SAND2019-7344C

The SAW Next Generation Workflow System

Ernest J. Friedman-Hill
ejfried@sandia.gov

SAND2019-2986 C

Motivation

Our Users

- Engineers and scientists
- Computer-literate
- Know scripting or programming
  - Not "programmers"
- Run lots of complex, in-house research codes
  - Experts with some, novices with others
- Need to follow best practices
  - Verification and validation

Motivation

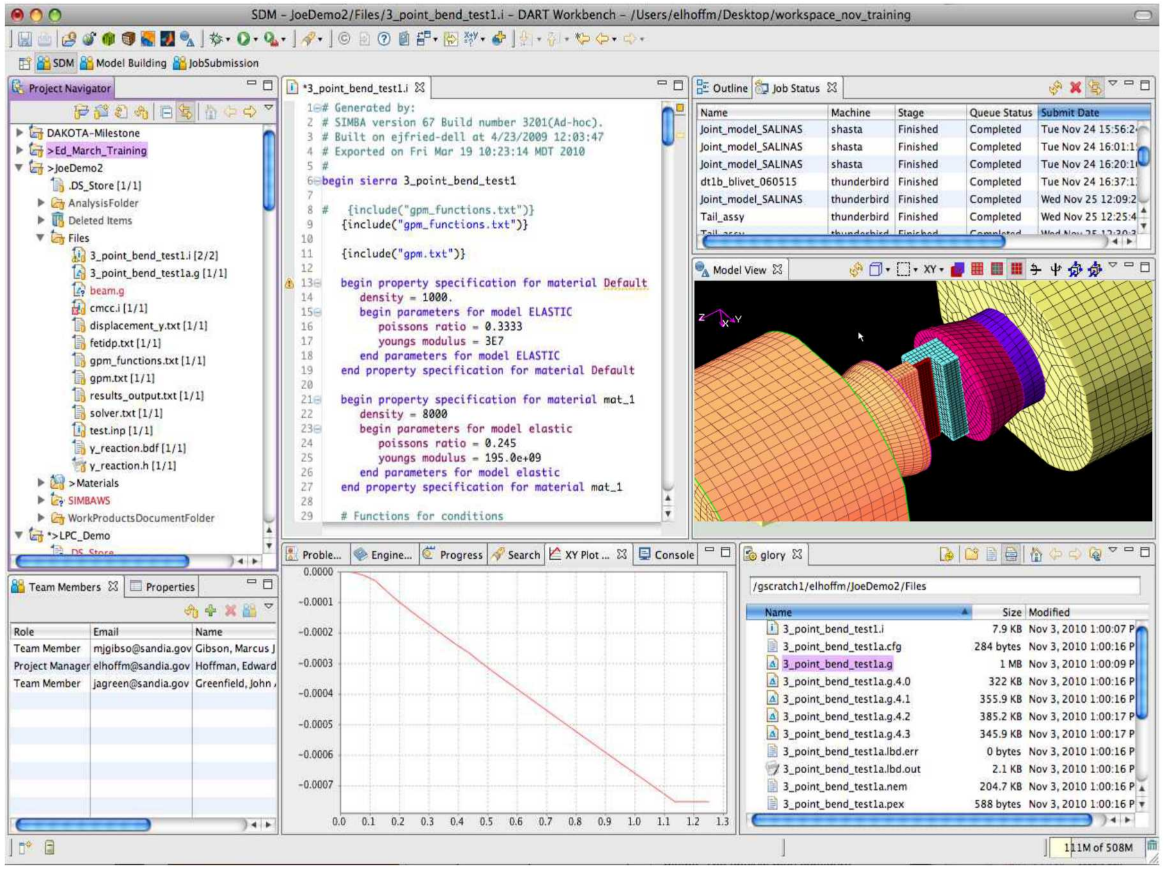
Environment

- Heterogeneous hardware
- Multiple, secure networks
- Need-to-know
- *Decentralized*
- Compartmentalized

The Sandia Analysis Workbench

An integrated environment for engineering analysis

- Graphical model building
- Job submission
- Distributed file management
- Scientific data management
- Requirements management



```
# Generated by:
# SIMBA version 67 Build number 3201(Ad-hoc).
# Built on ejfried-dell at 4/23/2009 12:03:47
# Exported on Fri Mar 19 10:23:14 MDT 2010
#
begin sierra 3_point_bend_test1

#   {include("gpm_functions.txt")}
   {include("gpm_functions.txt")}

   {include("gpm.txt")}

   begin property specification for material Default
      density = 1000.
      begin parameters for model ELASTIC
         poissons ratio = 0.3333
         youngs modulus = 3E7
      end parameters for model ELASTIC
   end property specification for material Default

   begin property specification for material mat_1
      density = 8000
      begin parameters for model elastic
         poissons ratio = 0.245
         youngs modulus = 195.0e+09
      end parameters for model elastic
   end property specification for material mat_1

   # Functions for conditions
```

