SNAP User Interface

Ken Jones
Applied Programming Technology, Inc.
Outline

- Brief Overview of System Architecture
- Extensibility
- GUI Features
- Current Development Efforts
- SNAP Demo
SNAP System Architecture

Clients

GUI Client Application

- CAFEAN Plugin API
  - CRUD/PostCRUD
  - TRACE
  - MELCOR
  - RELAP5
  - Animation
- CAFEAN Plugin API
- Pipe Component Beans
- MELCOR Component Beans
- Display Beans

Servers

Calculation Server

- CAFEAN Plugin API
  - PARCS
  - TRACE
  - RELAP5
  - MELCOR
  - EXTDATA
- PARCS
- TRACE
- RELAP5
- MELCOR
- NRCDB

Analysis Codes

Experimental Data

Job Status

jEdit Plug-in

Batch Submit

AcGrace (Plotting)

Configuration Tool
Design Features

- Platform independent
  - Pure Java (JRE 1.4 - 1.5)
  - Supported on Windows, MAC, Linux, SUN, HP, etc...
- Common Environment for Performing Engineering Analysis
  - Can Be Adapted to Any Engineering Analysis Code
  - Supports Model Development, Maintenance, and Analysis
  - Highly Extensible and Flexible
- Consistent and Intuitive User Interface
  - Minimize Learning Curve for Analysis Codes
  - Logical Organization of Model Components and Interconnections
  - 2D and 3D Visualization
  - Model Validation Tests
Extensible Design

• CORBA Used for All System Component Communication
  ‒ Strictly Defined, Well Documented Interfaces
  ‒ Easy to Implement New Client Applications
• Plug-in Design Used in Client GUI and Calculation Server
  ‒ Analysis Code Plug-ins
  ‒ Feature Plug-ins
  ‒ No Modification of the Base Code Required
• JavaBean Component Design
  ‒ Custom Beans can be independently developed
  ‒ Shared Repository for Contributed Beans
• Python Scripting
  ‒ User Defined Functions – Calculate Model Input
  ‒ Python Data Channels – Post-Processing Calculations, Animations
Plug-ins

- Plug-in API for Adding New Analysis Codes & New Features
  - available at: http://www.nrcsnap.com/snap
- Plug-in Contains All Analysis Code Specific Functionality
  - File I/O
  - Custom Editors
  - Component Drawing
  - Documentation Links
  - Component Renodalization
  - Initial Condition Extraction
  - etc.
- SNAP Code Plug-ins (Partial List)
  - TRACE
  - RELAP5 (MOD 3.3 & RELAP5-3D©)
  - CONTAIN
  - FRAPCON3
  - MELCOR 1.8.6
  - PARCS
  - COBRA-IE
- SNAP Feature Plug-ins
  - RELAP5 to TRACE Vessel Conversion Wizard
  - TRACE Data Channel Naming Conversion Utility
  - RELAP5 Legacy Model Conversion Utility
Component Data Model

• All Component Data is Declared Private
• Views Implement a ComponentListener Interface
• Changes to Component Data are Automatically Reflected in all Views.
Multiple-Window Mode

2D Views

Component Navigator

ASCII Views

Message Window

Property View
Component Navigator

- Logical Representation of Model
- All Model Data / Views
- Create, Edit, Delete Components
- Create 2D&3D Views

Main Categories

- Model Node
- Sub-Categories

Components

- Connections: Hydraulic, Heat Structures, etc.

User Defined Constants & Equations

Plug-in Nodes
The Main Property View

- Reflects the Current Selection
- Supports Multi-Selection-Edit

Mini-Navigator

Attribute Descriptions

Attribute Values

Attribute Groups

Display Options

Help Buttons

Attribute Popup Help

Custom Editors
Custom Editors

- Used to provide intuitive interface for editing complex data
- Table data can be copied to/from spreadsheet apps
2D Model Views

- May Contain any Component Type.
- Create/Edit/Delete Components and Connections.
- Any component can be displayed in multiple views.
- Embed a View into another View to Link Views.
- Export to Raster or Vector Image Formats.
- Add Text, Line, Shape and Image Annotations.
- Model Views Can be Copied to create Animation Views
- Layouts can be saved as Templates
3D Visualization

- Currently Supported by RELAP5 & TRACE Plugins
- Generate 3D Coordinates
- Rotate Pan & Zoom View
- Pivot & Shift used to Move Components about Vertical Axis
- Export Display for 3D Animation
## ASCII Views

- Component “Show ASCII” Menu Item
- Automatically Updates When Component Data Changes
- Syntax Highlighting

