

Smell Detection in Context

Kontextualisierung von automatisierten
Designmängelwarnungen

Sebastian Jancke, 2010

MOTIVATION
STATE OF THE ART
PROBLEME
STRUKTUREN
RELEVANZ
INTERLINKING
CASE STUDY

Smell Detection in Context

MOTIVATION

Ziel: Langfristige Evolvierbarkeit

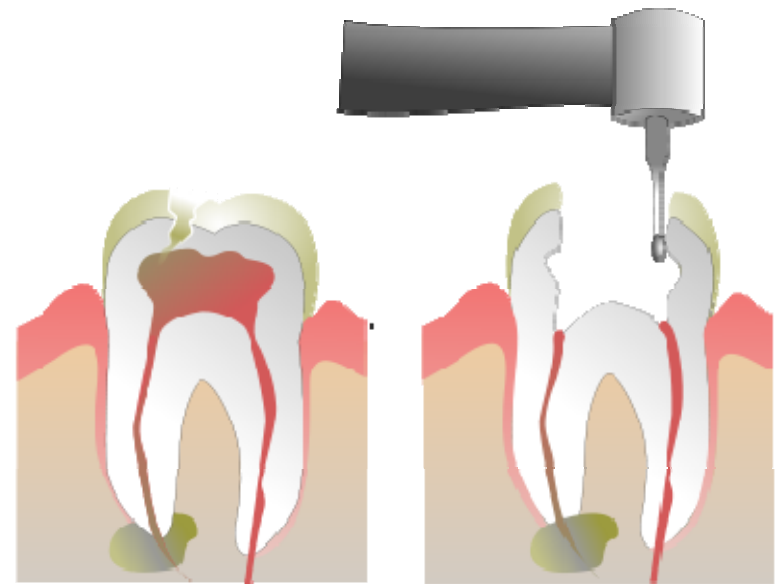
- Refaktorisieren
- Langfristig: Systeme effizient weiterentwickeln
- Kurzfristig: höherer Aufwand

 Werkzeuge zur Refaktorisierung und **Analyse**

Refaktorisieren



**Zahnseide
(Floss)**



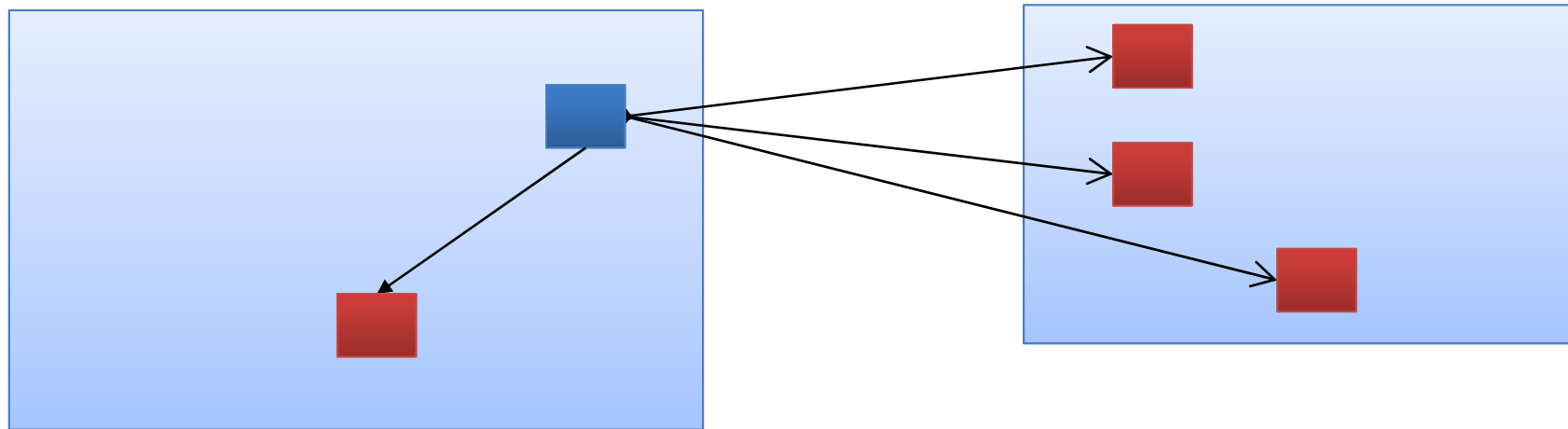
**Wurzelbehandlung
(Root Canal)**

MOTIVATION
STATE OF THE ART
PROBLEME
STRUKTUREN
RELEVANZ
INTERLINKING
CASE STUDY

Smell Detection in Context

STATE OF THE ART

Beispiel: Neid / Feature Envy



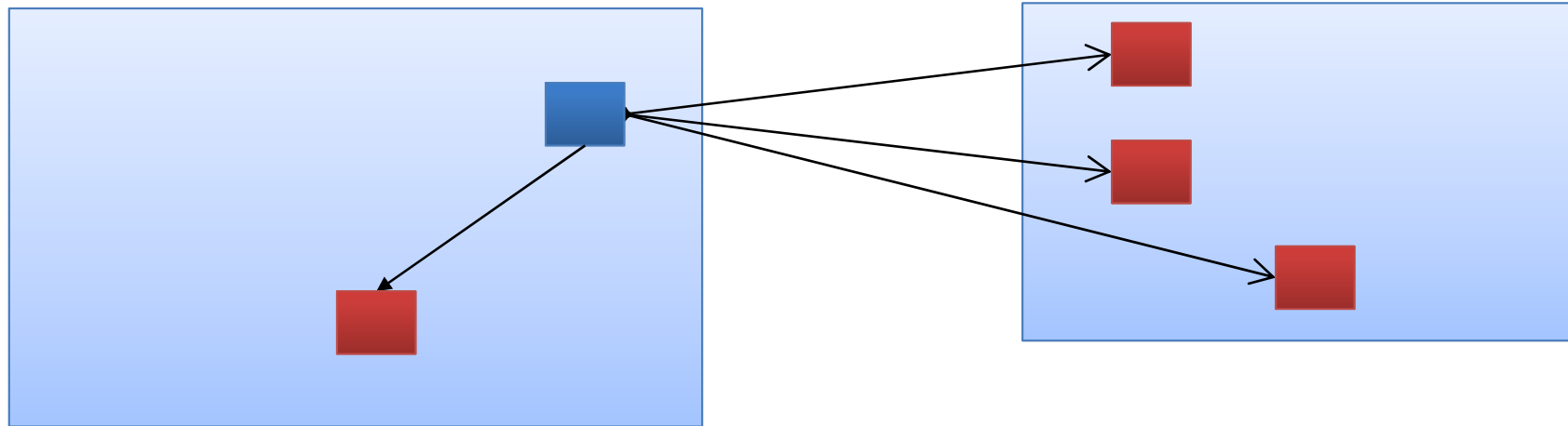
 Methode

 Daten

Design Probleme Erkennen

- Beck & Fowler: Smells
 - Ziel für Anwendung einer Refaktorisierung
 - Basiert auf „Intuition“
- Marinescu: Design Flaw
 - Abweichen von einem Qualitäts-Kriterium auf höherem Abstraktions-Niveau (als nur eine Metrik)
 - Basiert auf Metriken

Beispiel: Neid



Einige (3) fremde Attribute

Verhältnis: $0.25 < \mathbf{Drittel}$

Daten gehören zu **Einigen** (1) Klassen

Detection Strategies

- Zusammengesetzte logische Bedingung
- Basiert auf Metriken
- Elemente mit speziellen Eigenschaften erkennen

- Filter: Schwellwerte aus Heuristiken ableiten
- Komposition: UND, ODER

Beispiel: Neid (Feature Envy)

```
smell_(feature_envy, MethodId) :-
```

```
    internal_method(MethodId),
```

```
    metric(method_access_to_foreign_data, MethodId, AFTD),
```

```
    AFTD > 2, %2-5 = few
```

```
    metric(method_locality_of_attribute_access, MethodId, LAA),
```

```
    LAA < 0.3, % 0.3 = third
```

```
    metric(method_foreign_data_providers, MethodId, FDP),
```

```
    FDP =< 5 %2-5 = few
```

```
    .
```

Schwellwerte

- Studie über 45 Java und 37 C++ Projekte
- “Welche Komplexität / Größen / ... sind üblich”

Metric	Java			
	Low	Ave- rage	High	Very High
CYCLO/Line of Code	0.16	0.20	0.24	0.36
LOC/Method	7	10	13	19.5
NOM/Class	4	7	10	15

Table 2.1. Statistical thresholds of 45 Java and 37 C++ projects for the size and complexity metrics used in this study.

