

OOPSLA Eclipse News

Moritz Eysholdt
Werkstudent bei Itemis
Student des Masterstudiengangs
Informatik der Uni Oldenburg

27. November 2007

Agenda / Übersicht

- (1) Aspect Oriented Programming (AOP)
- (2) Statische Code Analyse: Findbugs
- (3) OSGi, Equinox auf der Serverseite
- (4) R-OSGi
- (5) Weiteres

Aspect Oriented Programming (AOP)

- Behandelt Cross-Cutting Concerns
 - Belange welche alle/mehrere Module einer Software betreffen und sich daher mit OOP schlecht selbst modularisieren lassen.
- Konzepte (von AspectJ)
 - Join Point
 - Pointcut
 - Advice
 - Introductions

AspectJ: Native vs. Annotated Syntax

- Bessere Syntaxüberprüfung

- Einfachere Einführung
- Echtes Java 5.0

“aspect” is like a class

Pointcut

```

aspect PersistenceAspect {
    pointcut stateChange(BankAccount ba):
        (call(void BankAccount.deposit(*)) ||
         call(void BankAccount.withdraw(*)))
        && target(ba);

    after(BankAccount ba) returning: stateChange(ba) {
        // persist ba.getBalance() value
    }
}
    
```

2 Join Points

“after” Advice

“aspect” is a class!

Pointcut

```

@Aspect
class PersistenceAspect {
    @Pointcut(“
        (call(void BankAccount.deposit(*)) ||
         call(void BankAccount.withdraw(*)))
        && target(ba)”)

    void stateChange(BankAccount ba) {}
    @AfterReturning(“stateChange(ba)”)
    void afterStateChange(BankAccount ba) {
        // persist ba.getBalance() value
    }
}
    
```

2 Join Points

“after” Advice

AOP Frameworks

- AspectJ.
 - eclipse.org/aspectj, Most sophisticated AOP solution.
- Spring Framework AOP.
 - springframework.org
- JBoss AOP.
 - jboss.org
- CGLIB, ASM, ...
 - Byte-code manipulation libraries.

Java - Introduction Example/src/introduction/HashablePoint.aj - Eclipse Platform - /Users/eysholdt/Eclipse/workspa...

```

public aspect HashablePoint {

    public int Point.hashCode() {
        return (int) (getX() + getY() % Integer.MAX_VALUE);
    }

    public boolean Point.equals(Object o) {
        if (o == this) { return true; }
        if (!(o instanceof Point)) { return false; }
        Point other = (Point)o;
        return (getX() == other.getX()) && (getY() == other.getY());
    }

    public static void main(String[] args) {
        Hashtable h = new Hashtable();
        Point p1 = new Point();

        p1.setRectangular(10, 10);
        Point p2 = new Point();

        p2.setRectangular(10, 10);

        System.out.println("p1 = " + p1);
        System.out.println("p2 = " + p2);
    }
}

```

Package Hierarchy

- Bean Example
- Introduction Example
 - src
 - introduction
 - CloneablePoint.aj
 - ComparablePoint.aj
 - HashablePoint.aj
 - Point.java
 - AspectJ Runtime Library
 - JRE System Library [JVM 1.5.0 (

Outline

- introduction
- import declarations
- HashablePoint
 - Point.hashCode()
 - Point.equals(Object)
 - main(String[])

Cross References

- HashablePoint
 - Point.hashCode()
 - declared on
 - Point
 - Point.equals(Object)
 - declared on
 - Point

Statische Code Analyse: Findbugs

- findbugs.sourceforge.net
 - Open source project
 - Large community
 - Easy to adapt and customize
 - Many defect detectors
 - Eclipse plugin support
 - Mostly searches for localized bugs
- Memory errors
 - array bounds / buffer overrun
 - illegal dereference
 - double free
 - memory leak
 - use uninitialized data
- Input validation
 - command injection
 - tainted data
- Concurrency
 - race conditions
 - deadlock
 - data protected by locks
- Resource/protocol errors
 - failure to free resources
- Exceptional conditions
 - integer over/underflow
 - not handling error cases
 - type conversion errors
- Code quality
 - unused variables

The screenshot displays the Eclipse IDE interface with the FindBugs plugin. The main editor shows the source code of `DevcomposProviderFactory.java`. The `getText` method is highlighted, showing a redundant null check for the `desc` variable. The Bug Tree View on the left lists several issues, with the selected issue being 'Redundant nullcheck of value known to be null'. The Bug Details view on the right provides a detailed description of this issue, including the class and method context and the specific line of code where the error occurred.

