

Diplomarbeit:
Qualitative Analyse der Abstraktionsniveaus von Driver und
Observer bei der Paarprogrammierung

Zwischenbericht

Diplomand: Jan Kettner

Betreuer: Stephan Salinger, Laura Plonka

Inhalt

Abstraktion

Reflective Practitioner

Reflective Ladder

Reflektion & Abstraktion

Todo



Paarprogrammierung - Rollen

Laurie Williams aus Strengthening the Case for Pair Programming:

*"One partner, the driver, controls the pencil, mouse, or keyboard and writes the code. The other partner continuously and **actively observes** the driver's work, **watching for defects, thinking of alternatives**, looking up resources, and **considering strategic implications**"*

Dick & Zarnett aus Paired Programming and Personality Traits:

*"The first is responsible for the typing of code (the driver); the second is responsible for **strategizing and reviewing** the problem currently being worked on (the navigator)"*

Abstraktion

Wie definiere ich Abstraktion?

Abstraktion als Komplexität mit der ein System betrachtet wird. Je höher das Abstraktionsniveau desto weniger Detail wird betrachtet.

Auf die Paarprogrammierung bezogen:

- Untersuchung auf Basis des Foundation Layers.
- Hierbei Focus auf Quotations in und um think_aloud Codes herum.
- Die Quotations der Partner wurden auf Komplexität bzgl des „Programmierens“ betrachtet.

Abstraktionsniveau bzgl. Produkt-Codes

Erster Ansatz bezog sich nur auf Komplexität der Betrachtung des Codes

Feld - Variablen, Instanzen, Werte

Loc – Breakpoints

Block - if then else, schleifen

Methode - Methoden, Konstruktoren

Klasse

Package

Applikation

Meta - IDE, offtopic, etc

Reicht Kodekomplexitätssicht für Abstraktion?

Aber: Noch kein Bezug zum Foundation Layer.

-Abstraktion lässt sich nicht allein mit Komplexität der Betrachtung des Codes beschreiben

-z.B. kann bei knowlegde Codes der Bezug zum Code für Abstraktionsniveau weniger wichtig sein

Deshalb: Abstraktionsniveaumodule

-eigene Abstraktionsniveaumodule für bestimmte „Bereiche“ des Foundation Layer

-z.B. Design (Kodekomplexitätssicht), Prozess, generische Codes

Weitere Abstraktionsniveaumodule - Ideen

Prozess

- im Foundation Layer existiert gewisse Einteilung (step, strategy)
- feinere Granularität notwendig?

Generische Codes

- ???

weitere Module notwendig?

Sollte im Verlauf des weiteren Kodierens untersucht werden



PP Rollen bei Orit Hazzan

Orit Hazzan, Technion, Haifa, Israel
- forscht im Bereich SE u.a. mit GT

zu Abstraktionsniveaus von Driver und Observer:

„Since the two individuals work on different levels of abstraction - the driver thinks locally; the Navigator thinks globally - the same task is considered on two different levels of abstraction at the same time.“

Abstraktionsniveaus:

- global view z.B. im Gespräch mit dem Kunden
- local view z.B. während des tatsächlichen Programmierens
- verschiedene Niveaus dazwischen möglich

zitiert Reflective Practitioner (D. Schön) und behauptet dass sich diese Theorie auch auf SE anwenden lässt

Reflective Practitioner

Donald Schön "*Educating The Reflective Practitioner*" (1983):

zentrale Idee:

bestimmte Berufe (z.B. Krankenschwester, Sozialarbeiter, Städteplaner) werden weniger durch theoretisches als durch praktisches (aus Erfahrung gewonnen) Wissen bestimmt

- Schön bezeichnet diese als Practitioner
- Practitioner vs Professionals (Ärzte, Juristen)
- weiteres Beispiel: Eingewöhnungszeit nach dem Studium
- Schön schlägt vor mehr Praxis in die Ausbildung einzuführen (reflektieren über spez. reale Situationen, anstatt theoretisches Wissen auswendig lernen)
- Lernen aus der Reflektion über den Umgang mit diesen Situationen
- Beispiel dafür: Übungen im Informatikstudium

