



Universität Hamburg

DER FORSCHUNG | DER LEHRE | DER BILDUNG

Tim Krämer

7kraemer@informatik.uni-hamburg.de

**Archimatrix -**  
Improved Software Architecture  
Recovery in the Presence of  
Design Deficiencies

08. Mai 2013





## Einordnung und Umfeld

- wissenschaftliches Paper zum „Archimetrix“-Projekt
- in Zusammenarbeit mit DFG und SFB „On-The-Fly Computing“  
Universität Paderborn

### Autoren:

- Platenius, M. C., von Detten, M., und Becker, S.
  - Mitarbeiter Fachgruppe Softwaretechnik, Heinz Nixdorf Institut
  - Mitarbeiter SFB 901 „On-The-Fly Computing“
  - Platenius u. von Detten: Informatik Master, Universität Paderborn
  - Becker: Juniorprofessor Softwaretechnik, Universität Paderborn

### Verwandte Arbeiten:

- Aufbauend auf: Platenius, M. C. (2011). Reengineering of design deficiencies in component-based software architectures.



# Gliederung

- Einleitung
- Grundlagen
  - Reverseengineeringmethoden
  - Ablauf des „Archimetrix“-Ansatzes
- Erweiterungen des Reengineeringprozesses
- Methode der Autoren
  - Erkennung / Identifikation
  - Ranking
  - Vorschau Korrekturauswirkungen
- Ergebnisse und Implementation
  - Aussichten und Grenzen
- Bewertung des Papers
  - Einordnung im wissenschaftlichem Umfeld
  - Kritische Betrachtung



# Grundlagen

- kompetentenbasierende Softwarearchitekturen
- Clustering-Based Reverse Engineering (SoMoX)
- Metriken
  - Coupling („Nähe zweier Gruppen von Klassen“)
  - Design Deficiencies (z.B. Interface Violations)
- Mängelbeseitigungsstrategien (removal strategies)
  - manuell vom Reengineerer
  - automatisch durch vordefinierte Strategien (z.B. anhand von Interfaces)



# Beispiel für Architektur mit Mängeln

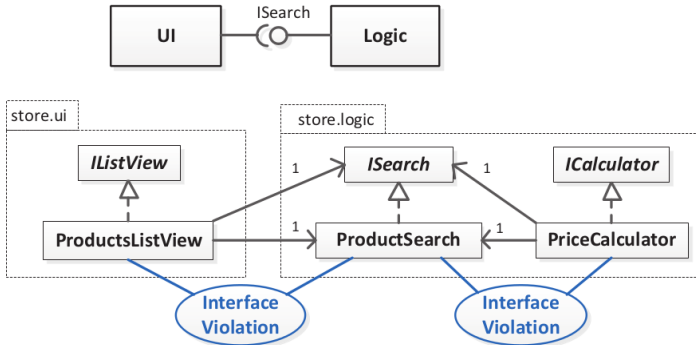


Abbildung : Running Example



# Archimatrix-Ansatz

## Annahmen und Anforderungen

- bestehende Implementationsmängel erkennen (z.B. Interface Violations)
- „clustering tools“ wesentlich schneller als menschliche Reverse Engineerer
- Idee: combined reverse and reengineering für besseres Architekturmodell

## Methode der Autoren

- Komponentenbasierten „clustering“ Ansatz um 3 Schritte erweitern:
  - Identifizieren von lohnenswerten Komponenten
  - Ranking der erkannten Mängel
  - Vorschau der Auswirkungen einer Korrektur



# Semiautomatischer Reengineeringvorgang mit Archimetrix-Ansatz

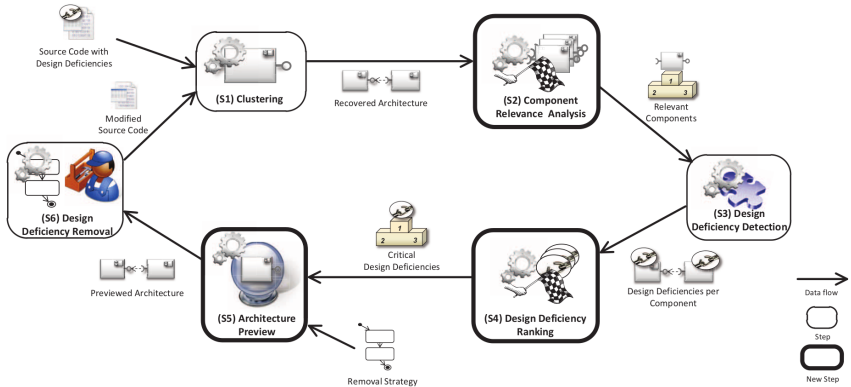


Abbildung : Reengineering process

# Schritt 1: „Component Relevance Analysis“



Bei welchen Komponenten lohnt sich eine genauere Betrachtung?

