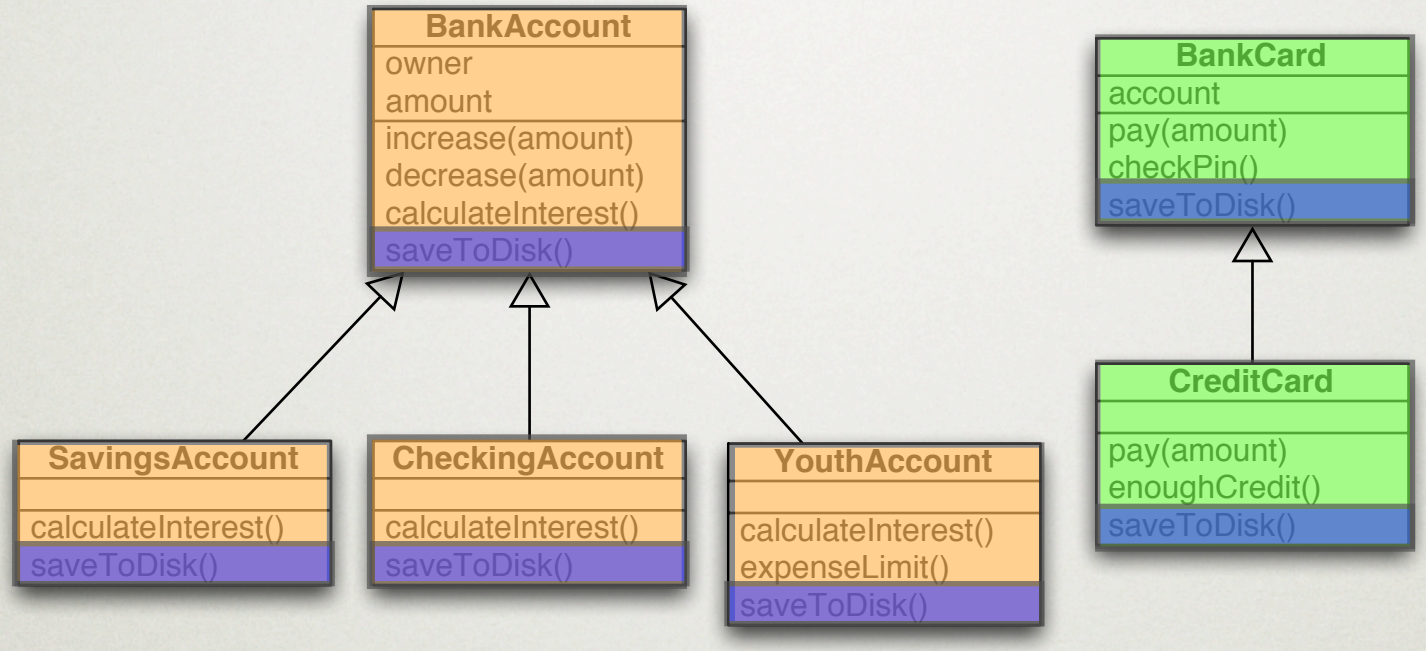
Intensional Views and relations

Andy Kellens
Kim Mens
Frédéric Pluquet
Roel Wuyts

