Applying and evaluating concern-sensitive design heuristics

Eduardo Figueiredo[1], Claudio Sant'Anna[2], Alessandro Garcia[3], Carlos Lucena[3]

[1] Computer Science Department, Federal University of Minas Gerais – UFMG, Brazil

[2] Software Engineering Lab (LES), Computer Science Department, Federal University of Bahia – UFBA, Brazil

[3] Opus Research Group, Computer Science Department, Pontifical Catholic University of Rio de Janeiro – PUC-Rio, Brazil

Article history:

Received 1 March 2010

Received in revised form 7 September 2011

Accepted 29 September 2011

Available online 20 October 2011

The results of this exploratory analysis give evidences that the heuristics offer support for:

I. Addressing the shortcomings of conventional metrics-based assessments,

II. Reducing the manifestation of false positives and false negatives in modularity assessment,

III. Detecting sources of design instability, and finding the presence of design modularity flaws in both object-oriented and aspect-oriented programs.

Division

1. Introduction
2. Design modularity assessment
3. Concern-driven metrics
4. Concern-sensitive heuristics
5. ConcernMorph
6. Evaluation settings
7. Results and discussion
8. Study constraints
9. Concluding remarks

The contributions of this paper are fourfold:

First : revisiting existing assessment mechanisms, it discusses the limitations of conventional metrics-based heuristics

Second: presents a suite of heuristic rules with the distinguishing characteristic of exploiting concerns as explicit abstractions in the design assessment process.

1. Identification and classification of crosscutting concerns
2. Identification and classification of specific crosscutting patterns
3. Detection of classical bad smells

Third: presents a prototype tool, called ConcernMorph.

Finally: it provides an exploratory evaluation on the accuracy of the concern-sensitive heuristics in the context of seven applications.

Design Modularity assessment

Metrics for aspect-oriented designs

Table 1
Examples of traditional software metrics (Ceccato and Tonella, 2004; Chidamber and Kemerer, 2004; Sant'Anna et al., 2003).

Metrics	Definitions
Number of Components (NC) (Sant'Anna et al., 2003)	Counts the number of classes, interfaces and aspects of the system
Number of Attributes (NOA) (Sant'Anna et al., 2003)	Counts the number of attributes of each class, interface or aspect of the system
Number of Operations (NOO) (Sant'Anna et al., 2003)	Counts the number of methods, constructors, advice, and intertype methods of each class, interfaces, or aspect
Coupling between Components (CBC) (Ceccato and Tonella, 2004; Chidamber and Kemerer, 2004; Sant'Anna et al., 2003)	Counts the number of other classes, interfaces, and aspects which a component is coupled to

Conventional heuristic assessment

Detecting a specific kind of modularity flaw, namely the Shotgun Surgery bad smell

Shotgun Surgery:=((CM, TopValues(20%))and(CM, HigherThan(10))) and (CC, HigherThan(5))

CM: Changing Method

CC: Changing Classes

Limitation of conventional heuristics

Many design modularity flaws are related to the inadequate modularisation of concerns, most of the current quantitative assessment approaches, such as AO metrics, do not explicitly consider concern as a measurement abstraction.