■ Identifikation durch:

- Komplexitätsmetrik (Complexity Metric)
- Angrenzungsmetrik (Closeness to Threshold Metric)

Ansatz der Autoren: Beide Metriken gleich gewichtet





# Schritt 1: „Component Relevance Analysis“

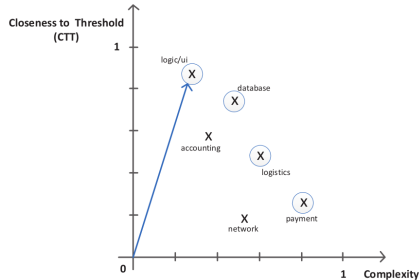


Abbildung : Relevance analysis result calculation

höchste Relevanz nach Pareto-Optimum und geometrischer Distanz:  
logic/ui Komponente



## Schritt 2: „Design Deficiency Ranking“

Ranking der relevanten Komponenten durch Metriken:

- „Class Locations Metric“
- „External Accesses Metric“
- „Higher Interface Adherence Metric“

Berechnung des Rankings durch Pareto-Optimum



## Schritt 3: „Architecture Preview“

Wie beeinflusst die gewählte Mängelbeseitigungsstrategie die resultierende Architektur?

- Mängelbeseitigung durch:
  - vordefinierte Strategien
  - manuelle Beseitigung durch Reengineerer (keine Vorschau)

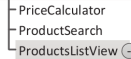


## Schritt 3: „Architecture Preview“

	Original Architecture	Predicted Architecture
Total Number of Components	1	2
Number of Primitive Components	1	2
Number of Composite Components	0	0
Total Number of Interfaces	3	3

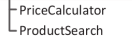
Original Architecture:

< comp 1 > (3 classes)



Predicted Architecture:

< comp 1' > (2 classes)



< comp 2 > (1 class) +

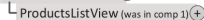


Abbildung : Architecture preview example

Unterschiede zwischen der ursprünglichen Architektur und der Vorschau sind hervorgehoben.

# Ergebnisse und Implementation



Autoren implementieren Ansatz in gleichnamiges Tool „Archimetrix“

- Anwendung auf Referenzimplementation des Common Component Modeling Example (CoCoME)
- 127 Klassen, über 5000 Zeilen Java Code, mehrere Designmängel
- zum Vergleich: dokumentierte angedachte Architektur

# Ergebnisse und Implementation



Fragestellung:

1. Ist die berechnete Relevanz der im ersten Schritt ausgewählten Komponente ein guter Indikator für die Erkennung von lohnenswerten Designmängeln?
2. Führt das Beheben der Mängel, die im zweiten Schritt ein hohes Ranking erfahren haben, zu Veränderungen in der resultierenden Architektur? Und im Gegensatz, führen die niedrig gerankten zu keiner Veränderung?
3. Ist die resultierende Architektur, nach Behebung der relevanten Designmängel näher an der dokumentierten, angedachten Architektur?

# Ergebnisse



1. Ist die berechnete Relevanz der im ersten Schritt ausgewählten Komponente ein guter Indikator für die Erkennung von lohnenswerten Designmängeln?
  - In den jeweiligen Top 3 Komponenten wurden Designmängel entdeckt, jedoch unter großem Performance Verlust. (Größe der Komponenten beachten!)

# Ergebnisse



2. Führt das Beheben der Mängel, die im zweiten Schritt ein hohes Ranking erfahren haben, zu Veränderungen in der resultierenden Architektur? Und im Gegensatz, führen die niedrig gerankten zu keiner Veränderung?
  - Die Auftreten von Designmängeln, deren Korrektur zu keiner Architekturveränderung geführt haben, sind niedrig gerankt worden.



# Ergebnisse



3. Ist die resultierende Architektur, nach Behebung der relevanten Designmängel näher an der dokumentierten, angedachten Architektur?
  - Der Vergleich mit der CoCoME dokumentierten Referenzarchitektur zeigt eine Verbesserung zum vorherigen Verfahren.

# Aussichten und Grenzen



1. Die meisten der Fragen positiv beantwortet
2. aber: Ansatz basiert auf Annahme, dass eine Art von Architekturmodell vorliegt
3. Prozess basiert auf Heuristiken, aber endgültige Entscheidung bleibt beim Entwickler
4. die größte Komponente ist immer die relevanteste, beide Metriken beziehen Größe ein



# Bewertung des Papers

- Interessanter Einstieg (Abstract, Einleitung)
- gut verständlich und nachvollziehbares Paper
- Detaillierte Erklärung des Ablaufs
- umfangreiche Erläuterung von Metriken
- passende Zusammenfassung mit angemessener Diskussion
- gute Einordnung in wissenschaftliches Umfeld
- genaue Quellenangaben



Danke für die Aufmerksamkeit!