Eclipse to the Rescue

Austin Riddle
Jim Wall, Ph.D., Keith Biggers, Cole Markham, Paul Bilnoski
At TCAT we like to keep our eye on Eclipse RT
A Tale of Two Realities

Can be extended for an operational environment

Real-world and simulated feeds can be separately managed

Training environment for decision-makers

Software Requirements:
- Collaborative toolkit for information authoring and dissemination
- Common Operating Picture (COP)
Eclipse to the Rescue

Yes, a shameless decapitation of Mighty Mouse, we know, the tail gives it away

- Pluggable tools
- Novel interfaces for collaboration and visualization
- Rich internet capabilities that leverage existing investments
- Fault-tolerant communications for distributed systems
- Federation of pluggable components into organic frameworks

Rich Client Platform (RCP)
Rich Ajax Platform (RAP)
Equinox
Graphical Editing Framework (GEF)
Eclipse Communication Framework (ECF)
Practice, Practice, Practice

Full Spectrum Threat Response ★ Simulation (FSTR*SIM)
United States Air Force
Emergency Operations Center
All AETC bases

Emergency Management ★ Exercise System (EM*ES)
Department of Homeland Security
Incident Command Post
More than 5500 trainees
Texas Division of Emergency Mgmt.
Multi-Agency Coordination Center
All 26 MACCs in State of Texas

• Scenario-driven simulation
• Flexible training audiences and incidents
• Supports decision making and command and control
• Increased situational awareness
• User-customizable Common Operating Picture

Creating “virtual veterans” of large scale disasters, both natural and human initiated
FSTR-SIM – Full Spectrum Threat Response Simulation

Fully operational client-server desktop application:
• Resource management
• Geospatial information management
• Temporal information management
• Collaborative scenario authoring
• Common Operating Picture

Eclipse Technologies:
• RCP (3.0 – 3.1.2)
• GEF
• ECF Shared Objects
• uDig (User-friendly Desktop Internet GIS)

First attempt at RCP development
EM*ES – Emergency Management Exercise System

Fully operational distributed desktop application:
- Resource management
- Geospatial information management
- Temporal information management
- Collaborative scenario authoring
- Common Operating Picture
- Exercise recording and playback

Eclipse Technologies:
- RCP (3.2 – 3.3)
- GEF
- Lucene Plugins
- BIRT

Our most robust RCP-based simulation offering
Moving Forward

Eclipse frameworks

Eclipse ecosystem

Desktop

Focus shift

Domain framework

3rd party contributions

Reuse

Web

Leverage existing capabilities

Pluggable components

Component generalization
Enter the Dashboard

Information Dashboard Framework (IDF):
- Component based federation framework
- Allows scoped information sharing between dashboard users, installations and echelons

Levels of Integration
- Visual
- Middleware (converging data streams)
- Application to Application Data Sharing
- Hybrid (any combination of the above)

Decision Support Tools
- Manual – visual integration of data
- Assisted – visualization development using visual programming
- Automated – monitoring agents

Our organic ideas and domain-based extensions of Eclipse paradigms
**Pushing the Limits**

<table>
<thead>
<tr>
<th>Dynamic Preparedness System</th>
<th>Biosurveillance Common Operating Picture</th>
<th>Coast Guard Display System</th>
<th>USDA Dashboard System</th>
</tr>
</thead>
<tbody>
<tr>
<td>DHS Foreign Animal and Zoonotic Disease Center</td>
<td>Department of Homeland Security</td>
<td>United States Coast Guard</td>
<td>United States Dept. of Agriculture</td>
</tr>
<tr>
<td>Animal Disease Mgmt. Prototype</td>
<td>Active Global Biosurveillance 20,000 users planned</td>
<td>Emergency Preparedness On-the-water exercise in Seattle, WA.</td>
<td>Threat Response In Development</td>
</tr>
</tbody>
</table>

**Common Characteristics**

- Information sharing—selective group and user scopes.
- Common Operating Picture—user customizable view of information.
- Decision support—collaborative decision making and at a glance summary.

*Supporting training, analysis, active surveillance and threat response*
DPS – Dynamic Preparedness System

Prototype/Demo Application:
- Based on EM*ES
- Echelon and incident switching
- Thumbnail swapping
- Scripted via timeline

Eclipse Technologies:
- RCP
- GEF
- Equinox HTTP Registry
- Jetty

First dashboard prototype
BCOP – Biosurveillance Common Operating Picture

Fully operational Rich Internet Application:
- URL-based components with automatic thumbnail generation
- Widget-based components
- Report uploading/data entry
- Report->Event association
- Robust filtering of reports
- Event-based component contents/title
- Map/Timeline integration
- Account management and reporting

Eclipse Technologies:
- RAP (1.1 – 1.2)
- Server-side Equinox

First attempt at RAP development
CGDS – Coast Guard Display System

Fully operational Rich Internet Application:
- Quick user customization of layout
- Profile switching
- Component swapping
- Dynamic contributions to map and timeline
- Selective information sharing
- Map drawing and marker placement
- Scenario recording and playback
- External services connectivity

Eclipse Technologies:
- RAP (1.2 – 1.3, CVS Head)
- Equinox HTTP Registry

Rapidly prototyped and developed
Single Sourcing Issues

- User-specific Workbenches/Displays
- Getting a display outside the UI thread
  - No more:
    - `PlatformUI.getWorkbench().getDisplay()`
    - `Display.getDefault()`
- Model listeners need a display reference to call `asyncExec()`
- Timely asynchronous updates from the server
  - Polling from the client-side (not built-in, not as responsive)
    - UICallbacks (limited by proxy constraints, etc.)
- SWT Resources (e.g., Color, Image, Font) were handled differently, use JFace and Theme support
Current Challenges

- **RCP Desktop Systems**
  - Single sourcing after the fact
  - Printing view contents

- **RAP Dashboard Systems**
  - Memory footprint issues on client and server
  - Scalability (20,000 users?)
  - Client software, IE 6.0!
  - Avoiding script timeouts (data push from server)
  - Reconnect to same session, workbench state
Contributions to the Community

RAP

• Bug reports
• Widget improvements
• Patches
• Rigorous government security evaluation
• Custom widgets
Lessons Learned

Look before you leap! Eclipse projects can save...

Try to understand the paradigm, not just the API.

Be in collaboration with Eclipse project teams via Newsgroups and Bugzilla.
Demo/Questions?

Special thanks to the RAP team.