Name	Machine	Stage	Queue Status	Submit Date
Joint_model_SALINAS	shasta	Finished	Completed	Tue Nov 24 15:56:2
Joint_model_SALINAS	shasta	Finished	Completed	Tue Nov 24 16:01:1
Joint_model_SALINAS	shasta	Finished	Completed	Tue Nov 24 16:20:1
dt1b_blivet_060515	thunderbird	Finished	Completed	Tue Nov 24 16:37:1
Joint_model_SALINAS	thunderbird	Finished	Completed	Wed Nov 25 12:09:2
Tail_assy	thunderbird	Finished	Completed	Wed Nov 25 12:25:4

Role	Email	Name
Team Member	mjgibso@sandia.gov	Gibson, Marcus J
Project Manager	elhoffm@sandia.gov	Hoffman, Edward
Team Member	jagreen@sandia.gov	Greenfield, John /

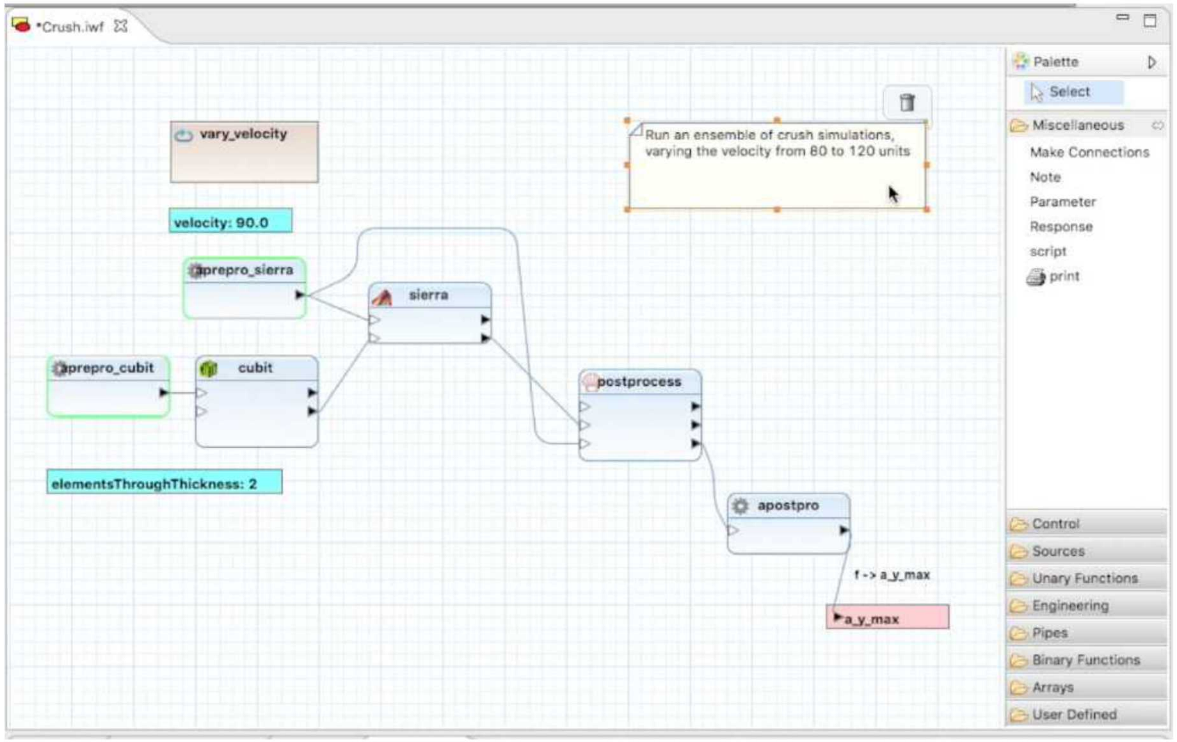
/gscratch1/elhoffm/JoeDemo2/Files

Name	Size	Modified
3_point_bend_test1.i	7.9 KB	Nov 3, 2010 1:00:07 P
3_point_bend_test1a.cfg	284 bytes	Nov 3, 2010 1:00:16 P
3_point_bend_test1a.g	1 MB	Nov 3, 2010 1:00:09 P
3_point_bend_test1a.g.4.0	322 KB	Nov 3, 2010 1:00:16 P
3_point_bend_test1a.g.4.1	355.9 KB	Nov 3, 2010 1:00:16 P
3_point_bend_test1a.g.4.2	385.2 KB	Nov 3, 2010 1:00:17 P
3_point_bend_test1a.g.4.3	345.9 KB	Nov 3, 2010 1:00:17 P
3_point_bend_test1a.lbd.err	0 bytes	Nov 3, 2010 1:00:16 P
3_point_bend_test1a.lbd.out	2.1 KB	Nov 3, 2010 1:00:16 P
3_point_bend_test1a.nem	204.7 KB	Nov 3, 2010 1:00:16 P
3_point_bend_test1a.pex	588 bytes	Nov 3, 2010 1:00:16 P

SAW Next Gen Workflow (NGW)

Automated workflow that

- is integrated with everything else in SAW
- works with with external codes
- runs inside or outside SAW
- is intuitive



Run an ensemble of crush simulations, varying the velocity from 80 to 120 units

