# CO-EVOLVING CODE AND DESIGN WITH INTENSIONAL VIEWS

#### A CASE STUDY

KIM MENS (UCL)
ANDY KELLENS (VUB)
FRÉDÉRIC PLUQUET & ROEL WUYTS (ULB)









## PROBLEM: CONTEXT

- Intensional Views (and Relations)
- have been proposed as a way to support
  - program understanding
  - software maintenance
  - co-evolution of source code and design
- by "actively" documenting high-level structural regularities in source code

## PROBLEM: GOAL

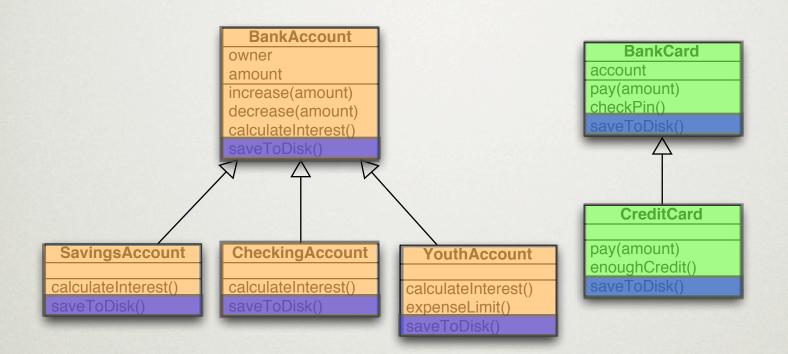
- Verify claims by conducting case study :
  - Document initial version
  - Apply to small evolution (1 month)
  - Apply to large evolution (+/-1 year)
- Draw conclusions about IntensiVE
  - w.r.t. software understanding,
     maintenance and (co-)evolution

# INTENSIONAL VIEWS (AND RELATIONS)

PROBLEM
INTENSIONAL VIEWS
INTENSIVE
THE CASE
CASE STUDY
LESSONS LEARNED
FUTURE WORK

- Motivating example
- Extensional vs. intensional views
- Alternative intensions
- Relations between views

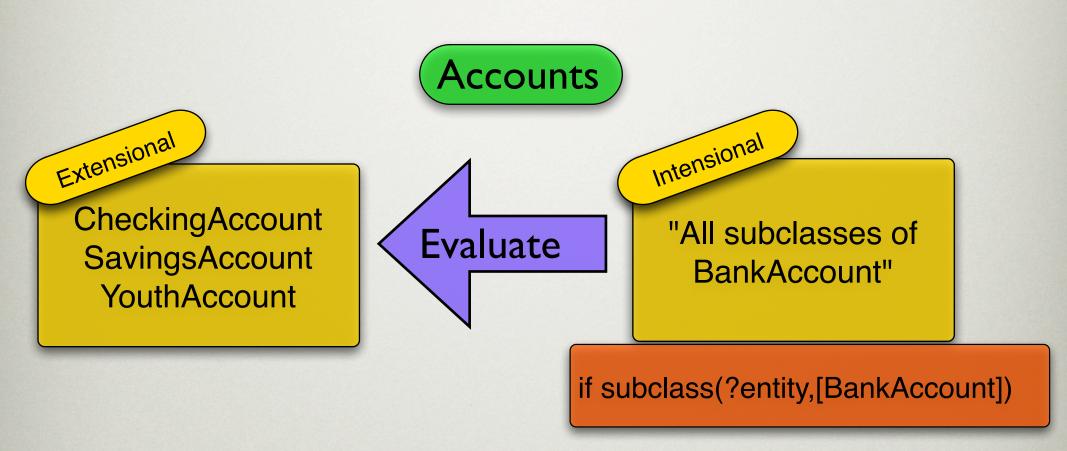
# MOTIVATING EXAMPLE



Source-code View
Collection of source-code entities

# EXTENSIONAL VS. INTENSIONAL VIEWS

PROBLEM
INTENSIVE
INTENSIVE
THE CASE
CASE STUDY
LESSONS LEARNED
FUTURE WORK



# **ALTERNATIVE**INTENSIONS

PROBLEM
INTENSIONAL VIEWS

INTENSIVE

THE CASE
CASE STUDY

LESSONS LEARNED

**FUTURE WORK** 

Accounts

"All subclasses of BankAccount"