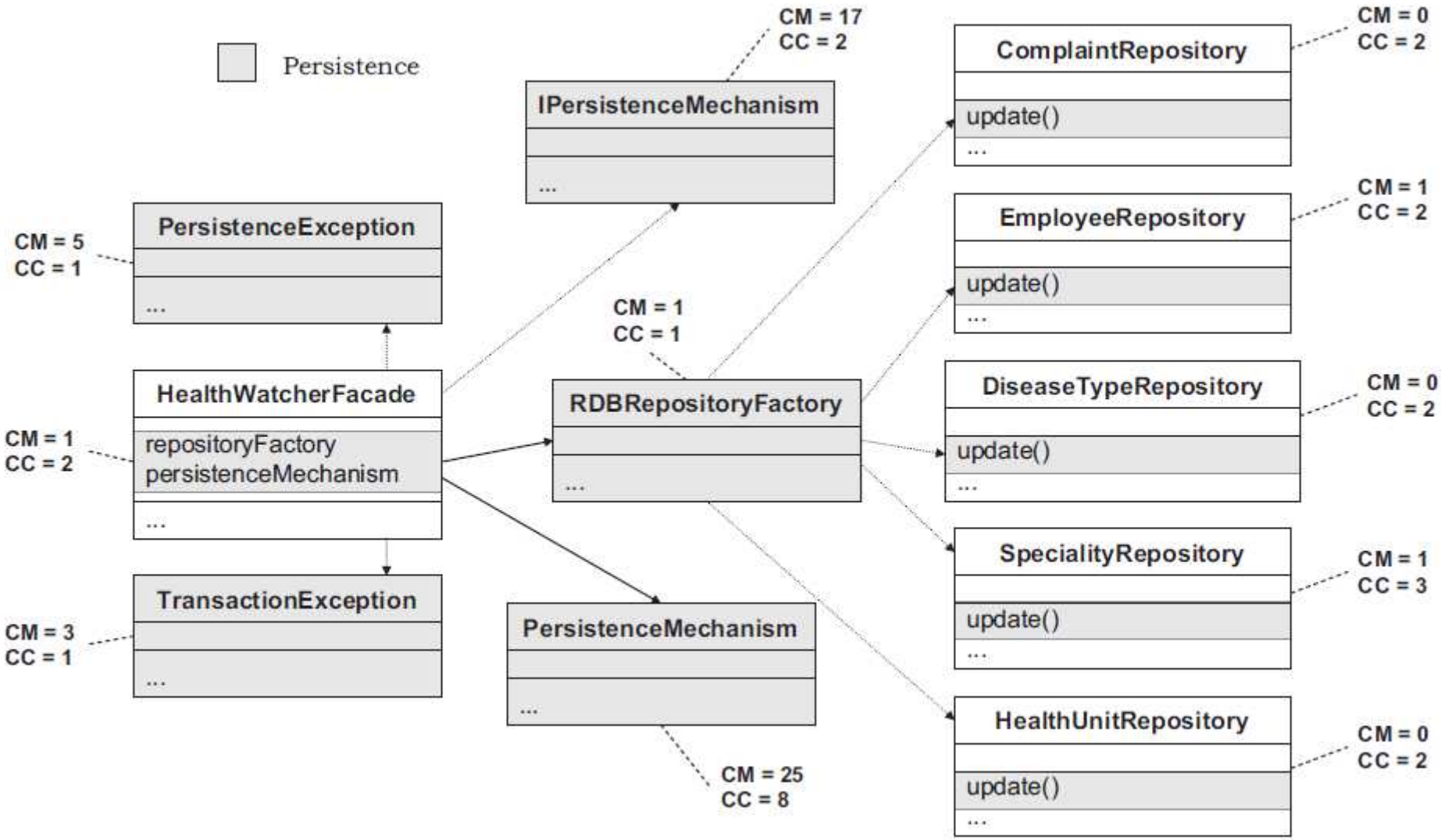


Fig. 1. Scattering of the Persistence concern over Health Watcher classes.

Concern-driven metrics

The concern metrics are computed based on the mapping of concerns to design elements.

ConcernMapper for supporting the mapping of concerns and ConcernMorph for concern measurement.

Concern scattering and tangling

Concern Diffusion over Components (CDC)

Number of Concerns per Component (NCC)