NGW System Requirements

Low cognitive load
- Users can think in the domain, not about the workflow system
- You don't *program* workflow, you *build* it

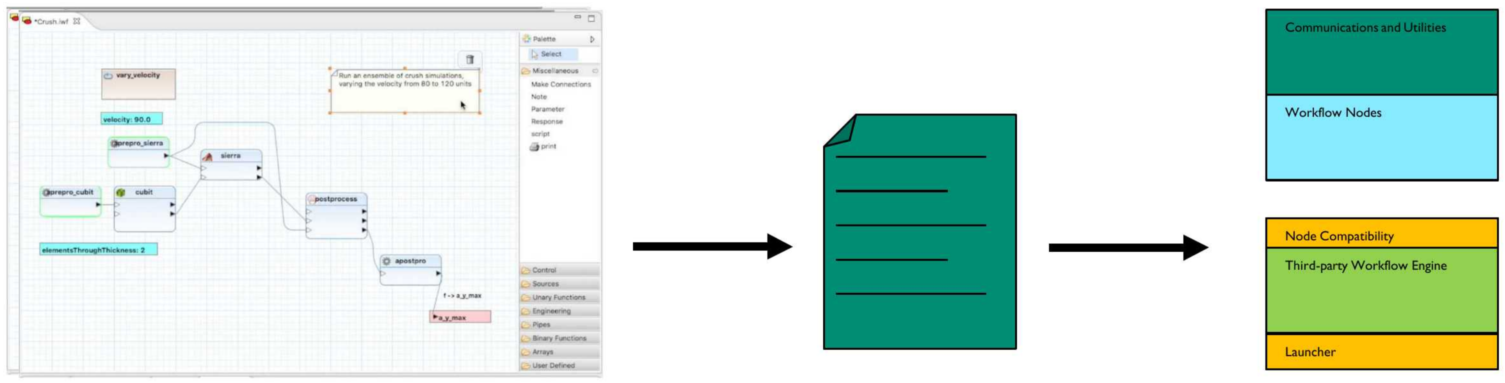
Transparent and "escapable"
- It's always obvious what the system is doing
- It's always possible for user to customize behavior

Flexible system architecture
- Everything can execute anywhere

Zero-install
- *Can't assume root access on runtime system*
- *Can't install a persistent server*

Workflow engine independence
- Can be implemented on multiple products

Next-Gen Workflow Architecture



1. Graphical Editor 2. Workflow File 3. Layered Runtime

Graphical editor and runtime are independent and can be used separately in time and space. Editor runs on desktop, while runtime can execute anywhere from desktop to HPC clusters

NGW Definitions

A workflow is an executable graph of connected components

A component or "node" represents an activity

- Executing an external code
- Extracting a column of numbers from a table
- Components are user-configurable
- Some very specific, some very generic

Connections represent data interchange and control flow

- A number or a path to a file
- An identifier for a shared memory segment
- *Connections are themselves configurable*

The home directory is where you build a workflow

The work directory is where you execute a workflow

- They may be coincident, or separated in time and/or space

NGW Graphical Editor

Based on Eclipse IDE framework
- Graphiti / EMF
- Borrows some ideas from Triquetrum/ESWG