Überblick: Detection Strategies

- „Gottklasse“
- Datenklasse
- Neid
- „Hirnklasse“
- Hirnmethode
- Eisberg
- Vermittler

- Schrotkugeln
Operieren
- Intensive
Kopplung
- Gestreute
Kopplung

- Ausgeschlagenes
Erbe
- Tradition brechen

Floss Refactoring Unterstützen (Murphy-Hill et al, 2008)

- Kontext-Sensitivität
 - Relevanz in der konkreten Entwicklungs-Aufgabe
- Verfügbarkeit
 - automatisiert, im Hintergrund
 - „Null-Aufwand“ (auch beim Verständnis)
- Skalierbarkeit
 - Nutzer nicht mit Information überladen
- Beziehungen
 - Zwischen Smells und Code

MOTIVATION
STATE OF THE ART
PROBLEME
STRUKTUREN
RELEVANZ
INTERLINKING
CASE STUDY

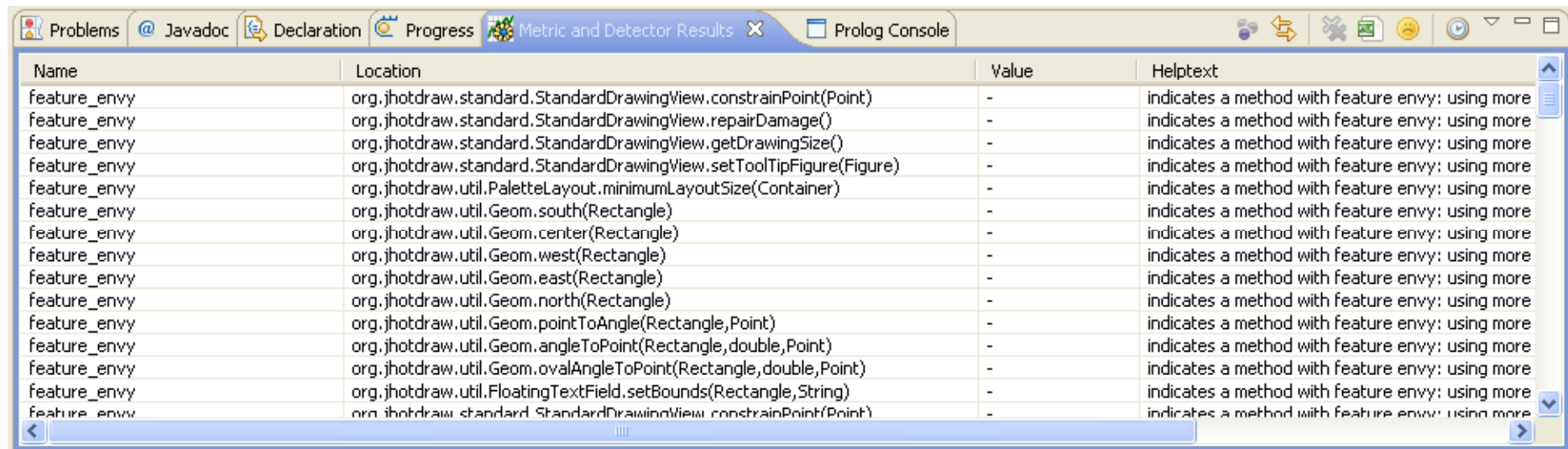
Smell Detection in Context

PROBLEME

Offene Probleme

- Umgang mit Falsch-Positiven
- Überflutung mit Informationen
- Ein Design Problem kann irrelevant sein
 - im Kontext der aktuellen Aufgabe
 - durch Stabilität über die Zeit
- Design Probleme werden isoliert vorgestellt

Aktuelle Werkzeugunterstützung



The screenshot shows an IDE window with several tabs: Problems, Javadoc, Declaration, Progress, Metric and Detector Results, and Prolog Console. The 'Metric and Detector Results' tab is active and displays a table with the following columns: Name, Location, Value, and Help text. The table lists 15 entries, all with 'feature_envy' as the name and a hyphen as the value. The help text for all entries is 'indicates a method with feature envy: using more'. The locations are various class methods from the 'jhotdraw' package.

Name	Location	Value	Help text
feature_envy	org.jhotdraw.standard.StandardDrawingView.constrainPoint(Point)	-	indicates a method with feature envy: using more
feature_envy	org.jhotdraw.standard.StandardDrawingView.repairDamage()	-	indicates a method with feature envy: using more
feature_envy	org.jhotdraw.standard.StandardDrawingView.getDrawingSize()	-	indicates a method with feature envy: using more
feature_envy	org.jhotdraw.standard.StandardDrawingView.setToolTipFigure(Figure)	-	indicates a method with feature envy: using more
feature_envy	org.jhotdraw.util.PaletteLayout.minimumLayoutSize(Container)	-	indicates a method with feature envy: using more
feature_envy	org.jhotdraw.util.Geom.south(Rectangle)	-	indicates a method with feature envy: using more
feature_envy	org.jhotdraw.util.Geom.center(Rectangle)	-	indicates a method with feature envy: using more
feature_envy	org.jhotdraw.util.Geom.west(Rectangle)	-	indicates a method with feature envy: using more
feature_envy	org.jhotdraw.util.Geom.east(Rectangle)	-	indicates a method with feature envy: using more
feature_envy	org.jhotdraw.util.Geom.north(Rectangle)	-	indicates a method with feature envy: using more
feature_envy	org.jhotdraw.util.Geom.pointToAngle(Rectangle,Point)	-	indicates a method with feature envy: using more
feature_envy	org.jhotdraw.util.Geom.angleToPoint(Rectangle,double,Point)	-	indicates a method with feature envy: using more
feature_envy	org.jhotdraw.util.Geom.ovalAngleToPoint(Rectangle,double,Point)	-	indicates a method with feature envy: using more
feature_envy	org.jhotdraw.util.FloatingTextField.setBounds(Rectangle,String)	-	indicates a method with feature envy: using more
feature_envy	org.jhotdraw.standard.StandardDrawingView.constrainPoint(Point)	-	indicates a method with feature envy: using more

Hypothesen der Diplomarbeit

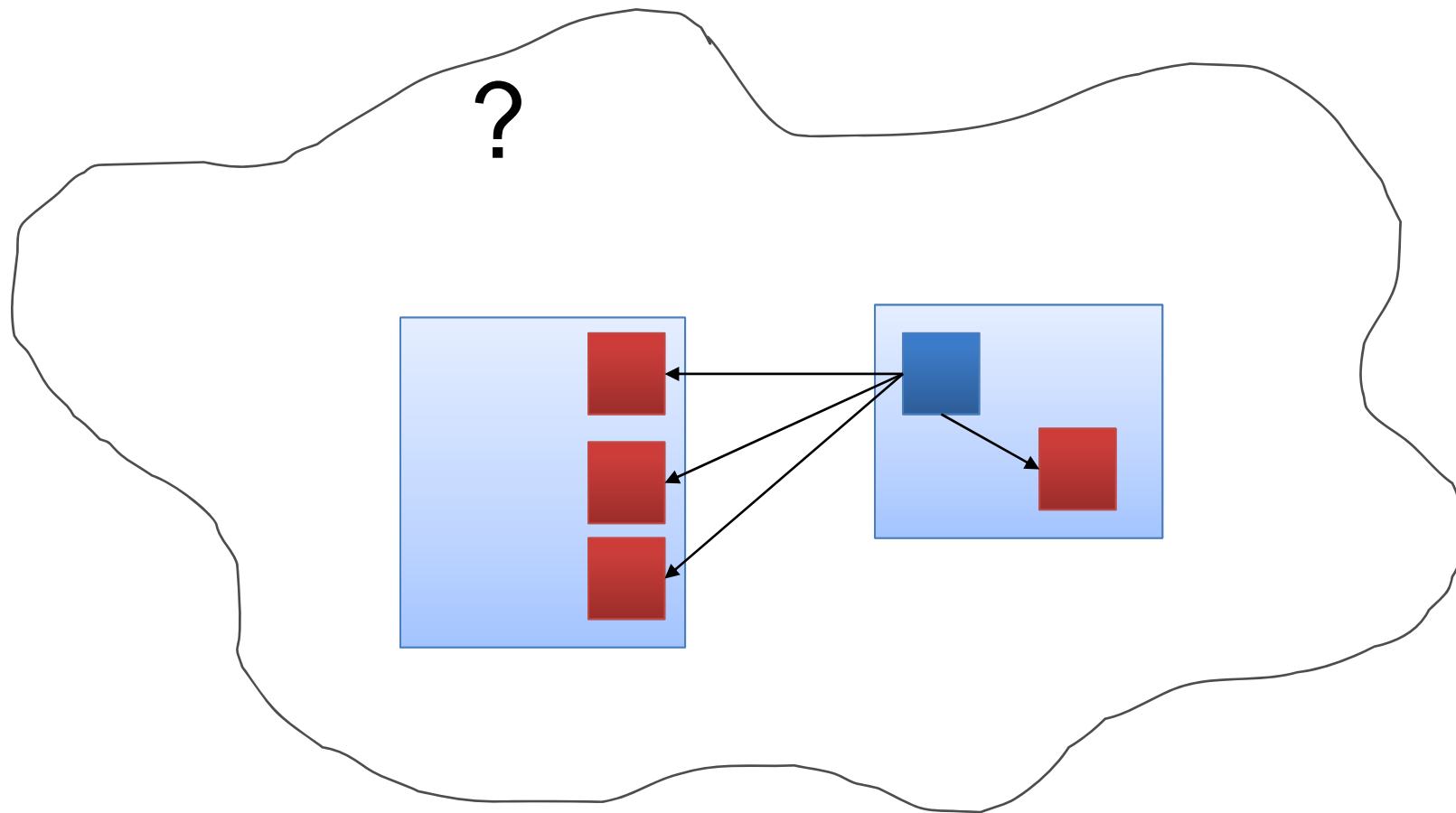
- Kontext-Bezug erleichtert Problem-Identifikation während des Floss Refactorings.
 - Strukturelle Hinweise reduzieren Falsch-Positive.
 - Gewichtung der Smells durch u.a. Historie und Task liefert nur relevante Smells im Projekt
 - Beziehungen zwischen Smells verbessern die Verständlichkeit eines Design Problems.

