selfish – a new message send for Smalltalk

John Aspinall – JPMorgan Kapital Project

selfish – a new message send for Smalltalk

- Kapital
  - JPMorgan risk-management system
  - Since 1994 (20th Birthday this month!)
  - VisualWorks Smalltalk + GemStone

- ENVY source code management
  - Same as VA Smalltalk
  - Used under licence
  - Maintained and enhanced in-house

selfish – a new message send for Smalltalk

- **Method Overrides**
  - **Not the same as subclass method overrides!**
  - **VisualWorks concept**
  - **Replaces current version of a method with a different implementation from another Package**
  - **Increasingly used in non-core/add-on Packages**

selfish – a new message send for Smalltalk

- Method Overrides for ENVY
  - Create _*override_method* in class extension Application (Package)
  - Replaces implementation of *method* in class's method Dictionary
  - Original implementation stored at: *#_overridden_method*
  - Overridden implementation can still be invoked: *self _overridden_method*

selfish – a new message send for Smalltalk

- **Example – Adding a guard clause:**

```
_override_method

    self shouldUseOriginalImplementationOfMethod
        ifTrue: [self _overridden_method]
        ifFalse: [self doSomethingElse]
```

selfish – a new message send for Smalltalk

- ClassA

```
doThing
    self doThingA
```

- ClassA subclass: ClassB

```
doThing
    self doThingB.
    super doThing
```

selfish – a new message send for Smalltalk

- ClassA

```
    _override_doThing
        self doAnotherThing.
        self _overridden_doThing
```

- ClassA subclass: ClassB

```
    doThing
        self doThingB.
        super doThing
```

selfish – a new message send for Smalltalk

- ClassA

```
_override_doThing
    self doAnotherThing.
    self _overridden_doThing
```

- ClassA subclass: ClassB

```
_override_doThing
    self doYetAnotherThing.
    self _overridden_doThing
```

selfish – a new message send for Smalltalk

- ClassB new doThing

```
_override_doThing [ClassB]
    » self doYetAnotherThing
    » self _overridden_doThing
        » self doThingB
        » super doThing

_override_doThing [ClassA]
    » self doAnotherThing
    » self _overridden_doThing
```

selfish – a new message send for Smalltalk

- ClassB new doThing

```
_override_doThing [ClassB]
        » self doYetAnotherThing
        » self _overridden_doThing
              » self doThingB
              » super doThing

_override_doThing [ClassA]
        » self doAnotherThing
        » self _overridden_doThing
```

...recursive loop!

selfish – a new message send for Smalltalk

- What do we actually want to do?
  - Invoke the implementation of *_overridden_doThing* in ClassA
  - i.e. the implementation in the class of the *_override_* method being executed
  - Not (always) the same class as the receiver object!

selfish – a new message send for Smalltalk

- The Solution – “selfish”
  - It’s “a bit like self”
  - It’s a “selfish” send

- The Implementation
  - A new bytecode? Difficult…
  - An existing bytecode? Easier.

selfish – a new message send for Smalltalk

- **The (meta) Solution – “super”**
  - **It’s “a bit like selfish”**
  - **Invokes the implementation in the superclass of the current implementation:**

    <44> push self

    <1C> push {current method’s implementing class}

    <F2 01> super send *method*

  - **Just need to push a subclass instead:**

    <1C> push (Class new superclass: {current method’s implementing class})

selfish – a new message send for Smalltalk

- ClassB new doThing

    - _override_doThing [ClassB]
        - » self doYetAnotherThing
        - » selfish _overridden_doThing
            - » self doThingB
            - » super doThing

    - _override_doThing [ClassA]
        - » self doAnotherThing
        - » selfish _overridden_doThing
            - » self doThingA

...success!