Why Integrate Test, Trace, Log and Performance Tools with Hyades

Mike Norman
CEO, Scapa Technologies, Hyades Project Lead
Hyades, a part of Eclipse

- Built as Eclipse Plug-ins
- Uses key Eclipse subsystems
  - JFace, SWT, GEF, EMF, XSD
- Imperceptible boundary with Eclipse
- Same open source engagement
  - Committers, Developers etc.
- Same vendor value-add model
Hyades and the Eclipse I*E
Outlining the head of Taurus the Bull, stars in the Hyades cluster are important to us. Oh, they give us pleasure to behold, but they also enable us to measure the universe.

Anthony G. A. Brown, Universidad Nacional Autónoma de México.
What is Hyades?

- An open cluster of stars
  - Mentioned by Hesiod between 1000 and 700 B.C
  - Located in the Taurus constellation

- A famous “measuring tool” in astronomy and physics
  - During the 1919 Eclipse, Hyades enabled scientists to collect the first empirical proof of Einstein's theory of relativity

- A groundbreaking Open Source project
  - Hyades is an integrated test, trace, log and monitoring environment, based on Eclipse, that provides standards, tools and tool interoperability across the test process
Eclipse Project Relationship (at Jan 04)

- **Project within Tools PMC**
  - Project Lead reports to Tools PMC Lead
  - Weekly committer meetings
  - Project management structure
  - Quarterly Releases since Q2/03, 1.2 at end 03

- **Occasional face-to-face meetings**
  - December 4/5 2003 Lexington
  - Fokus, IBM Rational, Intel, SAP, Scapa
  - Revising scope & engagement strategy to align with Eclipse Foundation launch
Open Source and Commercial Tools

- Open source integration framework
- Open source tools – test, trace, log, perfmon…
- Commercial tools
  - Volume development market – test, trace…
  - Enterprise Management Frameworks?
  - Application Performance Management?
  - Embedded?

- Potentially disruptive to existing players
Hyades and the Eclipse Paradoxes

- **Tools Integration = Tools Fragmentation**
  - Data viewer (e.g. trace viewer, statistical monitor)
  - Data editor (e.g. test editor)
  - Collector of data (e.g. monitoring agent)
  - Remote actuator (e.g. autonomic tuning agent)
  - Glue

- More tool function - around 80% less code

- Everybody builds, everybody benefits – an open-source and commercial ecosystem

- Hyades as an “Outreach” project
Hyades Scope

- Test definition
- Test execution history
- Trace – Java with C/C++ planned
- Statistical data (c.f. MS Perfmon)
- Log data (e.g. database & switch logs)
- Interoperability/Integration
  - User Interface look and feel
  - “Use Cases” – wizards, perspectives etc.
  - Data Integration
Lowering Tools Learning Curve

Users spend a significant time learning products
- Complexity
- Language barrier
- Lack of commonality

Hyades will provide a common user experience, common user paradigm, common terminology aimed at creating tools “habits” for testing tracing and logging

Research from Forrester shows that across all sites, visitors couldn't find what they are looking for as often as 60% of the time.
Driving the Pace of Innovation

- 80% of the effort of tool vendors spend today is duplicative infrastructure work, while only 20% produces value.

- Hyades will reverse these ratios directly benefiting both users and vendors.
Driving Standardization

- Implements OMG UML 2 Testing Profile standard
- Trace and Log standards are also leveraged/defined
- Consistent approach through the application life cycle
  - Developers, testers, system integrators, operations all use the same infrastructure and tools
  - Consistent access to traces, tests, test results, monitoring data, logs

*The nicest thing about standards is that there are so many of them to choose from.*

Ken Olsen (1926 - ), founder of Digital Equipment Corp., 1977
Hyades extends the Eclipse Platform

- Hyades depends on 3 Eclipse plugins part of the Eclipse Tools sub-project
  - EMF: Eclipse Modeling Framework
  - JDT: Java Development Toolkit
  - GEF: Graphical Editing Framework

- Each plugin provides extensibility points (■) enabling extensions such as
  - Test generators
  - Additional views
  - Additional analysis
  - etc
Hyades has Three Tiers

1) Reference Perspective and User Interface
   - Foundation for Hyades based testing, profiling and logging tools
   - “Sample implementations” providing testing and profiling features

2) Data Models/Assets Repository
   - Models for storing and persisting test and profiling data
   - Implemented using EMF

3) Data Collection and Execution Framework
   - Execution Framework
     - An extensible mechanism for executing and controlling tests and monitoring agents
     - Data Collection and Communication Infrastructure
     - Cross process / machine communication
     - Provides interfaces and implementations of data collectors
Asset Repository and Data Models

- Based on EMF
  - EMF persistence is XML/XMI
  - Alternate EMF persistence being planned for RDBMS

- Hyades assets inherit Eclipse team-sharing model

- There are 5 asset models
  - Test (Extends OMG Test Profile)
  - Trace
  - Log
  - Statistical
  - Execution History
Hyades metamodels are defined using UML
- Concepts, relationships and properties of the assets Hyades tools need to manage
- They include persistence granularity information

EMF automatically generates code from models

Test and Trace metamodels are independent at v1.2
- Hyades Test model is based on the MOF (Meta-Object Facility) Standalone Testing Profile meta-model
- Hyades Trace model is a de-facto standard created by Hyades members
- Hyades Log Model derives from Common Base Event

Hyades v2.x is targeting the unification between models
Data Collection and Execution Framework

**Execution Framework**
- Collection of interfaces for controlling tests and monitoring applications on local and remote machines
- An implementation known as the RAC
  - Start/stop/control features
  - High-bandwidth data collection

