Safe Metaclass Composition Using Mixin-Based Inheritance

Noury Bouraqadi
Computer Science Laboratory (CSL)
Ecole des Mines de Douai
France

Metaclasses

- Classes which instances are also classes
  - Allow defining class properties
  - Abstract, Singleton,
  - Multiple Inheritance, Final,
  - Lazy memory allocation, Persistent instances,…

- Implicit in Smalltalk
  - Solution for safe usage…
  - But, no reuse/composition

Need for Mixin-Based Inheritance

- Context
  - Single inheritance
  - Unrelated hierarchies
  - Same Properties

- Goal
  - Reuse shared properties
  - Avoid code duplication
  - Alternative to multiple inheritance

Outline

- Mixin Based Inheritance
- Metaclass Composition Using Mixins
- Conclusion

Outline

- Mixin Based Inheritance
- Metaclass Composition Using Mixins
- Conclusion

Single Inheritance Behind the scene

Mixin = Subclass Generator
[Bracha & Cook 90]
Example of Inheritance from Different Mixins

ColoredBoundedPoint inherits from Colored and Bounded

Explicit Linearization on Definition:
- ColoredBoundedPoint mixins: (Colored. Bounded)

Lookup list:
- ColoredBoundedPoint, Colored, Bounded, Point

Outline

- Mixin Based Inheritance
- Metaclass Composition Using Mixins
- Conclusion

Main Idea and Issues

- Singleton
- LazyMemoryAllocation

But, we need also:
- Compatibility (inheritance + inter-level messages)
- Class specific properties

The Upward Compatibility issue

foo is NOT understood by B

The Downward Compatibility issue

bar is NOT understood by aZ

No Class Specific Properties in Smalltalk

Unwanted property propagation: B becomes abstract!
Compatibility & Class Specific Properties

Class level

Metaclass level

Compatibility

foo

b

self new bar

AProperties

new

self error: 'Abstract class'

BProperties

subclass of

AbstractClass

B

foo

self class foo

bar

Class level

Metaclass level

Compatibility

foo

b

AProperties

new

self new bar

BProperties

instance of

subclass of

1.

Compatibility, Class Properties & Mixins

Class level

Metaclass level

AbstractClass

AProperties

new

Misc

BProperties

subclass of

B

A

1.

Outline

- Mixin Based Inheritance
- Metaclass Composition Using Mixins
- Conclusion

Summary

- Metaclasses are useful
  - Class properties = new "kinds" of classes
  - e.g. Mixin-based inheritance (3 class properties)
- Metaclass Composition using Mixins
  - Class-Metaclass compatibility
  - No undesirable class properties propagation
- Implementation on top of Squeak
  - Mixin-Based Inheritance = 3 metaclasses
  - No Performance Loss!

Some Future Works

- Extend mixin-based inheritance
  - Traits approach for methods composition
  - Instance variables composition
- OO Programming without "traditional" inheritance
  - Mixin-based inheritance only!
- Refactoring Squeak/Smalltalk libraries
  - Explicit metaclasses + Mixins
  - New kernel (bootstrap)

Thanks for your attention
Questions? Comments?

http://csl.ensm-douai.fr/MetaclassTalk