

Perspektivierung mit Anaphern in Java

Sebastian Lohmeier
BSE, 21.7.2017

4 Runden

Demo & Einführung
Überblick

Webcam-Eye-Tracking
Design der 1. Fallstudie

Runde 1/4

Demo & Einführung

Runde 1/4

Demo & Einführung

(Demo)

Runde 1/4

Demo & Einführung

- Textweltmodell vs. AST
- Text vs. Code: 0:1
 - Individuelle Perspektivierung
- Edit-time resolution
 - Benutzerschnittstelle (View): je Benutzer optimiert
 - Wissensrepräsentation (Datei): Compiler-optimiert
- 2 Modelle von Anaphern: kognitiv vs. “formal”

Runde 2/4

Überblick

Runde 2/4

Überblick

Vorherige Arbeiten, Fragestellungen, Karte

Runde 2/4

Überblick

Vorherigen Arbeiten

- Garrod & Terras (2000)
- O'Reilly & McNamara (2007)
- Masterarbeit

Runde 2/4

Überblick

Vorherige Arbeiten

- Garrod & Terras (2000): indirekte Anaphern führen zu längeren Fixationdauern auf dem Folgewort als direkte Anaphern, aber **nicht**, wenn die Relation zwischen Anapher und Bezugsausdruck dominant ist
- “The teacher was busy *writing* an exercise on *the blackboard*. However, she was disturbed by a loud scream from the back of the class and *the chalk* dropped on the floor.”
- vs. letter + pen

Runde 2/4

Überblick

Vorherige Arbeiten

- Garrod & Terras (2000): indirekte Anaphern führen zu längeren Fixationdauern auf dem Folgewort als direkte Anaphern, aber **nicht**, wenn die Relation zwischen Anapher und Bezugsausdruck dominant ist
- “The teacher was busy *writing* an exercise on *the blackboard* [*with chalk*]. However, she was disturbed by a loud scream from the back of the class and *the chalk* dropped on the floor.”
- vs. letter + pen

Runde 2/4

Überblick

Vorherige Arbeiten

- Garrod & Terras (2000): indirekte Anaphern führen zu längeren Fixationdauern auf dem Folgewort als direkte Anaphern, aber **nicht**, wenn die Relation zwischen Anapher und Bezugsausdruck dominant ist
- “The teacher was busy writing an exercise on the blackboard [*with chalk*]. However, she was disturbed by a loud scream from the back of the class and the chalk dropped on the floor.”
- vs. letter + pen
- Menschen können Unterspezifikation verstehen

Runde 2/4

Überblick

Vorherige Arbeiten

- O'Reilly & McNamara (2007): *reverse-cohesion effect*:
Leser mit viel Hintergrundwissen können
Verständisfragen besser beantworten, wenn der
gelesene Text nicht so viele bereits bekannte Fakten
wiederholt.
- Erklärung: Aktivierung auf weniger Knoten verteilt
- Könnte auch mit indirekten Anaphern funktionieren

Runde 2/4

Überblick

Vorherige Arbeiten

- Masterarbeit: 19 Programmierer lesen 2x20 Einheiten Java-Code mit/ohne Anaphern
- Wissensaktivierung wird über Wiederholung von Teil-Ganzes-Beziehungen im Code variiert
- Eye-Tracking und Verständnisfragen
 - quasi Garrod & Terras + O'Reilly & McNamara
- Eye-Tracking-Daten in kognitives Modell eingeben
- Rund 1/3 der Eye-Tracking-Daten verworfen
- Mein erstes Experiment, u.a. Aufgaben zu schwer

Runde 2/4

Überblick

Fragestellungen

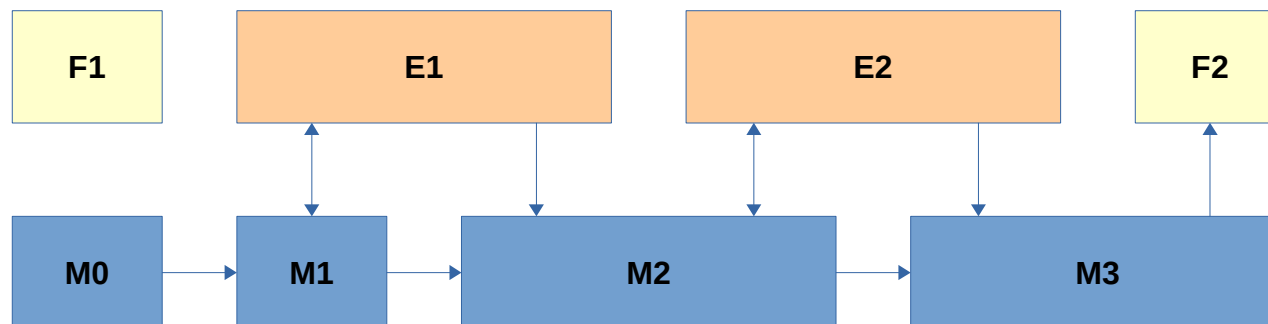
- Wie oft und wann verwenden Programmierer direkte und indirekte Anaphern?
- Unterstützen oder behindern Anapher das Code-Verstehen?
- Können RFL's die Genauigkeit von Eye-Trackern so weit verbessern, dass sie während der Programmierung genutzt werden können?
- Können Experimente mit Programmierern und Code Duffy & Rayner (1990), Garrod & Terras (2000) und O'Reilly & McNamara (2007) replizieren?
- Kann ein kognitives Modell die Ergebnisse der Experimente zu Fragen vorhersagen/erklären?
- Wie gut sagt das Modell zu 3.1 vorher, ob ein Programmierer eine Anapher mühelos verstehen wird?