Bug Tree View:

- de.oldenburg.uni.informatik.diem.device
 - Dead store to local variable
 - M D DLS: Dead store to d in de.oldenburg.uni.info
 - Field isn't final but should be
 - Redundant nullcheck of value known to be null
 - M D RCN: Redundant nullcheck of desc which is k
 - Should be a static inner class
 - M P SIC: Should de.oldenburg.uni.informatik.diem
 - M P SIC: Should de.oldenburg.uni.informatik.diem
 - M P SIC: Should de.oldenburg.uni.informatik.diem
 - M P SIC: Should de.oldenburg.uni.informatik.diem
 - Unread field
 - Unused field
 - Use the nextInt method of Random rather than nextD
 - M P Dm: Method de.oldenburg.uni.informatik.diem
 - Write to static field from instance method
 - H D ST: Write to static field de.oldenburg.uni.infor
 - H D ST: Write to static field de.oldenburg.uni.infor

Code Snippet:

```
public String getText(Object object) {
    Task t = (Task) object;
    String desc = null;
    // TODO: make this work
    /*
     * if (t.getDevCompos() != null) { desc =
     * t.getDevCompos().getName(); } else if (t.getMode() != null)
     * Resource r = t.getMode().eResource(); Device d = (Device)
     * t.getMode().eContainer(); desc = d.getName() + " - " +
     * t.getMode().getName(); }
     */
    return desc == null ? t.getName() : t.getName() + "(" + desc
}
```

Bug Details:

Medium Priority Dodgy

In class de.oldenburg.uni.informatik.diem.device.views.DevcomposProviderFactory\$TaskProvider
 In method de.oldenburg.uni.informatik.diem.device.views.DevcomposProviderFactory\$TaskProvider.e
 Local variable named desc
 Redundant null check at DevcomposProviderFactory.java:[line 60]

Redundant nullcheck of value known to be null

This method contains a redundant check of a known null value against the constant null.

OSGi, Equinox auf der Serverseite

- Weg 1: Einen HTTP Server in Equinox einbetten
 - Die OSGi-Anwendung wird zum eigenständigen Webserver
- Weg 2: Equinox in ein Servlet einbetten
 - Kompatibel zu Tomcat, Jetty etc.
- <http://www.eclipse.org/equinox/server/>

Weg 1: In Equinox eingebetteter HTTP Server

- Zwei vollständige OSGi Http Service Implementierungen:
 - `org.eclipse.equinox.http`
 - Servlet-API 2.4 kompatibel
 - Geringer Ressourcenverbrauch
 - `org.eclipse.equinox.http.jetty`
 - Baut auf die Serverletengine von Jetty auf.
 - Servlet-API 2.4 kompatibel

Weg 2: Equinox in WAR einbetten

- **/WEB-INF**
 - **/web.xml** (with one servlet entry assigning all incoming requests to the BridgeServlet)
 - **/lib/servletbridge.jar** (the classes associated with the equinox.servletbridge)
 - **/eclipse** (the eclipse platform directory)
 - **launch.ini** (contains framework properties that will allow override of any eclipse specific System Properties)
 - **/configuration** (contains config.ini which lists the bundles you want to have available)
 - **/features**
 - **/plugins**
- **Automatische Erstellung atm durch Ant-Tasks. Direkte Unterstützung durch Plugins ist geplant.**

Webanwendung implementieren

- Verzeichnis freigeben:

```
<plugin>
  <extension point="org.eclipse.equinox.http.registry.resources">
    <resource
      alias="/files"
      base-name="/web_files"/>
    </extension>
  </plugin>
```

- Servlet registrieren:

```
<extension point="org.eclipse.equinox.http.registry.servlets">
  <ervlet
    alias="/test"
    class="com.example.servlet.MyServlet"/>
</extension>
```

