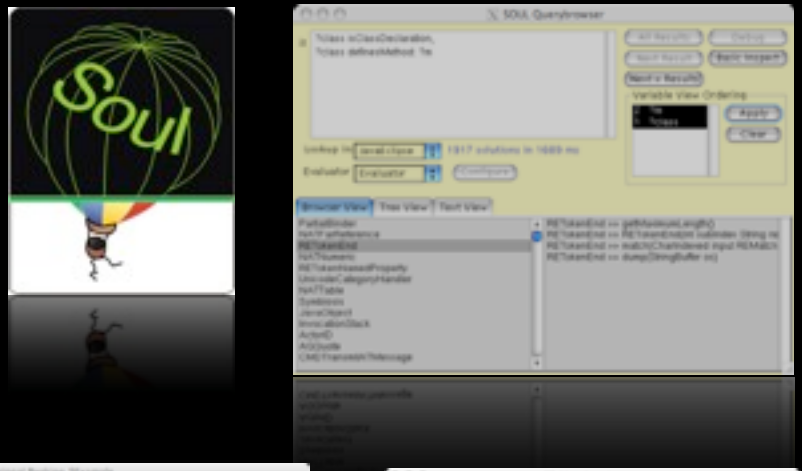
Visual(ized) Source-code Querying

Sattose 2009

Département d'Ingénierie Informatique - Université catholique de Louvain

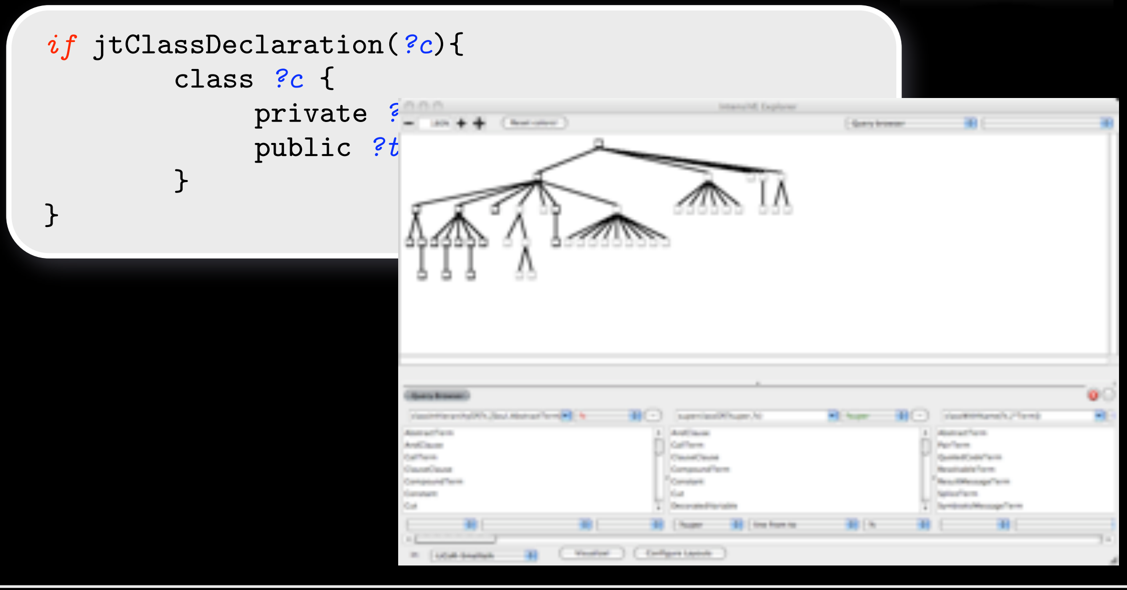
Background

- **Logic-based Program Reasoning**
  - Declarative Code Queries



