Extending Eclipse Test and Performance Tools Platform (TPTP)

Eugene Chan, IBM Tivoli Software
Agenda

- Overview of TPTP
  - TPTP Profiling tools
  - TPTP Monitoring
  - Extending TPTP
  - Challenges
  - What’s new?
Overview - Eclipse TPTP

Eclipse top level project
  • 2002: Eclipse tools subproject - Hyades
  • 2004: Eclipse top-level project - TPTP

Mission:
  • To build a generic, extensible, standards-based platform for test and performance tracing tools.

Goals:
  • Platform of choice for test, performance, and monitoring tools.
  • Exemplary tooling.
  • Enable value-added third-party tooling through extensibility and high-quality APIs.
Overview - TPTP project structure

- Comprised of four projects
  - Platform
  - Test
  - Trace
  - Monitoring

- Developed by a number of strategic contributors
  - Active contributors: Intel, IBM, OC Systems
  - Inactive contributors: Scapa Technologies, Computer Associates, Compuware, FOKUS

- For more information visit http://www.eclipse.org/tptp/
Overview - TPTP framework

- **TPTP offers a common, extendable framework** as well as reference implementations for the following functions:

  - Testing
  - Profiling
  - Static code analysis
  - Static and dynamic Bytecode Instrumentation (BCI)
  - Application monitoring and log analysis
Agent Controller Architecture

Remote TPTP Client

Local TPTP Client

Socket

RAC

Sh.Mem

Socket

Bindings

Agent(s)

Config

DAT

CMD
Agenda

✓ Overview of TPTP
  ➢ TPTP Profiling tools
  □ TPTP Monitoring
  □ Extending TPTP
  □ Challenges
  □ What’s new?
Overview - TPTP Profile tools

➢ Common framework for profiling simple Java applications or complex Web applications running on multiple platforms, on different hosts.
   • Common perspectives and views for interacting with target systems and resources.
   • User actions to interact with the profiled application.
     • Start, stop, attach, detach from the profiled application.
   • Views framework to analyze the profiled application.
     • Views can be extended and customized using extension points.
   • Standard EMF data model, query framework and assets repository.
   • Common data collection and execution framework on local and remote targets.

➢ Reference implementation of the Profiling framework:
   • Java Profiling tools based on JVMPI and JVMTI.
   • Actions to interact with the profile application: run garbage collection, collect object references, filter and sort data
   • Views to analyze collected data: execution, memory and thread analysis.
Profiling Tools

- Profiling Perspective
Profilling Tools

- Start a profiling session…
Profiling Tools

- Profiling Filter

![Image of Profiling Tools interface]

- Select the data collectors and analysis types for the launch.
- Press F2 to see a quick description of the selected data collector or analysis type.
- Use a filter set to filter out and customize profiled data.
- Supported wildcard expressions: "*", "?.", "??.", "??", "*.class", "*.class.*", "*.class.??".
Profiling Tools

- Profiling Option – Memory Analysis

![Diagram showing memory analysis options in Eclipse Profiler]

- Level of detail:
  - Track object allocation sites
Profiling Tools

- Open Profiling View
Profiling Tools

- Profiling View - Object Allocations view
Profiling Tools

- Profiling View – Object Allocation View
Profiling Tools

- Profiling Option – Execution Time Analysis
Profiling Tools

- Profiling View – Execution Statistic View
Profiling Tools

- Profiling View – Execution Statistics View
Profiling Tools

- Profiling Option – Thread Analysis
Profiling Tool

- Profiling View – Thread Analysis View
Profiling Tool

- Profiling View – Call Stack
Profiling Tool

- Profiling Views – Toolbar Actions
Agenda

✓ Overview of TPTP
✓ TPTP Profiling tools
➢ TPTP Monitoring
▪ Extending TPTP
▪ Challenges
▪ What’s new?
Monitoring - Overview

- Disparate pieces and parts
- Tools focused on individual products
- No common interfaces among tools
- No synergies in building tools or creating log entries
- Common format for log files
- Common set of tools
- Common interfaces among tools
Monitoring - Overview

➤ Logging Agent

- XML message-based agent for real-time monitoring of message generating facilities (e.g. loggers).
- Extensible architecture for crafting proprietary Logging Agents:
  - Defined APIs and programming model.
  - C implementation with a Java wrapper for supporting C/++ and Java run-times.
  - TPTP provides Logging Agent support for several popular Java-based logging and tracing facilities.

➤ Common Base Event

- Open-source specification (OASIS) to provide a common and standardized taxonomy for events occurring in hardware and software
- Unified format and terminology for the standardized exchange and consistent interpretation of problem determination data to circumvent varying vendor, product and version representations.
- Common Base Event XML schema defines the overall structure of the event, format of each property and all mandatory properties for completeness.
- TPTP provides EMF consumer (e.g. model artifacts) Java, EMF producer (e.g. native logging) Java, non-EMF Java and C/C++ implementations
Monitoring – Generic Log Adapter

Generic Log Adapter

• A component of TPTP
• In TPTP, Generic Log Adapter transforms log messages into Common Base Event objects. Common Base Event is the common format for messages in TPTP.
• Has flexible architecture that is easily extendable.
• Uses a configuration XML file to describe the transformation
  • Includes Standalone run-time
  • Adapter configuration files and sample execution scripts
  • Editor for creating, modifying and testing configuration files
Monitoring – GLA adapters

GLA

- Transforms are defined in adapter files:
  - Contains one or more contexts and the components for each context.
  - Stored as XML based on a defined schema.
  - Parsing component incorporates mapping proprietary log and trace record properties to Common Base Event properties using mapping rules.