MOTIVATION
STATE OF THE ART
PROBLEME
STRUKTUREN
RELEVANZ
INTERLINKING
CASE STUDY

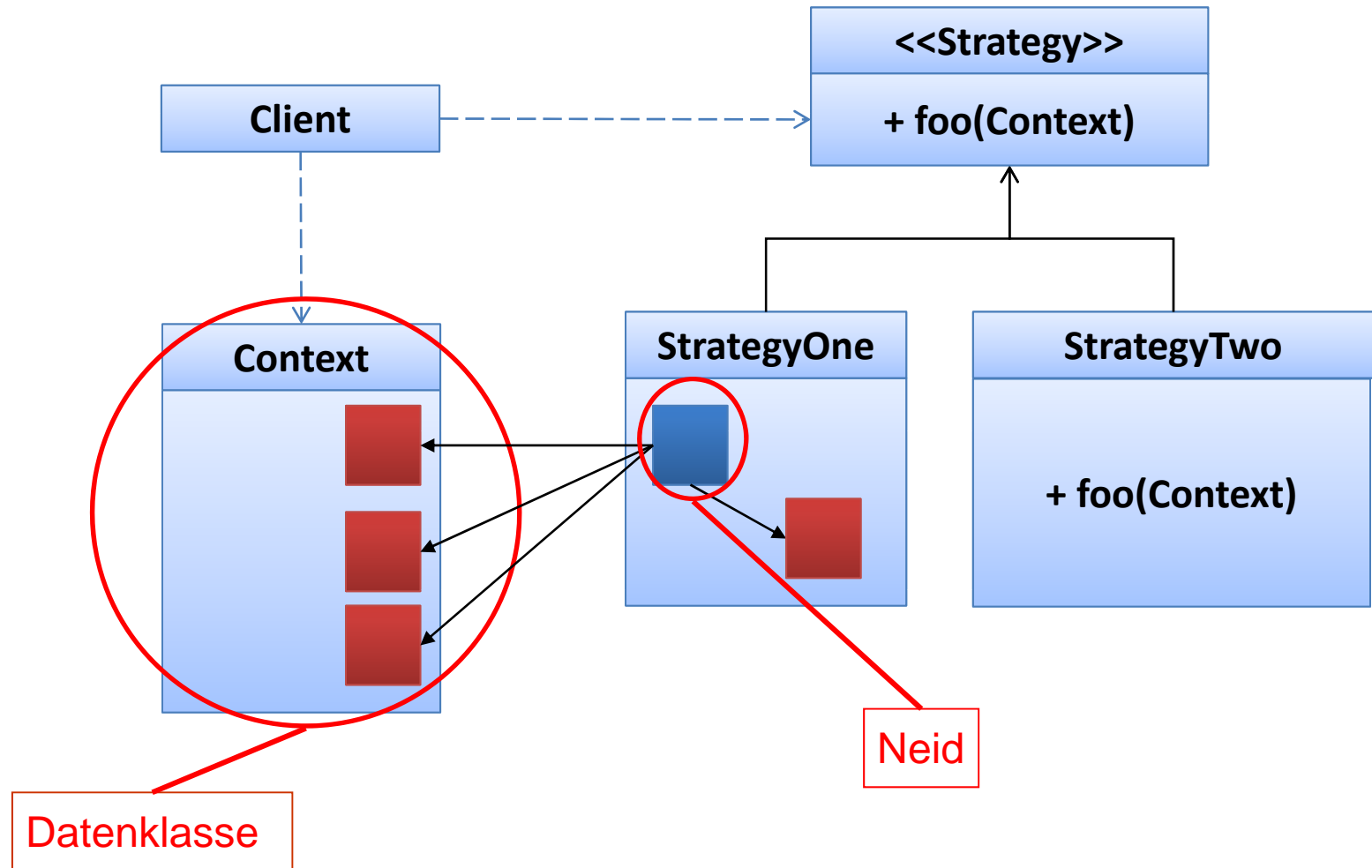
Smell Detection in Context

STRUKTUREN

Einflüsse auf Neid



Strategy Pattern impliziert Neid



Patterns haben Smells

Visitor	Neid, Datenklasse
Strategy	Neid, Datenklasse
(Abstract) Decorator	Vermittler, Ausgeschlagenes Erbe
(Abstract) Composite	Ausgeschlagenes Erbe
Adapter	Vermittler
Facade	Vermittler, Intensive oder Gestreute Kopplung
Mediator	
Creation Methods	Neid

Weitere betroffene Strukturen

- Embedded DSLs
- Prozedurale Komponenten
- Transaction Scripts
- Presentation Patterns
- APIs
- Tests
- ...

Embedded DSLs

- Auch: Internal DSL, Fluent API (Fowler, Evans)
- Stile:
 - Geschachtelte Funktionen
 - Methodenkettens
- Ziel: Lesbarkeit

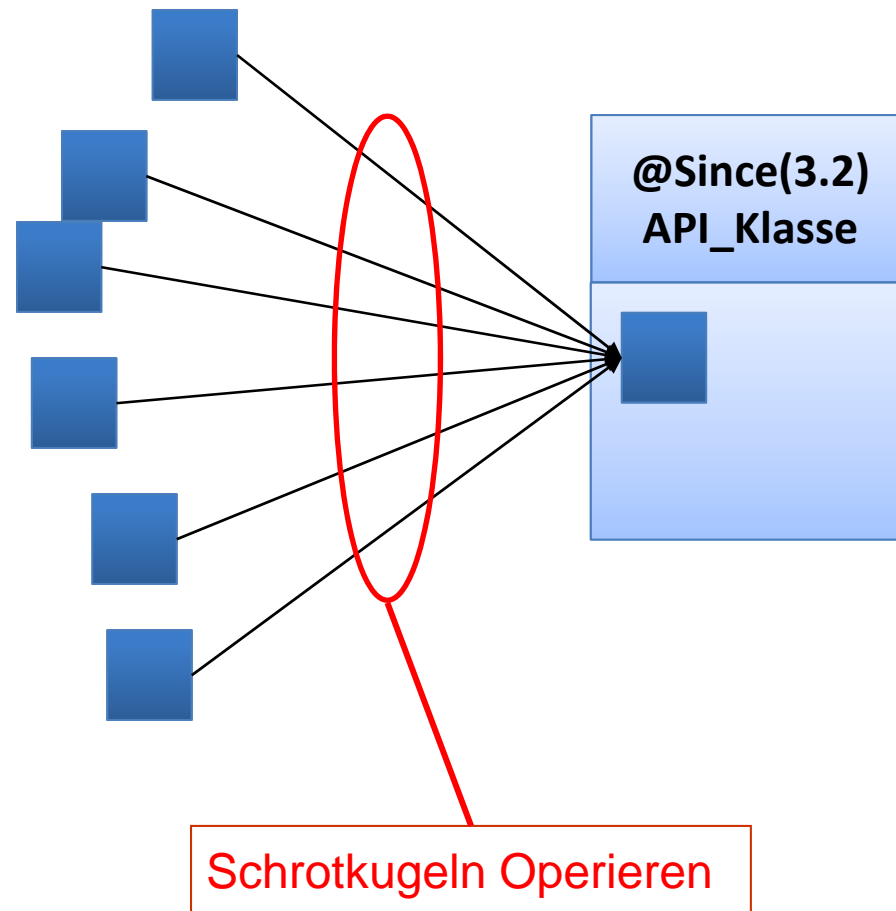
```
bind(CreditCardProcessor.class)  
    .annotatedWith(Names.named("Checkout"))  
    .to(PayPalCreditCardProcessor.class);
```

Embedded DSLs haben Smells

```
bind(CreditCardProcessor.class)
    .annotatedWith(Names.named("Checkout"))
    .to(PayPalCreditCardProcessor.class);
```

- Methoden Ketten
- „Law of Demeter“ Verletzung
- Möglicherweise auch
 - Gestreute Kopplung
 - Intensive Kopplung
- Abwägung: Smell - Lesbarkeit

APIs haben Smells



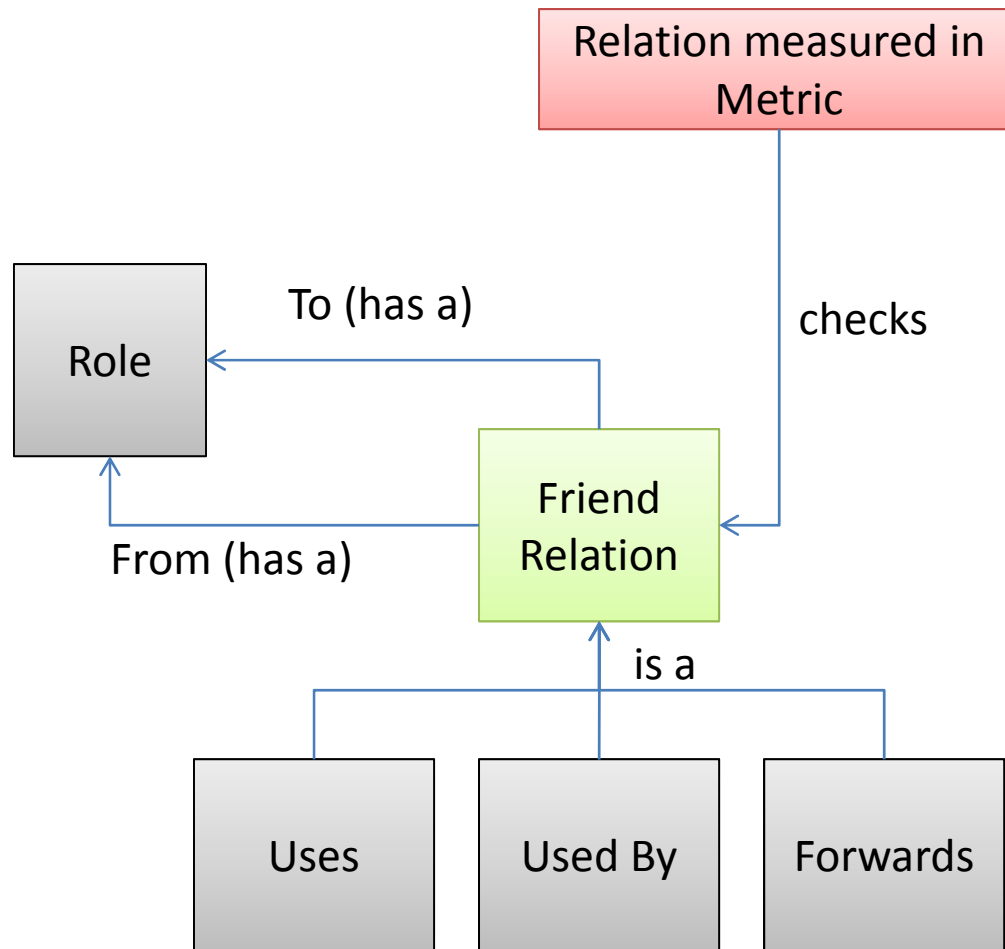
Unittests haben Smells

- Durch Erzeugung des Szenarios für den Test
 - Gestreute Kopplung
 - Neid
- Methode mit vielen Tests -> Viele Aufrufer
 - Schrotkugeln Operieren
- Falsch-Negative: Komplexität eines Tests ?!