Data-driven
- XML file to describe available component types

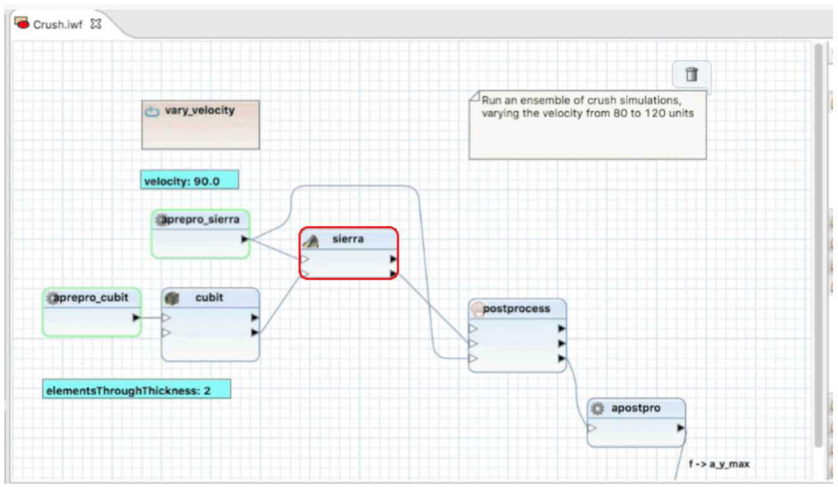
Easy to extend
- Can add component types
- Can add component-specific editing
- Can add special behaviors for embedded execution
- Programmers work with editor’s EMF model

NGW Graphical Editor

Eclipse integration

Ease of use

- Drag file from project, creates appropriate component to use that file
- File-type-specific editors automatically invoked

