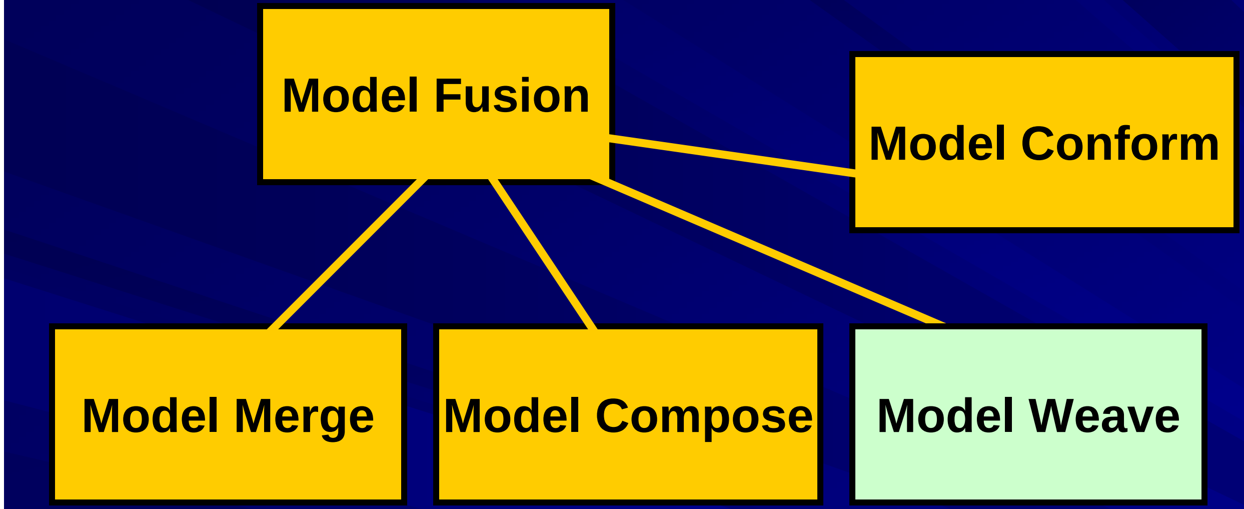
MiSE 2007 Panel on: Model Manipulation and Management Challenges

Jon Whittle
Dept of Computer Science
George Mason University, Fairfax, VA, USA



Model Fusion

Model Conform

Model Merge

Model Compose

Model Weave

- Aspect-Oriented Modeling:
  - *model* cross-cutting concerns
  - *relate* cross-cutting models to “base” models
  - *may or may not actually compose*
  - Special case of model fusion (or is it?)

Composition in AOM

- Requirements for composition
  - **Expressive**
    - Captures all practical compositions
  - **Scalable**
    - Models are large
  - **Intuitive**
    - Graphical, easy to learn and matches existing processes/languages
  - **Formal**
    - For execution and analysis (e.g., aspect interactions)
- For AOM, broadly, 2 classes of approaches so far:
  - **AspectJ-like (explicit)**
    - Define joinpoint model for the modeling language
    - Compose only at these joinpoints
  - **HyperJ-like (implicit)**
    - Default merge algorithm (match names, match states)
    - Override if necessary

Are these enough?

Composition Category	Occurrence Percentage	Implicit (HyperJ-like)	Explicit (AspectJ-like)
1-1 match	13	+	-
many-to-many match	39	?	-
refactor	46	-	?
refine	2	-	?

Consequences

- The bad news:
  - We need more expressive composition languages that allow complex compositions to be specified in a scalable, intuitive way
- The good news:
  - Model weaving is just model transformation
  - May want specific things for weaving (concrete syntax, patterns), but a lot of model transformation ideas can be reused

Some thoughts

- Expressiveness of composition language versus need for refactoring
  - Refactoring eases composition
  - but does not promote reuse
- (obvious?) weaving/merging just a model transformation
  - Do we need special composition languages?
- Composition at different abstraction levels
- Semantic-based composition