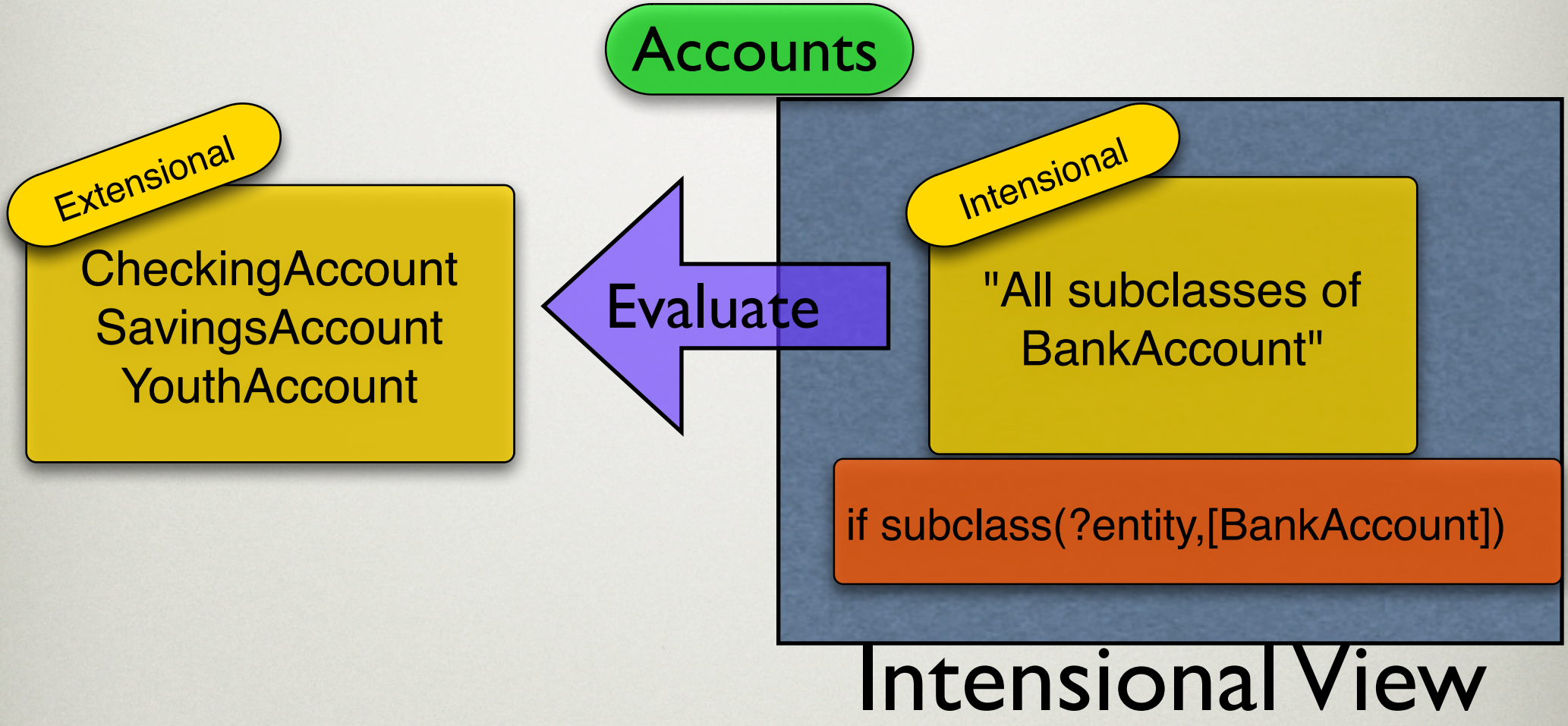
ULB

Documentation

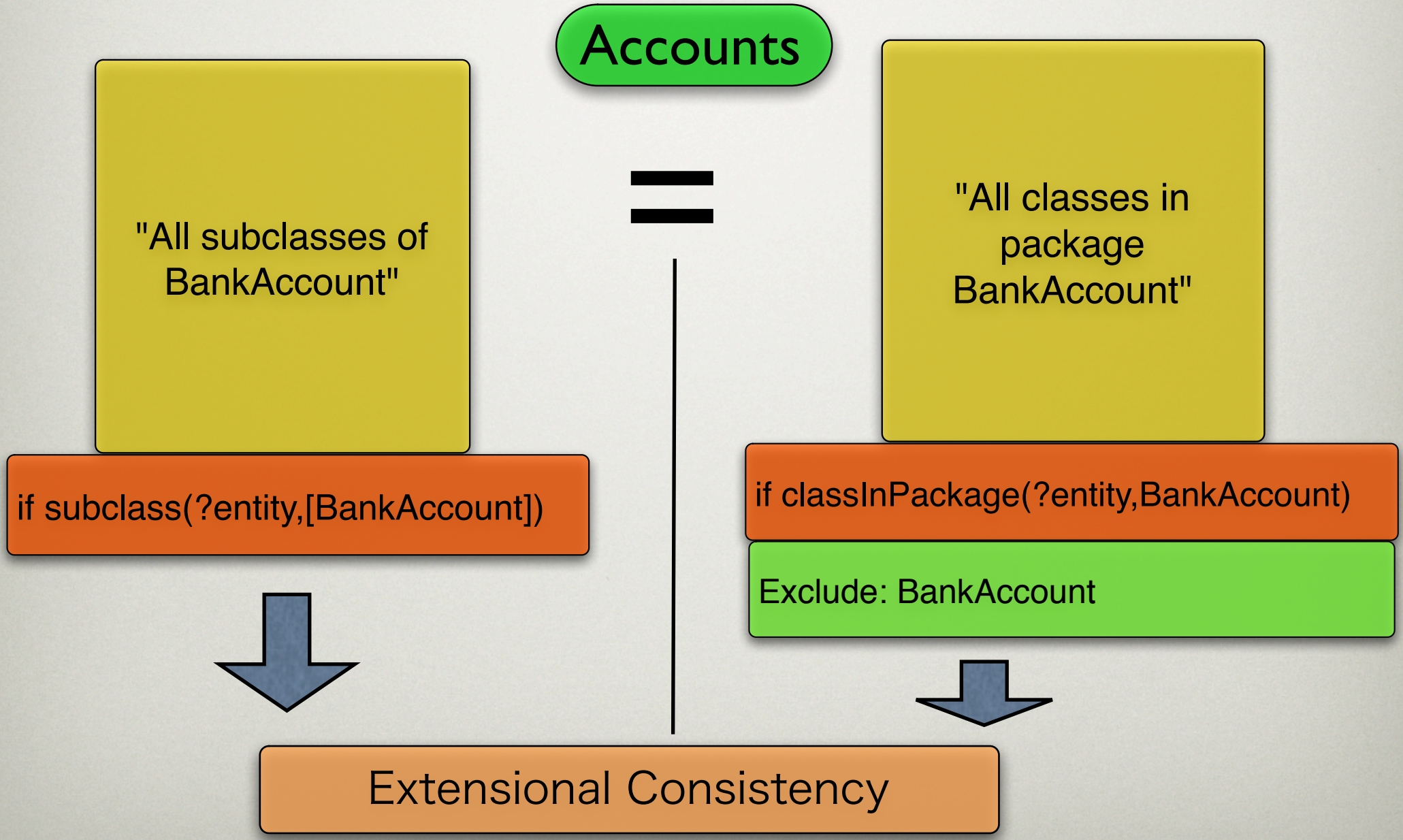


Source-code View
Collection of source-code entities