- **Intensional Software Views**
  - Software Evolution Support

- **Tool Integration**
  - Ease of use
  - General-purpose
  - Scalability



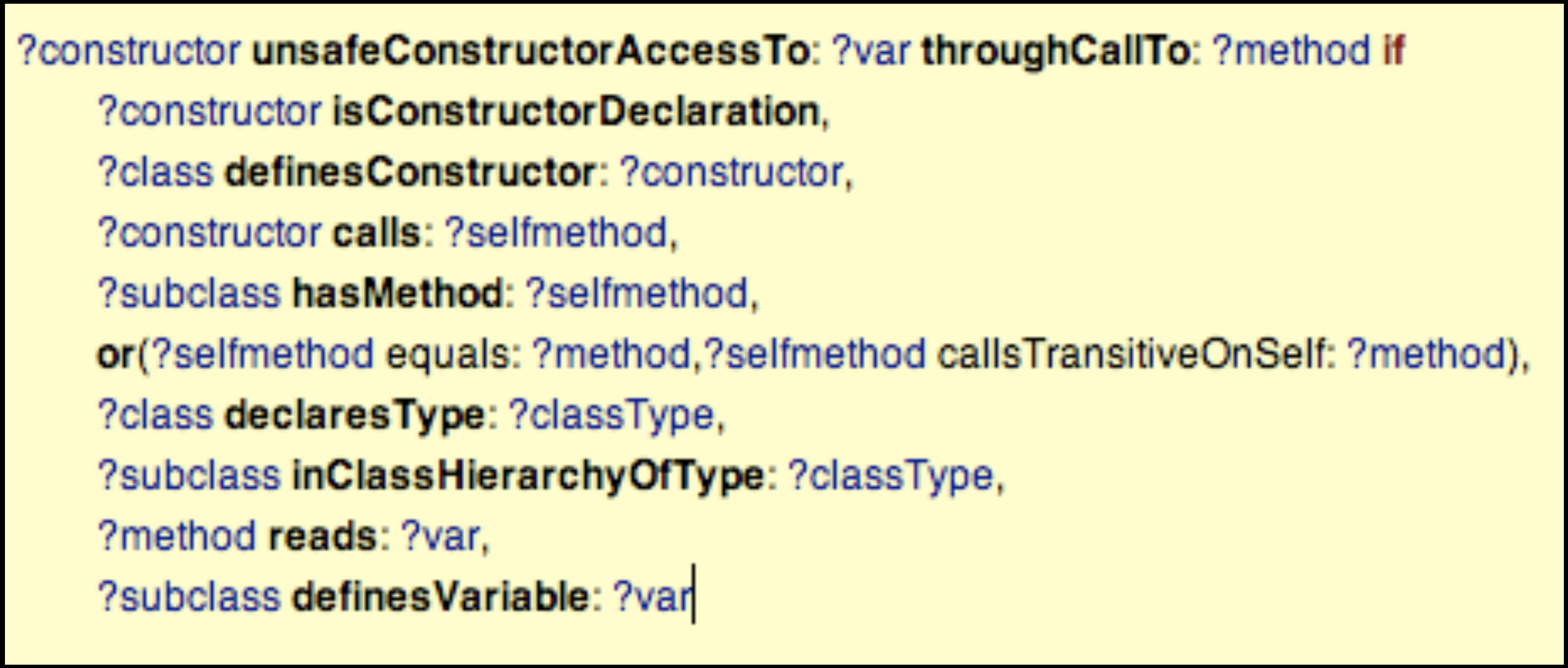
```
if jtClassDeclaration(?c){
        class ?c {
                private ?
                public ?t
        }
}
```