Reflective Practitioner – zentrale Begriffe

knowing in action

- entwickelt von implizitem Wissen
- Wissen das unreflektiert während einer Aktion angewendet wird
- schwer zu dokumentieren

reflection in action

- Schön: *"think what they are doing while they are doing it"*
- unmittelbare Gedanken zur aktuellen Aktion,
- kann diese Aktion beeinflussen

reflection on action

- Reflektion nach dem Ereignis
- bewusst durchgeführt

Reflective Ladder

- Schön beschreibt die verschiedenen Niveaus der Reflektion im Gespräch zwischen Lehrer und Schüler als verschiedene Stufen einer sog *"Ladder of Reflection"*
- Auf diesen Stufen kann sich das Reflektionsniveau der Gesprächspartner bewegen.
- Eine Stufe höher entspricht der Änderung des Niveaus von Kommentaren zu einer Aktion hin zu Kommentaren die diese Aktion reflektieren. umgekehrt funktioniert dieser Schritt analog.
- Bei Schön hat die Ladder of Reflection 3 - 4 Stufen: *Designing, Description of Designing, Reflection of Description of Designing, (Reflection of Reflection of Description of Designing)*

Anwendung der Reflective Ladder

Designing

reflection in Action, verbalisieren der Handlung ähnlich dem thinkaloud, unmittelbar und unreflektiert, aber auch Kommentare, die auch vom Observer kommen können

Beispiele:

„Wir sollten mal einen Breakpoint hier machen“

„Dann schiess ich es gleich mal ab“, „Ich mach jetzt mal ein Komma“

Description of Designing

reflection on Action, Beschreibungen, Kritik, Fragen, Kommentare, Annerkennung , Erklärungen bzw. Begründungen

Beispiele:

„Breachpunkte kannst du einfach so setzen“

„wenn er überhaupt in die Methode reingeht, kann ja sein, dass er...“

„Exactly“

Anwendung der Reflective Ladder 2

Reflection of Description of Designing

reflection on Action, Aha Erlebnis, Erkenntnis zu der Handlung, Reflektion über die Äußerungen des Anderen

Beispiele:

„Können wir uns gleich mal abkucken wie das bei ... gemacht wird und dann...“

„reinschmeissen muss ich aber Key und nur Key“

„man könnte fast denken, dass die durch Komma...(getrennt sind)“

Reflection of Reflection of Description of Designing

sinnvoll?



Reflektion = Abstraktion ?

Reflektion ist nicht gleichzusetzen mit Abstraktion. Schön verwendet den Begriff Abstraktion nicht.

Hazzan formuliert die These, dass in einem Gespräch bei steigendem Reflektionsniveau auch das Abstraktionsniveau steigt.

In der Literatur wird ausser von Hazzan, der Begriff Abstraktionsniveau im Zusammenhang mit der PP nicht verwendet.

Problem: In der Literatur werden Unterschiede in Verhaltens- und Denkweise zwischen Driver und Observer beschrieben. Zu Beginn der Arbeit wurde davon ausgegangen, dass es sich hierbei um unterschiedlichen Abstraktionsniveaus handelt.

Beim Kodieren ist aufgefallen, dass das sich mit dem Abstraktionsniveau diese Unterschiede nicht hinreichend erfassen lassen.

Reflektion vs Abstraktion ?

Noch einmal die Rollenverteilung in der Literatur:

- „*strategizing and reviewing the problem*“
- „*observes the driver’s work, watching for defects, thinking of alternatives, ..., considering strategic implications*“

beschreibt eher Reflektions- als Abstraktionsniveau

Ich möchte daher Reflektions- und Abstraktionsniveau untersuchen.

„Kreuzprodukt“ aus den Abstraktionsniveaumodulen und Reflective Ladder

- unterschiedliche Niveaus für zB Loc möglich

Todo

Im Laufe des weiteren Kodierens

- Modell überprüfen
- welche Module sind notwendig
- Modelle für Module entwickeln