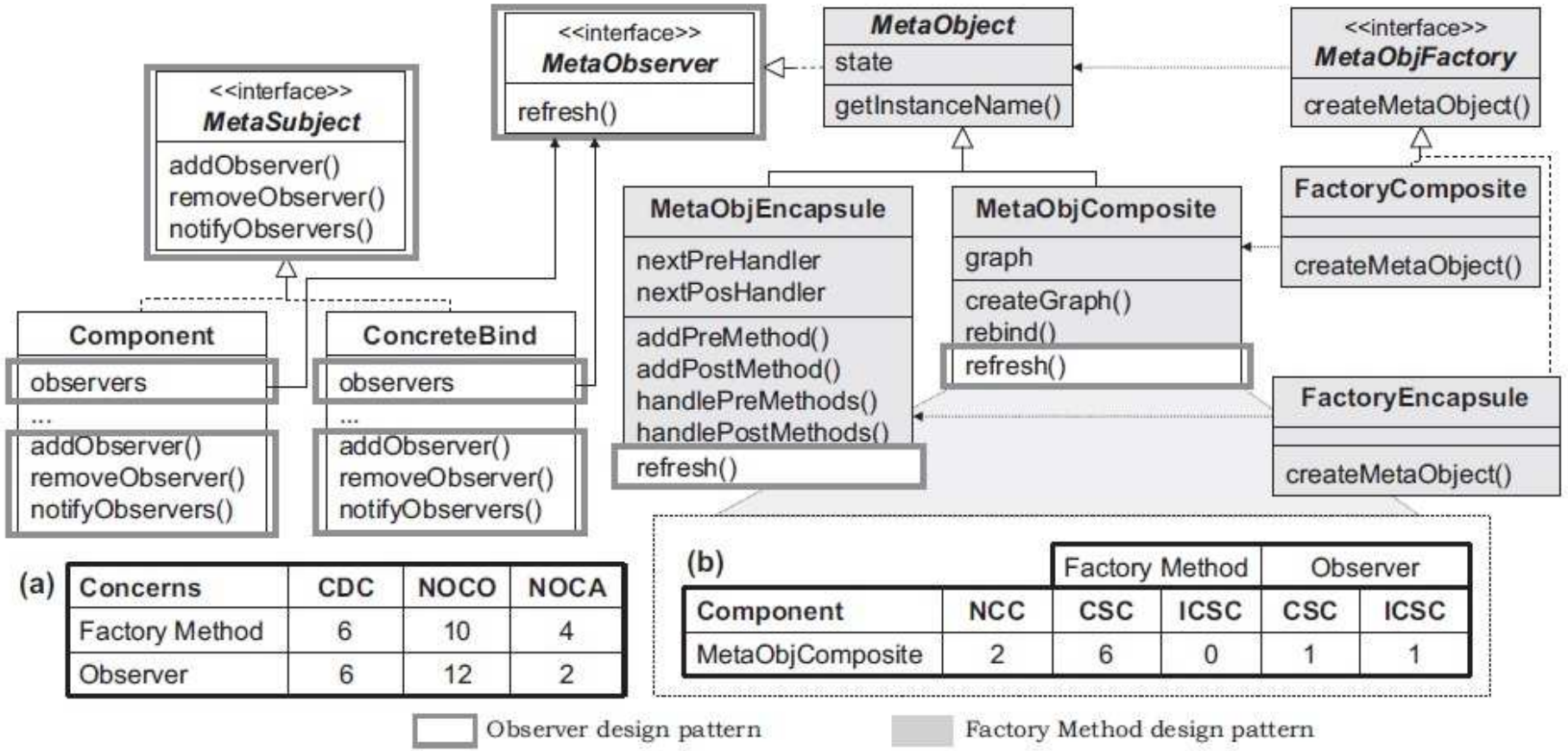
Concern materialisation and coupling

Number of Concern Attributes (NOCA)

Number of Concern Operations (NOCO)

Concern-Sensitive Coupling (CSC)

Intra-Component Concern Sensitive Coupling (ICSC)



Concerns	CDC	NOCO	NOCA
Factory Method	6	10	4
Observer	6	12	2

		Factory Method		Observer	
Component	NCC	CSC	ICSC	CSC	ICSC
MetaObjComposite	2	6	0	1	1

Fig. 2. Concern metrics applied to the Observer and Factory Method patterns.

Concern-sensitive heuristic

The proposed heuristics, called concern-sensitive heuristic rules, are defined in terms of combined information collected from both concern metrics and conventional modularity metrics.

IF <condition> THEN <consequence>

Concern diffusion

- Isolated,
- Tangled,
- Little Scattered,
- Highly Scattered,
- Well Localised,
- Crosscutting

Definitions of the concern diffusion heuristic rules (left) denoting transitions between two concern classifications (diagram on the right).

R01 - *Isolated*:
if NCC = 1 **for every component realising** CONCERN
then CONCERN is ISOLATED

R02 - *Tangled*:
if NCC > 1 **for at least one component then** CONCERN is TANGLED

R03 - *Little Scattered*:
if (CDC / NC) of CONCERN < 0.5 **then** TANGLED CONCERN is LITTLE SCATTERED

R04 - *HighlyScattered*:
if (CDC / NC) of CONCERN ≥ 0.5 **then** TANGLED CONCERN is HIGHLY SCATTERED

R05 - *Well Encapsulated*:
if (NOCA / NOA ≥ 0.5) **and** (NOCO / NOO ≥ 0.5) **for every component**
then LITTLE SCATTERED CONCERN is WELL-LOCALISED

R06 - *Crosscutting*:
if (NOCA / NOA < 0.5) **and** (NOCO / NOO < 0.5) **for at least one component**
then LITTLE SCATTERED CONCERN is CROSSCUTTING

R07 - *Well Encapsulated*:
if (NOCA / NOA ≥ 0.5) **and** (NOCO / NOO ≥ 0.5) **for every component**
then HIGHLY SCATTERED CONCERN is WELL-LOCALISED

R08 - *Crosscutting*:
if (NOCA / NOA < 0.5) and (NOCO / NOO < 0.5) **for at least one component**
then HIGHLY SCATTERED CONCERN is CROSSCUTTING

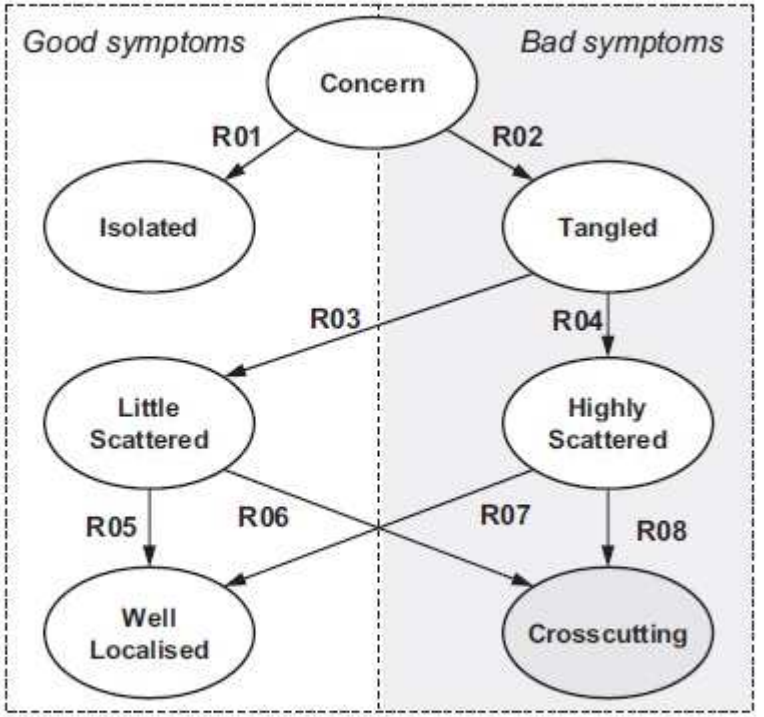


Fig. 3. Concern classification and heuristics for concern diffusion.

