UML 2.0:
Why Should I Care?
And What Should I Do?
UML 2.0: Why Should I Care?

- Reason #1: structure modelling
- Better architectural communication
  - A nice-to-have, but not a fundamental change...
- MDA enabler!
  - Generate better applications faster
Structures Enable Model-Driven Architecture

Full benefits of model-driven development require structural code generation from the model

• Faster development
• Higher quality
Automating Structure Generation Enables Control

• It’s not what you draw, it’s what you do with it
• Code Generation
  – Shorter time-to-market, better quality
• Model-Driven Testing
  – Full control of stubbing
• Thread Assignment
  – Inheritance of controlling threads
• Build Minimization
  – Full two-way encapsulation
• And more…
UML 2.0: Where Are We?

- We’re not there yet
- Official UML 2.0 adoption ~October 2003
- Official UML 2.0 availability ~October 2004
  - I.e., no more changes
UML 2.0: What Should I Do?

• Learn UML 2.0
  – Forewarned is forearmed
  – But be careful of what you’re taught today…

• Be wary of “UML 2.0 today!” claims

• Understand how to apply UML 2.0, not just draw it

• Get ready for the future of software development
Dirk Epperson
Kabira
www.kabira.com
Levels of Abstraction
Before UML modelers spent as much time arguing whether arrows should be round or pointy, as they did about the viability of modeled systems.

Current programming practice is tied too much to lines of code.

Model Driven Architecture (MDA) gives us the level of abstraction required to get above the fray of the lines of code!

UML is virtually necessary for MDA (see the first point) and is nearly sufficient for MDA.
The Good News
- More ways to express a concept
- Action Semantics
  - Foundation for MDA

The Bad News
- Expression comes at the cost of Complexity
  - How many developers can create a fully semantically correct, verifiable, executable model?
- More than one way to do the same thing
  - Java learned a lesson from C++ about this.
Shakespeare and Dickens were able to use the full expression of the English Language.

But newspaper reporters use a vocabulary of only about 1,000 words to get the story across.

We are in danger of ending up with a UML that only a Shakespeare can use!
- This does not bode well for the longevity of UML.
What is needed...

- A profile which describes UML for mere mortals.
- An executable context which provides a normal mode of MDA.
- An audit context which will guarantee the executability of a model.
Why Should I Care?

- Obscurity, complexity, and over-ambition
- Consistency, capability, and quality
- Minor quirks and major advances

Lovelace
“I gave desperate warnings against the obscurity, the complexity, and the over-ambition of the new design but my warnings went unheeded. I conclude that there are two ways of constructing a design:

One way is to make it so simple that there are obviously no deficiencies and the other way is to make it so complicated that there are no obvious deficiencies.”

_Tony Hoare_
_Turing Award Lecture, 1980_
Consistency: great care has been taken, to good effect

Capability: significant gains, building on other established standards

Quality: much improved
No objects in UML 2 models [but instanceSpecifications will fill the void]

**InstanceSpecification**

+classifier

0..*

**Classifier**

**Instance**

Minor quirks

Lovelace
Type distinct from specification [but only one type per element]

Associations may be typed

Major advancements
Equilateral

- « instance »
  - x:A2S1Triangle
    - angleA = 60°
    - angleB = 60°
    - sideC = 4
    - draw ()

- « instance »
  - y:S3Triangle
    - sideA = 4
    - sideB = 4
    - sideC = 4
    - draw ()

Same type, different implementation
Different type, same implementation
And What Should I Do?

As you use UML 2, think about UML 3

CommunityUML

www.cuml.org

Lovelace
This is a workshop.

Let’s work!