"All classes in package BankAccount"

if subclass(?entity,[BankAccount])



if classInPackage(?entity,BankAccount)

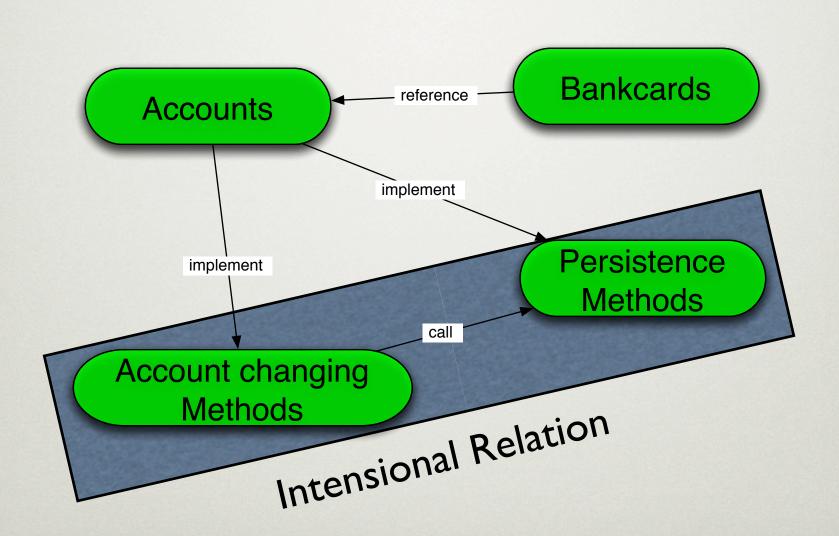
Exclude: BankAccount



**Extensional Consistency** 

# INTENSIONAL RELATIONS

PROBLEM
INTENSIVE
INTENSIVE
THE CASE
CASE STUDY
LESSONS LEARNED
FUTURE WORK



# INTENSIONAL RELATIONS

PROBLEM
INTENSIONAL VIEWS
INTENSIVE
THE CASE
CASE STUDY
LESSONS LEARNED
FUTURE WORK

Account changing Methods

- call -

Persistence Methods

All account changing methods must call a persistence method

 $\forall x \in \text{"Account changing methods"}$   $\exists y \in \text{"Persistence Methods"}$ x calls y