Patterns of crosscutting concerns

The identification of such crosscutting patterns complements the tangling and scattering analysis.

Five of these crosscutting patterns:

- Black Sheep,
- Octopus,
- God Concern,
- Data Concern,
- Behavioural Concern

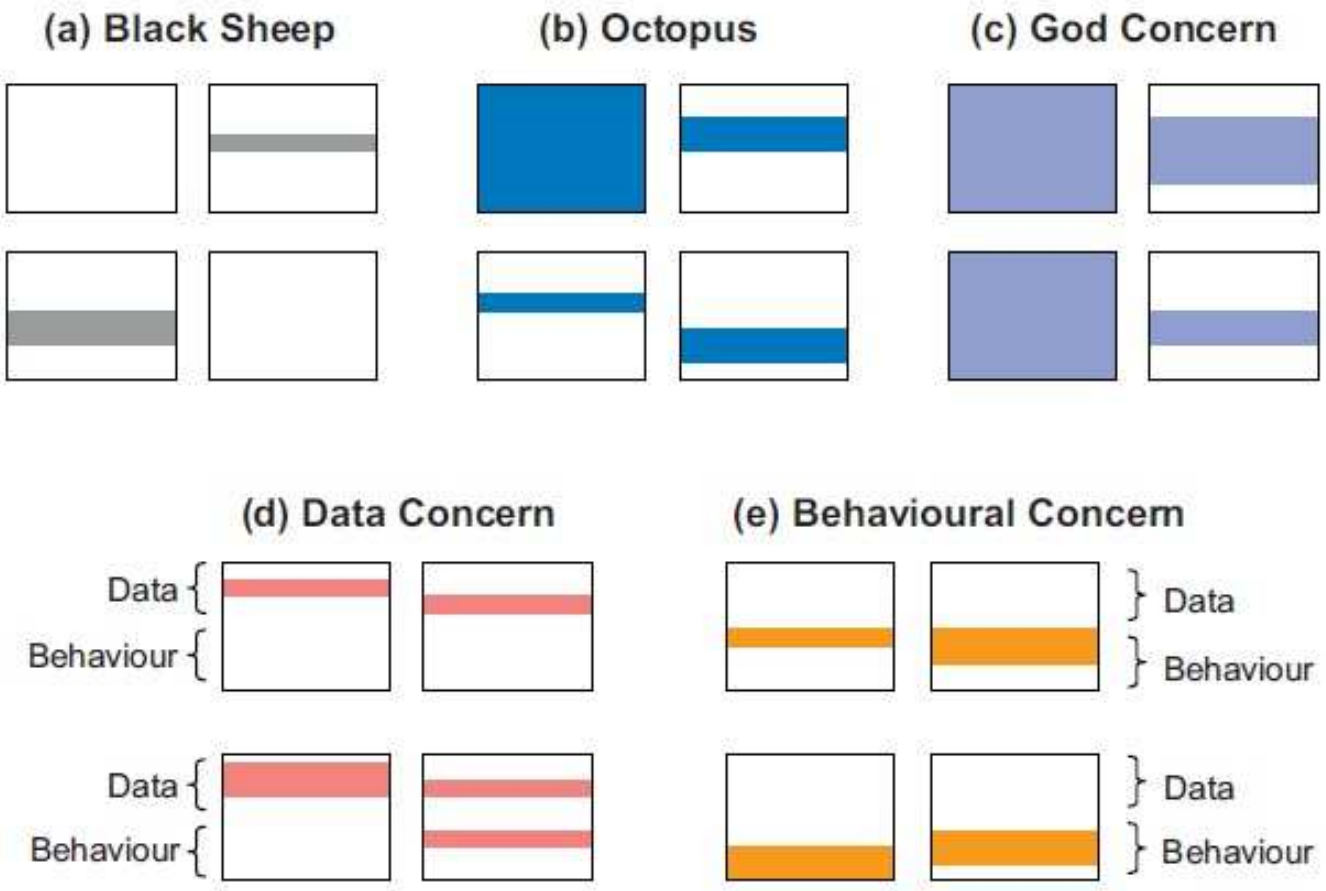


Fig. 4. Abstract representation of crosscutting patterns.

The heuristic rules, R09 to R13, which aim at identifying each of the selected crosscutting patterns.

Condition A - *Little Dedication*: (NOCA / NOA < 0.33) **and** (NOCO / NOO < 0.33)

Condition B - *High Dedication*: (NOCA / NOA ≥ 0.67) **and** (NOCO / NOO ≥ 0.67))

R09 - *Black Sheep*:
if (*Little Dedication*) **for every component with** CONCERN **then** CROSSCUTTING CONCERN is BLACK SHEEP

R10 - *Octopus*:
if ((*Little Dedication*) **or** (*High Dedication*) **for every component with** CONCERN)
and ((*Little Dedication*) **for at least 2 components and** (*High Dedication*) **for at least 1 component with** CONCERN)
then CROSSCUTTING CONCERN is OCTOPUS

R11-*God Concern*:
if ((CDC / NC) > 0.33) **and** (High Dedication) **for most components with** CONCERN
then CROSSCUTTING CONCERN **is** GOD CONCERN

R12 - Data Concern:
if ((NOCA/NOA) > (NOCO/NOO)) **for every component with** CONCERN
thenC ROSSCUTTING CONCERN is DATA CONCERN

R13 - Behavioural Concern:
if (NOCA = 0) and (NOCO > 0) **for every component with** CONCERN
then CROSSCUTTING CONCERN is BEHAVIOURAL CONCERN

Fig. 5. Heuristics for crosscutting patterns.

