

UCL

Université
catholique
de Louvain



USAGE CONTRACTS*

KIM MENS

UNIVERSITÉ CATHOLIQUE DE LOUVAIN (UCL)

JOINT WORK WITH

ANGELA LOZANO

ANDY KELLENS

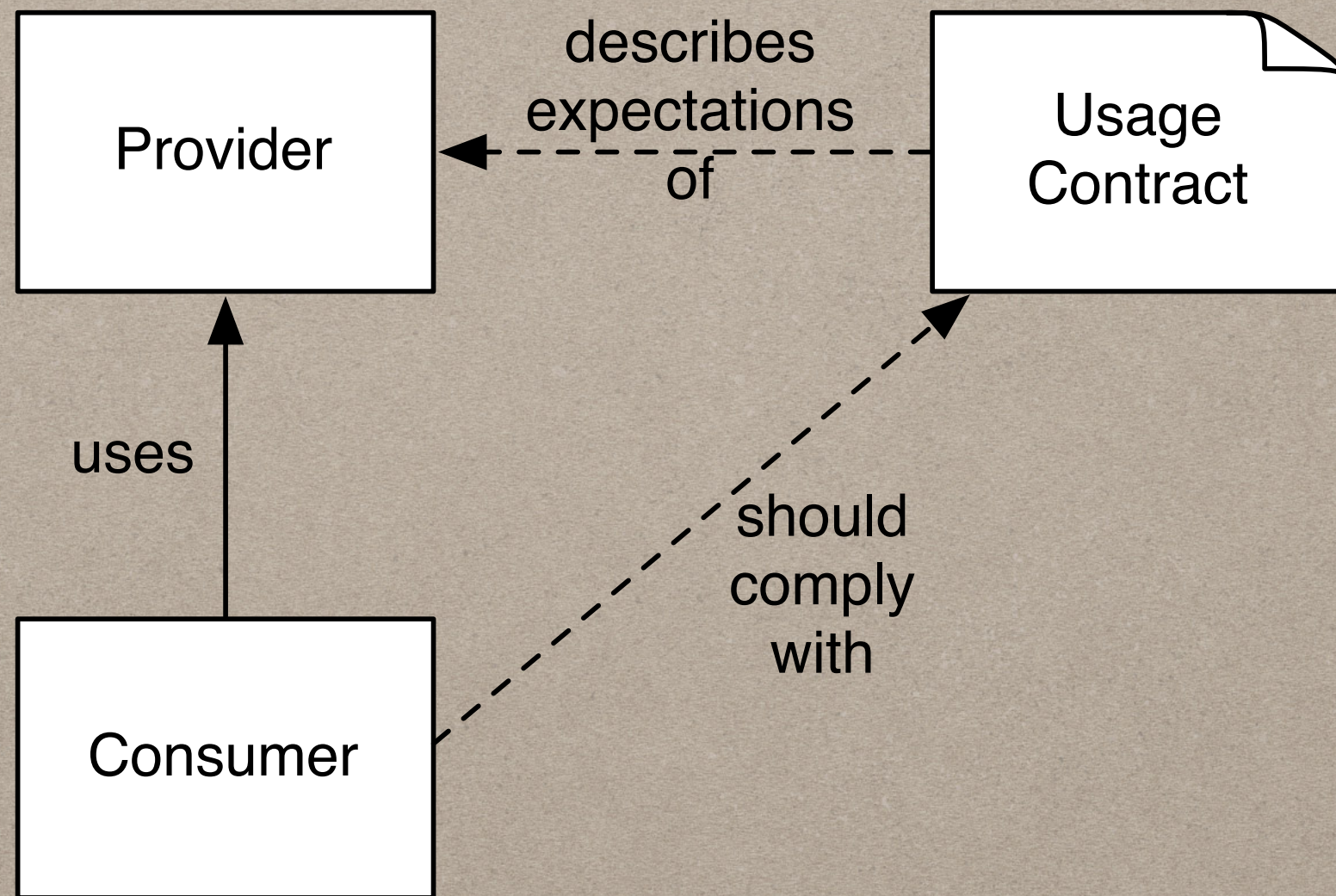
DAGSTUHL SEMINAR - "THE FUTURE OF REFACTORING"
LIGHTNING TALK - 21.05.2014

USAGE CONTRACTS

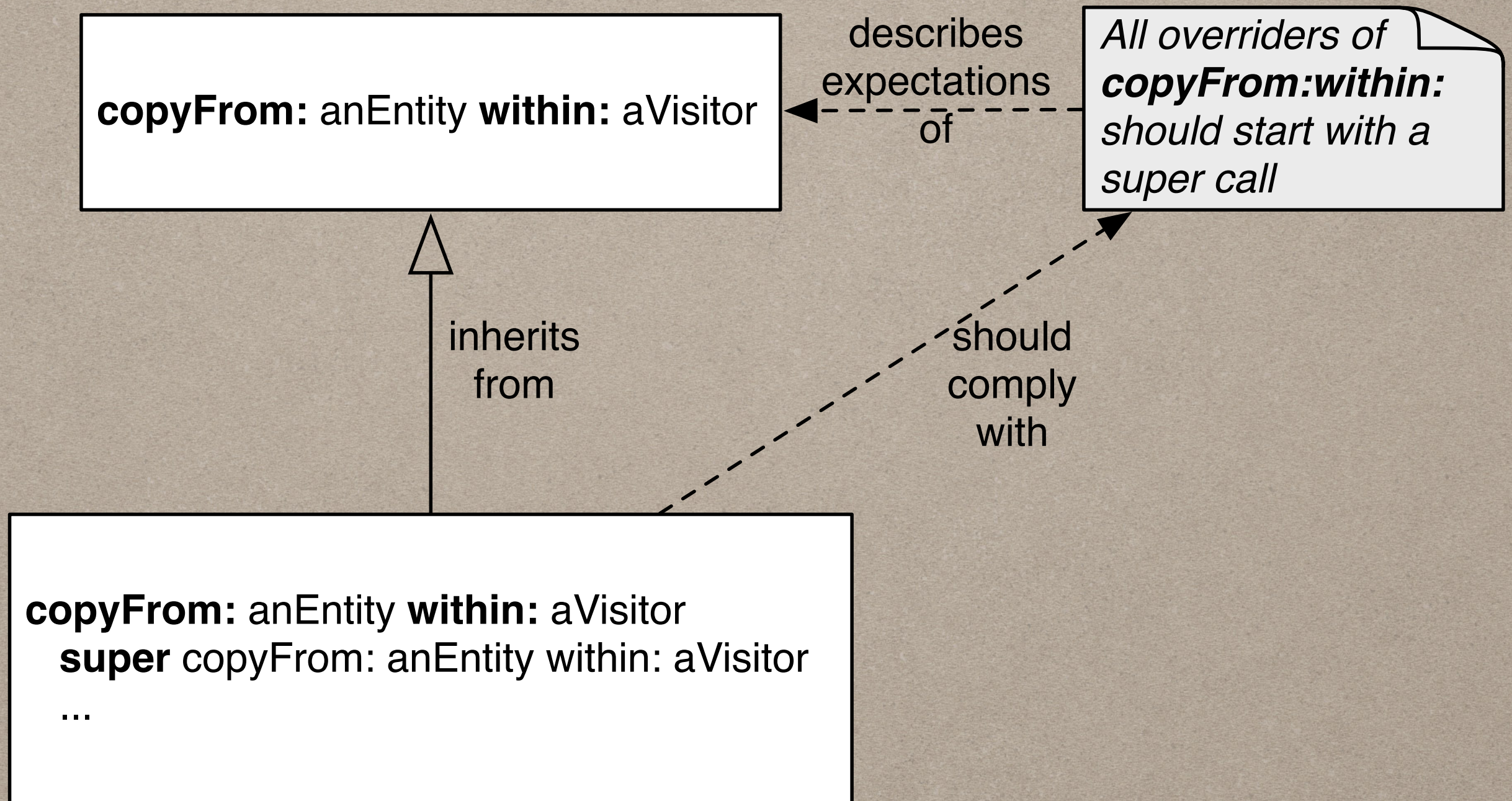
```
/**
 * Deactivates the tool. This method is called whenever the user switches to another tool
 * Use this method to do some clean-up when the tool is switched.
 * Subclassers should always call super.deactivate.
 * An inactive tool should never be deactivated.
 */
public void deactivate() {
    if (isActive()) {
        if (getActiveView() != null) {
            getActiveView().setCursor(new AWTCursor(java.awt.Cursor.DEFAULT_CURSOR));
        }
        getEventDispatcher().fireToolDeactivatedEvent();
    }
}
```