R-OSGi

- Erweiterung von OSGi
- Lässt eine OSGi-Anwendung als verteiltes System laufen
 - Für Kommunikation werden dynamisch Proxies erstellt, welche welche selbige transparent über Netzwerk tunnelt
 - SLP als service discovery protokoll
 - Nachrichtenaustausch im SLP-Stil, leichtgewichtiger als Jini/RMI
 - Load-Balancing
 - Bei Bedarf kann auch ein gesamtes Bundle transferiert werden
 - Fault-Tolerance
 - z.B. können Netzwerkfehler auf OSGi-Bundle-Unload-Events gemapped werden

RDT

R-OSGi Deployment Tool

- Entwicklung, Deployment und Monitoring von R-OSGi basierten Systemen
- Befindet sich noch stark in der Entwicklung
- Grafische Analyse von Abhängigkeiten zwischen OSGi Bundels
- Konfiguration von automatischem Load-Balancing und Fault-Tolerance
- Visualisiert Struktur und Status eines verteilten Systems in echtzeit
- Protokolliert alle über Netzwerk ausgetauschten Nachrichten

Weiteres #1

- Tut3: Findbugs
- Tut6: AspectJ
- Tut11: Best Practices for Model-Driven Development
- Tut22: Building Embedded and Stand-alone Domain-specific Languages: Principles & Practice
- Tut31: Green Bar for C++ - Unit Testing & Refactoring for C++
- Tut49: Creating Plug-ins and Applications on Eclipse Platform
- Tut52: Building Service-Oriented Architectures with Web Services

Weiteres #2

- ismm/p55: Accordion Arrays: Selective Compression of Unicode Arrays in Java
- oopsla/p1: The JastAdd Extensible Java Compiler (!!)
- oopsla/p281: WebRB: Evaluating a Visual Domain-Specific Language For Building RelationalWeb-Applications
- oopsla/p623: Living it up with a Live Programming Language
- oopsla/p734: X3DWeb Software Visualization in Action!
- oopsla/p749: Eclipse Technology Exchange Workshop (ETX2007)
- oopsla/p779: Ruby Refactoring Plug-In for Eclipse
- oopsla/p781: Refactoring Support for the C++ Development Tooling
- oopsla/p783: CUTE: C++ Unit Testing Easier
- oopsla/p791: Green – A Flexible UML Class Diagramming Tool for Eclipse
- oopsla/p805: Using FindBugs On Production Software
- oopsla/p834: A Flexible UML Class Diagramming Tool for Eclipse
- oopsla/p854: TuningFork: A Platform for Visualization and Analysis of Complex Real-time Systems
- oopsla/p856: Finding Bugs in Eclipse (Demonstration)
- oopsla/p864: Ready for Distribution? Turning Modular into Distributed Applications with the R-OSGi Deployment Tool
- oopsla/p870: DEMOCLES: A Tool for Executable Modeling of Platform-Independent Systems
- oopsla/p874: Model-driven Development with Predictable Quality
- oopsla/p878: Improving Quality Together
- oopsla/p880: Improve Software Quality with SemmleCode —an Eclipse Plugin for Semantic Code Search—
- oopsla/p882: Lagrein: Tracking the Software Development Process (!!)
- oopsla/p917: CodeGenie: a Tool for Test-Driven Source Code Search
- oopsla/p923: Automatic Support for Model-Driven Specialization of Object-Oriented Frameworks
- oopsla/p925: Activating Refactorings Faster
- oopsla/p927: Round-Trip Engineering Using Framework-Specific Modeling Languages
- oopsla/p945: SmartEMF

Quellen

- AspectJ
 - <http://www.eclipse.org/aspectj>
 - Principles of Aspect-Oriented Design in Java and AspectJ by Dean Wampler (Tutorial #6, OOPSLA'07)
- Findbugs
 - <http://findbugs.sourceforge.net>
 - Revolutionizing Software Quality through Static Analysis Tools by Jonathan Aldrich (Tutorial #3, OOPSLA'07)
- Server-Side Equinox
 - <http://www.eclipse.org/equinox/server/>
- R-OSGi
 - <http://r-osgi.sourceforge.net/>
 - Ready for Distribution? Turning Modular into Distributed Applications with the R-OSGi Deployment Tool by Jan S. Rellermeyer, Gustavo Alonso und Timothy Roscoe (OOPSLA'07 Proceedings Seite 864)

Vielen Dank für Ihre Aufmerksamkeit!

Moritz Eysholdt
Werkstudent bei Itemis
Student des Masterstudiengangs
Informatik der Uni Oldenburg

27. November 2007

Moritz Eysholdt