflow systems.
- Please visit: [http://shiwa-repo.cpc.wmin.ac.uk](http://shiwa-repo.cpc.wmin.ac.uk)
- and Browse among the available workflows
Outline

- Background
  - Workflow Lifecycle
- Workflow Submission: Basic Steps
- Hands-On Session
- Supplementary Material
Supplementary Material

- SCI-BUS Project: http://www.sci-bus.eu
- Er-Flow Project: http://www.erflow.eu
- WSPgrade/gUse: http://www.guse.hu
- VisIVO Science Gateway: http://visivo.oact.inaf.it:8080
- SHIWA Repository: http://shiwa-repo.cpc.wmin.ac.uk