 $Q_1 x \in V_1$ ;  $Q_2 y \in V2$ : x r y

 $Q_1,Q_2 \in \{\forall,\exists,\exists !,\ldots\}$ 

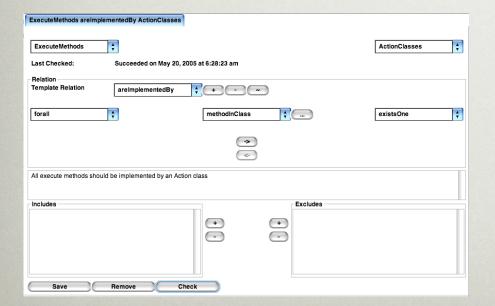
V<sub>1</sub>,V<sub>2</sub> ∈ Views

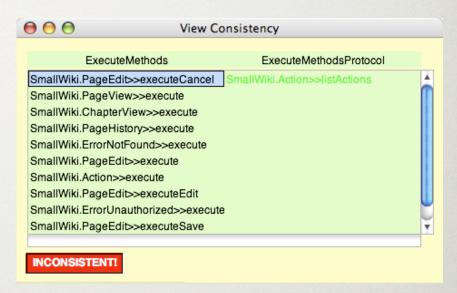
r = predicate over sourcecode entities

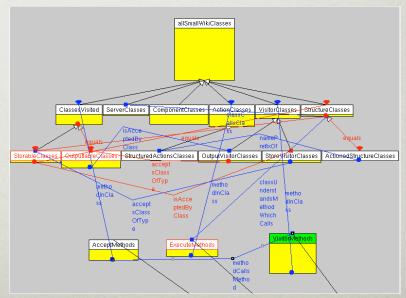
# THE INTENSIONAL VIEW ENVIRONMENT

PROBLEM
INTENSIONAL VIEWS
INTENSIVE
THE CASE
CASE STUDY
LESSONS LEARNED
FUTURE WORK

B ★ Root ■		ExecuteMethods	
The institution of the instituti	뫁	Execute Methods	
		Name ExecuteMetDodsPF Persame Delete Add att.	
		Language Soul	
StoreVisitorClasses ServerClasses		All methods in the action protocol	
StructuralClasses  ⇒ w allSmallWikiMethods		Intension:	<u>-</u>
- ₩ AcceptMethods		methodInProtocol(?entity,action)	_
■ ExecuteMethods ■ VisitorMethods			
AcceptMethods allCall Visitoriv     ClassesVisited subset Outputtal			
⊕ R ClassesVisited subset Structure			
⊕ B ExecuteMethods areImplements     ⊕ B OutputtableClasses allAreAcce			v
⊕ B OutputVisitorClasses allAccepte		Includes Excludes	
StoreVisitorClasses allAcceptC  StructureClasses subset Action  StructureClasses allUnderstand  StructureClasses calls ServerC  StructureClasses calls ServerC  StructureClasses subset Storab  VisitorMethods areImplemented  VisitorMethods someMethodsC-  VisitorMethods someMethodsC-		SmallWiki Actions-slistActions Save Extension	
∰-∳ Relations (1)		Check	







# THE INTENSIONAL VIEW ENVIRONMENT

PROBLEM
INTENSIONAL VIEWS
INTENSIVE
THE CASE
CASE STUDY
LESSONS LEARNED
FUTURE WORK

Seamlessly integrated with Visual works 7.3

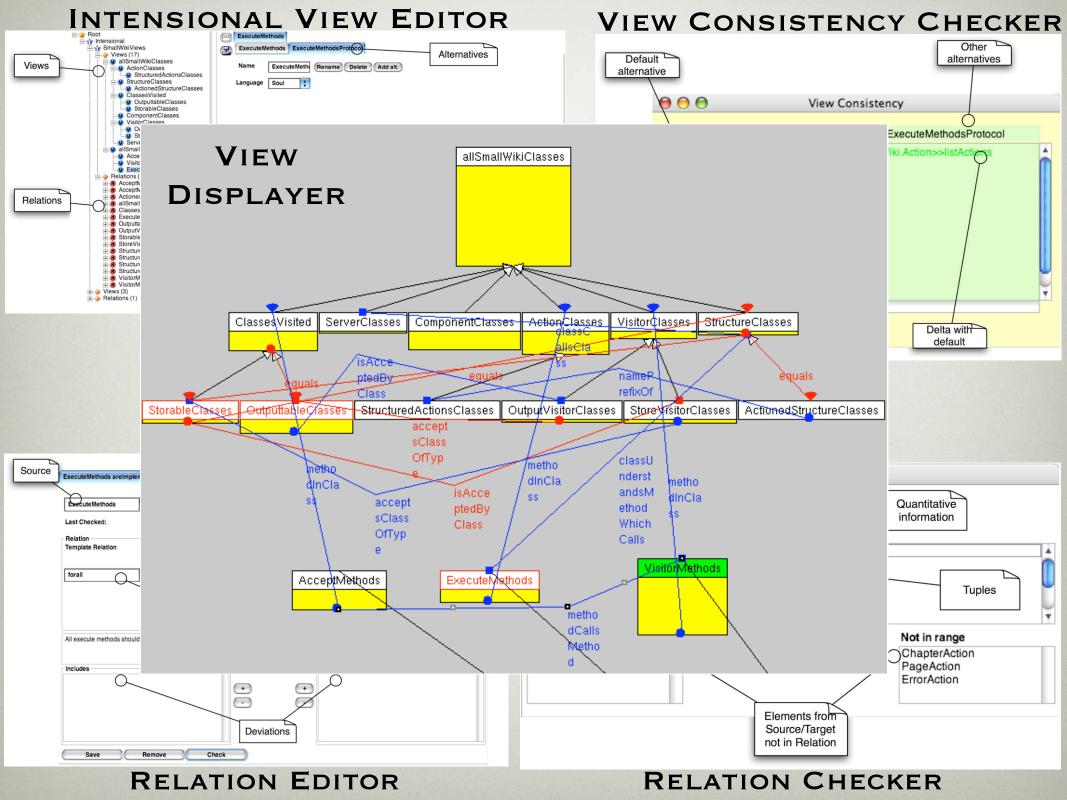
VisitorMethods Relations (15) AcceptMethods areImplementer AcceptMethods allCall Visitoriv	ensiV	E comp	orises
6: © Classes Visited subset Outputal 5: © Classes Visited subset Outputal 6: © Classes Visited subset Structura 6: © ExecuteMethods are implementa 7: © Output Visited Classes all Accept 7: © Output Visited Classes all Accept 7: © Structure Classes subset Strab 7: © Structure Classes subset Strab 7: © Structure Classes calls Server C 7: © Structure Classes subset Strab 7: © Visited Methods a releiplemented 7: © Visited Methods one Methods C. 7: © Visited Methods Strabet Medical 7: © Visited Methods Strabet Methods Methods 7: © Visited Methods Strabet Methods 7: © Visited Me			

ExecuteMethods	A T			ActionClasses	· · · · · · · · · · · · · · · · · · ·
ast Checked:	Succeeded on May 2	20, 2005 at 6:28:23 am			
Relation Template Relation					
	areImplementedBy				
forall	*	methodinClass	<b>†</b>	existsOne	A Y
T		• 7 7 7	1 •1		
<ul><li>Tr</li></ul>	nten	siVE	huil	ds 111	noi
• I1	ntens	siVE	buil	ds u	<b>po</b> 1
	ntens		buil	ds u	p01
			buil	ds uj	<b>po</b> 1
All execute methods sh			buil	ds uj	poı
				ds uj	poı

Save Remove Check

ExecuteMethods ExecuteMethodsProtocol
SmallWiki.PageEdit>>executeCancel SmallWiki.Action>>listActions
SmallWiki.PageView>>execute
Intensional View Editor
SmallWiki.ErrorNotFound>>execute
View Consistency Checker
Relation Editor
Relation Checker
View Displayer
viece Bispiliger
Star Browser Action Classes   Action Cla
equals Crass refisor
Storable Chases
SOUL + NewLicor

MOOSE + CodeCrawler



**FUTURE WORK** 

## THE CASE: SMALLWIKI

• SmallWiki

• 15 views

- 15 nesting views
- 17 relations
- Document initial version
- Apply to small evolution (1 month)
- Apply to large evolution (+/- 1 year)

GOAL::
Support:co-evolution:of:
documentation:and:
implementation

### **METHODOLOGY**

- Manual code inspection
  - Initial views/relations
- Check conformance:
  - Detected small "bugs"
  - Refine documentation
- Iterative process

# APPLYING TO LATER VERSIONS

PROBLEM
INTENSIONAL VIEWS
INTENSIVE
THE CASE
CASE STUDY
LESSONS LEARNED
FUTURE WORK

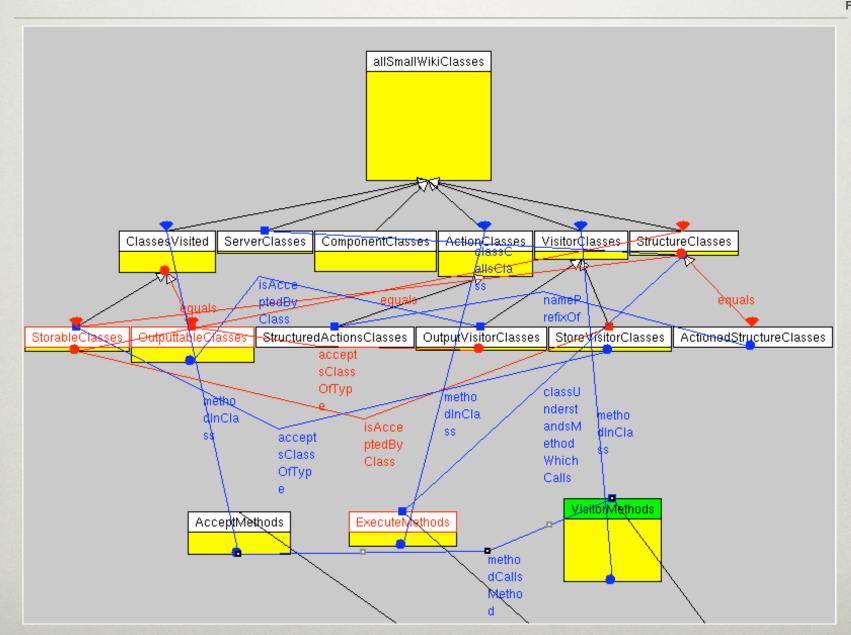
- Check views/relations to new version of code
- Inspect conflicts
  - Conflicts because of restructuring the code
- Update documentation when needed

### LESSONS LEARNED

- Visualization
- Understanding (Evolution)
- Query languages
- Other

PROBLEM

### VISUALIZATION



# UNDERSTANDING (EVOLUTION)

PROBLEM
INTENSIONAL VIEWS
INTENSIVE
THE CASE
CASE STUDY
LESSONS LEARNED
FUTURE WORK

- Start: little knowledge about SmallWiki
- Incremental documentation:
  - building views/relations in small steps
  - increased insights into the internals of the system
- Checking conformance: insights into evolution of the system

Incremental approach leads to better understanding of the system

### QUERY LANGUAGE

```
namesClasses |
namesClasses := SmallWiki.VisitorOutput allSubclasses inject: Set new
     into:
        [:tas:nextclass |
        tas
          addAll: (((Soul.MLI current methodsInClass: nextclass)
                 select: [:method | 'accept*:' match: method selector])
                    collect: [:method | method selector copyFrom: 7 to: method selector size - 1]);
          yourself].
namesClasses inject: OrderedCollection new
    into:
      [:tas:next |
       | value |
      (value := (Soul.MLI current classesFor: next asSymbol)
             detect: [:class | SmallWiki allClasses includes: class]
             ifNone: [nil] ) isNil ifFalse: [tas add: value].
      tas
```

classInHierarchyOf(?c, [SmallWiki.VisitorOutput]),
methodWithNameInClass(?,?s, ?c),
[?s = (#accept, ?entity name, ':') asSymbol]

### OTHER

- Coverage
- Deviations
  - some evolution conflicts due to obsolete deviations
- Dynamic vs. Static
- Do we need logic?
- Scaleability

### **FUTURE WORK**

- Mining:
  - Views (FCA + ILP?)
  - Relations (Brute-force technique)
- Document dynamic structure
- Support for documenting versions
- Aspect Documentation

### WANNA LEARN MORE?

- Contact:
  - Kim Mens
     (km@info.ucl.ac.be)
  - Andy Kellens
     (akellens@vub.ac.be)
  - Frédéric Pluquet
  - Roel Wuyts
- Publications:
  - TOOLS1999, SEKE2002, ICSM2003, ESUG2005, ICSM2005 (2)

#### Wanna learn more?

- Innovation Awards (tonight)
  - Andy Kellens
  - (✓ vote for us •)
- Technical track (tomorrow)
  - Frédéric Pluquet
  - Tools for Intensional Views
- Procheck our Website

www.intensional.be