Runde 2/4

Überblick

Karte

- F1: Anaphern schreiben, Eye-Tracking-Korrektur
- E1: Abstand zwischen Anapher und Bezugsausdruck
- E2: Einfluss von Hintergrundwissen (N=100)
- F2: individuell passende Anaphern anzeigen



Runde 3/4

Webcam-Eye-Tracking

Runde 3/4

Webcam-Eye-Tracking

WebGazer (Papoutsaki et al., 2016)

PACE (Huang et al., 2016)

Runde 3/4

Webcam-Eye-Tracking

(Demo)

Runde 4/4

Design der 1. Fallstudie

Runde 4/4

Design der 1. Fallstudie Anaphernverwendung & Eye-Tracking

- Eclipse-Plugin
- 4 Wochen, 30+ Entwickler im Job
- Fragestellungen
 - Wie oft und wann verwenden Programmierer direkte und indirekte Anaphern?
 - Können RFL's die Genauigkeit von Remote-Eye-Trackern so weit verbessern, dass sie während der Programmierung genutzt werden können?

Runde 4/4

Design der 1. Fallstudie Anaphernverwendung

- Wie oft und wann verwenden Programmierer direkte und indirekte Anaphern?
- 3 Gruppen/Referentialisierungsstrategien:
 - Alle lokalen Variablen werden als direkte/indirekte Anaphern angezeigt.
 - Nur vom jeweiligen Programmierer eingegebene Anaphern werden als Anaphern angezeigt.
 - Es wird nicht unterspezifiziert. Anaphernresolution nur zur Code-Generierung.
- Dürfen Benutzer selbstbestimmt zwischen Gruppen wechseln?
- Können Identifier übertragen werden?
- Kann "Kontext" erhoben werden?
- Kann mit den Teilnehmern kommuniziert werden?
- Distanz, aber dafür mehr Ergebnisse – auch für seltene Arten von Anaphern, oder besser vor Ort sein und mehr Kontext zu bekommen?

Runde 4/4

Design der 1. Fallstudie Anaphernverwendung

- Wie oft und wann verwenden Programmierer direkte und indirekte Anaphern?
- Welche Arten von Anaphern werden wie oft verwendet?
- Verwenden Programmierer mit der Zeit andere Arten?
- Welche Wörter werden (wie oft) als Anaphern/Begzugsausdruck verwendet?
 - Kombinationen und Distanzen zwischen beiden.
- Wie viele Versuche sind notwendig, um eine Anapher erfolgreich aufzulösen?
- Wie lange sind Anaphern im Code enthalten, bevor sie gelöscht/geändert werden?
- Wenn der Bezugsausdruck gelöscht wird:
 - Ergibt das einen Compilerfehler oder wird korrekt neu aufgelöst?
 - Korrigiert der Programmierer dann die Anapher?
 - Löscht der Programmierer die Anapher?
- Auswirkungen der Referentialisierungsstrategien

Runde 4/4

Design der 1. Fallstudie Eye-Tracking

- Können RFL's die Genauigkeit von Remote-Eye-Trackern so weit verbessern, dass sie während der Programmierung genutzt werden können?
- Alle Experimente brauchen auch Eye-Tracking
 - um Aktivierung zu ermitteln
 - Alternativ: Texteingabe analysieren (manual module)
 - um die Ergebnisse zu messen.
- Können die Entwickler mit tollen Features motiviert werden, um das Eye-Tracking zu verwenden?
- Wie viel Kontext soll/kann ermittelt werden, um später das Tracking zu verbessern?
 - Bilddaten
 - Klickdaten
 - Rechte Maus, um Kontextmenü zu öffnen
 - <BS>/ um Text zu löschen
 - Doppelklick, der Wort markiert
 - DnD, um Wörter zu markieren

Runde 4/4

Design der 1. Fallstudie Eye-Tracking

- Können RFL's die Genauigkeit von Remote-Eye-Trackern so weit verbessern, dass sie während der Programmierung genutzt werden können?
- Zeitliche und räumliche Verteilung der Klicks
- Wie lange dauert es, ausreichend viele Klicks zu sammeln?
- Wie genau kann kalibriert werden?
- Kann auch mit schlechter Kalibrierung gearbeitet werden?
 - Müssen dann mehr Daten gesammelt werden?
 - Verträgt die Anwendung false positives?

Runde 5/4

Fragerunde