- We want a tool that allows encoding such regularities and offering immediate feedback on violations of such structural source-code regularities
- The tool should be proactive (violations reported 'on the fly' during coding)
- The tool should be "developer-friendly" (like unit testing but for usage expectations)
 - desired regularities expressed in the same programming language
 - tight integration with the integrated development environment
 - not coercive

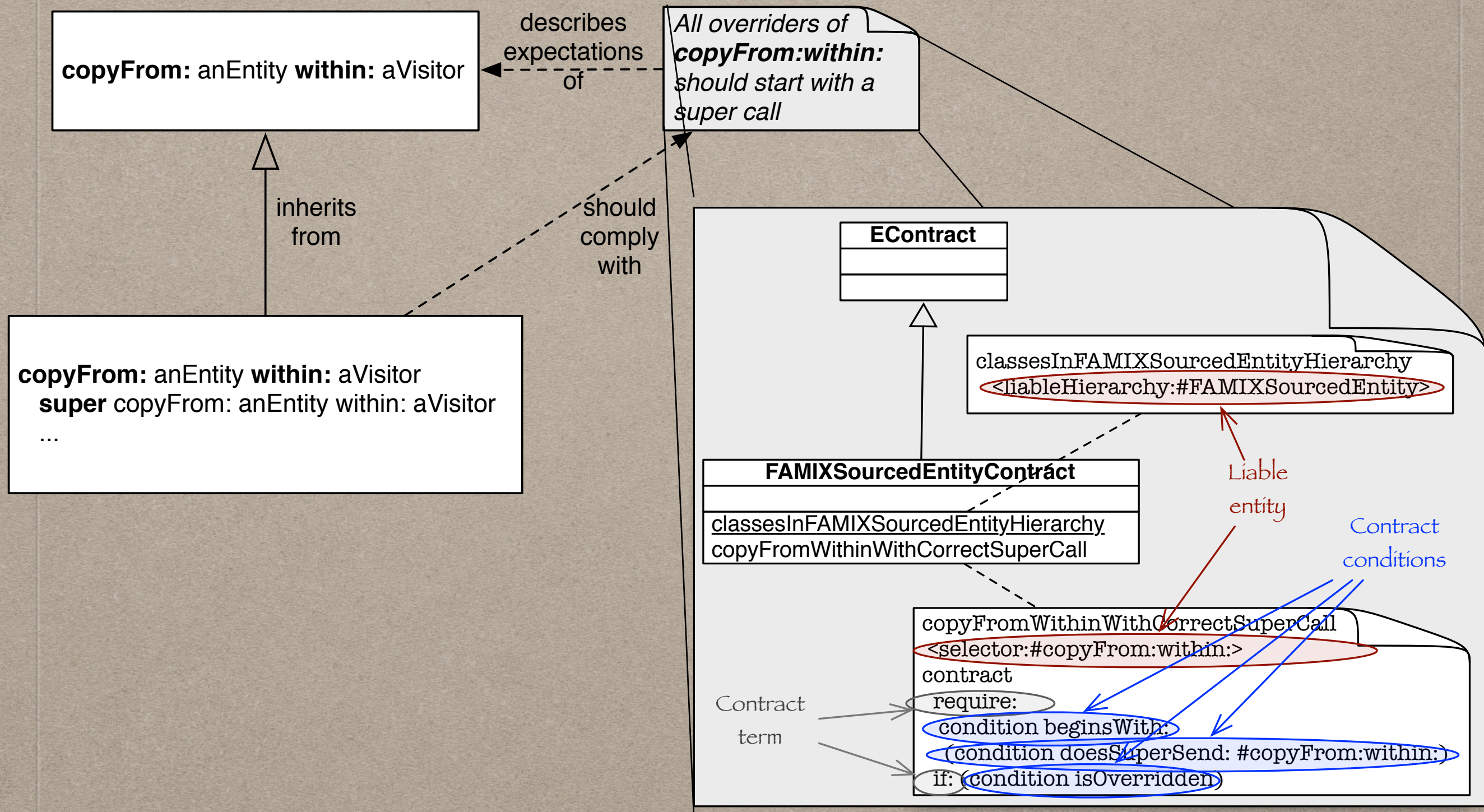