Concern-aware design flaws

Four bad smells to illustrate this kind of concern-sensitive heuristics:

- Shotgun Surgery
- Feature Envy
- Divergent Change
- God Class.

Condition C - *High Inter-Component Coupling*: (CSC / CBC) > ((NOCA+NOCO) / (NOA+NOO))

Condition D - *Low Intra-Component Coupling*: (ICSC / ((NOA+NOO)-(NOCA+NOCO))) < ((NOCA+NOCO) / (NOA+NOO))

R14 - *Feature Envy*:
if (*High Inter-Component Coupling*) **and** (*Low Intra-Component Coupling*) **and** (NCC > 1)
then CROSSCUTTING CONCERN is FEATURE ENVY

R15 - *Shotgun Surgery*:
if CONCERN is (*Tangled*) **and** (*HighlyScattered*) **and** (*Crosscutting*) **then** C ROSSCUTTING CONCERN is SHOTGUN SURGERY

R16 - *Divergent Change*:
if (NCC > 3) and (CBC > 5) **then**C OMPONENT has Divergent Change

R17 - *God Class*:
if (NCC >2) **and** (NOA > (NOAsystem/NC)) **and** (NOO > (NOOsystem/NC)) **then**C OMPONENT is a GOD CLASS

Fig. 6. Heuristic rules for bad smells.

ConcernMorph: extensible metrics-based heuristic analysis

A tool to automatically apply the proposed concern-sensitive heuristic rules. ConcernMorph is implemented as an Eclipse plug-in and supports the heuristic rules.

ConcernMorph and its relationships with ConcernMapper and the Eclipse Platform

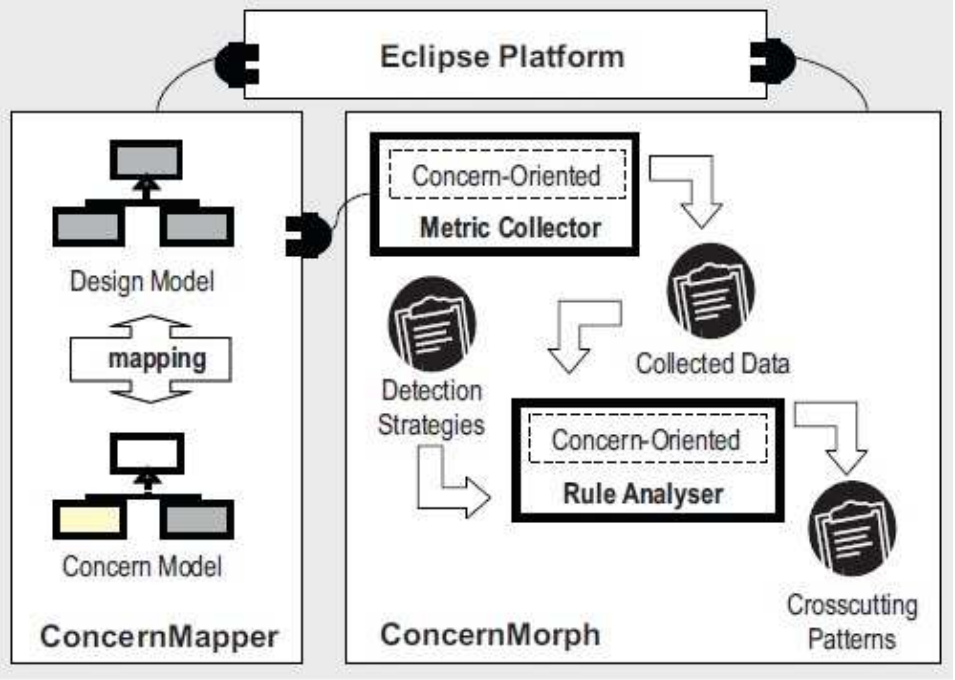


Fig. 7. The ConcernMorph Architecture.

ConcernMorph

The Concern Metrics view shows concern measurements while the Crosscutting Patterns view shows all instances of crosscutting patterns in the target system.