---

**ASCII View - PUMP 113 (IPUMP)**

<table>
<thead>
<tr>
<th>*</th>
<th>name</th>
<th>type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1130000</td>
<td>&quot;ipump&quot;</td>
<td>pump</td>
</tr>
<tr>
<td>*</td>
<td>area</td>
<td>length</td>
</tr>
<tr>
<td>1130101</td>
<td>15.723</td>
<td>0.0</td>
</tr>
<tr>
<td>*</td>
<td>az-angle</td>
<td>inc-angle</td>
</tr>
<tr>
<td>1130102</td>
<td>0.0</td>
<td>33.0</td>
</tr>
<tr>
<td>*</td>
<td>flags</td>
<td></td>
</tr>
<tr>
<td>1130103</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>*</td>
<td>ebt</td>
<td>press</td>
</tr>
<tr>
<td>1130200</td>
<td>003</td>
<td>2246.0</td>
</tr>
<tr>
<td>*</td>
<td>to</td>
<td>area</td>
</tr>
<tr>
<td>1130108</td>
<td>11205002</td>
<td>15.723</td>
</tr>
<tr>
<td>*</td>
<td>flow</td>
<td>nfl</td>
</tr>
<tr>
<td>1130201</td>
<td>1</td>
<td>3.046455e4</td>
</tr>
<tr>
<td>*</td>
<td>to</td>
<td>area</td>
</tr>
<tr>
<td>1130109</td>
<td>114010001</td>
<td>12.3741</td>
</tr>
<tr>
<td>*</td>
<td>flow</td>
<td>nfl</td>
</tr>
<tr>
<td>1130202</td>
<td>1</td>
<td>3.046455e4</td>
</tr>
<tr>
<td>*</td>
<td>phase</td>
<td>twophase</td>
</tr>
<tr>
<td>1130301</td>
<td>-2</td>
<td>0</td>
</tr>
<tr>
<td>*</td>
<td>pvel</td>
<td>pratlo</td>
</tr>
<tr>
<td>1130302</td>
<td>1189.0</td>
<td>1.0057</td>
</tr>
<tr>
<td>*</td>
<td>rtorq</td>
<td>inmoment</td>
</tr>
<tr>
<td>1130303</td>
<td>9.48e4</td>
<td>2.46e5</td>
</tr>
</tbody>
</table>
User Defined Functions

**Constants/ Variables:**
- Can be used to define values in editing dialogs
- Include Engineering Units
- Can be displayed and edited in 2D Views
- Parametric Constants generate a set of Calculations

**Functions:**
- Python Interpreter
- Calculate Variables based on Constants and Variables

---

```python
# Define some constants
PI = 3.1415926
r = GetConstant("CHAN_CORNER_RADIUS")
w = GetConstant("CHAN_INSIDE_WIDTH")

# Calculate the Channel Inside Perimeter
p = 4 * ( w - 2*r ) + ( 2*PI*r )

# Set the Result
SetVariable("CHAN_INSIDE_PERIM", p)

# Now Calculate the flow area
chanarea = w*w - (r*r - ( 2 * PI * (r^2))
rodsPerRow = GetConstant("RODS_PER_ROW")
umFuelRods = rodsPerRow*rodsPerRow*GetConstant("RODS_PER ROW")

# Calculate the Water Rod Area
waterRodArea = PI*GetConstant("CLAD_OUT")

# chanFlowArea = chanarea - ( numFuelRods * fuelRodArea )

# Set the Result
SetVariable("CHAN_FLOW_AREA", chanFlowArea)
```

**Calculated Variables**

- CHAN_INSIDE_PERIM: 0.490505m
- CHAN_FLOW_AREA: 4.985552E-3m^2
Runtime & Post-processing

- Support for Interactive and Batch Modes as well as Importing Completed Runs
- Provides Access to Plot Data for Client Applications (Animation & Plotting)
- AcGrace (Legacy Plotting Package)

Job Status Tool:
- View Status of All Runs
- Interactive Commands
- View ASCII Output
- Delete Runs
**Animation Models**

- Can be Created Directly from Model Views
- Multiple Simultaneous Data Sources
  - Analysis Code Calculations
  - Experimental Data
  - Python Calculations
- 2D & 3D Animation
- Dynamic T/H Property Range Selection
- Interactive Capability
- JavaBean Display Elements
  - Easy to Add New User Defined Beans
Test Suite Analyzer (TSA)

- Collect and Analyze Model Metrics
- Identify Holes in the Test Suite
- Generate summary and detailed reports.
- Embedded SQL Database (DERBY)
- Custom and free-form queries may be used to explore the data.
Currently Under Development

- **Runtime Improvements**
  - On-Demand Startup & Auto-Shutdown (idle timeout)
  - Remote Connections
  - Multiple Jobs in One Folder & Multiple Root Folders
  - On-Demand Folder Scanning (Improved Performance)
  - Automatic Job Loading/Unloading
  - Runtime Server Configuration (Without Server Restart)

- **Animation Plug-in**
  - Data Source Job Sequences
  - Auto-Import Using Local Plot File Selector

- **New Java-Based Plotting Package & Demultiplexers**

- **Animation Recording & AVI Movie Generation**

- **Updated User's Manual**
Contacts

Website: http://www.nrcsnap.com

USNRC Project Manager:
Chester Gingrich
Safety Margins and Systems Analysis Branch
Division of Systems Analysis and Regulatory Effectiveness
Office of Nuclear Regulatory Research
email: cgg@nrc.gov
voice: (301)415-6780

APT Project Manager:
Ken Jones
Applied Programming Technology, Inc.
240 Market St., Suite 208
Bloomsburg, PA 17815
email: krjones@appliedprog.com
voice: (570)204-4052