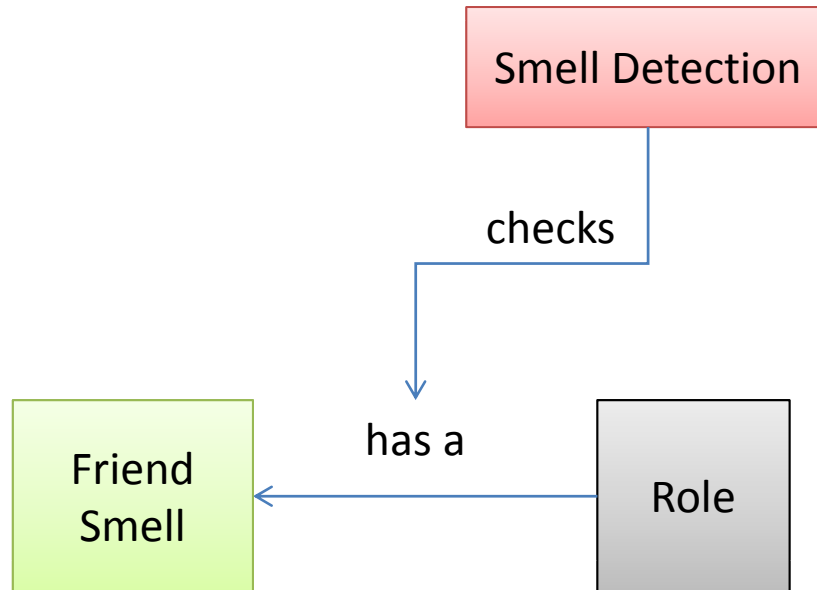
Falsch-Positive Reduzieren

- Strukturen können uns dabei helfen
- Strukturelle Hinweise:
 - Metadaten / Java-Annotationen („@Strategy“)
 - Namenskonventionen
 - Von Nutzer definierte Regeln
 - Pattern Detection

Friend Relations

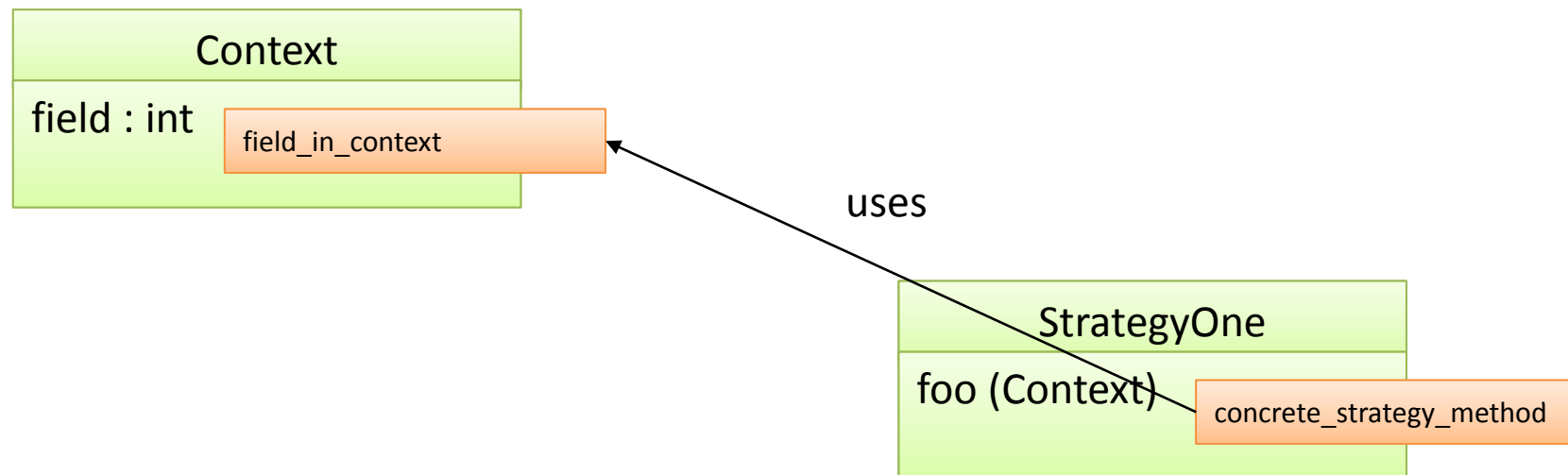


Friend Smells



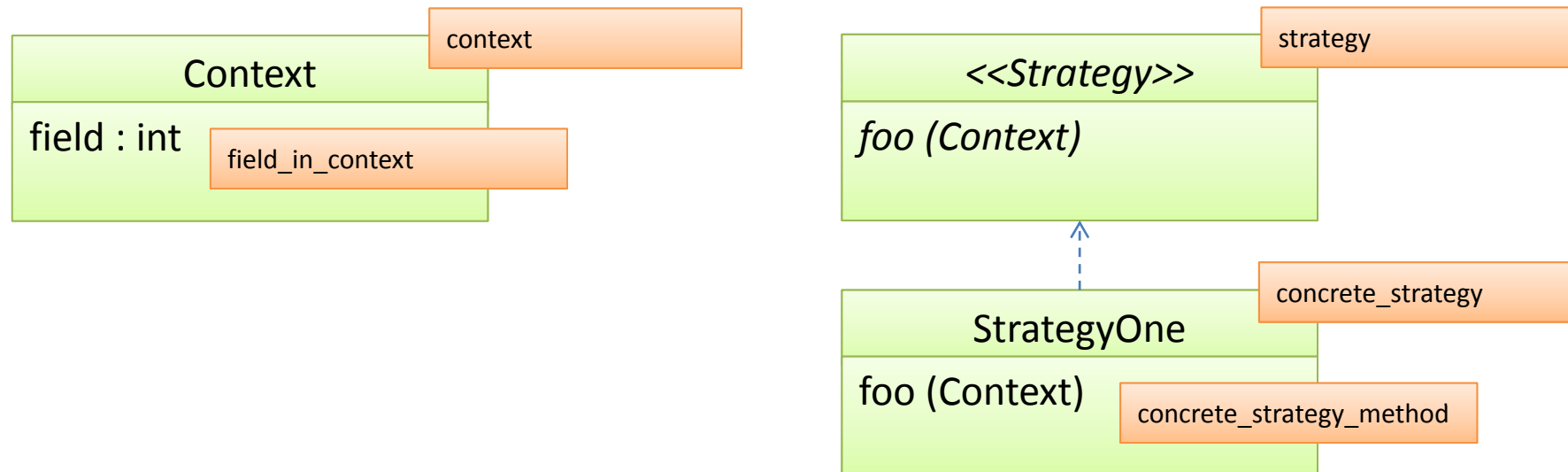
Strukturen definieren

```
structure (strategy_pattern) .  
role_uses (strategy_pattern,  
  concrete_strategy_method,  
  field_in_context) .
```



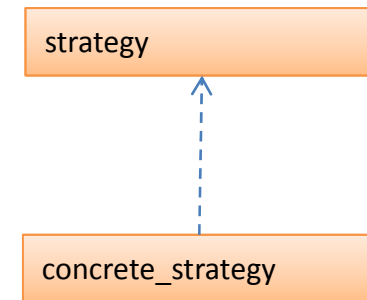
Annotierung mit Hinweisen

- `fqn_role(strategy_pattern, strategy, 'org.jhotdraw.framework.Connector')`



Ableitung der Rollen definieren

```
process_leaf_(strategy_pattern,  
             concrete_strategy,  
             strategy,  
             Instance, Instance,  
             ConcreteStrategy,  
             (  
                 subtype(ConcreteStrategy, Instance),  
                 not(is_interface(ConcreteStrategy))  
             )  
             ).
```



Falsch-Positive ausschließen

```
uses(method, attribute, MethodId,  
      AccessedAttribute) :-  
  
    . . . . ,  
not(friend_usage(method, attribute,  
                 Method, AccessesAttribute)).
```

Falsch-Positive ausschließen

```
friend_usage(method, attribute,  
             FromMethod, ToAttribute) :-  
  
    role(Structure, UsingRole, Instance,  
         FromMethod),  
  
    role(Structure, UsedRole, Instance,  
         ToAttribute),  
  
    role_uses(Structure, UsingRole,  
              UsedRole).
```