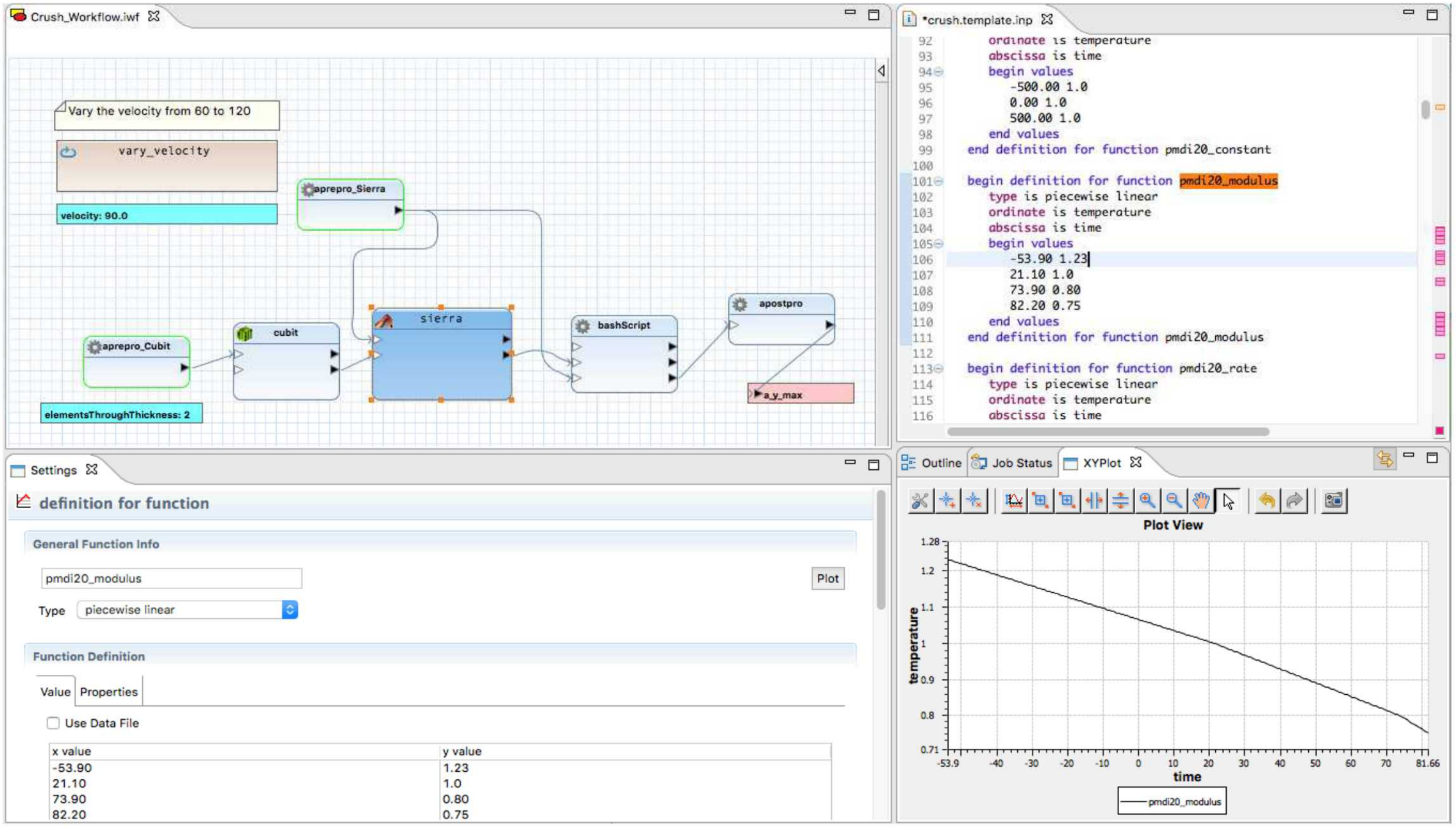


Run an ensemble of crush simulations, varying the velocity from 80 to 120 units

Integrates with other tools in SAW

NGW Graphical Editor

Edit model components naturally using familiar tools



```
      ordinate is temperature
      abscissa is time
      begin values
         -500.00 1.0
         0.00 1.0
         500.00 1.0
      end values
   end definition for function pmdi20_constant

   begin definition for function pmdi20_modulus
      type is piecewise linear
      ordinate is temperature
      abscissa is time
      begin values
         -53.90 1.23
         21.10 1.0
         73.90 0.80
         82.20 0.75
      end values
   end definition for function pmdi20_modulus

   begin definition for function pmdi20_rate
      type is piecewise linear
      ordinate is temperature
      abscissa is time
```

Settings

definition for function

General Function Info

pmdi20_modulus

Type: piecewise linear

Function Definition

Value | Properties

☐ Use Data File

x value	y value
-53.90	1.23
21.10	1.0
73.90	0.80
82.20	0.75

Workflow File Format

Based on EMF native format (XML)

Simple file format contains
- … component configuration
- … connectivity
- … optional graphical (layout) information

Does *not* contain component implementations
- Different runtime implementations can execute the same workflow definition

Does *not* contain rendering information
- Appearance of components easy to change

Can be generated or consumed by any code
- i.e., workflows can be defined by code or scripts
- Eclipse parser library not required
- No Eclipse code (EMF) in server

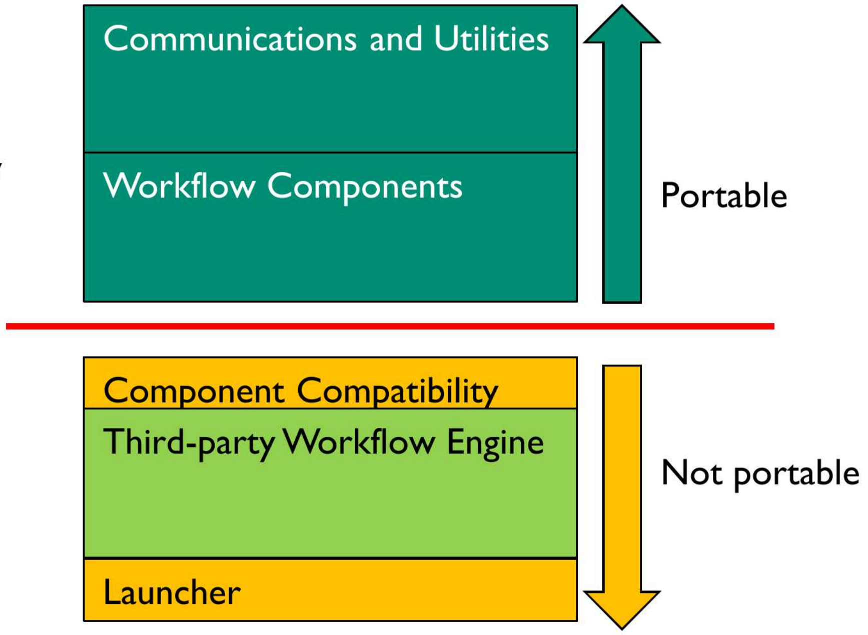
Workflow File Format

```
<gov.sandia.dart.workflow.domain:WFNode name="print" properties="/4 /5 /6 /7”
     inputPorts="/2" outputPorts="/3" start="true" type="print" label="print"/>
<gov.sandia.dart.workflow.domain:InputPort name="x" type="default" node="/1"/>
<gov.sandia.dart.workflow.domain:OutputPort name="f" type="default" node="/1"/>
<gov.sandia.dart.workflow.domain:Property name="formatString" type="default" value="Hello, world!" node="/1"/>
<gov.sandia.dart.workflow.domain:Property name="async" type="boolean" value="false" node="/1"/>
<gov.sandia.dart.workflow.domain:Property name="privateWorkDir" type="boolean" value="false" node="/1"/>
<gov.sandia.dart.workflow.domain:Property name="clear private work directory”
      type="boolean" value="false" node="/1"/>
```

