ASPEKTORIENTIERTE PROGRAMMIERUNG IN .NET

Hilmar Bunjes

20. August 2009

Agenda

- Motivation „Warum soll ich AOP lernen?“
- Was ist AOP?
- Tools für .NET
- Beispiele und Demos
- Zusammenfassung

Motivation

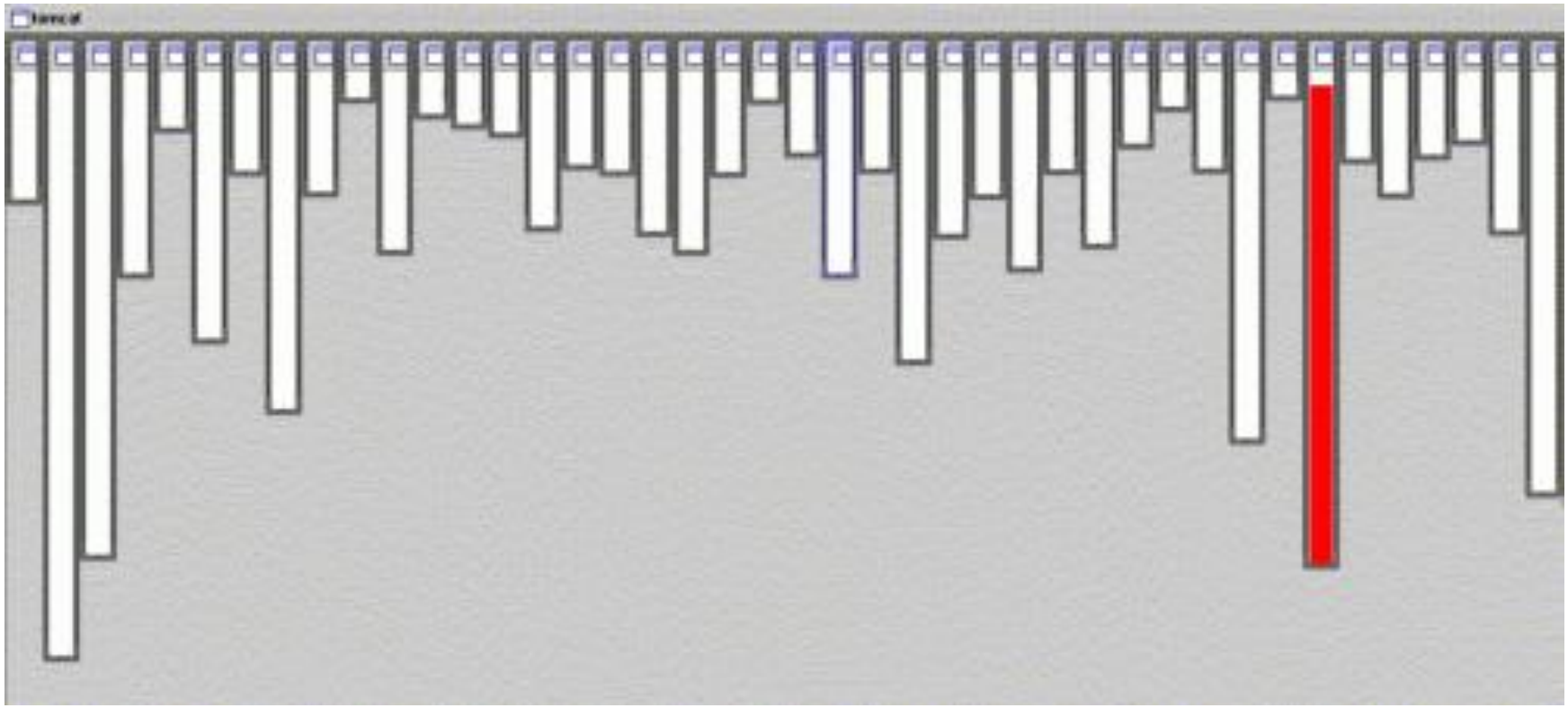
„Warum soll ich AOP lernen?“

Motivation

- Seperations of Concern
  - Concern: Angelegenheit / Konzept
  - Fachlich: Geschäftslogik
  - Nichtfachlich: Logging, Sicherheit, …
- Komplexität der Software sinkt nicht ☺

