Modernizing Simulation Input Generation and Post-Simulation Data Visualization with Eclipse ICE

Alex McCaskey  
Research Staff  
Oak Ridge National Laboratory  
mccaskeyaj@ornl.gov  
@amccaskey2223

Taylor Patterson  
Research Associate  
Oak Ridge National Laboratory  
pattersontc@ornl.gov  
@TCPatt

Jay Jay Billings  
Research Staff, Project Lead  
Oak Ridge National Laboratory  
billingsjj@ornl.gov  
@jayjaybillings  
+jayjaybillings

EclipseCon North America 2015  
Burlingame, CA, USA  
2015-03-12
Outline

• ICE and the Standard Model of Scientific Computing
• Streamlining XML-based scientific code integration in ICE
• Visualization through VisIt integration

Additional Resources:

GitHub: http://github.com/eclipse/ice
Eclipse Wiki: http://wiki.eclipse.org/ICE
YouTube: http://youtube.com/user/jayjaybillings
**Standard Model of Scientific Computing**

All users must do these things...

<table>
<thead>
<tr>
<th>Define the Problem</th>
<th>Run the Simulator</th>
<th>Analyze Output</th>
<th>Archive Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Write an input file in a format reminiscent of a dead language</td>
<td>Manually launch jobs on impressively terrifying machines</td>
<td>Analyze simulation output in its most raw and unlimited form</td>
<td>Store data... somewhere!</td>
</tr>
</tbody>
</table>

*Super-users think these are easy tasks, but most users are overwhelmed!*
A cooler model of Scientific Computing

It would be better to have a computer program handle all of that...

Most of the stuff we need to do can be encapsulated for ease of use and/or automated entirely with improvements.
ICE in Action

An early video of the triso results in ICE's visualization perspective (thus the bugs!)

Fully interactive 3D support in VisIt

JMonkeyEngine for Mesh and Geometry
ICE in Action

3D Model of a Nuclear Plant (TMI)
ICE in Action

SNS Phonon Scattering Data
ICE in Action

Scientific Software Development in ICE
Today’s Focus

Streamlining the implementation of input generation tools for developers

VisIt provides the tools for generating 2D and 3D visualizations
ICE Architecture: How we do it...

The platform is easily modified:
- Developers write *plugins* for their scientific codes
- We handle data representations, processing, presentation, etc...

Plugin code goes here!

Plugins are:
- Dynamic Services - Completely reusable components!
- “Item” Subclasses - Most of the work is already done by the platform
- Self-contained, business logic - **ONLY** your code, not UI, etc.
- Tools - Reusable components, tools, or things other
ICE Architecture: Data Structures

ICE parses each Form and presents each component graphically to the user
- TreeComposite, DataComponent UI View

Each Item provides a hierarchical Form
- Each Form contains Components
- Tree Pattern

```
- DataComponent
- MatrixComponent
- TreeComposite
```

```
- «interface» Component
  - addComponent ( child : Component )
  - removeComponent ( childId : int )
  - GetComponent ( childId : int ) : Component
  - getNumberofComponents ( ) : int
  - getComponents ( ) : Component[*]
```
XML and Scientific Community

Quantum Computing

XML widely used in scientific community:
- Standardized
- Readers/Writers
- Hierarchical

Neutron Scattering

Advanced Batteries
For each of these, and others, we have to write a new ICE Item plugin…
We want something like this...

How can we leverage existing Eclipse tools to streamline this???

ICE XML Support

Battery XML schema and/or schema instance

Quantum Computing XML schema and/or schema instance

Neutron Scattering XML schema and/or schema instance
Eclipse Modeling Framework

- XML Schemas define input **models** for these scientific codes...
- The EMF already handles XML well
- Reflection API

**Key Ecore Components for ICE (simplified view for our purposes)**

- **EClass**
- **EReference**
- **EAttribute**
ICE Leveraging the EMF - High Level View

- **Goal**: Instead of writing N Item plugins for N XML-input based codes, write just 1.
- We can do this by leveraging the EMF and existing ICE data structures - TreeComposite.

**Diagram:**

1. **Use EMF to read in XML Schema**
   - Read-in schema with existing EMF XMLProcessor
   - Let EMF do the heavy lifting!

