Evolution

Jan Vraný

Department of Computer Science and Engineering
Czech Technical University In Prague
Faculty of Electrical Engineering

November 18, 2008

Outline

Problems

Software evolves

How to deal with new versions?

How to deal with new library versions?

How to change the code?

Outline

Version Control Systems

File-based

- CVS
- SVN
- GIT/Mercurial/Bazaar

Version Control Systems

Metamodel-based

StORE (Visual Works)

Monticello 2 (Squeak)

ENVY (IBM VAST/VA Smalltalk)

Outline

Problems

Is it OK to subclass existing class?

Is it OK modify a foreign method?

Is it OK add a method to foreign class?

Classboxes

Classboxes is a module system supporting local class refinements.

Classboxes – Example

```
classbox greetings-cb {
  class hello {
    method say-hello { print ``Hello!''  }
    method say-good-bye { print ``Good bye!''  }
  }
}

classbox spanish-greetings-cb {
  import class hello from classbox greetings-cb
  refine class hello {
    refine method say-hello { print ``Hola!''  }
  }
}
```

Classboxes – Example (cont.)

```
classbox application-cb {
  import class hello from classbox greetings-cb {
  class app {
    method main { hello.say-hello; hello.say-good-bye
  }

classbox spanish-application-cb {
  import class app from classbox application-cb
  import class hello classbox spanish-greetings-cb
}
```

Outline

Hints

Many many classes

Many many methods

Be carefull with private/protected methods

Be carefull with final/sealed classes

One responsibility per class

Write libraries as set of traits, if possible :-)

Outline

Refactoring

Changing the code without changing the functionality.

Simple examples

- Rename a parameter or temporary
- Rename the method
- Extract to method

Complex refactoring

at:ifAbsent → at:ifAbsentPut:

```
propertyAt: key

  ↑dict
    at: key
    ifAbsent:
      [dict
        at: key
        put: nil]
```

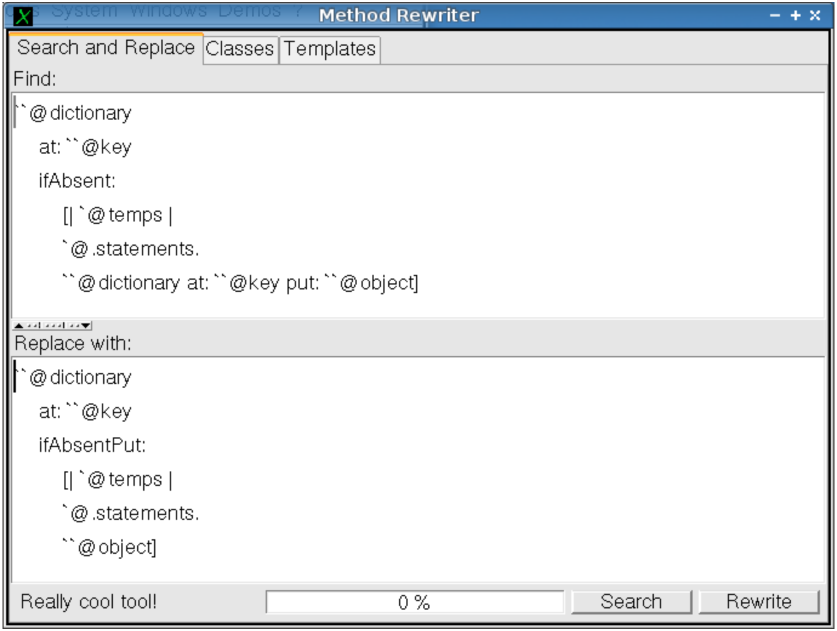
Complex refactoring II

at:ifAbsent → at:ifAbsentPut:

```
propertyAt: key

  ↑dict
    at: (self normalizeKey: key)
    ifAbsent:
      [|default|
      default←self defaultValueForProperty: key.
      dict
        at: (self normalizeKey: key)
        put: default]
```

Complex Refactoring – Method Rewriter



Find:

```
``@dictionary
   at: ``@key
   ifAbsent:
      [| `@temps |
      `@.statements.
      ``@dictionary at: ``@key put: ``@object]
```

Replace with:

```
``@dictionary
   at: ``@key
   ifAbsentPut:
      [| `@temps |
      `@.statements.
      ``@object]
```

Really cool tool!

Changing the code