**Data Collectors (Agents)**
- JVMPI, method coverage, heap dumps
- Apache log collection
- Other logging features
- etc

**Available on a number of platforms**
- Linux (x86, 390), Windows, AiX (ppc), zSeries, iSeries, HP-UX, Solaris (sparc)
Testing & Profiling Perspective (1/2)

**Testing Perspective UIs:**
- Test generation wizards
- Test management Uis
- Test editors
- Execution log viewer
- Report publishing (HTML)

**Integrated with Profiling**
- Profiling data can be collected on a per test basis
- Traceability between tests and profiling enables greater control over completeness assessment, etc.
- Profiles can be used by some tools as the basis of tests
Testing & Profiling Perspectives (2/2)

**Profiling Perspective UIs**
- Log Viewer
- Package, Class, Instance, Method statistics
- Execution flow
- Method Invocation
- Object references
- Sequence Diagram Viewer

**Integrated with testing**
- Profiling is associated with its corresponding tests when collected from tests execution
Log Analysis

- Event logs e.g. database, O/S, Switch etc.
- Based around Common Base Events (CBE)
  - Passed as XML fragments with standard loaders
- Generic Log Adaptor (GLA)
  - Tooling to convert existing logs into CBE streams and into the Hyades Models
  - Uses Regular Expressions
- Runs dynamically or post-mortem
- Log event correlation service
Statistical Viewer

- Sampled data such as CPU-useage
- Aggregated data such as throughput, average response time, I/O counters etc.
- Measured by local agents or computed from event data (e.g. test execution history)
- Simple hierarchy of names and typed time/value pairs, with filters for collection and display
- Aggregation hierarchy/logic in the model
- Control/Dependent pairs for control systems
- Static analysis post-mortem
- Flexible live analysis tool (PerfMon++ for Eclipse)
Scapa StressTest (a Hyades Migration Case Study)

Why?

- To get some pieces we didn’t have
  - Although we had to set up Hyades to get them
  - … and then build some ourselves!
  - Most benefits are still to accrue
- To throw away code (reduce maintenance)
- To drive and conform to Standards
- To bridge development and production
  - Although production use of Eclipse is rare
- Eclipse is a better way to build software
Scapa StressTest - V2.x

**Agents/Interfaces**
- Trace capture
- Test data collection
- System data collection
- Testability wrappers

**Artefacts**
- Test
- Test execution history
- Trace
- Log
- Performance Statistics

**Engines**
- Test execution
- Intelligent test construction
- Data collection

**User Interface**
- Artefact management
- Test editing
- Test execution/control
- Performance monitoring
- Test result analysis
Scapa StressTest- V3.x Hyades Adoption

**Agents/Interfaces**
- Trace capture
- Test data collection
- System data collection**
- Testability wrappers

**Artefacts**
- Test
- Test execution history
- Trace
- Log
- Performance Statistics

**Engines**
- Test execution
- Intelligent test construction
- Data collection

**User Interface**
- Artefact management
- Test editing
- Test execution/control
- Performance monitoring**
- Test result analysis
Hyades for systems vendors

✔ Build a log agent
  - Manage your product in Eclipse

✔ Build a Statistical Performance Data Agent
  - Monitor your product in Eclipse

✔ Build some workspaces/configurations
  - Manage/monitor your product and hyades-enabled dependencies in Eclipse

✔ Build/Integrate with a test & probe engine
  - A complete Application Performance Management solution in Eclipse
Hyades for test/management/profile vendors

- Leverage the framework to get instant synergy
- Glue some agents into the framework for an instant value-added product
- Access the Eclipse volume market
- Move everything to Hyades - Fragment your code and throw away 80% (for the brave only!)
- Defend yourself – this thing could be disruptive
Hyades Resources

- Mike Norman (Hyades Project Lead) mgn@scapatech.com
- IBM technology Lead – Harm Sluiman sluiman@ca.ibm.com
- Steve Lauzon (WebSphere Studio ISV Enablement) lauzon@ca.ibm.com
- EclipseCon Technology Exchange
- http://www.eclipse.org/hyades/
- https://dev.eclipse.org/mailman/listinfo/hyades-dev
- news.eclipse.org:eclipse.tools.hyades
- http://dev.eclipse.org/viewcvs/indextools.cgi/~checkout~/hyades-home/docs/Installing%20Hyades.html
Current Hyades Structure and Committers

(enabled)

Project Lead
- Mike Norman (Scapa Technologies), mgn@scapatech.com

Model Group
- Harm Sluiman (IBM Toronto), sluiman@ca.ibm.com

Execution Group
- Kent Siefkes (IBM Rational), ksiefkes@rational.com
- Joe Toomey (IBM Rational), jptoomey@us.ibm.com

Data Group
- Richard Duggan (IBM Toronto), rduiggan@ca.ibm.com

User Interface Group
- Valentina Popescu (IBM Toronto), popescu@ca.ibm.com
Hyades Resources (cont)

- **MOF reference**
  - MOF 1.4 [http://cgi.omg.org/docs/formal/02-04-03.pdf](http://cgi.omg.org/docs/formal/02-04-03.pdf)

- **OMG Testing Profile**
  - RFP [http://cgi.omg.org/docs/ad/01-07-08.pdf](http://cgi.omg.org/docs/ad/01-07-08.pdf)
  - Submissions
    - [http://www.fokus.gmd.de/research/cc/tip/projects/u2tp/EMF](http://www.fokus.gmd.de/research/cc/tip/projects/u2tp/EMF)
The End