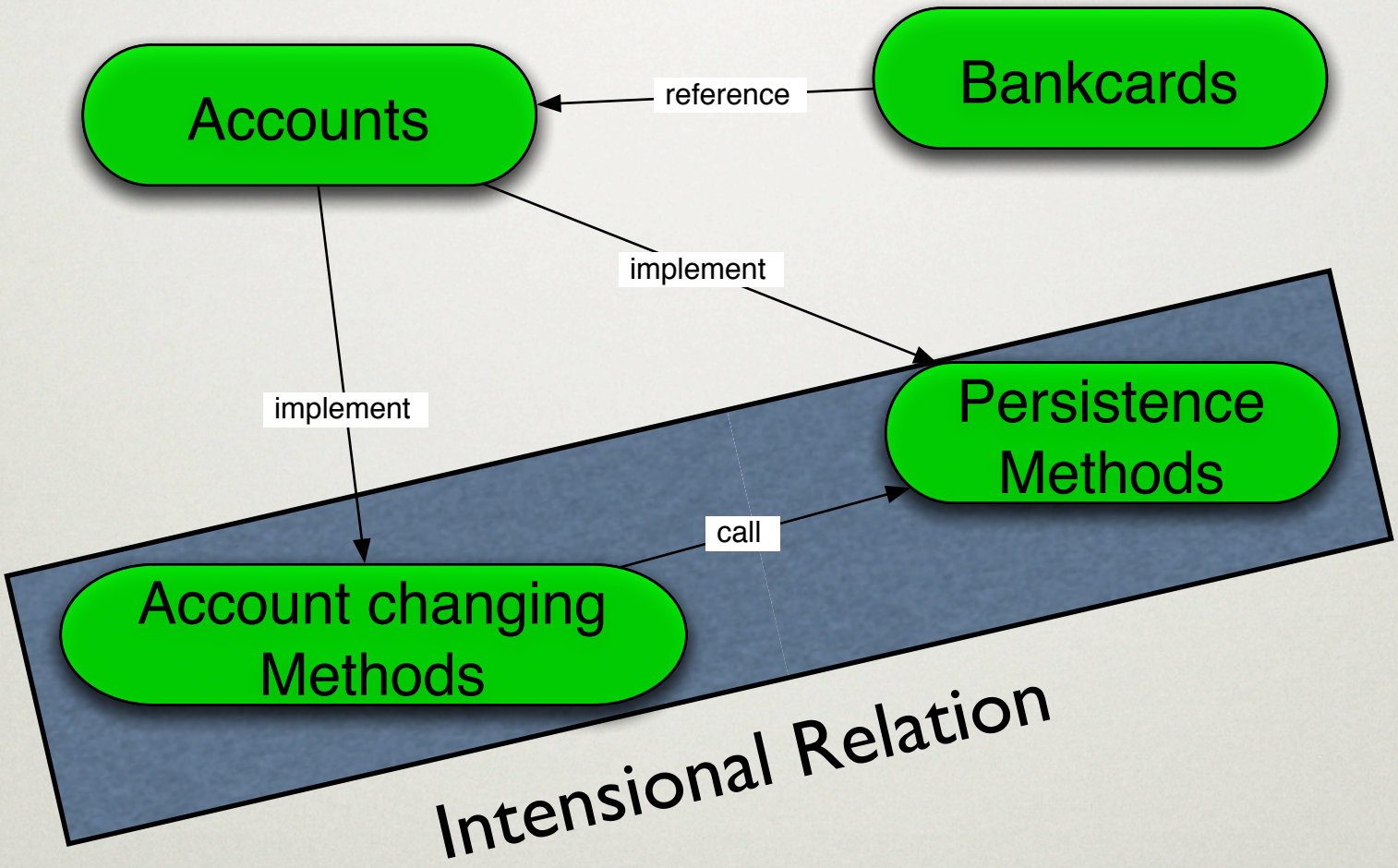
Extensional vs. Intensional



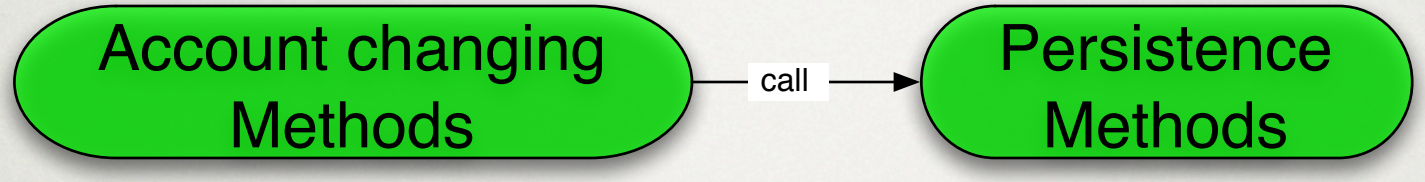
Alternative views



Relations


Accounts
reference
Bankcards
implement
implement
Persistence Methods
call
Account changing Methods
Intensional Relation