Motivation – Beispiel

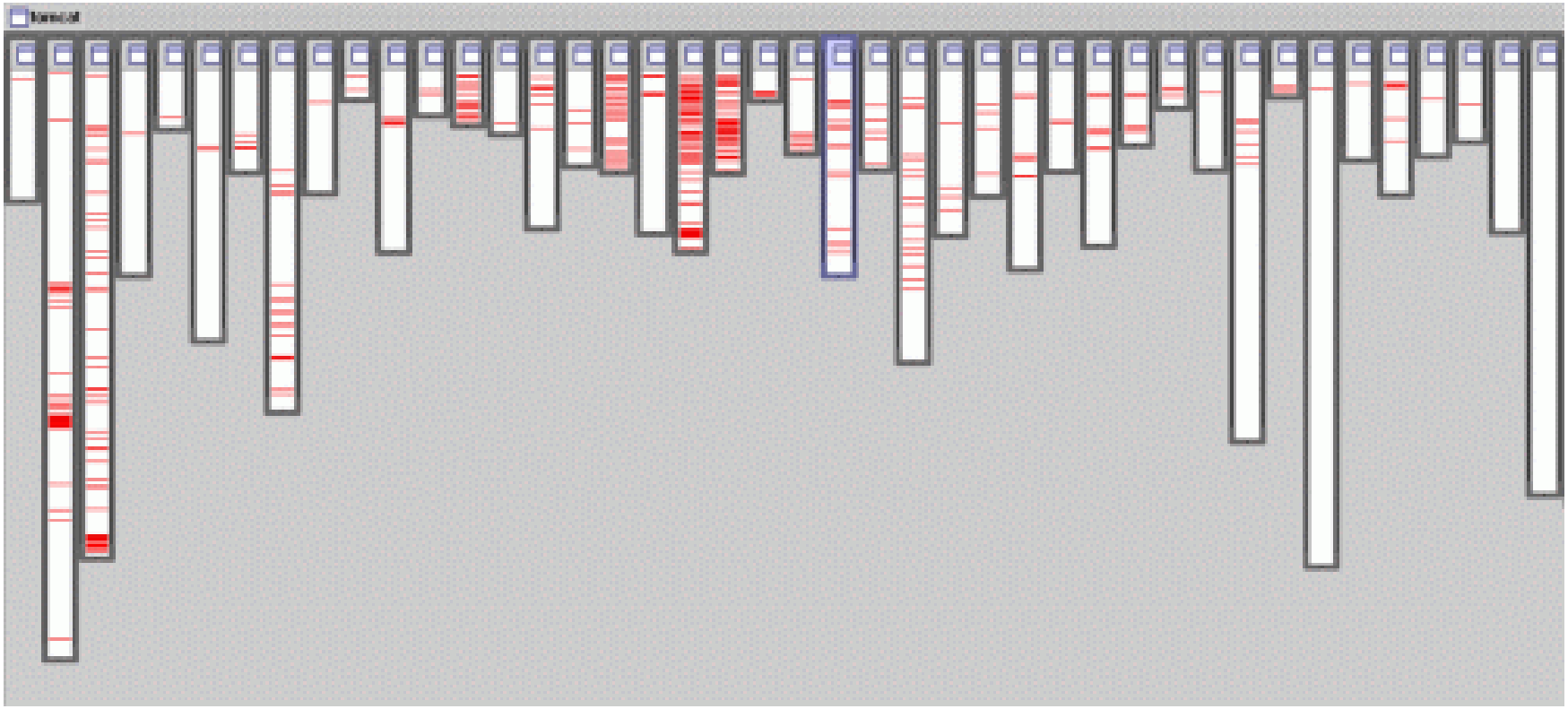
- XML-Parsing im Apache Tomcat Server



www.aspectj.org

Motivation – Beispiel

- Logging im Apache Tomcat Server



www.aspectj.org

Motivation – Beispiel

- Was passiert, wenn Log Framework geändert wird?
  - Namensänderungen -> Refactoring
  - Tiefere Änderungen -> Händische Arbeit
- Was passiert, wenn Autorisierung geändert wird?
- …

Was ist AOP?

Was ist AOP?