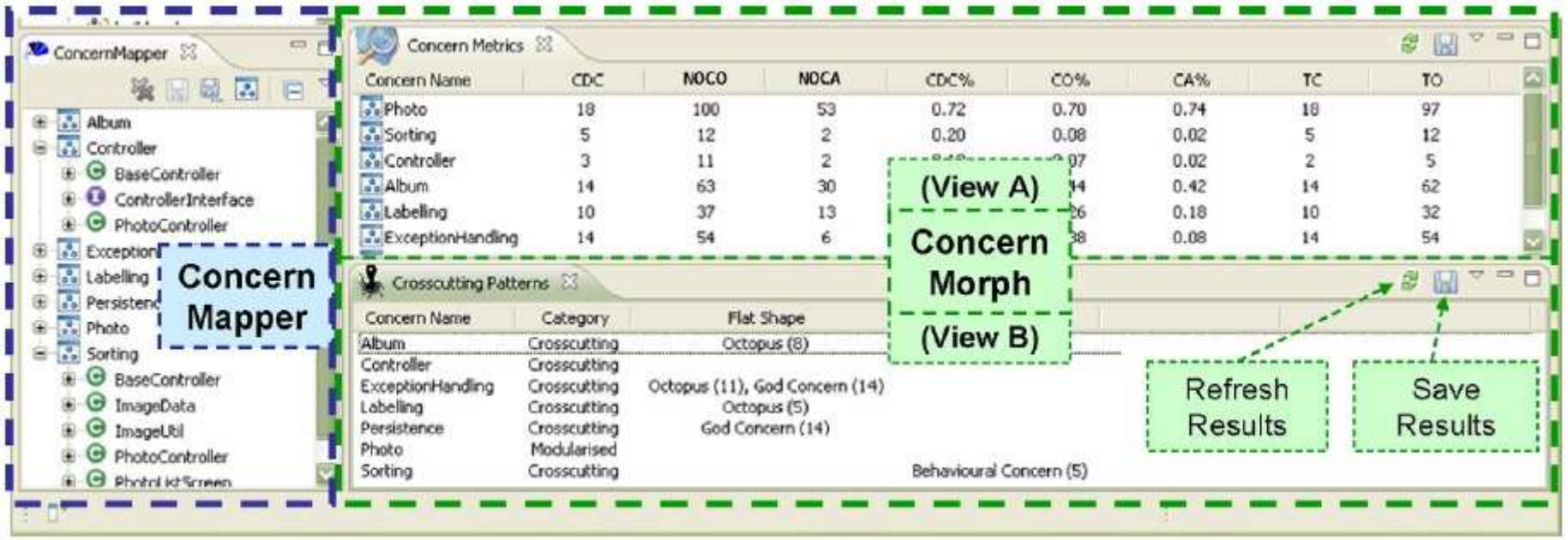


Fig. 8. Screenshot showing concern measurements (View A) and crosscutting patterns (View B).

Evalution settings

Hypothesis: Heuristic assessment techniques which are based on concern properties enhance traditional quantitative analysis.

Research Question 1. If concern-sensitive assessment techniques properties enhance traditional quantitative analysis, in which cases do they succeed or fail?

Research Question 2. Do concern-sensitive heuristics provide more accurate results compared to traditional detection strategies?

Research Question 3. Does a high number of crosscutting patterns impact positively, negatively or have no impact on design stability?

Research Question 4. Which bad smells can be detected by concern-sensitive heuristic rules?

Selection of the target applications

Table 2
Selected system releases.

Systems		LOC	No. of releases	No. of developers
(1) A Design Pattern library (Hannemann and Kiczales, 2002)	Builder	168 (OO)/177 (AO)	2	2
	Chain of Resp.	213 (OO)/234 (AO)	2	2
	Factory Method	135 (OO)/146 (AO)	2	2
	Mediator	253 (OO)/253 (AO)	2	2
	Observer	265 (OO)/363 (AO)	2	2
(2) OpenOrb Middleware (Cacho et al., 2006)		~1.5 KLOC (OO) ~2 KLOC (AO)	1	3
(3) AJATO (Cacho et al., 2006; Figueiredo et al., 2006)		~1 KLOC (OO) ~1.1 KLOC (AO)	1	1
(4) Eclipse CVS Plugin (Filho et al., 2006)		~19 KLOC (OO) ~22 KLOC (AO)	5	8
(5) Health Watcher, R9 (Greenwood et al., 2007)		~8 KLOC (OO) ~7 KLOC (AO)	10	>10
(6) Portalware (Garcia et al., 2004)		~1.4 KLOC (OO)~1.6 KLOC (AO)	1	6
(7) MobileMedia, R6 (Figueiredo et al., 2008)		~2.2 KLOC (OO)~2.6 KLOC (AO)	9	11

Selection of the target applications

Table 3
Selected concerns per system.

Systems	Nature of the concerns	Concern names
(1) A Design Pattern library	Roles of design patterns	Director role (Builder), Handler role (Chain of Responsibility), Creator role (Factory Method), Subject and Observer roles (Observer), Colleague and Mediator roles (Mediator)
(2) OpenOrb Middleware (3) AJATO	Design patterns and their compositions	Factory Method, Observer, Facade, Singleton, Prototype, State, Interpreter, and Proxy
(4) Eclipse CVS Plugin (5) Health Watcher	Recurring architectural crosscutting and non-crosscutting concerns	Business, Concurrency, Distribution, Exception Handling, and Persistence
(6) Portalware	Domain-specific concerns	Adaptation, Autonomy, and Collaboration
(7) MobileMedia	Product line features	Controller, Sorting, Exception Handling, Favourites, Persistence, and Labelling

Results and discussion

Table 4
Results of the heuristics for concern diffusion and crosscutting patterns in the OO systems.