Relations



All account changing methods must call a persistence method

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

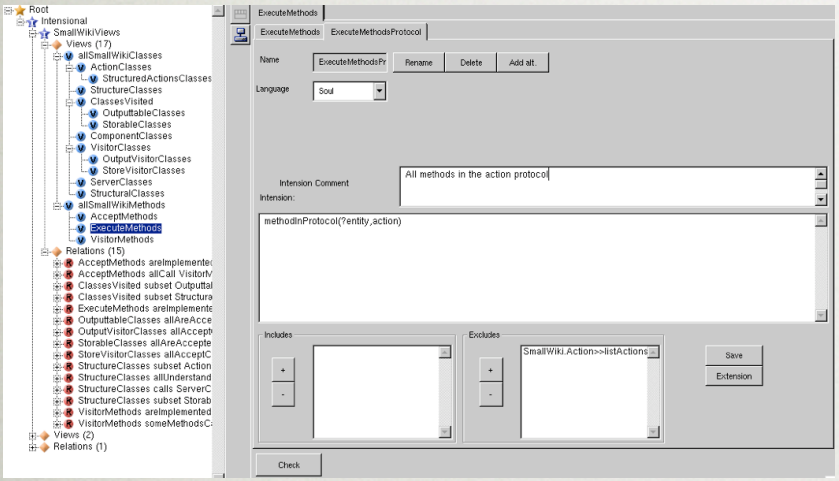
$Q_1 x \in V_1; Q_2 y \in V2: x\ r\ y$

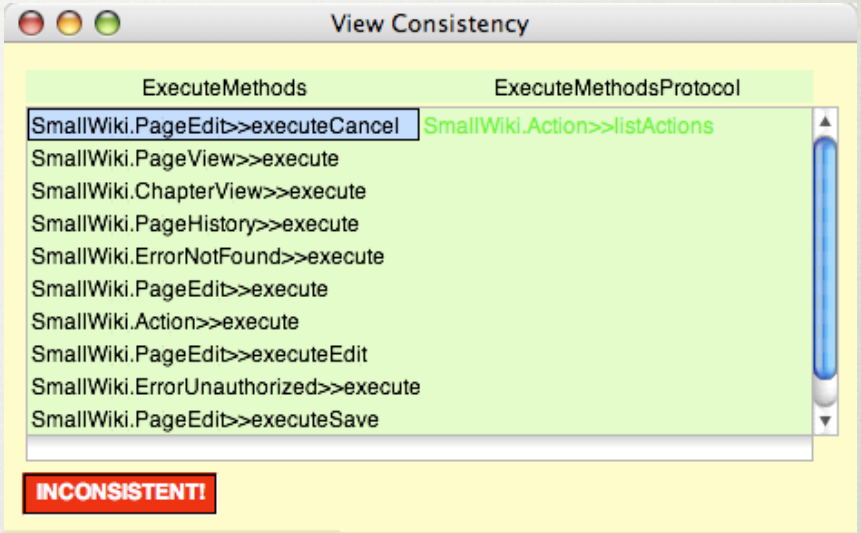
$Q_1, Q_2 \in \{\forall, \exists, \nexists, \exists !, \ldots\}$