TPTP provides Common Base Event and Logging Agent standalone and plug-in support for the following popular logging facilities:

- Jakarta Apache Commons
- Java Logging (JSR-047)
- Jakarta Apache Log4J
Monitoring – Symptom Catalog

Symptom Catalog

• Symptoms are the sign of a malfunction, a problem or just simply a change of state in a resource that is the subject of our problem determination analysis.

• A database consisting of matching patterns indicating known problems, explanations and resolution steps.

• Persisted as an XML file with a defined schema.

• Crafted using the Symptom Editor of TPTP

• Vendors and organizations may provide multiple hierarchal symptom catalog based on logical or business divisions.

• Local and remote (FTP/HTTP) importing functionality to ensure content freshness.
Monitoring – Import Log

- Import Log Wizard
- Log Type (Adapter)
- Log Location
  - Local
  - Remote
- Log details
Monitoring – Log View
Monitoring – Import Symptom

- Import Symptom Wizard
  - Local
  - Remote

- Log Analysis
  - Allows users to easily detect and solve problems that have already been previously encountered, and persisted in symptom catalog
  - Analysis consists of lexicographically comparing varying Common Base Event properties with match patterns in one or more symptom databases.
  - Extensible architecture to allow users to define vendor and product specific analysis engines.
Monitoring – Analysis Results View
Monitoring - Correlation

- Computing system maintainers require a detailed understanding of an entire computing system in order to detect and resolve cascading problems.
- Correlation determines one or more sets of related events to visualize control flow within and between computing systems.
- A correlation engine or schema associates varying Common Base Event properties (log events) based on a predetermined criteria (e.g. time).
- Extensible architecture to allow users to define vendor and product specific correlation engines or schemas.
Monitoring - Correlation
Agenda

✓ Overview of TPTP
✓ TPTP Profiling tools
✓ TPTP Monitoring
   ➢ Extending TPTP
☐ Challenges
☐ What’s new?
Extending TPTP - Profiling

Host B

Host A

TPTP Remote Agent Controller

Control + Data

Trace events

LTTng Agent

January 29, 2008

Eclipse Foundation - www.eclipse.org

Content provided under the terms and conditions of the Eclipse Public License Version 1.0
Extending TPTP - Monitoring

- LTTng
- TextDump
- GLA
- CBE
- Symptom Catalog
- Eclipse WB
- Application Log
- System Log
Extending TPTP

- TPTP defines extension points that allow consumer plug-ins to contribute behaviors to existing UI.

- Context Language Extension
  - Define a user interaction scenario which is applied for specific languages.
  - The content, description and actions for a profiling view can be customized based on the type of the profiled application.
  - Custom columns can be added to the existing views to show language specific information.
  - Custom implementation for the ‘Open source’ action which opens the source code on a specified resource.
Extending TPTP
Extending TPTP

- Multiple contexts support
Extending TPTP

[org.eclipse.hyades.ui.contexts]
• Defines the language context so that custom content can be applied for this type of language.
• The contextKey attribute, which uniquely identifies the language type, is persisted into the data model.

```xml
<extension
    point="org.eclipse.hyades.ui.contexts">
  <context
     id="C/C++"
     contextKey="C/C++"
     name="%_1"
     description="%_2"
     icon="/icons/full/obj16/compileunit.gif">
  </context>
</extension>
```

[org.eclipse.hyades.ui.contextProviders]
• Defines the content to be displayed in the statistical views
• Icons, labels and views content can be customized using this extension point
Extending TPTP

[org.eclipse.hyades.ui.contextOpenSourceProviders]

• Custom implementation for the ‘Open source’ action which opens the source code on a specified resource
• In the Web Services scenario, it can open the WSDL file defining the web service

➢ Trace event declares context language:
  • <traceStart traceId="1" agentIdRef="1" time="…." language="C/C++" />
  • Default language is Java.
  • Mixture of languages is supported, language specified at object def level.
Extending TPTP

- Example: OC Systems Hitchhiker
  - Trace, profile or find memory leaks in any C/C++ application
Agenda

✓ Overview of TPTP
✓ TPTP Profiling tools
✓ TPTP Monitoring
✓ Extending TPTP
➢ Challenges
☐ What’s new?
Challenges

- TPTP Profiler tool has been initially designed for profiling Java applications
  - Java centric data model
    - Host, Package, Class and Method hierarchy
  - Java centric visualization
  - new UI extensions since TPTP 4.1 for visualizing non-Java languages

- Size of Data
  - Filtering support – collection time and UI level
  - Profile to file
Agenda

✓ Overview of TPTP
✓ TPTP Profiling tools
✓ TPTP Monitoring
✓ Extending TPTP
✓ Challenges
➢ What’s new?
What’s New in TPTP 4.5

- Profiling Tools:
  - Binary data format to increase profiler performance and scalability.
  - Improved profiling filters (filter sets).
  - Secure profiling (JVMTI) including authentication and encryption.
  - Simplified stand-alone profiling (JVMTI).
  - Thread contention analysis for locating monitor and data contention.
  - Support for Java 6 and IPv6 networks.
References

➢ TPTP home page
   http://www.eclipse.org/tptp/

➢ TPTP download page
   http://www.eclipse.org/tptp/home/downloads/

➢ TPTP documentation
   http://www.eclipse.org/tptp/home/documents/

➢ TPTP tutorials and demos
   http://www.eclipse.org/tptp/home/downloads/quicktour/v44/quick_tour.html

➢ TPTP Eclipse corner article
   http://www.eclipse.org/articles/Article-TPTP-Profiling-Tool/tptpProfiling