Queries and Visualizations

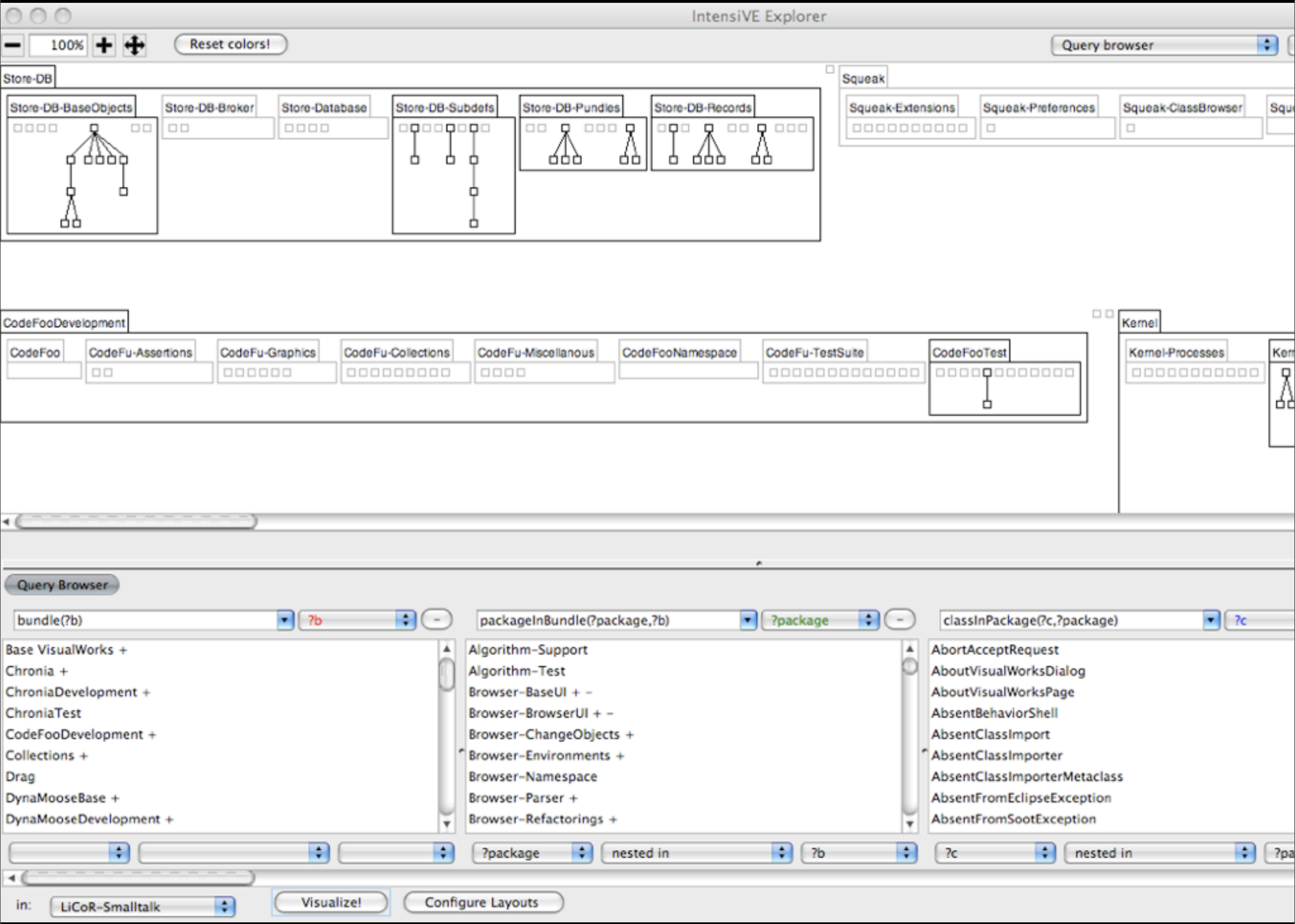
- **Declarative Program Queries**
  - Search program parts adhering to well-defined conditions
    - Bad smells and potential bug patterns
    - Design patterns
    - Coding conventions
    - Idioms
- **Structural Software Visualization**
  - Asses an entire system through a concise overview, exposing one or several properties
    - Modular coupling
    - System complexity
    - Metrics