Ergebnis des Struktur-Kontext

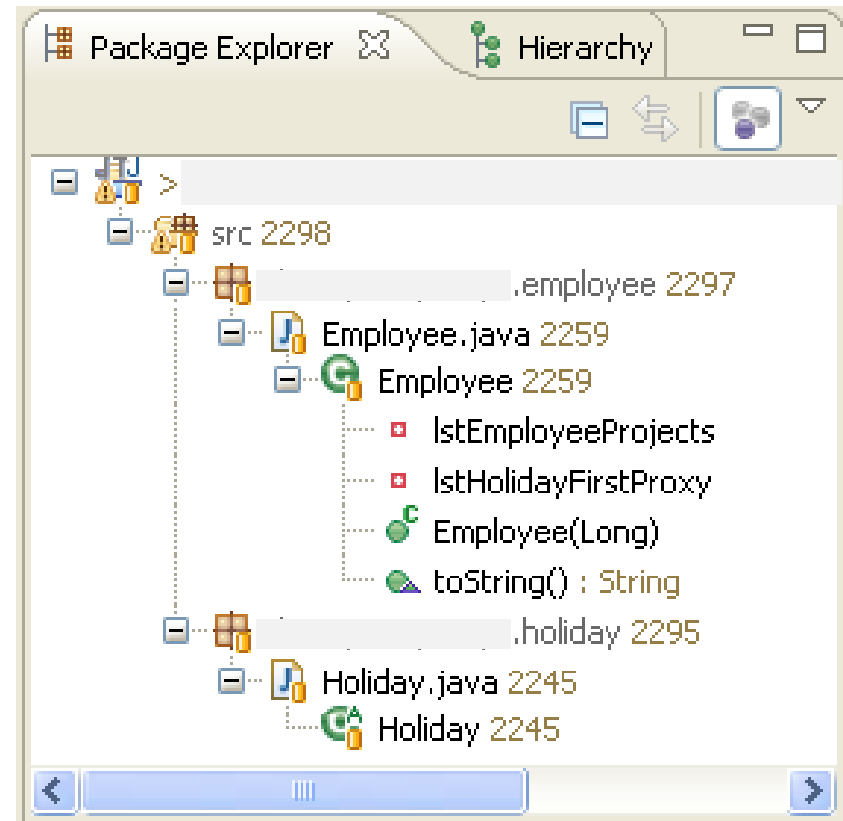
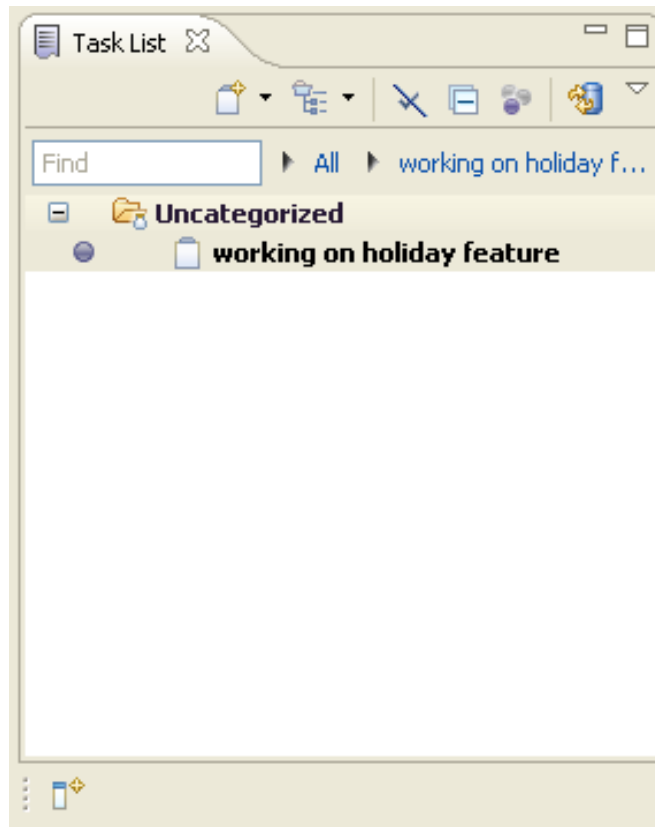
- Zusammenhang von Strukturen und Smells wird sichtbar
- Aussagekräftige Annotation
- Strukturen verursachen nun keine Falsch-Positiven

MOTIVATION
STATE OF THE ART
PROBLEME
STRUKTUREN
RELEVANZ
INTERLINKING
CASE STUDY

Smell Detection in Context

RELEVANZ

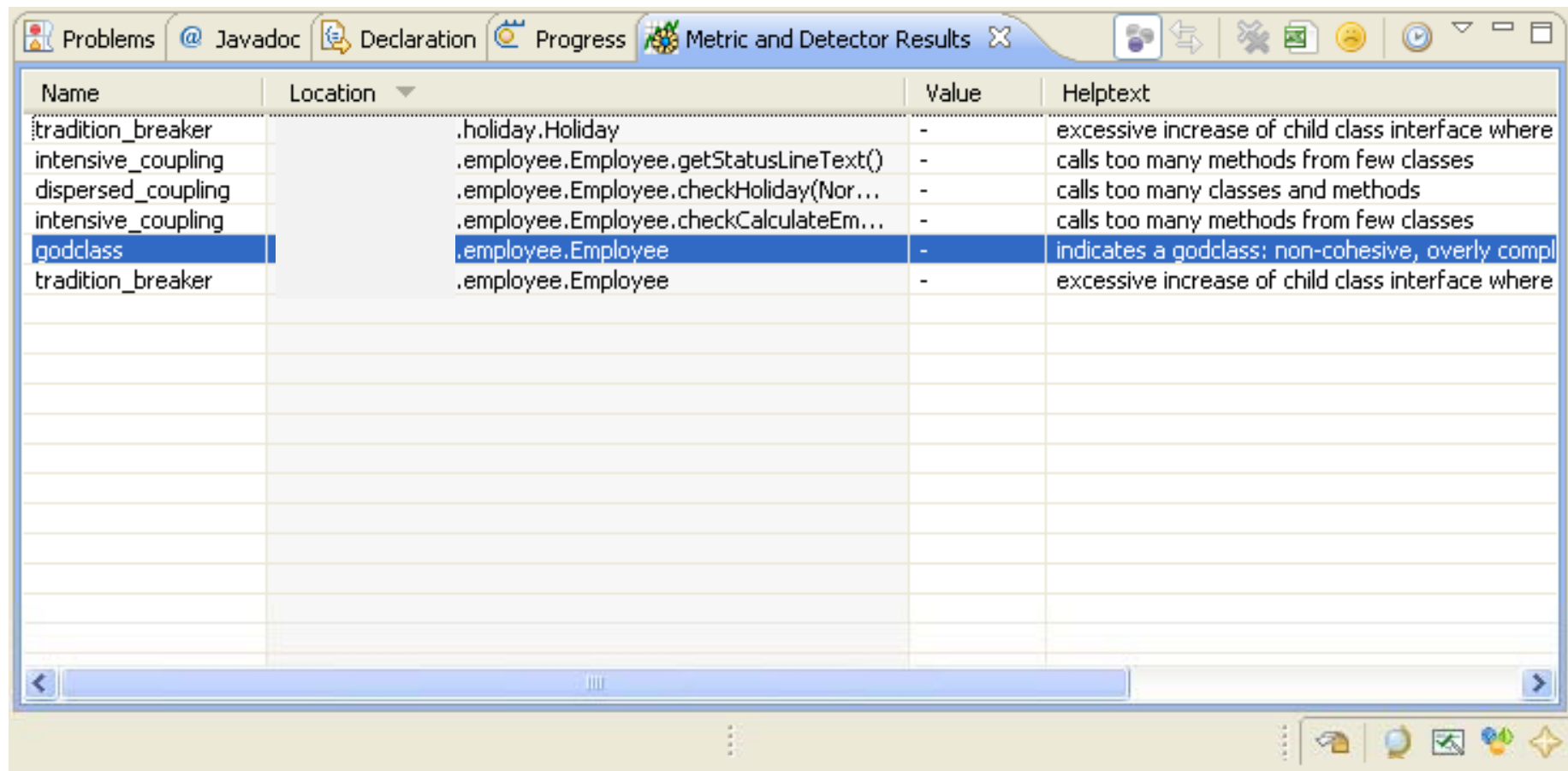
Fokussierung auf aktuelle Aufgabe



„Degree of Interest“ als Relevanz

- Interaktion mit Element A
 - erhöht DOI(A)
- Interaktion mit anderen Elementen
 - verringert DOI(A)
 - „Decay“
- Wenn $DOI(A) < 0$
 - Bei nächster Interaktion: DOI(A) startet mit 0

Filtern nach Relevanz



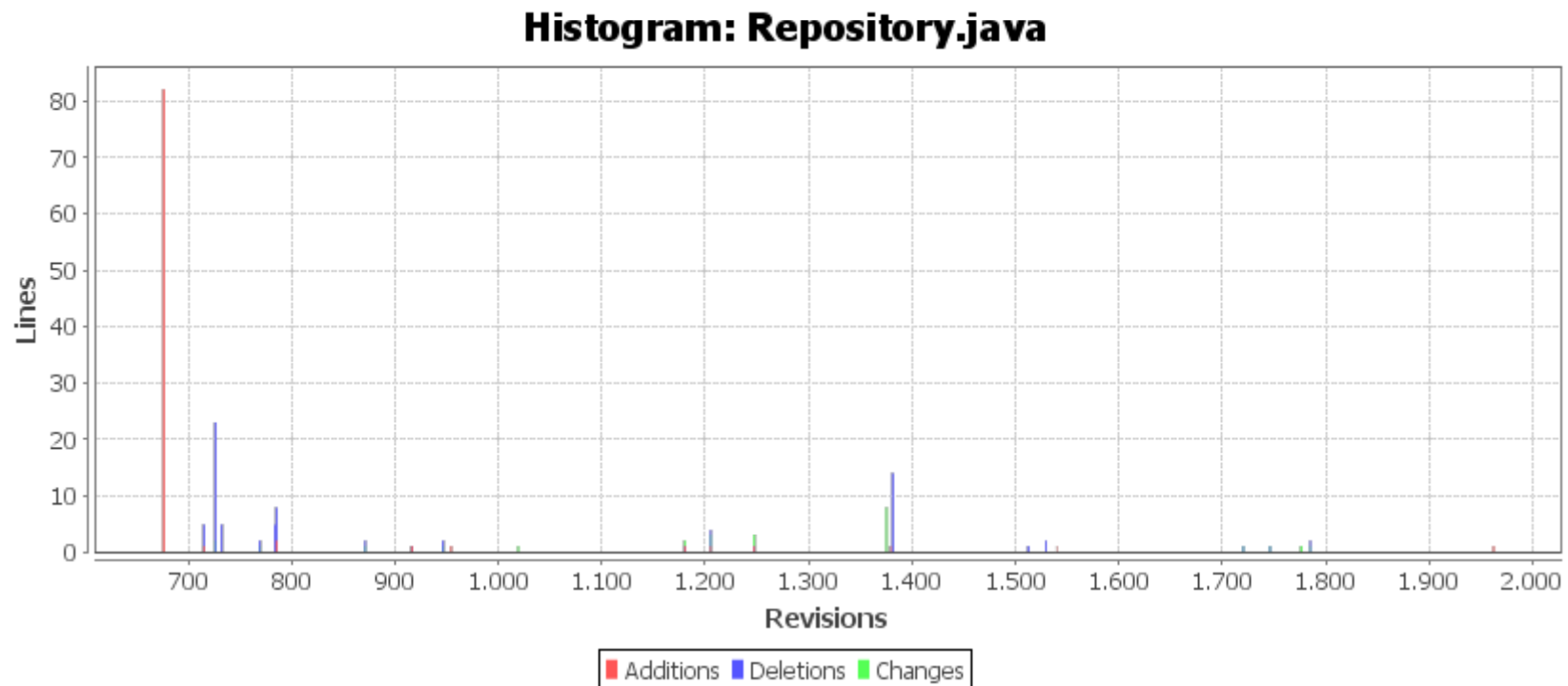
Name	Location	Value	Helptext
tradition_breaker	.holiday.Holiday	-	excessive increase of child class interface where
intensive_coupling	.employee.Employee.getStatusLineText()	-	calls too many methods from few classes
dispersed_coupling	.employee.Employee.checkHoliday(Nor...	-	calls too many classes and methods
intensive_coupling	.employee.Employee.checkCalculateEm...	-	calls too many methods from few classes
godclass	.employee.Employee	-	indicates a godclass: non-cohesive, overly compl
tradition_breaker	.employee.Employee	-	excessive increase of child class interface where