Studies	Concerns	Heuristic rules													Concern classifications
		01	02	03	04	05	06	07	08	09	10	11	12	13	
(1) Patterns library (Hannemann and Kiczales, 2002)	Director role	n	y	y	n	y	n								well-loc.
	Handler role	n	y	n	y			y	n						well-loc.
	Creator role	n	y	n	y			n	y	n	n	n	n	y	BC
	Observer role	n	y	n	y			n	y	n	y	n	n	y	Oct + BC
	Subject role	n	y	n	y			n	y	n	y	n	n	n	Oct
	Colleague role	n	y	n	y			n	y	n	y	n	n	n	Oct
	Mediator role	n	y	n	y			y	n						well-loc.
(2) OpenOrb Middleware (Cacho et al., 2006)	Fact. Method	n	y	y	n	y	n								well-loc.
	Observer	n	y	y	n	n	y			n	y	n	n	n	Oct
	Façade	y	n												isolated
	Singleton	n	y	y	n	n	y			y	n	n	n	n	BS
(3) Measurement tool (Cacho et al., 2006; Figueiredo et al., 2006)	Prototype	n	y	y	n	n	y			y	n	n	n	y	BS + BC
	State	n	y	y	n	y	n								well-loc.
	Interpreter	n	y	y	n	n	y			n	y	n	n	n	Oct
	Proxy	n	y	y	n	n	y			n	n	n	n	n	crosscutting
(4) CVS plugin (Filho et al., 2006)	Exc. Handling	n	y	n	y			n	y	n	y	n	n	y	Oct + BC
(5) Health Watcher, R9 (Greenwood et al., 2007)	Business	n	y	n	y			y	n						well-loc.
	Concurrency	n	y	n	y			n	y	n	y	n	n	n	Oct
	Distribution	n	y	n	y			n	y	n	y	n	n	n	Oct
	Persistence	n	y	n	y			n	y	n	y	y	n	n	Oct + GC
(6) Portalware (Garcia et al., 2004)	Adaptation	n	y	y	n	n	y			n	n	n	n	n	crosscutting
	Autonomy	n	y	y	n	n	y			n	n	n	n	n	crosscutting
	Collaboration	n	y	n	y			n	y	n	y	n	n	n	Oct
(7) MobileMedia, R6 (Figueiredo et al., 2008)	Controller	n	y	n	y			n	y	n	y	n	n	n	Oct
	Sorting	n	y	y	n	n	y			y	n	n	n	n	BS
	Exc. Handling	n	y	n	y			n	y	n	y	n	n	n	Oct
	Favourites	n	y	y	n	n	y			y	n	n	n	n	BS
	Persistence	n	y	n	y			n	y	n	y	y	n	n	Oct + GC
	Labelling	n	y	n	y			n	y	n	y	n	n	n	Oct

Solving measurement shortcomings

Six of problems (labelled A–F):

- False crosscutting warnings: (A) false scattering and tangling warnings and (B) false coupling or cohesion warnings.
- Hiding concern-sensitive flaws: (C) metrics are not able to reveal an existing modularity problem.
- Lack of mapping to flaw-causing concerns: (D) measurement does not show where (which design elements) the problem is and (E) measurement does not relate design flaws to concerns causing them.
- Controversial outcomes from concern metrics: (F) the results of different metrics do not converge to the same outcome, making the designer’s interpretation difficult.

Accuracy of the concern-sensitive heuristic

Table 6
Results of the heuristics for concern diffusion in the OO and AO systems.

Studies	Concerns	Best solution	Concern classification		Results	
			OO	AO	OO	AO
(1) Patterns library (Hannemann and Kiczales, 2002)	Director role	OO	well-loc.	well-loc.	hit	hit
	Handler role	AO	well-loc.	well-loc.	FAIL	hit
	Creator role	OO	crosscutting	well-loc.	FAIL	hit
	Observer role	AO	crosscutting	crosscutting	hit	FAIL
	Subject role	AO	crosscutting	crosscutting	hit	FAIL
	Colleague role	AO	crosscutting	crosscutting	hit	FAIL
	Mediator role	AO	well-loc.	crosscutting	FAIL	FAIL
(2) OpenOrb Middleware (Cacho et al., 2006)	Fact. Method	OO	well-loc.	isolated	hit	hit
	Observer	AO	crosscutting	isolated	hit	hit
	Façade	OO	isolated	isolated	hit	hit
	Singleton	AO	crosscutting	isolated	hit	hit
(3) Measurement tool (Cacho et al., 2006; Figueiredo et al., 2006)	Prototype	AO	crosscutting	isolated	hit	hit
	State	AO	well-loc.	isolated	FAIL	hit
	Interpreter	AO	crosscutting	isolated	hit	hit
	Proxy	AO	crosscutting	isolated	hit	hit
(4) CVS plugin (Filho et al., 2006)	Exc. Handling	AO	crosscutting	crosscutting	hit	hit[a]
(5) Health Watcher, R9 (Greenwood et al., 2007)	Business	OO	well-loc.	well-loc.	hit	hit
	Concurrency	AO	crosscutting	isolated	hit	hit
	Distribution	AO	crosscutting	well-loc.	hit	hit
	Persistence	AO	crosscutting	well-loc.	hit	hit
(6) Portalware (Garcia et al., 2004)	Adaptation	AO	crosscutting	well-loc.	hit	hit
	Autonomy	AO	crosscutting	crosscutting	hit	FAIL
	Collaboration	AO	crosscutting	isolated	hit	hit
(7) MobileMedia, R6 (Figueiredo et al., 2008)	Controller	AO	crosscutting	crosscutting	hit	hit[b]
	Sorting	AO	crosscutting	isolated	hit	hit
	Exc. Handling	AO	crosscutting	crosscutting	hit	hit[a]
	Favourites	AO	crosscutting	isolated	hit	hit
	Persistence	AO	crosscutting	crosscutting	hit	hit[b]
	Labelling	AO	crosscutting	crosscutting	hit	hit[b]