- Programmierparadigma
- Getrennte Entwicklung logischer **Aspekte**
- Spätere Zusammenführung (Weaving/Weben)

Was ist AOP?

- Beispiel Bankkonto

fachlich	nichtfachlich
-Einzahlung	-Logging
-Auszahlung	-Performance
-Überweisung	-Sicherheit
-…	-…

Was ist AOP - Beispiel


```
public class Bankkonto {
    public int Kontostand { get; private set; }
    private static Logger _log = Logger.GetLogger(typeof(Bankkonto));
    public void Einzahlen(int eurocent) {
            CheckPermission();
            _log.info("Einzahlung: alt: {0}, neu: {1}", Kontostand, Kontostand + eurocent);
            Kontostand += eurocent;
    }
    public void Auszahlen(int eurocent) {
            CheckPermission();
            _log.info("Auszahlung: alt: {0}, neu: {1}", Kontostand, Kontostand + eurocent);
            Kontostand -= eurocent;
    }}
```

Was ist AOP - Beispiel


```
public class Bankkonto {
    public int Kontostand { get; private set; }
    private static Logger _log = Logger.GetLogger(typeof(Bankkonto));
    public void Einzahlen(int eurocent) {
            CheckPermission();
            _log.info("Einzahlung: alt: {0}, neu: {1}", Kontostand, Kontostand + eurocent);
            Kontostand += eurocent;
    }
    public void Auszahlen(int eurocent) {
            CheckPermission();
            _log.info("Auszahlung: alt: {0}, neu: {1}", Kontostand, Kontostand + eurocent);
            Kontostand -= eurocent;
    }}
```

Was ist AOP - Beispiel

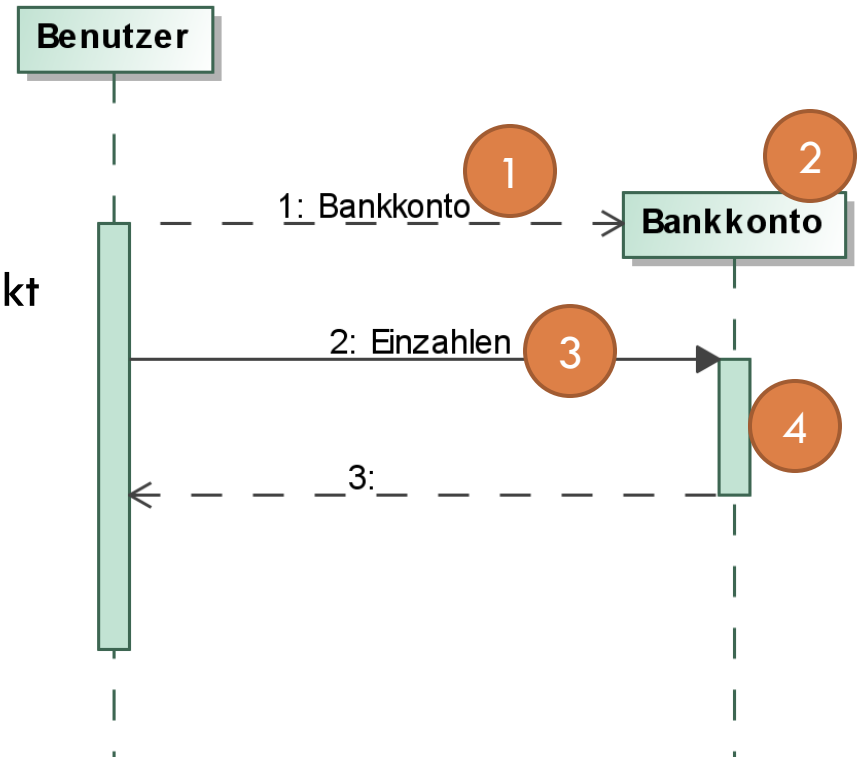

```
public class Bankkonto {
    public int Kontostand { get; private set; }
    [LogCall]
    [CheckPermission]
    public void Einzahlen(int eurocent) {
            Kontostand += eurocent;
    }
    [LogCall]
    [CheckPermission]
    public void Auszahlen(int eurocent) {
            Kontostand -= eurocent;
    }}
```

