



Comments on LCO ARB's CS577a Fall 2001

Ed Colbert

USC Center for Software
Engineering



Outline

- Why UML?
- Common Problems
- Design Workshop



Student Asked “Why Do We Need UML? Its Only Used In Defense Industry.”

- So can model system before construction
 - Acts like blueprint for a large building
 - Supports communication among project team & customer(s)
 - Supports verification & validation of construction
- Must have:
 - Model elements (concepts with semantics)
 - Notation for visualizing model elements
 - Guides for usage of model elements



“Why UML?” (cont.)

- Pre-UML “Tower of Babel”
 - Each OO method had own
 - Definition of Concepts
 - Incomplete
 - Conflicting definitions
 - Notation
 - Some commonality
 - Many not well integrated
 - 1 OO Tool supported about 50 different notations
- UML has substantially improved state



“Why UML?” (cont.)

Is It Only Used By Defense Industry?

- UML is standard of Object Management Group
 - Industry consortium
 - Approximately 300 member companies
 - Industries such as healthcare, manufacturing, telecommunications
- Many more non-member companies use OMG standards including UML, CORBA



Outline

- Why UML?
- Common Problems
- Design Workshop



Common Problems Seen At ARB

- Not Using MBASE 2.3.3 Guide for
 - Notation
 - SSAD document structure
- Many of you didn't
 - Listen to Dan when he said don't use previous years notation as a guide
 - Follow Full-Text Title Database example I presented as a guide
 - Come ask questions
 - Document your variations in your Exit criteria



Common Problems Seen At ARB

- Confusing *Components* with *Entity Classes* in OCD or *Object Classes* in SSAD
 - Problem isn't fully your fault
 - MBASE guide isn't always clear
 - Conflicting definitions (“models”) of *Components* : MBASE, UML, RUP, ...
 - But, you could have
 - Follow Full-Text Title Database example I presented as a guide
 - Come ask questions
 - Mentoring is a time-honored way of teaching/learning



Common Problems Seen At ARB (cont.)

- Not identifying any “Entities”
 - Most teams just repeated *Components* and *Actors*
 - Maybe added a few attributes
 - Maybe added a few operations (typically wrong)
 - Most teams didn’t identify
 - Classes of things which are [known,] inspected, manipulated, or produced



Common Problems Seen At ARB (cont.)

- Missing stereotypes for *Actors*, *Component*, & *System* classifiers
- Not recognizing that each use-case is a behavior
- Representing *Components* as
 - Use-case on use-cases diagrams
 - Hardware (“nodes”) on deployment diagrams
- Missing stereotypes for *Actors*, *Component*, & *System* classifiers
- Did design work that wasn’t warranted by risk



Outline

- Why UML?
- Common Problems
- Design Workshop