Temporal Degree of Interest

- Adaptiert Kersten's DOI auf Projekt-Historie
- Projekt-Historie: Alle Revisionen im SCM
- Interaktion = Änderung eines Elements
 - Zeilenweise Differenz zwischen Revisionen
 - Hinzufügen, Ändern, Löschen

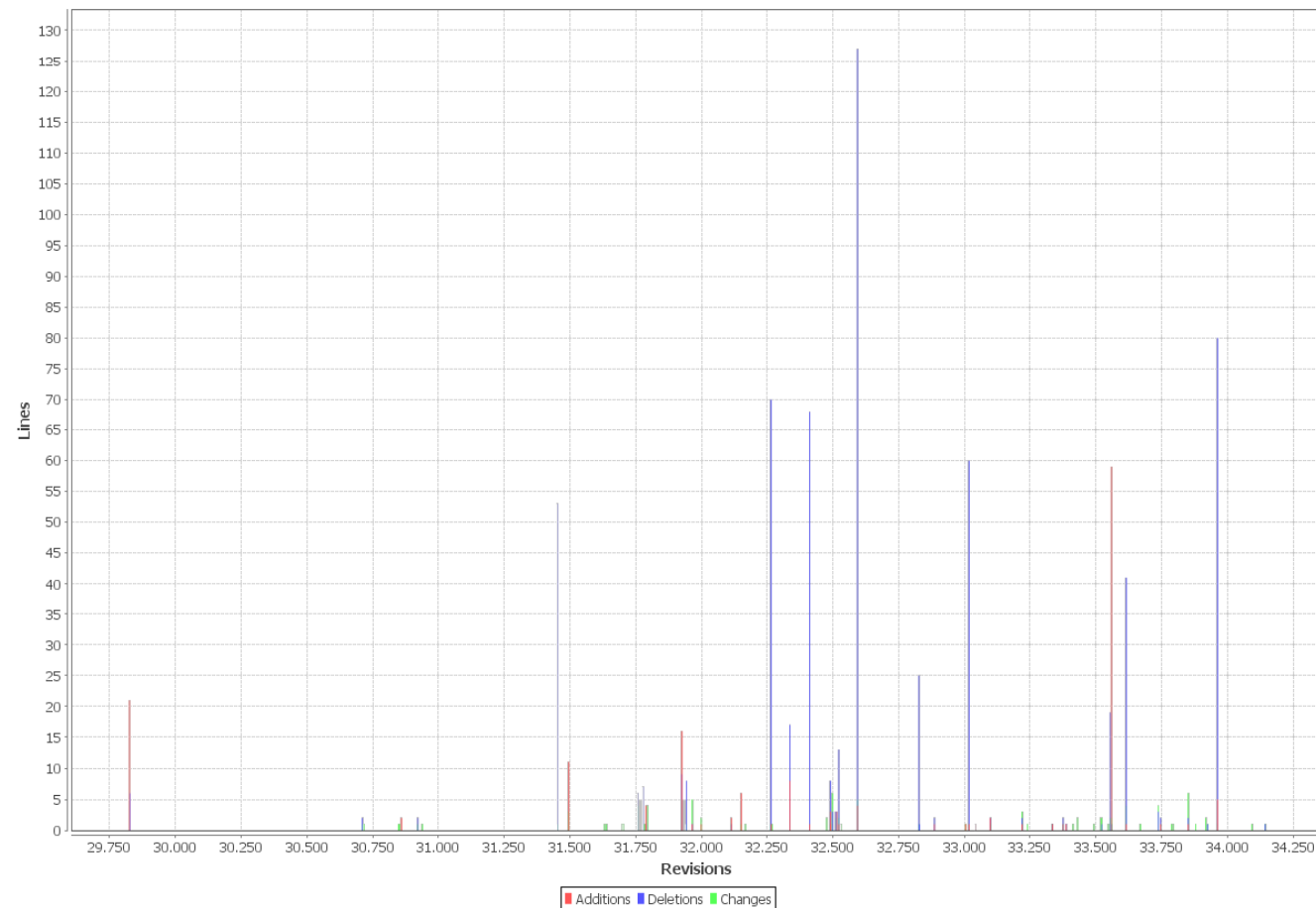
Relevanz gemessen an der Historie

- Temporal Stabiles Element
- $DOI \ll 0$

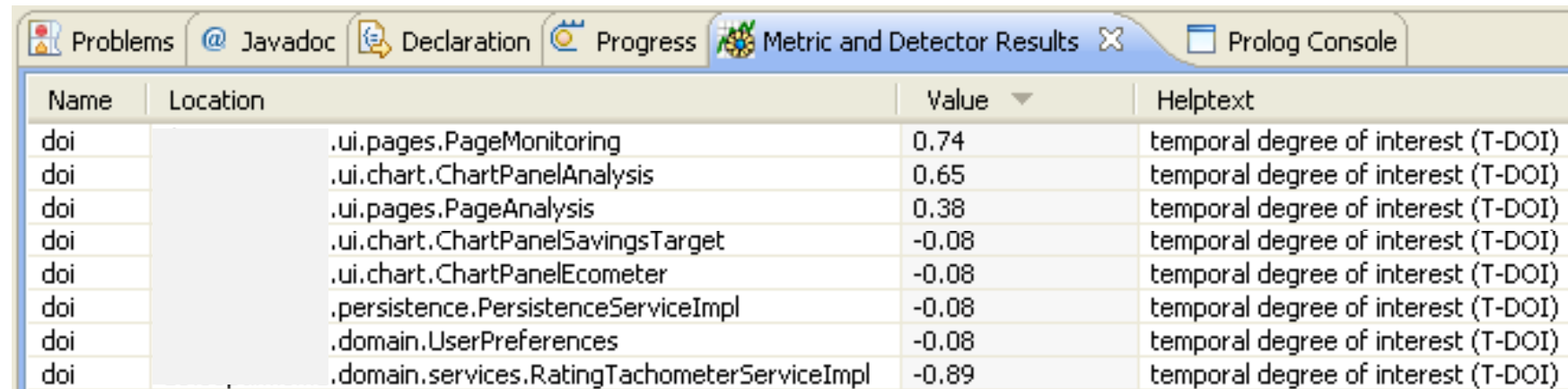


Relevanz gemessen an der Historie

- Temporal Instabiles Element
- $DOI > 0$

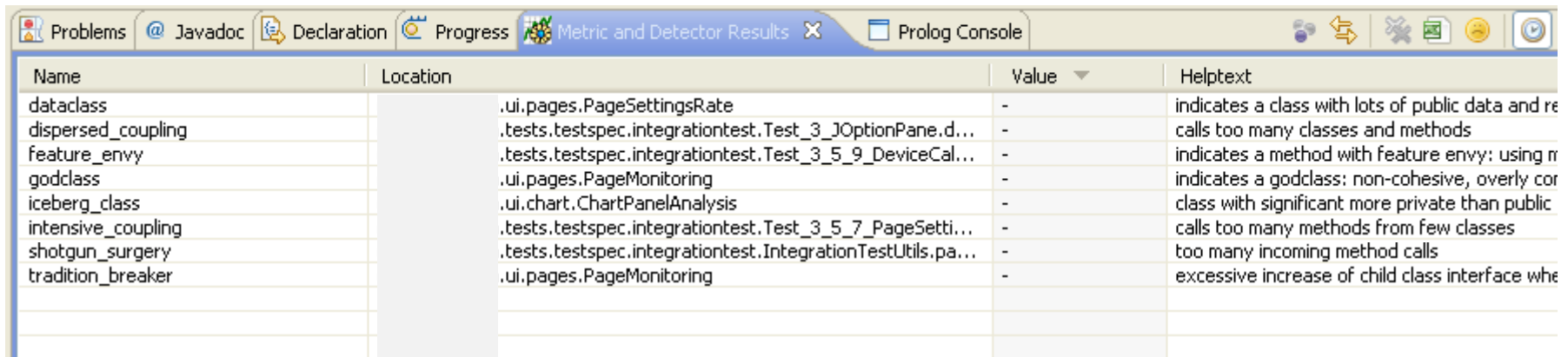


Filtern nach Relevanz



The screenshot shows an IDE window titled 'Metric and Detector Results'. The window has several tabs: 'Problems', 'Javadoc', 'Declaration', 'Progress', 'Metric and Detector Results', and 'Prolog Console'. The 'Metric and Detector Results' tab is active and displays a table with the following data:

Name	Location	Value	Help text
doi	.ui.pages.PageMonitoring	0.74	temporal degree of interest (T-DOI)
doi	.ui.chart.ChartPanelAnalysis	0.65	temporal degree of interest (T-DOI)
doi	.ui.pages.PageAnalysis	0.38	temporal degree of interest (T-DOI)
doi	.ui.chart.ChartPanelSavingsTarget	-0.08	temporal degree of interest (T-DOI)
doi	.ui.chart.ChartPanelEcometer	-0.08	temporal degree of interest (T-DOI)
doi	.persistence.PersistenceServiceImpl	-0.08	temporal degree of interest (T-DOI)
doi	.domain.UserPreferences	-0.08	temporal degree of interest (T-DOI)
doi	.domain.services.RatingTachometerServiceImpl	-0.89	temporal degree of interest (T-DOI)



The screenshot shows an IDE window titled 'Metric and Detector Results'. The window has several tabs: 'Problems', 'Javadoc', 'Declaration', 'Progress', 'Metric and Detector Results', and 'Prolog Console'. The 'Metric and Detector Results' tab is active and displays a table with the following data:

Name	Location	Value	Help text
dataclass	.ui.pages.PageSettingsRate	-	indicates a class with lots of public data and re
dispersed_coupling	.tests.testspec.integrationtest.Test_3_10OptionPane.d...	-	calls too many classes and methods
feature_envy	.tests.testspec.integrationtest.Test_3_5_9_DeviceCal...	-	indicates a method with feature envy: using r
godclass	.ui.pages.PageMonitoring	-	indicates a godclass: non-cohesive, overly cor
iceberg_class	.ui.chart.ChartPanelAnalysis	-	class with significant more private than public
intensive_coupling	.tests.testspec.integrationtest.Test_3_5_7_PageSetti...	-	calls too many methods from few classes
shotgun_surgery	.tests.testspec.integrationtest.IntegrationTestUtils.pa...	-	too many incoming method calls
tradition_breaker	.ui.pages.PageMonitoring	-	excessive increase of child class interface whe

MOTIVATION
STATE OF THE ART
PROBLEME
STRUKTUREN
RELEVANZ
INTERLINKING
CASE STUDY

Smell Detection in Context

INTERLINKING

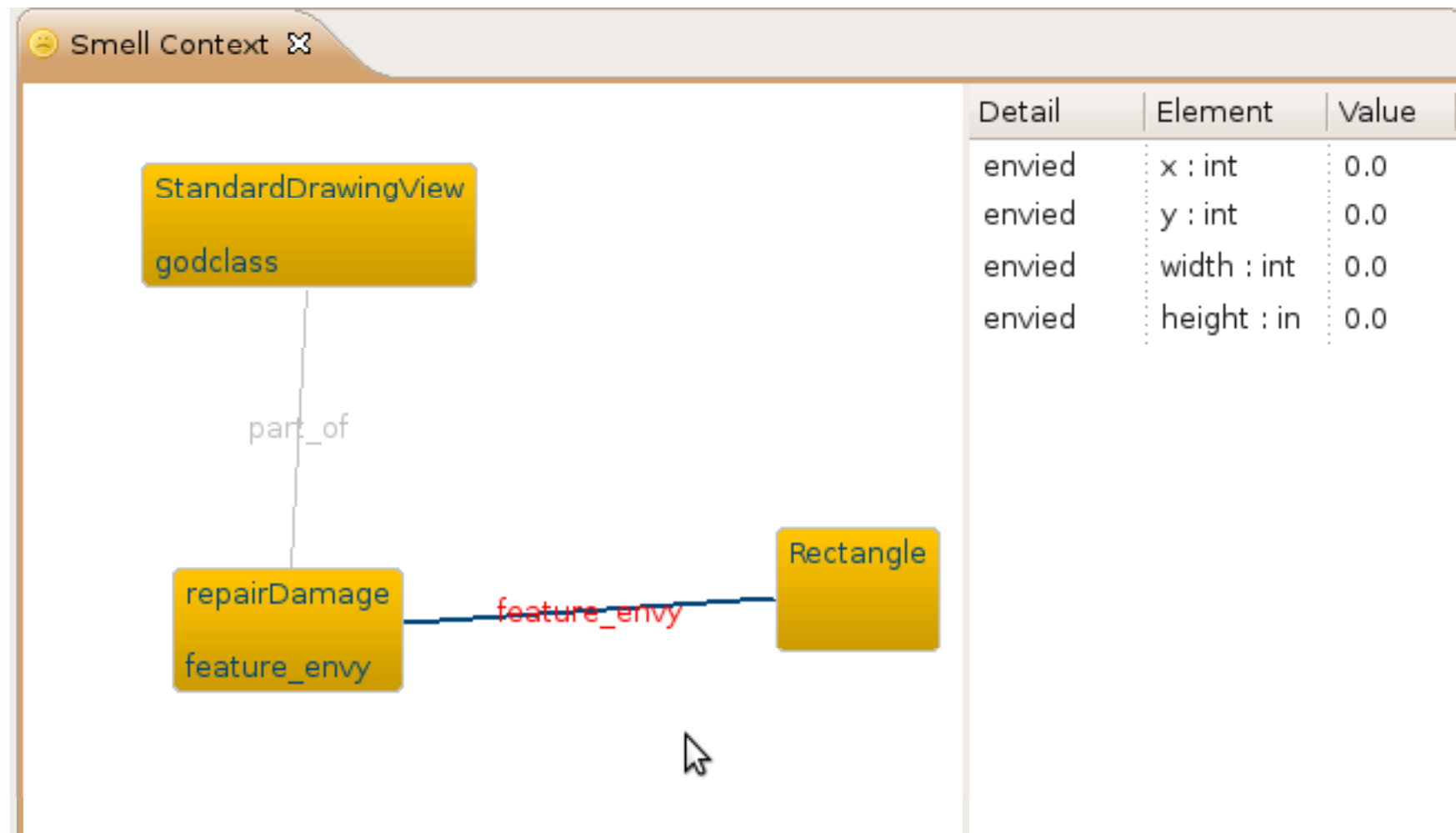
Interlinking: Smell im Kontext darstellen

- Fast kein Design Problem existiert in Isolation
- Neid
 - Beneide ich eine Datenklasse ?
- Schrotkugeln Operieren
 - Welche Typen und Methoden sind Aufrufer ?
- „Gottklasse“
 - Was sind die komplexen Methoden ?

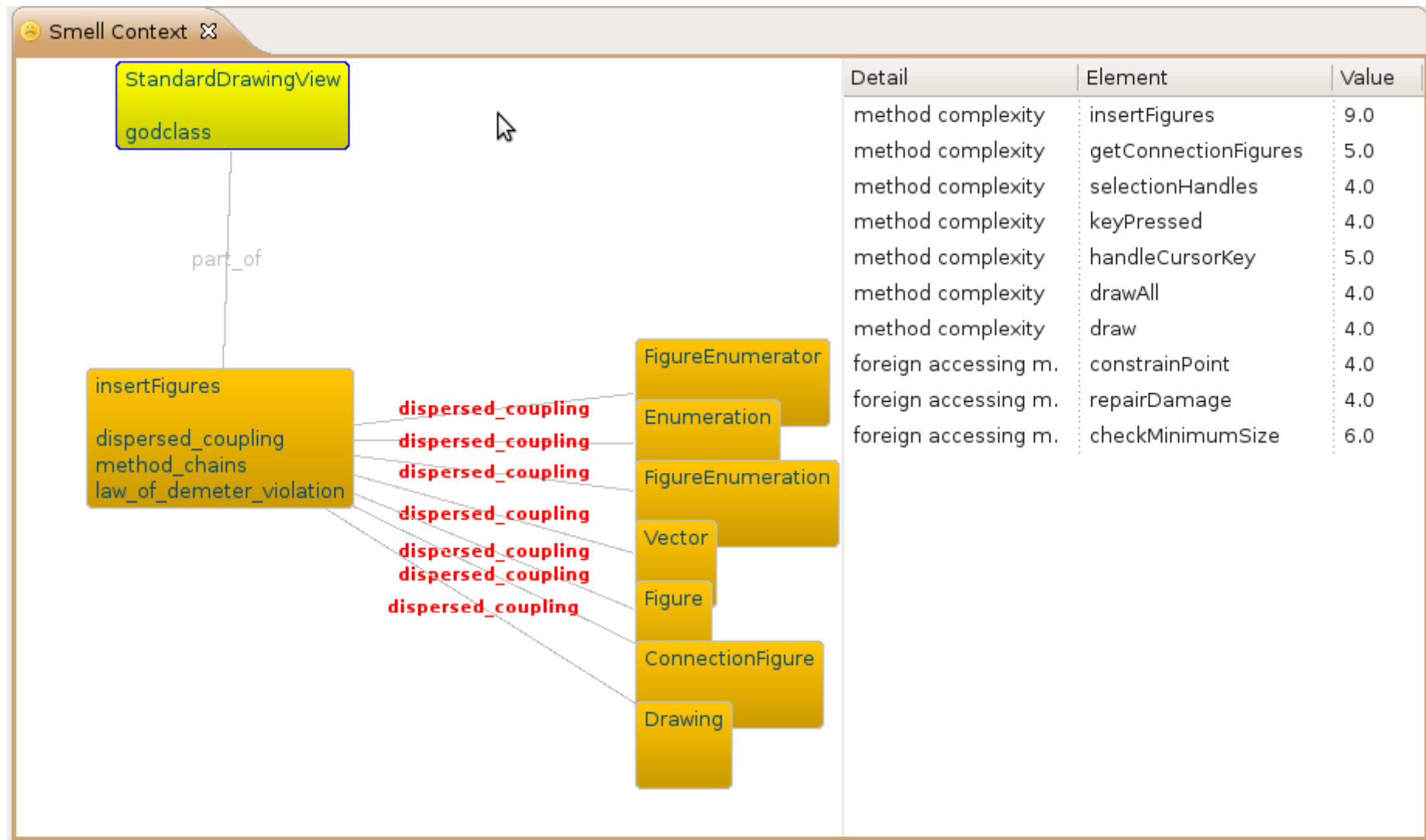
Kontext eines Smells

- Relationen und Details sind von Interesse
- Jeder Smell definiert Relationen
 - auf denen zB Metriken der Detection Strategies arbeiten
- Jeder Smell definiert Details
 - für ein Element (mit Smell)
 - für Relation