Was ist AOP - Begriffe

- Concern
  - Angelegenheit / Konzept (z.B. Logging)
- Joinpoint
  - Eingriffspunkt im Programm
- Pointcut
  - Selektion bestimmter Joinpoints
- Advice
  - Ratschlag/Code, der an Pointcut eingebaut wird
- Aspect
  - Gruppierung Pointcut/Advices, die zu Concern gehören

Was ist AOP - Begriffe

- Joinpoint
  - Aufrufen einer Methode
  - Ausführen einer Methode
  - Zugriff auf Variable
  - Werfen einer Exception
  - Initialisierung einer Klasse/Objekt



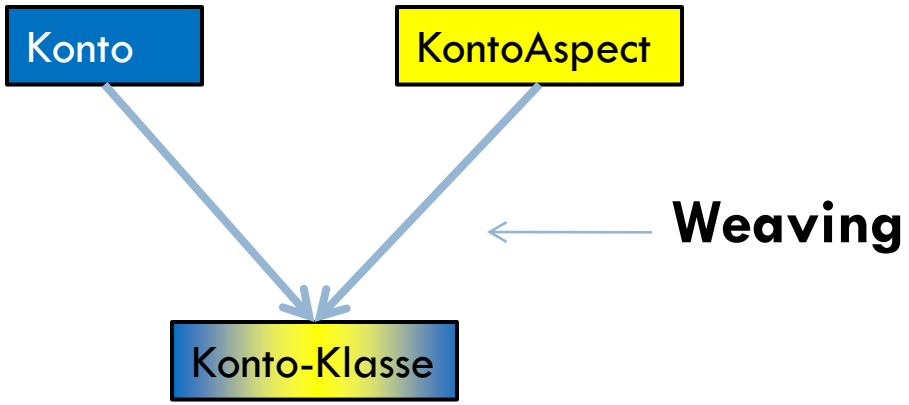
Was ist AOP - Begriffe

- Advice
  - Code, der am Joinpoint eingebaut wird

```
public override void OnEntry(MethodExecutionEventArgs eventArgs)
{
        Trace.WriteLine("Entry method " + eventArgs.Method);
}
```

Was ist AOP - Begriffe

- Weaving
  - Zusammenführung fachliche Klasse und Aspekt



Tools für .NET

Tools für .NET

- PostSharp
  - Version 1.5 vom 28. Juli 2009
  - Aktueller Webcast: AOP in Silverlight (http://www.microsoft.com/germany/msdn/webcasts/library.aspx?id=1032422768)
- Spring.NET AOP
  - Auch ohne Spring Framework nutzbar
- dotspect
  - Seit 2006 kein commit
- Aspect#
  - Letzte Änderung 2005

Tools für .NET – PostSharp

- .NET Assemblies nach Compilerlauf ändern
- PostSharp Laos: AOP-Plugin
- Ausführung nach C#/VB Compiler
- Leichte Verlängerung der Build-Zeit (5-100%)
- Leichtgewichtig zur Laufzeit
- Wenig Reflection-Kenntnisse notwendig

Tools für .NET – PostSharp

- Einstiegpunkte
  - **Methoden**: Vorher, Nachher, Exception, Erfolg
  - Auftreten einer **Exception**
  - **Felder**: Lesen, Schreiben
  - **Methodenaufruf** (ohne Zugriff auf Methode)
  - **Interface/Implementierung** zu Klasse hinzu
  - Implementierung **abstract/extern Methode**
  - **CustomAttribute** hinzufügen

Beispiele und Demos
-> Visual Studio

Zusammenfassung

Zusammenfassung

- Fachlichen/Nichtfachlichen Code trennen
- Einfache Änderung Implementierungsdetails
- Benötigt Eingewöhnung
- Kann Performance kosten, aber beschleunigt Entwicklung

Vielen Dank für die Aufmerksamkeit!

Kontakt:
Hilmar Bunjes
hilmar@bluecall.de

Ressourcen

- PostSharp:
  http://www.postsharp.org
- AOP Konferenz:
  http://aosd.net/
- Spring AOP Einstieg:
  http://www.developer.com/lang/article.php/3795031
- Spring AOP Guide:
  http://www.springframework.net/doc-1.1-P3/reference/html/aop-quickstart.html