Queries

- **Example:** **Constructors that call overridden instance methods that reference instance fields defined in the subclass may reference uninitialized fields (in Java)**

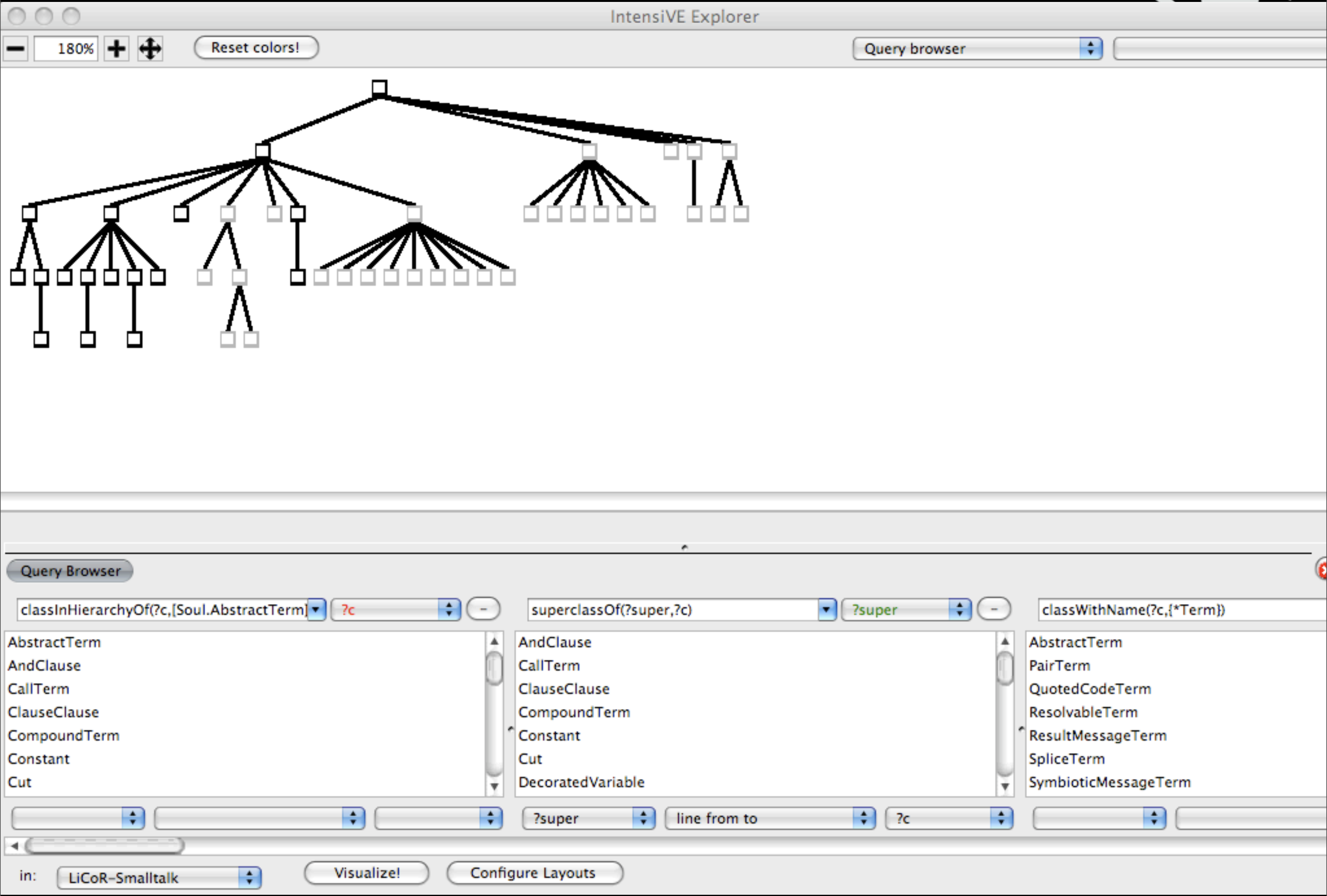


```
?constructor unsafeConstructorAccessTo: ?var throughCallTo: ?method if
      ?constructor isConstructorDeclaration,
      ?class definesConstructor: ?constructor,
      ?constructor calls: ?selfmethod,
      ?subclass hasMethod: ?selfmethod,
      or(?selfmethod equals: ?method,?selfmethod callsTransitiveOnSelf: ?method),
      ?class declaresType: ?classType,
      ?subclass inClassHierarchyOfType: ?classType,
      ?method reads: ?var,
      ?subclass definesVariable: ?var
```