NGW Engine

Layered Runtime Architecture

- NGW Components are written in Java to a vendor-neutral API. The compatibility layer adapts to a third-party rule engine
- Variant component implementations can be provided for different host environments

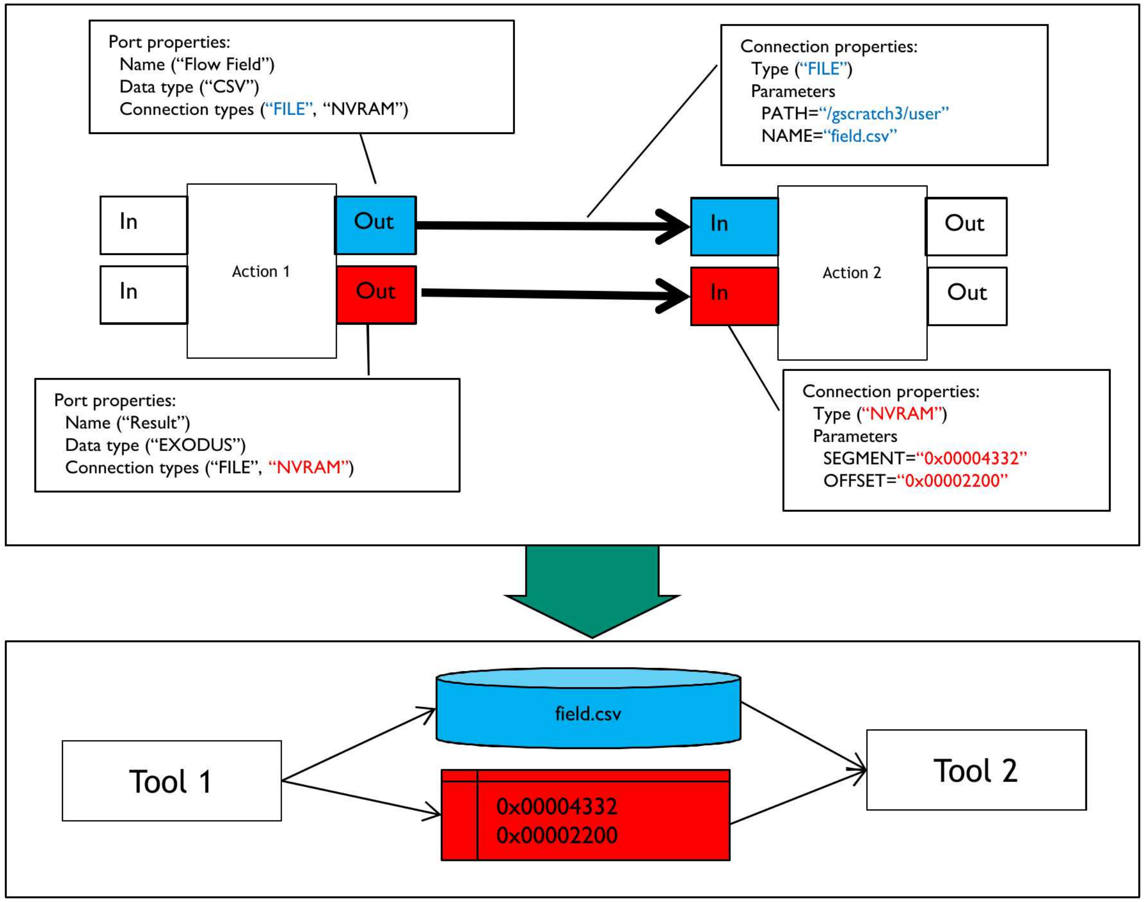


By using a thin compatibility layer, we can deploy on arbitrary third-party Java workflow engines with little effort

NGW Engine

Abstract Dataflow

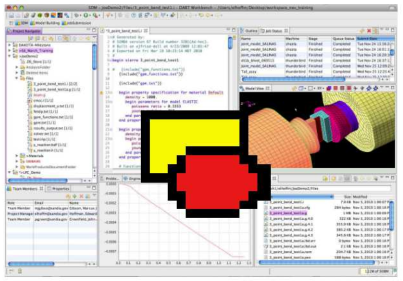
- Connections between components are configurable. Configuration options determine how data is exchanged at runtime
- Future work will involve adding support for more data exchange mechanisms
- In this notional diagram, the blue connection is configured to exchange data via a file, while the red is using a shared memory segment