„Smell Context View“



„Smell Context View“



MOTIVATION
STATE OF THE ART
PROBLEME
STRUKTUREN
RELEVANZ
INTERLINKING
CASE STUDY

Smell Detection in Context

CASE STUDY

„Case Study“ ist eine empirische Methode

- Untersucht zeitnah Phänomene in ihrem Kontext im echten Leben
- Single Case vs. **Multiple Case Study**
- Design einer Studie kann enthalten:
 - Triangulation (Datenquellen, Arten, Beobachter)
 - Beweisketten
 - Muster, Rivalisierende Erklärungen
 - „Critical Cases“, Reproduktion der Cases
- Dokumentation im „Case Study Protocol“

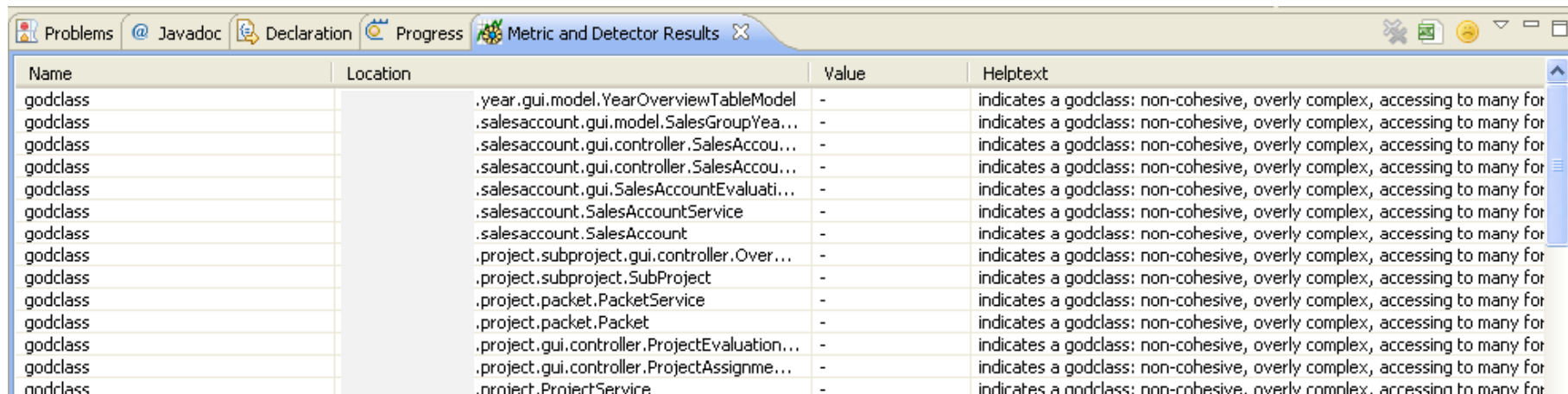
Case Study

- Smell Detection **ohne Kontexte...**
- ... **verglichen** mit Smell Detection **in Context**
- ... einem echten Industrie Projekt

- Insgesamt 4 Entwicklertage
- Resultate durch Interviews und Beobachtung ermitteln

1. Fall „Klassische“ Smell Detection

1. Interviews zum Hintergrund der Teilnehmer
2. Schulung „Design Smells“ + Tooling
3. Beobachtung der Entwickler
 - „Think-aloud“ Protokolle
4. Interviews am Ende



The screenshot shows a table with the following columns: Name, Location, Value, and Helptext. The table lists 14 instances of 'godclass' smells, each with a specific location and a detailed help text explaining the smell.

Name	Location	Value	Helptext
godclass	.year.gui.model.YearOverviewTableModel	-	indicates a godclass: non-cohesive, overly complex, accessing to many for
godclass	.salesaccount.gui.model.SalesGroupYea...	-	indicates a godclass: non-cohesive, overly complex, accessing to many for
godclass	.salesaccount.gui.controller.SalesAccou...	-	indicates a godclass: non-cohesive, overly complex, accessing to many for
godclass	.salesaccount.gui.controller.SalesAccou...	-	indicates a godclass: non-cohesive, overly complex, accessing to many for
godclass	.salesaccount.gui.SalesAccountEvaluati...	-	indicates a godclass: non-cohesive, overly complex, accessing to many for
godclass	.salesaccount.SalesAccountService	-	indicates a godclass: non-cohesive, overly complex, accessing to many for
godclass	.salesaccount.SalesAccount	-	indicates a godclass: non-cohesive, overly complex, accessing to many for
godclass	.project.subproject.gui.controller.Over...	-	indicates a godclass: non-cohesive, overly complex, accessing to many for
godclass	.project.subproject.SubProject	-	indicates a godclass: non-cohesive, overly complex, accessing to many for
godclass	.project.packet.PacketService	-	indicates a godclass: non-cohesive, overly complex, accessing to many for
godclass	.project.packet.Packet	-	indicates a godclass: non-cohesive, overly complex, accessing to many for
godclass	.project.gui.controller.ProjectEvaluation...	-	indicates a godclass: non-cohesive, overly complex, accessing to many for
godclass	.project.gui.controller.ProjectAssignme...	-	indicates a godclass: non-cohesive, overly complex, accessing to many for
nndrclass	.nnriert.ProjectService	-	indicates a nndrclass: non-cohesive, overly complex, accessing to many for

Hintergrund der Teilnehmer

- Statische Code Analyse: Checkstyle, Eclipse Warnings, Findbugs
- Refaktorisierung basiert auf Intuition
- Teilweise spezielle Probleme im Projekt bekannt

Ergebnisse „Klassisch“

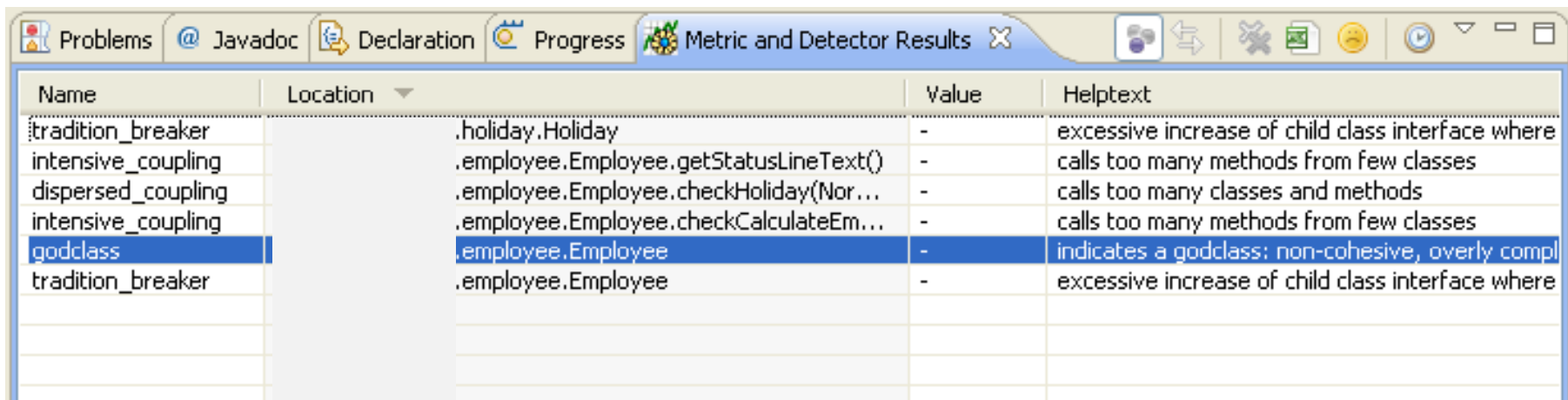
- Identifikation der Probleme „manuell“
 - Suchen im ResultView
 - Korrespondierenden Quelltext lesen
 - Eclipse Call Graph und Reference Search
- Falsch-Positive fallen kaum auf oder werden einfach ignoriert

“Iteration über alle Ergebnisse, Caller, usw.“

“Lästige Handarbeit“

2. Fall „Smell Detection in Context“

1. Schulung des Toolings
2. Beobachtung der Entwickler
3. Interviews am Ende
4. Weiteres Interview 2 Wochen später



The screenshot shows a window titled "Metric and Detector Results" with a table of metrics. The table has four columns: Name, Location, Value, and Helptext. The "godclass" metric is highlighted in blue.

Name	Location	Value	Helptext
!tradition_breaker	.holiday.Holiday	-	excessive increase of child class interface where
intensive_coupling	.employee.Employee.getStatusLineText()	-	calls too many methods from few classes
dispersed_coupling	.employee.Employee.checkHoliday(Nor...	-	calls too many classes and methods
intensive_coupling	.employee.Employee.checkCalculateEm...	-	calls too many methods from few classes
godclass	.employee.Employee	-	indicates a godclass: non-cohesive, overly compl
tradition_breaker	.employee.Employee	-	excessive increase of child class interface where

Ergebnisse „SDIC“

- Ausnutzung der Relevanz-Filterung (Aufgaben)
- Smell Context View für konkrete Probleme konsultiert
- Struktur-Definitionen werden ausgenutzt
- Temporales Filtern wird nicht genutzt
 - anscheinend ist die Ergebniss-Menge pro Aufgabe schon klein genug

Fazit

- Annahme „Kontext fehlt“ bestätigt sich in der Studie des klassischen Prototyps.
- Strukturen sind Abwägungen von Design-Prinzipien und haben Einfluss auf die Erkennung von Smells
- Relevanz-Filterung pro Aufgabe macht Smell Detection effizient genug für Floss Refactoring
- Kontext eines Smells unterstützt das subjektive Verständnis

Q & A