bundle(?b)	packageInBundle(?package,?b)	classInPackage(?c,?package)
?b	?package	?c
Base VisualWorks +	Algorithm-Support	AbortAcceptRequest
Chronia +	Algorithm-Test	AboutVisualWorksDialog
ChroniaDevelopment +	Browser-BaseUI + -	AboutVisualWorksPage
ChroniaTest	Browser-BrowserUI + -	AbsentBehaviorShell
CodeFooDevelopment +	Browser-ChangeObjects +	AbsentClassImport
Collections +	Browser-Environments +	AbsentClassImporter
Drag	Browser-Namespace	AbsentClassImporterMetaclass
DynaMooseBase +	Browser-Parser +	AbsentFromEclipseException
DynaMooseDevelopment +	Browser-Refactorings +	AbsentFromSootException
	?package nested in ?b	?c nested in ?pa

The Intensional Views Environment



classInHierarchyOf(?c,[Soul.AbstractTerm]) | ?c
superclassOf(?super,?c) | ?super
classWithName(?c,{*Term})

AbstractTerm
AndClause
CallTerm
ClauseClause
CompoundTerm
Constant
Cut

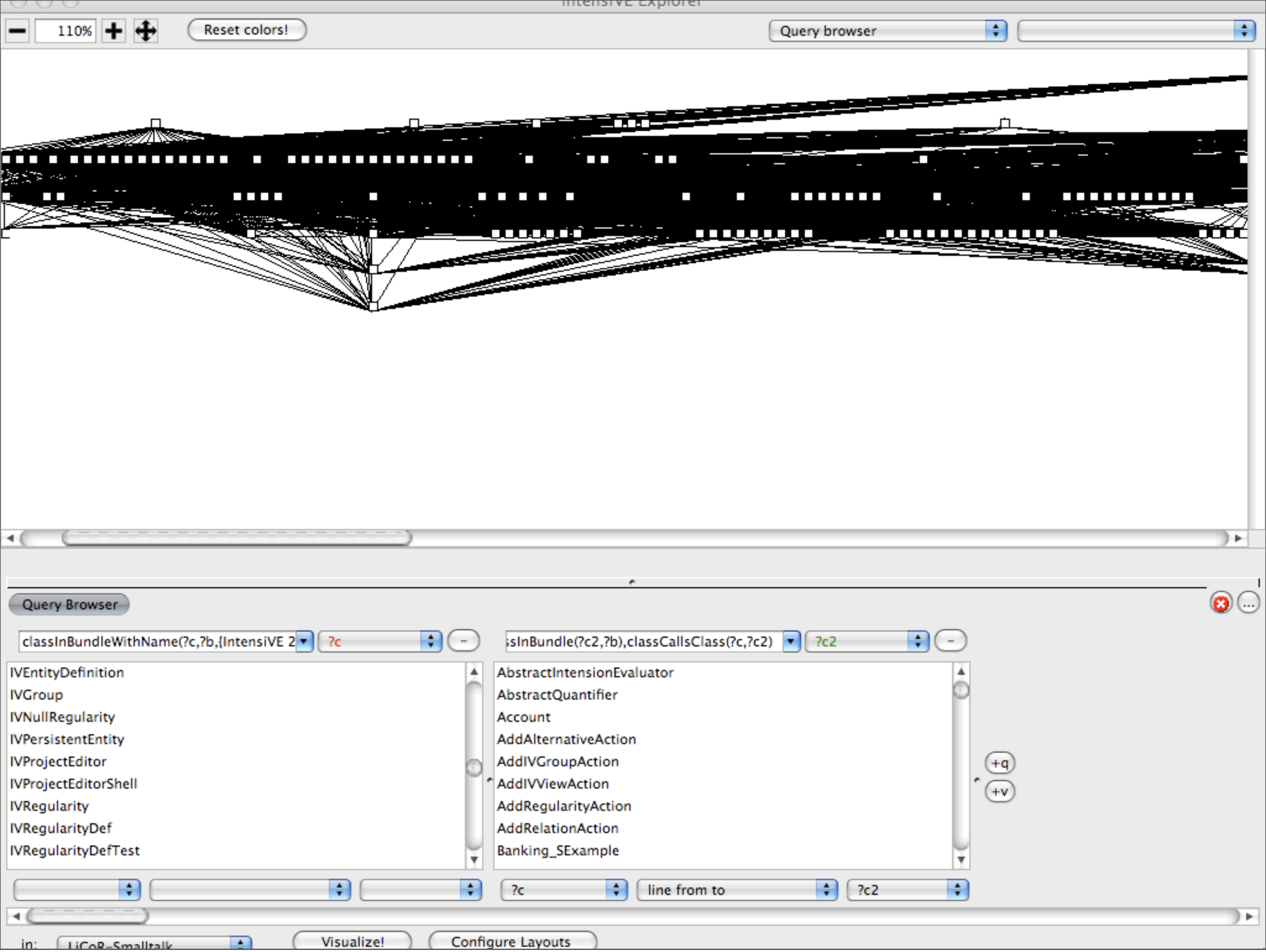
AndClause
CallTerm
ClauseClause
CompoundTerm
Constant
Cut
DecoratedVariable

AbstractTerm
PairTerm
QuotedCodeTerm
ResolvableTerm
ResultMessageTerm
SpliceTerm
SymbioticMessageTerm

?super | line from to | ?c

in: LiCoR-Smalltalk

The Intensional Views Environment



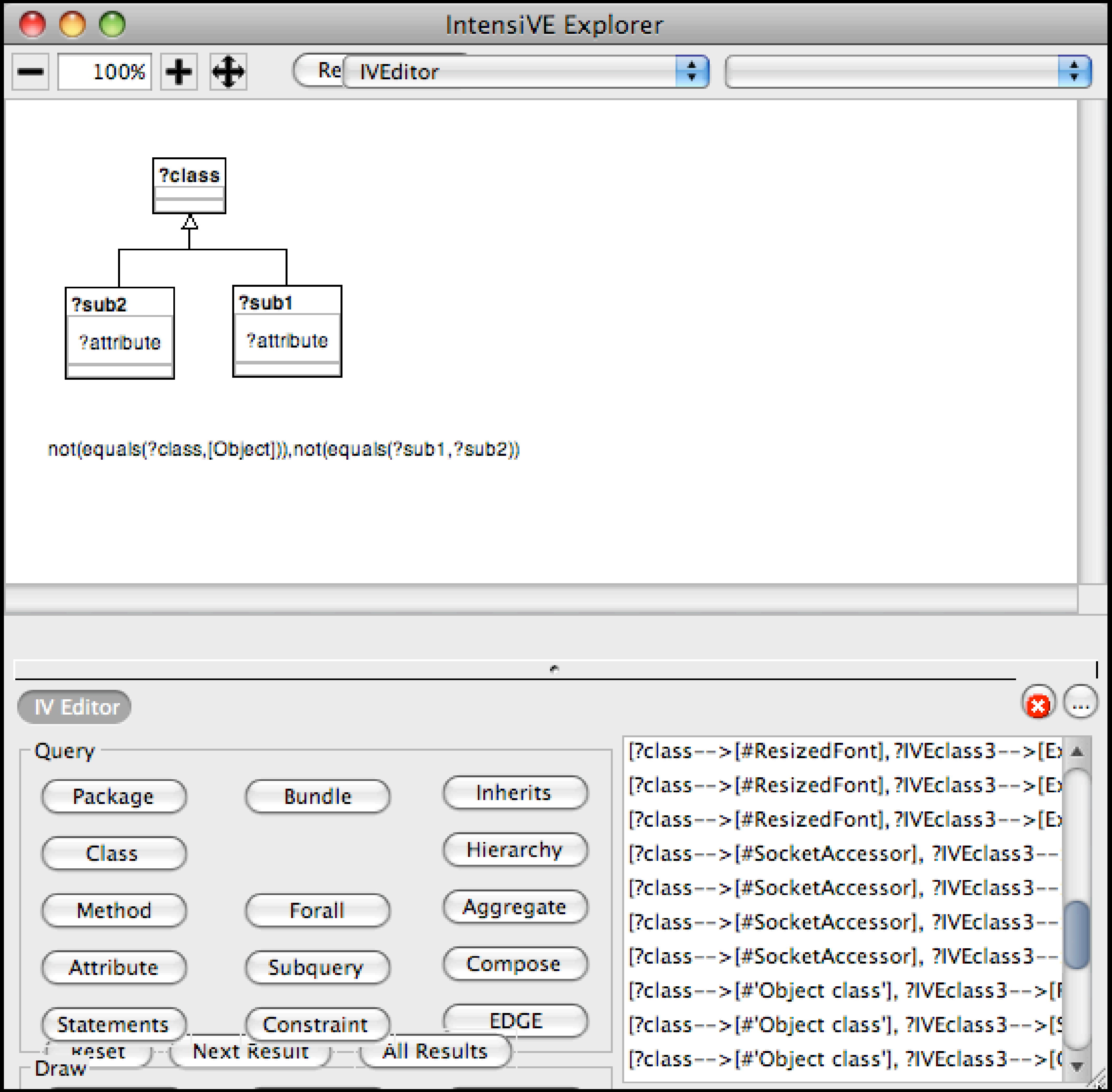
classInBundleWithName(?c,?b,{IntensiVE 2 | ?c

;sInBundle(?c2,?b),classCallsClass(?c,?c2) | ?c2

?c	?c2
IVEntityDefinition	AbstractIntensionEvaluator
IVGroup	AbstractQuantifier
IVNullRegularity	Account
IVPersistentEntity	AddAlternativeAction
IVProjectEditor	AddIVGroupAction
IVProjectEditorShell	AddIVViewAction
IVRegularity	AddRegularityAction
IVRegularityDef	AddRelationAction
IVRegularityDefTest	Banking_SExample