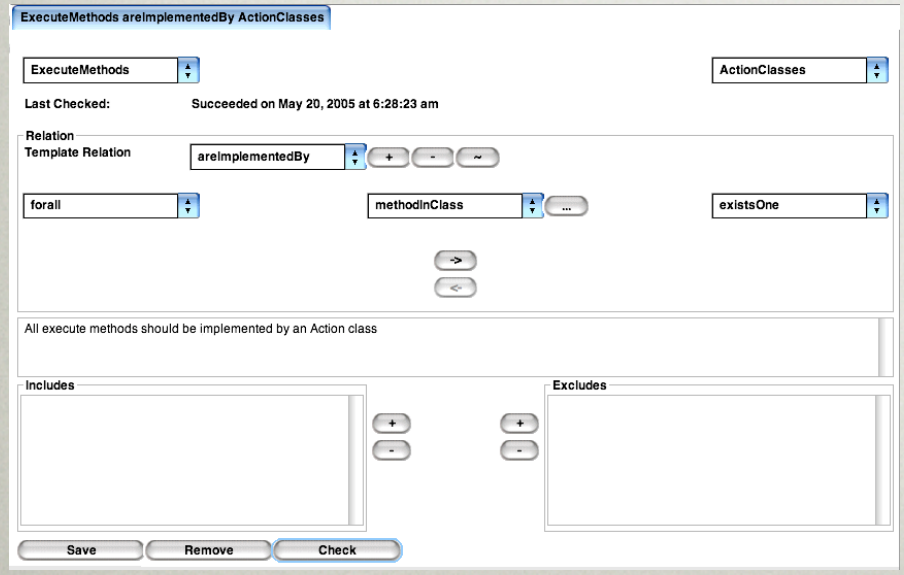
$V_1, V_2 \in$ Views

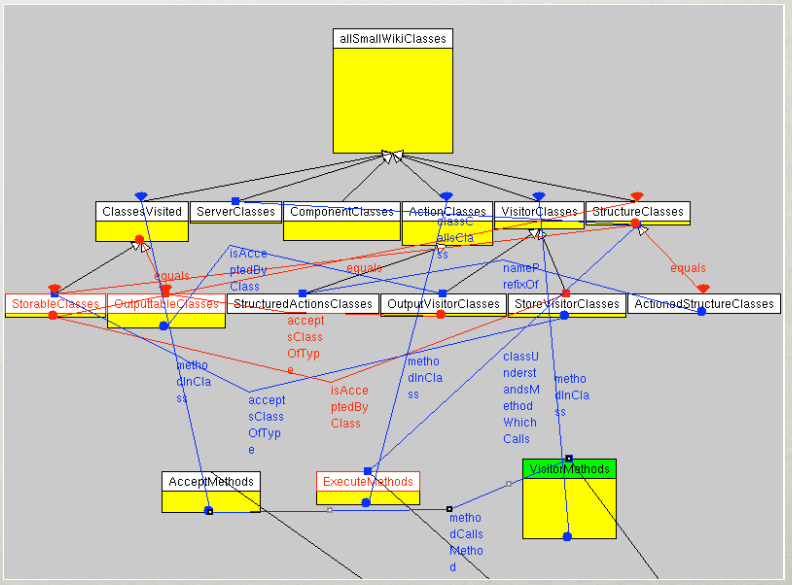
r = predicate over source-code entities

Tools


Root
Intensional
SmallWikiViews
Views (17)
allSmallWikiClasses
ActionClasses
StructuredActionsClasses
StructureClasses
ClassesVisited
OutputtableClasses
StorableClasses
ComponentClasses
VisitorClasses
OutputVisitorClasses
StoreVisitorClasses
ServerClasses
StructuralClasses
allSmallWikiMethods
AcceptMethods
ExecuteMethods
VisitorMethods
Relations (15)
Views (2)
Relations (1)
ExecuteMethods
ExecuteMethodsProtocol
Name
Rename
Delete
Add alt.
Language
Intension Comment
All methods in the action protocol
Intension:
methodInProtocol(?entity,action)
Includes
Excludes
SmallWiki.Action>>listActions
Save
Extension
Check


View Consistency
ExecuteMethods
ExecuteMethodsProtocol
SmallWiki.PageEdit>>executeCancel
SmallWiki.Action>>listActions
SmallWiki.PageView>>execute
SmallWiki.ChapterView>>execute
SmallWiki.PageHistory>>execute
SmallWiki.ErrorNotFound>>execute
SmallWiki.PageEdit>>execute
SmallWiki.Action>>execute
SmallWiki.PageEdit>>executeEdit
SmallWiki.ErrorUnauthorized>>execute
SmallWiki.PageEdit>>executeSave
INCONSISTENT!


ExecuteMethods areImplementedBy ActionClasses
ExecuteMethods
ActionClasses
Last Checked:
Succeeded on May 20, 2005 at 6:28:23 am
Relation
Template Relation
areImplementedBy
forall
methodInClass
existsOne
All execute methods should be implemented by an Action class
Includes
Excludes
Save
Remove
Check


allSmallWikiClasses
ClassesVisited
ServerClasses
ComponentClasses
ActionClasses
VisitorClasses
StructureClasses
StorableClasses
OutputtableClasses
StructuredActionsClasses
OutputVisitorClasses
StoreVisitorClasses
ActionedStructureClasses
AcceptMethods
ExecuteMethods
VisitorMethods
equals
isAcceptedByClass
namePrefixOf
acceptsClassOfType
methodInClass
classUnderstandsMethodWhichCalls
methodCallsMethod

Case Study

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

Methodology

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

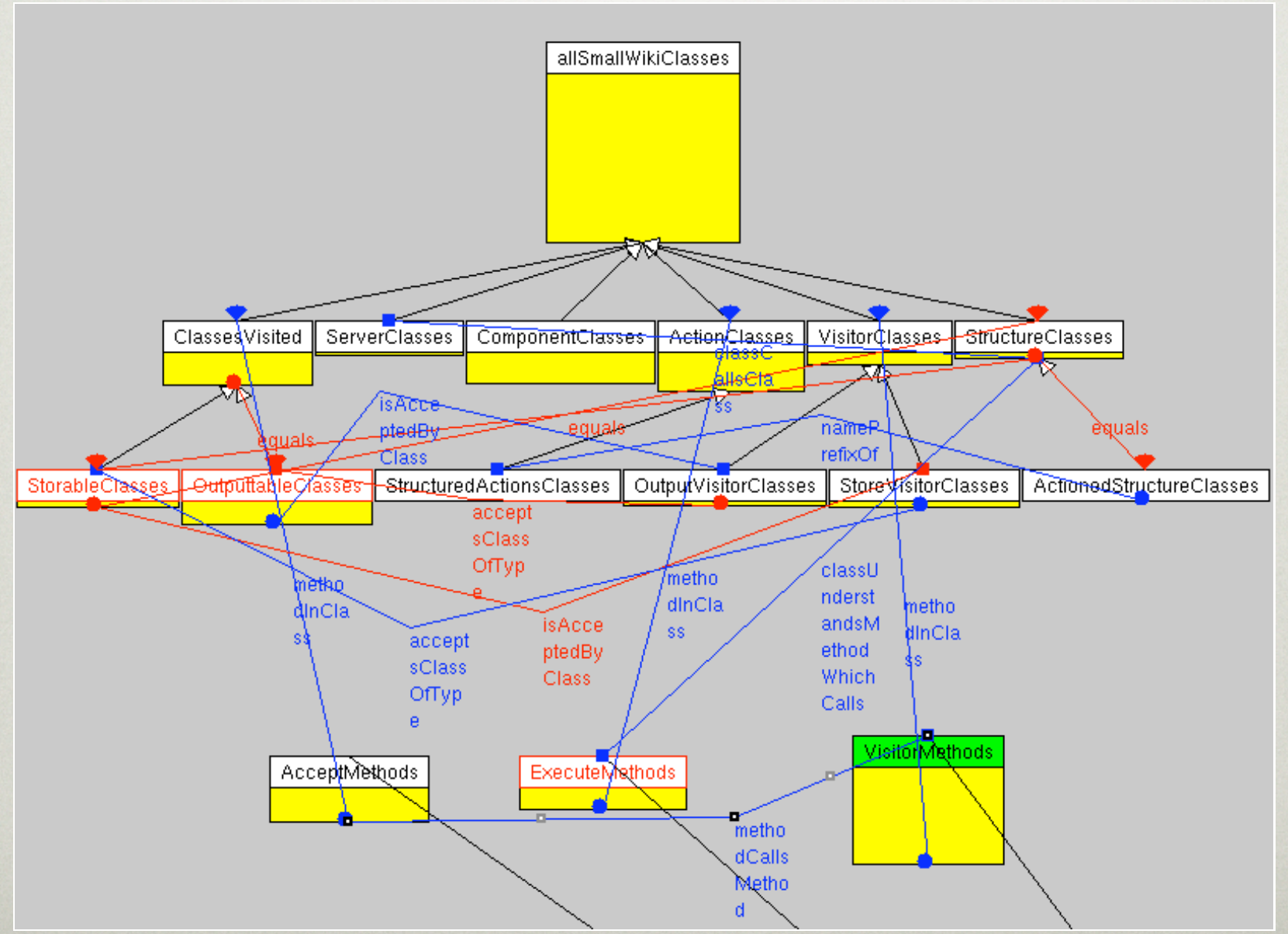
Applying to later versions

- 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

Visualization



Understanding (evolution)

- 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

- Completeness
- Deviations
- Dynamic vs. Static
- Do we need logic?

Future work

- Mining:
  - Views (FCA)
  - Relations (Brute-force technique)
- Document dynamic structure
- Support for assessing impact of evolution
- Aspect Documentation

Questions