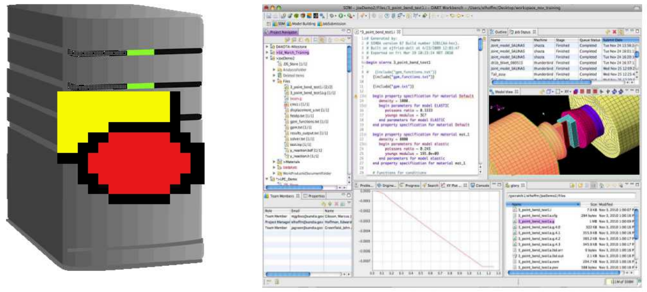
NGW Engine

Flexible Topology

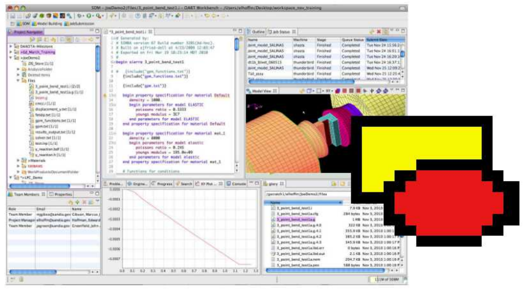
The same workflow can run in multiple environments. We can leverage our existing job submission and monitoring tools to implement this