METAPHOR



EXAMPLE



EXAMPLE



UContracts : THE LANGUAGE

Liabie classes

- **liableClass**: regExp / **exceptClass**: regExp
- **liableHierarchy**: className / **exceptHierarchy**: className
- **liablePackage**: regExp / **exceptPackage**: regExp

Liabie methods

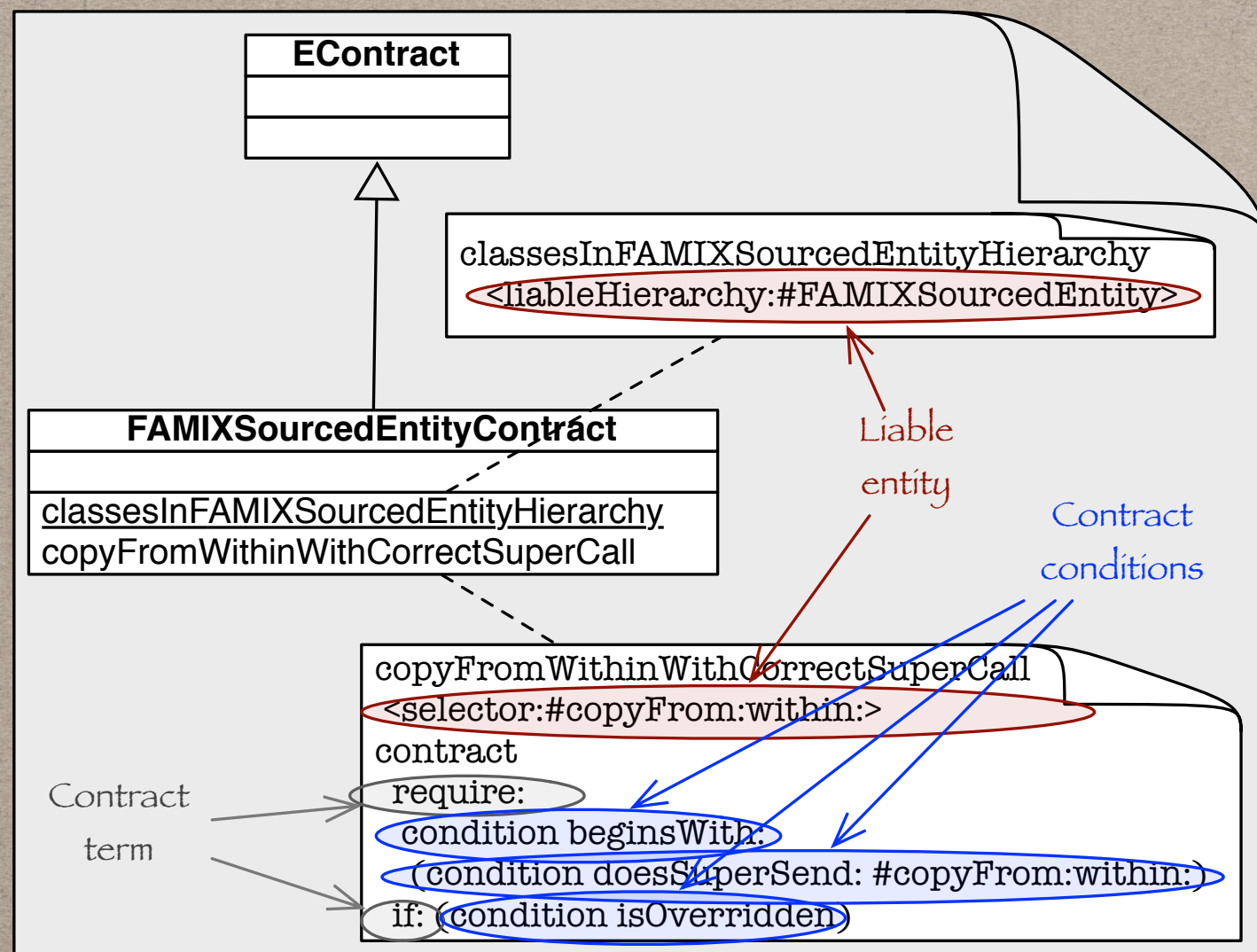
- **selector**: regExp / **exceptSelector**: regExp
- **protocol**: regExp / **exceptProtocol**: regExp
- / **exceptClass**: className **selector**: selector