[a] Exception Handling was only partially aspectised in the CVS plugin and MobileMedia.
[b] Controller, Persistence, and Labelling were not aspectised in MobileMedia.

Accuracy of the concern-sensitive heuristic

Table 7
Statistics for concern diffusion heuristics.

Studies	Hits (%)	FP (%)	FN (%)	Total (%)
OO	25 (86.2)	1 (3.5)	3 (10.3)	29 (100)
AO	24 (82.8)	5 (17.2)	0 (0.0)	29 (100)
OO + AO	48 (84.5)	6 (10.3)	3 (5.2)	58 (100)

According to data above, the heuristics failed in less than 20% of the cases (6 false positives and 3 false negatives)

Correlating crosscuting patterns and design stability

Table 8
Unstable MobileMedia components and crosscutting patterns (Release 6).

Components	Crosscutting patterns					Total of patterns	Total of changes
	BS	Oct	GC	DC	BC		
BaseController	0	4	1	0	0	5	5
MediaAccessor	0	3	1	0	0	4	5
MediaController	2	4	1	0	0	7	5
MediaUtil	2	4	1	0	0	7	5
PhotoViewScreen	0	3	0	0	0	3	4
Complete list is at Applying and Evaluating Concern-Sensitive Design Heuristics (2010)							
SmsReceiverThread	0	2	0	0	0	2	1
SmsSenderThread	0	2	0	0	0	2	1
UnavailablePhoto AlbumException	0	2	0	0	0	2	1
BaseThread	0	0	0	0	0	0	0
SplashScreen	0	0	0	0	0	0	0
Average	0.3	2.9	0.7	0.0	0.0	3.8	1.8
Correlation	41.2%	40.5%	27.6%	0.0%	0.0%	48.5%	

Table 9
Unstable Health Watcher components and crosscutting patterns (R09).

Components	Crosscutting Patterns					Total of patterns	Total of changes
	BS	Oct	GC	DC	BC		
HealthWatcherFacade	0	3	1	0	0	4	6
HWServlet	0	1	1	0	0	2	6
UpdateHealthUnitSearch	0	3	1	0	0	4	6
UpdateComplaintData	0	2	1	0	0	3	5
GetDataForSearchBy DiseaseType	0	2	1	0	0	3	4
Complete list is at Applying and Evaluating Concern-Sensitive Design Heuristics (2010)							
Subject	0	0	0	0	0	0	0
SymptomRecord	0	3	1	0	0	4	0
ThreadLogging	0	0	0	0	0	0	0
TransactionException	0	3	0	0	0	3	0
UpdateEntryException	0	1	0	0	0	1	0
Average	0	1.5	0.5	0	0	2.0	1.1
Correlation	0	31.1%	47.9%	0	0	46.9%	

Specific design flaws detection

Concern heuristics are useful to detect specific design flaws. Applied rules R14 to R17 to identify instances of four bad smells :

- Shotgun Surgery,
- Feature Envy,
- Divergent Change,
- God Class

Table 10
Statistics for the heuristics related to bad smells.

Bad smell	Concern-sensitive rules		Conventional rules	
	Hits (%)	FP (%)	Hits(%)	FP (%)
Shotgun Surgery	10 (76.9%)	3 (23.1%)	9 (47.4%)	10 (52.6%)
Feature Envy	6 (85.7%)	1 (14.3%)	2 (25.0%)	6 (75.0%)
Divergent Change	4 (57.1%)	3 (42.9%)	*Not supported*	
God Class	9 (81.8%)	2 (18.2%)	2 (25.0%)	6 (75.0%)

Study constraints

The goal of this paper was to propose and to investigate concern-sensitive heuristics.

Strategy based empirical evidence:

(i) 50% as default for the concern diffusion heuristics

(ii) a meaningful ratio represents the definition of black sheep and octopus and similar thresholds used by Marinescu in the heuristics for bad smells.

Concluding remarks

(i) presented a suite of concern-sensitive heuristic rules,

(ii) investigated the hypothesis that these heuristics offer enhancements over typical metrics-based assessment approaches.

Investigation indicated promising results in favour of concern-sensitive heuristics.

Additionally, the proposed tool, ConcernMorph, could be extended to incorporate more sophisticated means for mining and mapping concerns to design elements.