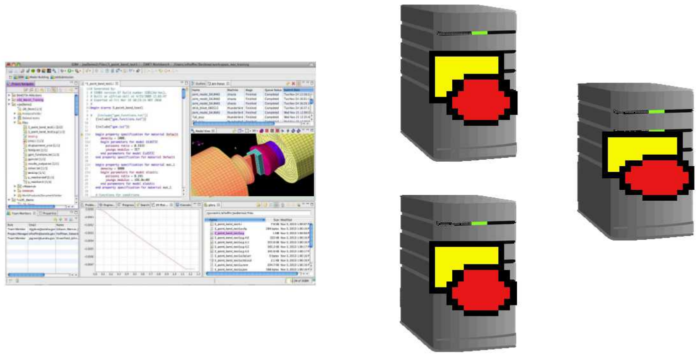
Interactive



Server



Batch

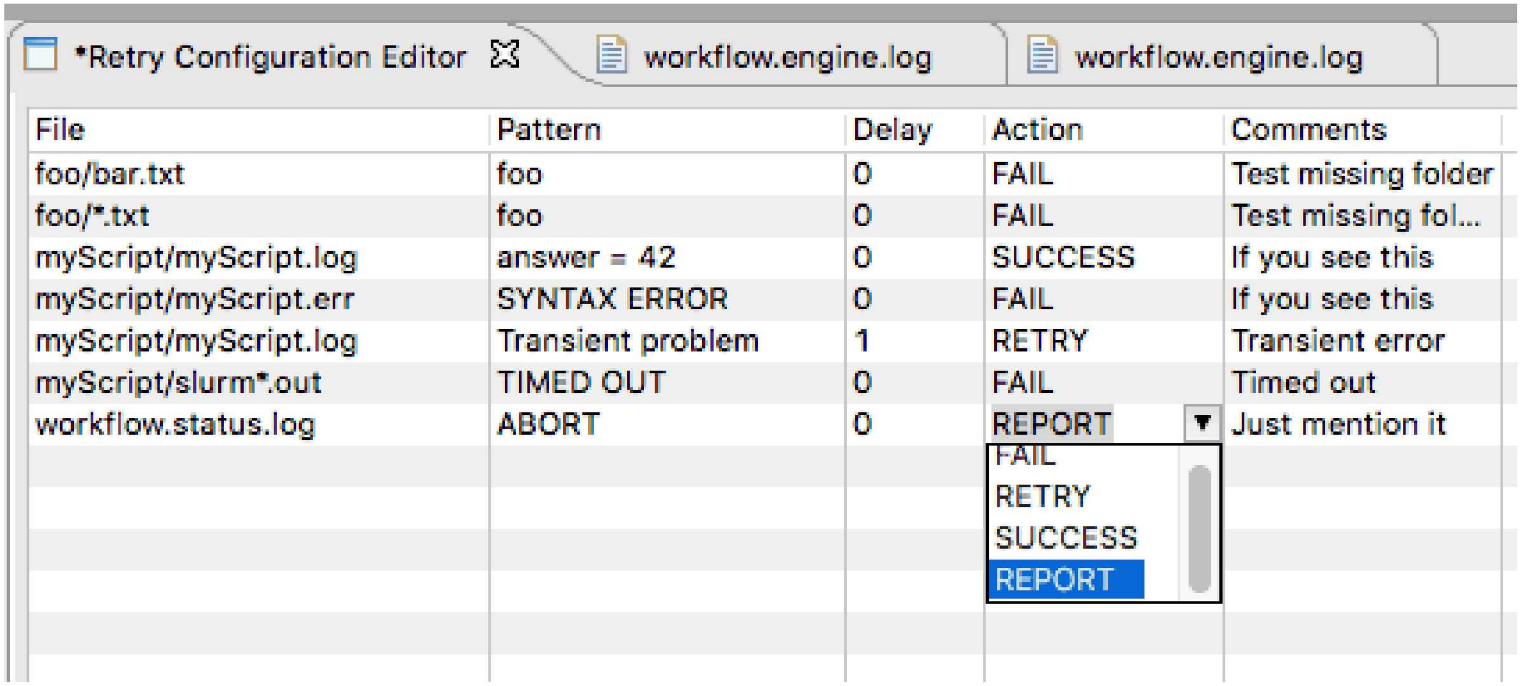


Distributed

NGW Engine

Automatic Failure Detection and Retry

"Foo Fighter" lets you detect and react to error conditions without scripting



File	Pattern	Delay	Action	Comments
foo/bar.txt	foo	0	FAIL	Test missing folder
foo/*.txt	foo	0	FAIL	Test missing fol...
myScript/myScript.log	answer = 42	0	SUCCESS	If you see this
myScript/myScript.err	SYNTAX ERROR	0	FAIL	If you see this
myScript/myScript.log	Transient problem	1	RETRY	Transient error
myScript/slurm*.out	TIMED OUT	0	FAIL	Timed out
workflow.status.log	ABORT	0	REPORT	Just mention it

[Sprint Wiki Page]

NGW Engine

Automatic Failure Detection and Retry

Workflow log excerpt showing AFDR aborting a run

```
INFO  (17:13:47,181) - Retrying as per retry strategy
INFO  (17:13:47,181) - About to run subworkflow, attempt 3
INFO  (17:13:47,202) - Sample DEFAULT, Status for node  nestedWorkflow: Running
node myScript in sample DEFAULT
INFO  (17:13:48,256) - Subworkflow ran OK.
INFO  (17:13:48,256) - Checking test Probe [file=/Users/ejfried.../myScript.err,
pattern=SYNTAX ERROR, delay=0, action=FAIL]
INFO  (17:13:48,256) -    No match
INFO  (17:13:48,257) - Checking test Probe [file=/Users/ejfried...pt/slurm*.out,
pattern=TIMED OUT, delay=0, action=FAIL]
INFO  (17:13:48,257) -    Match: TIMED OUT
INFO  (17:13:48,257) - Aborting as per retry strategy
INFO  (17:13:48,258) - Sample DEFAULT, Aborting node nestedWorkflow: Execution
failed.
ERROR (17:13:48,258) - Workflow terminated with error
```

Sprint Wiki Page

Extending NGW

"Wrapping" an external code for runtime use

- *No programming required*
- Can be done in the workflow editor
- Configure a script node or external process node
- Save to palette for personal use
- Export to JAR file for sharing

Extending NGW

Custom images for nodes in editor

- Register icon with nodeIcon extension point
- Package as Eclipse plugin

Providing files and resources to workflow creator

- Register file with resourceContributor extension point
- Package as Eclipse plugin

Extending NGW

Defining node types for editor

- Requires Java programming
- Implement one method in IWorkflowEditorNodeTypeContributor

Custom rendering for nodes in editor

- Requires Java programming
- Extend AbstractGARenderer and implement two methods
- Register with nodeRenderer extension point
- Package as Eclipse plugin

Extending NGW

Creating a more complex runtime node

- Requires Java programming
- Extend SAWCustomNode
- Implement (at minimum) one method
- Package as JAR file, in Eclipse plugin, or both

Extending NGW

Custom editors for specific node types

- Requires Java programming
- Implement ISettingsEditor and implement several methods
- Register with nodeTypeEditor extension point
- Package as Eclipse plugin

Impact

- Best Practices and New Capabilities
  - Workflows capture the best way to complete a task
- Reproducibility
  - It’s easy to make sure you’re performing the same steps
- Reliability
  - Errors handled robustly
- Reusability
  - No reinventing the wheel
- Team communication
  - Intrinsically documented processes
- Archival Information
  - Automatically preserve what was done and why

Future work

Façades

Web client

Alternate servers

- Python
- C++
- Other Java?

In situ workflow