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

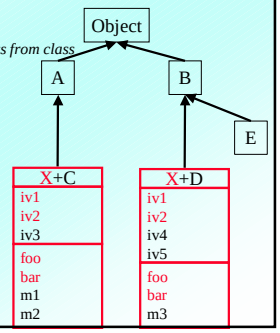
  [Cointe87/90, Briot89, Danforth94-96, Ledoux96]:
  - Abstract, Singleton,
  - Multiple Inheritance, Final,
  - Lazy memory allocation, Persistent instances,…
- **Implicit in Smalltalk**
  - Solution for safe usage…
  - But, no reuse/composition

Outline

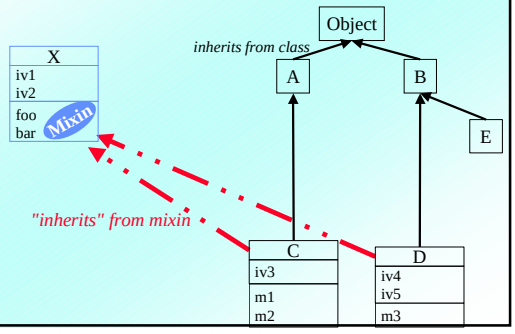
- **Mixin Based Inheritance**
- **Metaclass Composition Using Mixins**
- **Conclusion**

Need for Mixin-Based Inheritance

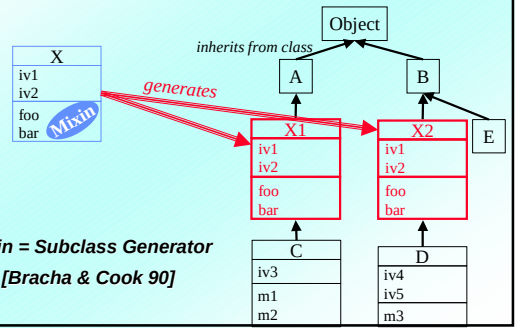
- **Context**
  - Single inheritance
  - Unrelated hierarchies
  - Same Properties
- **Goal**
  - Reuse shared properties
  - Avoid code duplication
  - Alternative to multiple inheritance



Mixin-Based Inheritance

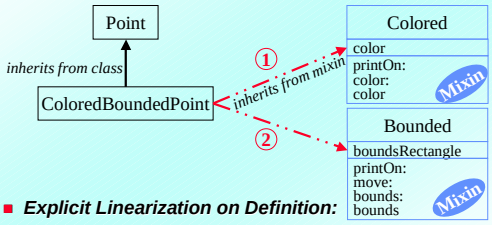


Single Inheritance Behind the scene



Mixin = Subclass Generator
[Bracha & Cook 90]

Example of Inheritance from Different Mixins

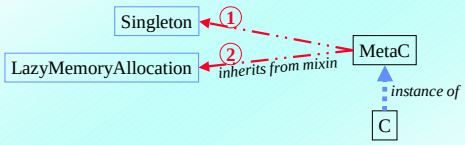


- ***Explicit Linearization on Definition:***
  - ColoredBoundedPoint mixins: {Colored. Bounded}
- ***Lookup list***
  - ColoredBoundedPoint, Colored, Bounded, Point

Outline

- ***Mixin Based Inheritance***
- ***Metaclass Composition Using Mixins***
- ***Conclusion***

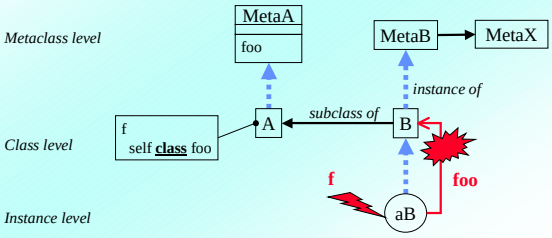
Main Idea and Issues



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

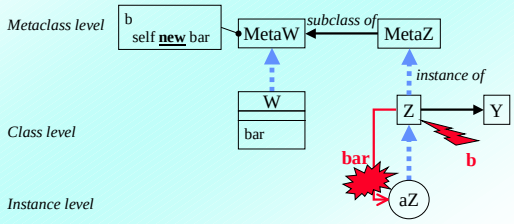
The Upward Compatibility issue

foo is NOT understood by B

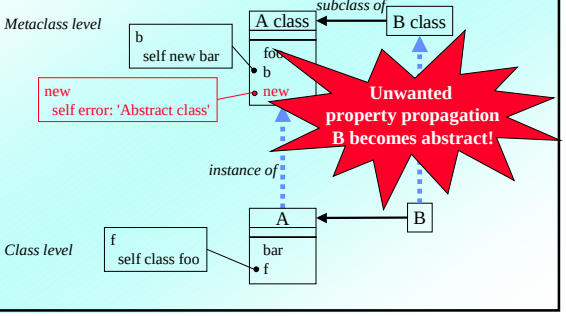


The Downward Compatibility issue

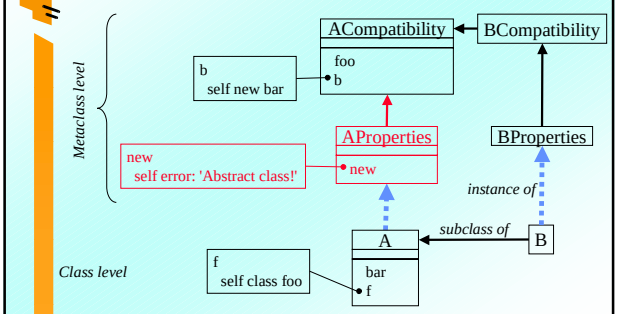
bar is NOT understood by aZ



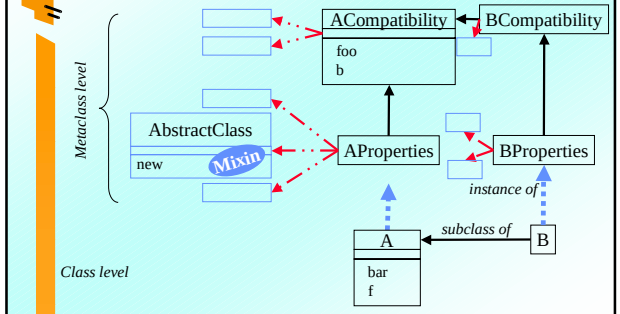
No Class Specific Properties in Smalltalk



Compatibility & Class Specific Properties



Compatiblity, Class Properties & Mixins



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?