



# Verteilte Paarprogrammierung mit Eclipse

Riad Djemili ([djemili@inf.fu-berlin.de](mailto:djemili@inf.fu-berlin.de))

Diplomarbeitsthema  
Freie Universität Berlin  
Februar 2006

# Inhalt

1. Wie ist der aktuelle Stand im Bereich des Distributed Pair Programming?
  - Welche Implementierungen existieren?
  - Worin unterscheiden sie sich untereinander und vom klassischen PP?
2. Entwicklung eines DPP-Plugins für die Eclipse-Umgebung.
3. Implementierung neuer Sensordaten für das ElectroCodeoGram.

# Paarprogrammierung

- Zwei Entwickler arbeiten an einem PC.
  - *Driver* programmiert und kontrolliert Maus und Tastatur.
  - *Observer* schaut zu, gibt Rat und schlägt Dokumentation nach.
- Vor allem beliebt als Prinzip agiler Softwareprozesse, insbesondere **Extreme Programming**.

# Warum Paarprogrammierung?

## ■ Vorteile

1. Höhere Software-Qualität
2. Wissensaustausch

## ■ Nachteile

1. „Erhöhte Kosten“
2. Skepsis gegenüber ungewohnter Teamarbeit

# Übergang von PP zu DPP

- Wie lässt sich Paarprogrammierung in eine verteilte Umgebung übertragen?
  - Welche Aspekte des PP lassen sich übernehmen?
  - Wo unterscheidet sich DPP oder ist sogar besser als PP?
- Es existieren bereits verschiedene Implementationen..

# Vergleich der Ansätze 1/2

- VNC-Tools (Virtual Network Computing)
- IDEs
  - Borland JBuilder
  - SUN Java Studio / NetBeans
  - Sangam - *Eclipse Plugin*
- SubEthaEdit – *Texteditor für Apple Mac*

# Vergleich der Ansätze 2/2

	Kollaboration	„Präsenz“	Was wird übertragen?	IDE-Features
VNC Tools	Observer hat keine Kontrolle	Observer hat keine Kontrolle	Alles	IDE des Hosts
Borland JBuilder	Observer kann selbständig browsen	Textstelle & Followmode	Code, GUI, Debug	Vollständige IDE
NetBeans/ SUN Java Studio	Mehrere Driver	Textstelle	Code, Run, Debug	Vollständige IDE
SubEthaEdit	Mehrere Driver	Positionen aller Teilnehmer	Text	Hauptsächlich Texteditor

# Umfrage

- Untersuchung der unterschiedlichen Ansätze wird mit eigener Umfrage ergänzt.
  - Welche Erfahrungen existieren mit DPP?
  - Welche Funktionen werden bei DPP als wichtig erachtet?
  - Wie stark werden Internetmedien (z.B. Instant Messenger, IRC, Foren) bereits *während* der Programmierung genutzt?
  - ..



# Saros

- Ein Plugin für Eclipse, zur verteilten Paarprogrammierung.
  - Teilnehmerlisten
  - Synchronisation von Dateien
  - Kollaboration
  - Kommunikation
  - Datensammlung

# Teilnehmerlisten

- In Eclipse wird eine Teilnehmerliste aller angemeldeten Entwickler angezeigt.
- Neue Sitzungen werden aus der Liste heraus eingeleitet. Laufende Sitzungen werden in der Liste angezeigt.
  - Keine zusätzlich benötigte Software, um den Kontakt herzustellen und Möglichkeit für spontane Sitzungen.

# Synchronisation von Dateien

- Üblicherweise „Remote Sharing“
  - Ein Teilnehmer gibt sein lokales Projekt frei.
  - Quelldateien werden bei Anfrage von dem „Hoster“ geladen.
- Nachteil
  - Teilnehmer können nicht selbstständig kompilieren, das Programm starten oder Unit Tests ausführen.
- Erweiterte Ansätze
  - Builds, Unit Tests etc. können entfernt aufgerufen werden.
  - Bibliotheken, die ohnehin lokal vorliegen, werden benutzt.

# Kollaboration

- Texteditor mit Echtzeit-Kollaboration
- Unterschiedliche Kooperationsmodelle
  - Driver/Observer und Driver/Driver (concurrent)
  - Schneller Wechsel zwischen Driver/Observer
- Hohes Bewusstsein über andere Teilnehmer. Position des anderen Teilnehmers wird gekennzeichnet.
  - Datei, Cursorposition, Bildausschnitt, Änderungen

# Kommunikation

- Verbreitet
  - Chat
  - Instant Messaging
- Wünschenswert
  - VoIP

# Evaluationen von PP/DPP

- Bisher zwei gegensätzliche Ansätze
  - „Black-Box“ Fallstudien bei denen nur die erbrachten Artefakte in Bezug auf ihre Qualität untersucht werden.
  - „Händische“ Untersuchungen bei denen DPP-Sitzungen als Video aufgezeichnet werden und von Menschen ausgewertet werden.

# ECG-Einbindung 1/3

## ■ Mikroprozesse

- Forschungsbereich in der Software-Engineering Gruppe. Betreut durch Sebastian Jekutsch.
- Beispiele
  - Codeänderungen
  - Browsen durch den Code
  - Ausführung des Programms
  - Arbeitsunterbrechungen
- Das ElectroCodeoGram von Frank Schlesinger ist ein Eclipse-Plugin, welches die Mikroprozesse des Programmierer aufzeichnet.

# ECG-Einbindung 2/3

- Saros erweitert den ECG-Sensor um PP/DPP-spezifische Handlungen
  - Einleiten/Beenden von Sitzungen
  - Driver/Observerwechsel
  - Kommunikationen (Chat, Nachrichten, VoIP)
  - Aufmerksamkeitsfokus



# ECG-Einbindung 3/3

- Zum Abschluß eine Fallstudie zu Saros, mit der die tatsächliche Einsetzbarkeit gezeigt werden soll.
- Was könnte man untersuchen?
  - Wie oft schauen Driver/Observer auf gemeinsame Codestellen?
  - Wie oft wechseln sich Driver/Observer ab?
  - Wie stark wird Kommunikation eingesetzt?
  - Welche Phasen (Episoden) können beobachtet werden?



Danke für die Aufmerksamkeit

Fragen?

Riad Djemili  
([djemili@inf.fu-berlin.de](mailto:djemili@inf.fu-berlin.de))