2. **Map EMF Ecore Tree to ICE TreeComposite**
   - Use EMF Reflection API to create ICE TreeComposite instances that correspond to EClass nodes in the Ecore Tree

3. **Use existing ICE tools to present tree to user**
   - Present with JFace TreeViewer with Properties Tab provider for data input
ICE Leveraging the EMF - Low Level View

```java
// Create a new XMLProcessor to be used in creating
// and persisting XML Resources
try {
    xmlProcessor = new XMLProcessor(URI.createFileURI(file
        .getAbsolutePath()));
} catch (SAXException e) {
    e.printStackTrace();
}

// Get the package containing the model
    .values().toArray()[0];

// Get the TreeIterator to walk over the elements
TreeIterator<EObject> tree = ePackage.eAllContents();
```

Creating the Ecore Tree - Easy with the tools provided by EMF!

Map to ICE tree structure

Specialized behavior for adding children and querying ability to add children
Results!

Sassena - Neutron Scattering Simulation for the ORNL Spallation Neutron Source.
Post-Simulation Data Visualization in ICE through VisIt
What is VisIt?

“… an open source, interactive, scalable, visualization, animation and analysis tool.”

“… a distributed, parallel visualization and graphical analysis tool for data defined on 2D and 3D meshes.”

“… originally developed… to visualize and analyze the results of terascale simulations… [but] has also proven to be well suited for visualizing smaller scale data…”

“… capable of visualizing data from over 120 different scientific data formats.”

Credit: Quotes and images from VisIt Development Team website (https://wci.llnl.gov/simulation/computer-codes/visit)
Why include VisIt in ICE?

Simulation requires 3D visualization

All-in-one approach

Familiar UI

Users of ICE also using VisIt
Features – Connections

Remote launch and rendering
Features – Connections

Single install capability
System Components

Eclipse ICE  VisIt Java Client  VisIt Application

User input

Maintain connection
Input as JSON

Rendered image
File information

Results as JSON
System Components

Eclipse ICE

VisIt Java Client

VisIt Application

Binary distribution

Source location

ICE executable

http://svn.code.sf.net/p/niceproject/code/trunk

https://github.com/visit-vis/visit_java_client.git

VisIt executable

https://wci.llnl.gov/simulation/computer-codes/visit/source

(contact us)

(no public repo)
ICE Components – Perspective

Modernizing Simulation Input Generation and Post-Simulation Data Visualization with Eclipse ICE
ICE Components – Perspective

Modernizing Simulation Input Generation and Post-Simulation Data Visualization with Eclipse ICE
ICE Components – Perspective

Modernizing Simulation Input Generation and Post-Simulation Data Visualization with Eclipse ICE
ICE Components – Viewers

![Visualization of a 3D model](image)

**DB: output-Battery_1.1.silo**

**Cycle: 1**

Pseudocolor

Var: Battery_/ThermaRFSP1

- 0.1383
- 0.06915
- 0.03467
- 0.000

Max: 0.1383
Min: 0.000

Thu Feb 26 16:56:43 2015
ICE Components – Viewers

Data provided by ICE

ViewPart subclasses with JFace TreeViewers

Data provided by VisIt
ICE Components – Viewers

Data provided by VisIt

Subclass of CheckedTreeSelectionDialog
ICE Components – Editor

Modernizing Simulation Input Generation and Post-Simulation Data Visualization with Eclipse ICE
ICE Components – Editor

SWT Canvas subclass in an EditorPart subclass

SWT Image created from a ByteArrayInputStream

Canvas mouse listeners use atomic positions for daemon thread
ICE Components – Python Interface

Immediate exposure of VisIt’s rich Python API

Save a collection of operations and execute with only a few clicks

Automate frequently executed analysis workflows

Temporary dialog, moving to Console
ICE Components – Additional Features

Support for the full range of plot types provided by VisIt

Animate time series data
Future Work

- Capabilities added to workbench
- Integrate other visualization tools - ParaView
- Extract interfaces and create an OSGi service
- Potentially spin-off a separate project
- Increase users and further collaboration with Science IWG (among others)
Future Work
Modernizing Simulation Input Generation and Post-Simulation Data Visualization with Eclipse ICE

Questions?

GitHub

Eclipse Wiki

YouTube

github.com/eclipse/ice

wiki.eclipse.org/ICE

youtube.com/user/jayjaybillings

Additional Authors: Andrew Bennett, Jordan Deyton, Hari Krishnan, Anna Wojtowicz

Author Email: mccaskeyaj@ornl.gov, pattersontc@ornl.gov, billingsjj@ornl.gov
Evaluate the sessions

Sign in: www.eclipsecon.org