

# Scalability

# Ask why it does not scale!

- Is there anything about scalability problems in MDE that cannot be addressed by any other existing solution?
- Is there anything about MDE that does not scale at all?
  - For collaboration among people
  - For algorithms
  - For computer resources (memory)
  - Technical limitations
    - XMI, EMF etc
    - Support for modularity, access, distribution

# Ask why we are trying to scale!

- What is the purpose of scalability?
- What kind of scalability are we trying to achieve?
  - (eg. solutions for quick search in Java (specific indices) are different to those for databases (generic))
  - Optimisation for transformation, weaving, merging, code generation ...

# Extra Slides

# Requirements?

- What size of models and metamodels
  - Are needed
  - Can be handled
- Expressiveness
  - UML has subsetting, inverses etc which cause extra problems
- Managing large numbers of properties
  - Constraint sets
- What purpose for the (scaled up) models
  - (eg. solutions for quick search in Java (specific indices) are different to those for databases (generic))
  - Optimisation for transformation, weaving, merging, code generation ...

# The problem of scalability

- Very large models
  - Time to load model much greater than time to check, transform etc. the models
- Can techniques from other areas help
  - Programming: caches, indices etc
    - Modular engineering principles
    - Incremental processing
  - Databases: EMF Teneo
  - Model checking, Logic inference engines
  - High performance computing: optimisations
- Understanding the scalability limitations of each characteristic or component and writing guidelines that respect these

# Linkage problem

- Conceptually, scalability is not an issue
  - Programs handle flexible structures ...
- Physically, the way models are stored causes a problem
  - Concrete linkage
  - Fragility – ordering is critical
  - Dependability – all model needed in memory
  - Ways to handle scoping
    - UML is not helpful here
    - Naïve matching algorithms are problematic
- Need ways to support modularity
- MDE's uniform conceptual core does not help scalability
  - Needs to support all possible model management activities
  - But tools should be able to sort optimisation?

# Key issues

- There are known solutions to all the problems we thought of
- We do not know how to generalise
  - eg. from programming to modelling support
- We do not design abstract languages (MOF) that respect known scaling problems in concrete languages
  - Inverses, subsetting etc.
- Confusing bad practice with bad principles
  - You can build bad databases and OO programs, too