Contract terms

- **require**: condition
- **suggest**: condition
- **require**: condition **if**: anotherCondition
- **suggest**: condition **if**: anotherCondition

Contract conditions

- **assigns**: regExp
- **calls**: regExp
- **references**: regExp
- **returns**: expression
- **doesSuperSend**: regExp
- **doesSelfSend**: regExp
- **inProtocol**: regExp
- **isOverridden**: selector
- **isOverridden**
- **isImplemented**: selector
- **custom**: visitor
- **and**: cond1 with: cond2
- **or**: cond1 with: cond2
- **not**: cond
- **beginsWith**: cond
- **endsWith**: cond
- **does**: cond1 **after**: cond2
- **does**: cond1 **before**: cond2



UContracts : THE TOOL

ECConcreteElementIncorrect3

eContracts-Tests-Classes
eContracts-Tests-Contracts
eContracts-Examples-ClassicalSche
eContracts-Examples-Visitor
eContracts-Examples-State
eContracts-Examples-Composite
eContracts-Examples-Template
eContracts-Examples-Command

ECAbstractElement
ECConcreteElementCorrect
! ECConcreteElementIncorrect
! ECConcreteElementIncorrect2
! ECConcreteElementIncorrect3
ECAbstractElement

-- all --
visiting

! accept:

File out cmd+o
New class template
Subclass template
Copy...
Remove class... cmd+x
Rename...
Browse cmd+b
Browse hierarchy cmd+h
Browse package cmd+p
Browse references shift+cmd+n
Chase variables
Contracts
Jump to Test cmd+j
Refactor
Refactor class
Refactor class variable
Refactor instance variable

Instance

Browse Hierarchy Variables

accept: aVisitor
^aVisitor wrongMessage: self

Inheritance Senders Versions View

Contracts

! ECAbstractElementContract

- ! ECConcreteElementIncorrect3 must override #accept:
- ! ECConcreteElementIncorrect3>>yourself must implement a double dispatch
- ! ECConcreteElementIncorrect3>>yourself must call do*
- ! ECConcreteElementIncorrect3>>yourself must be in protocol visiting

Code Critics: All checks on eContracts-Examples-Visitor (24 problems)

Contracts (3)
Lint checks (21)
Transformations

ECAbstractElementContract (3)
ECommandContract

must be in protocol visiting (1)
must call method named do* (1)
must implement a double dispatch

Browse Refresh

must be in protocol visiting

ECConcreteElementIncorrect2>>#accept: