Pipelines = Java based Engine + XML workflows + [Cluster] + Executables + XNAT

Mohana Ramaratnam

President, NRG India

XNAT Developer

What will you get out of this session?

Learn what Pipeline Engine can do for you and how to start using it today!

Agenda

- Need for Pipeline Engine
- Pipeline Engine Features
- Creating a pipeline
- Invoking pipelines
- Adding pipelines to a XNAT site
- Demo
- Compute Cluster
- Troubleshooting
- Q&A

Need for Pipeline Engine

- Project based workflows
- Status tracking as the workflow progresses
- Email notifications
- Pipelets for modular design

Pipeline Engine Features

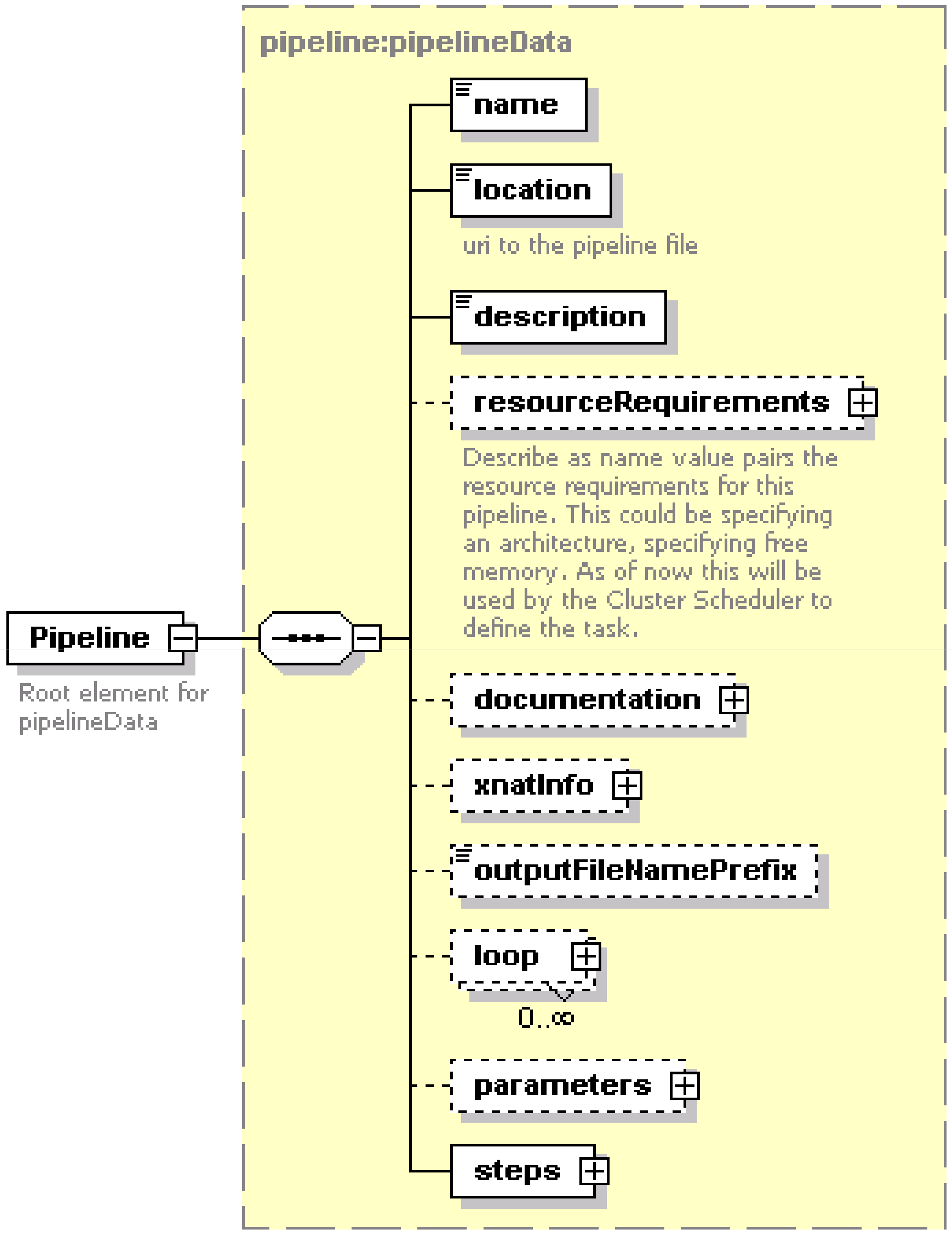
- XML based workflow – Pipeline Descriptor
- Pause and re-start from a specified step
- Exit status of each step is monitored
- Input Parameters can be specified inline, through command prompt and through a parameter file
- Support for using XPATH and custom functions to evaluate input parameters
- Email notifications on completion and failure

Pipeline Engine Installation

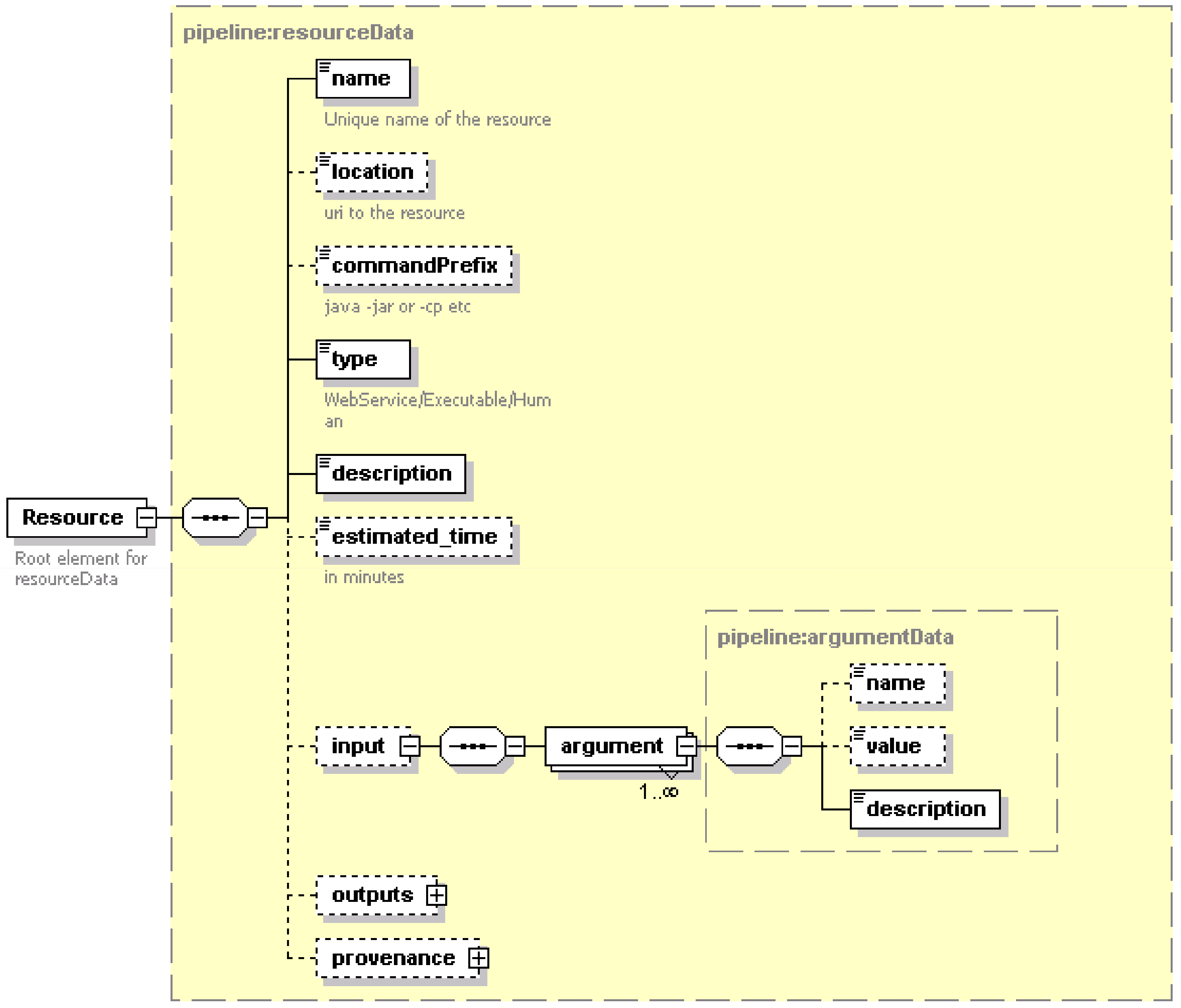
- Folder structure
  - bin
  - catalog
  - doc
  - lib
  - logs
  - *-tools
  - sample_pipelines
  - pipeline.config

Creating a pipeline – Part I

- Two step process
  - Create resource descriptors for each of the executables invoked in a step.
  - Create pipeline descriptor (workflow)



pipeline:pipelineData
Pipeline
Root element for pipelineData
name
location
uri to the pipeline file
description
resourceRequirements
Describe as name value pairs the resource requirements for this pipeline. This could be specifying an architecture, specifying free memory. As of now this will be used by the Cluster Scheduler to define the task.
documentation
xnatInfo
outputFileNamePrefix
loop
0..∞
parameters
steps


pipeline:resourceData
name
Unique name of the resource
location
uri to the resource
commandPrefix
java -jar or -cp etc
type
WebService/Executable/Human
description
estimated_time
in minutes
Resource
Root element for resourceData
input
argument
1..∞
pipeline:argumentData
name
value
description
outputs
provenance

Attributes of a Step

- Precondition
- Workdirectory
- GotoStepId
- AwaitApprovalToProceed
- ContinueOnFailure

Attributes of a step - resource

- ssh2Host
- ssh2Password
- ssh2Identity
- ssh2User
- pipeId

Creating a pipeline – Part II

- On your compute machine(s)
  - Install the executable(s)
- In your XNAT project
  - Create Action class
  - Create Screen class
  - Create Velocity template file for custom interface to launch and setup a pipeline
- Create pipeline xml
- Create resource xml(s)

Transfer Pipeline Explained

- Copy data from prearchive to archive
- Generate snapshots of the images
- Move prearchive folder to cache
- Notify user

RESOURCE DESCRIPTOR

```
<?xml version="1.0" encoding="UTF-8"?>
<pip:Resource xmlns:pip="http://nrg.wustl.edu/pipeline">
        <pip:name>AntCopy</pip:name>
        <pip:location>PIPELINE_DIR_PATH/ant-tools/bin</pip:location>
        <pip:type>Executable</pip:type>
        <pip:description>Uses Ant to copy files </pip:description>
        <pip:input>
                <pip:argument id="source">
                        <pip:name>src</pip:name>
                        <pip:description>Source directory to copy from</pip:description>
                </pip:argument>
                <pip:argument id="destination">
                        <pip:name>dest</pip:name>
                        <pip:description>Destination directory to copy to</pip:description>
                </pip:argument>
                <pip:argument id="overwrite">
                        <pip:name>overwrite</pip:name>
                        <pip:description>Flag to indicate if copy should overwrite
                                files</pip:description>
                </pip:argument>
        </pip:input>
</pip:Resource>
```

```
 <step id="1" description="Copy files from prearc to archive">
          <resource name="AntCopy" location="ant-tools">
          <argument id="source">
          <value>^/Pipeline/parameters/parameter[name='sourceDir']/val
ues/unique/text()^</value>
          </argument>
          <argument id="destination">
          <value>^/Pipeline/parameters/parameter[name='destinationDir'
]/values/unique/text()^</value>
          </argument>
          <argument id="overwrite"/>
          </resource>
     </step>
```

```
PIPELINE_HOME/ant-tools/bin/AntCopy –src  PATH1 –dest  PATH2 -overwrite
```

Effects of running Transfer pipeline

- Session data in Archive – Step 1 - AntCopy
- Snapshots for each scan – Step 2 - WebBasedQCImageCreator
- Move to cache – Step 3
- Notify user – Step 4
- Log files: error, stdout and a resolved representation of the pipeline <CACHE>/logs/transfer/<PROJECT_ID>/<SESSION_LABEL>

Launching a pipeline

- Two modes to launch
  - Interact with XNAT and update status

    Invoke XnatPipelineLauncher
  - Don’t interact with XNAT just execute

    Invoke PipelineRunner

  Specify:

  -pipeline : Path to pipeline descriptor

  -parameter <name>=<value1,value2,…,valueN>

  ……..

Adding pipeline to XNAT

- Site Admin makes site wide pipelines available
- Project owners add relevant pipelines to a project.
- Project members launch pipelines for a datatype

Demo

- Create a pipeline to generate NIFTI versions from DICOM files and insert the file at the SCAN level
- Create resource descriptor
- Add pipeline to site
- Setup pipeline for a project
- Invoke pipeline through Actions
- Invoke pipeline via a REST call

Cluster Support

- Pipeline engine ships with support for SGE using the DRMAA API
- A pipeline can specify resources for processing like machine architecture, free memory etc
- <PIPELINE_HOME>/bin/schedule – “overload” this shell script for your site/cluster

Invoking a pipeline via REST CALL

- REST/projects/{PROJECT_ID}/pipelines/{STEP_ID}/experiments/{EXPT_ID}
- Limitation
  - This is possible only if all input arguments are specified as XPATH

Features

- Custom XPATH extension functions
  - Create a static method
  - Create a jar which includes the method
  - Include the jar in the classpath of XnatPipelineLauncher
  - Set namespace eg.

xmlns:fileUtils=http://www.xnat.org/java/org.nrg.imagingtools.utils.FileUtils

```
<parameter>
        <name>nx</name>
        <values>

<unique>^fileUtils:getNXArgumentForUnpack4dfp(/Pipeline/parameters/parameter[name='nx_ny_catalog_file']/values/unique/text())^</unique>
        </values>
</parameter>
```

Monitoring site pipelines

- Get a summary of Site activity

  Administrator -> More options -> Summary
- Get a summary of Site workflows

  Administrator -> More options -> View All Workflows

Troubleshooting

- See

<TOMCAT_HOME>/webapps/<XNAT_PROJECT>/logs/application.log

<PIPELINE_HOME>/logs/

Q&A

THANK YOU!