?c | line from to | ?c2

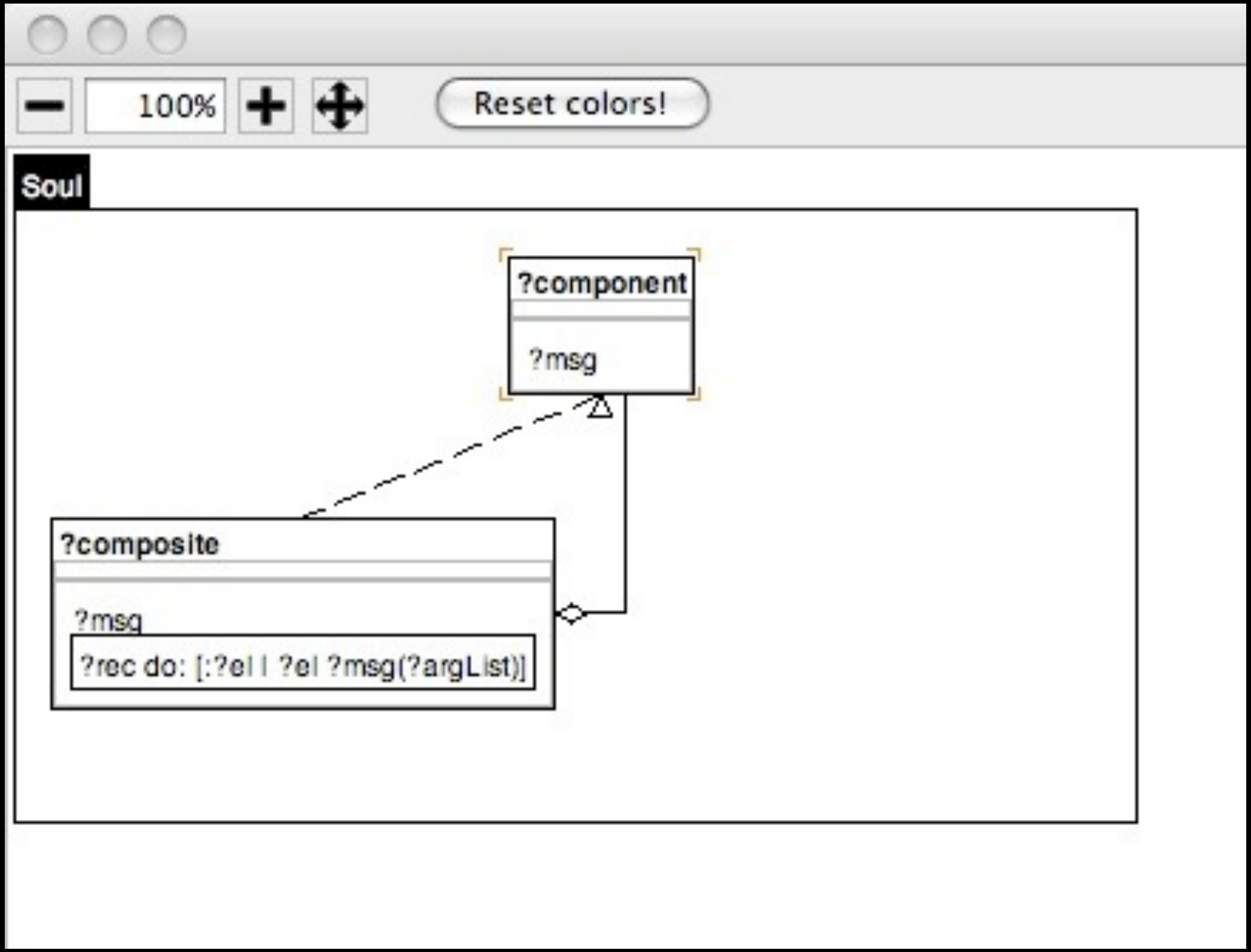
Visual Queries



```
not(equals(?class,[Object])),not(equals(?sub1,?sub2))
```

```
[?class-->[#ResizedFont], ?IVEclass3-->[E
[?class-->[#ResizedFont], ?IVEclass3-->[E
[?class-->[#ResizedFont], ?IVEclass3-->[E
[?class-->[#SocketAccessor], ?IVEclass3--
[?class-->[#SocketAccessor], ?IVEclass3--
[?class-->[#SocketAccessor], ?IVEclass3--
[?class-->[#SocketAccessor], ?IVEclass3--
[?class-->[#'Object class'], ?IVEclass3-->[
[?class-->[#'Object class'], ?IVEclass3-->[
[?class-->[#'Object class'], ?IVEclass3-->[
```

Composite DP



To conclude

- **Visualize *what* you query**
  - Entities (objects)
  - Relations (predicates)
- **Asses query results in a global context**
- **Narrowing focus metaphor**

- **Specify a visual query**
- **Example-based source-code queries**