



Scalability: The Holy Grail of MDE

Dimitrios Kolovos, Richard Paige, Fiona Polack

{dkolovos,paige,fiona}@cs.york.ac.uk

Department of Computer Science

The University of York



“We are interested in adopting MDE...”

Typical person from the industry



BUT

We have huge models!

- Order of 100s MB and many thousands elements
- Do current MDE tools support such models?
- If I do a small change in the model
 - Should I re-validate my entire model?
 - Should I re-generate all my code?
 - Should I re-transform my entire model?
- What about collaborative development?



What do **we** answer?

Optimistic answers

- Models can now be stored in databases!
- Tool/Language X supports incremental code generation/transformation/validation!
 - ... but it takes a PhD to use it and only applies to (dead) simple cases
- Tools such as EMF Compare can compare/merge changes in your model!

Pragmatic answers

- Huge models are really slow to load/save
- In the general case you'll need to revalidate/generate/transform them all over
- You can store your models in CVS/SVN but
 - ... committing/updating will take forever
 - ... tools cannot compare models that big



What should **we** be answering?



Software Engineering Principles

- High Cohesion
- Low Coupling
- Encapsulation
- Modularity
- ...



Having huge monolithic models
violates **all of them!**



Scalability = Modularity

Challenges (not)

- To implement another “incremental” model management language
- To further argue about the virtues of declarativeness
- To start storing models in data farms
 - or indexing them in Google

Challenges

- Learn how to design modeling languages that enforce modular models
 - ... and implement supporting tools
- Reduce coupling
 - Implement mechanisms that support lazy loading/resolution
 - Get rid of hard ID-based references
- Learn from programming languages and tools!
 - They've dealt with similar problems a long time ago

Questions and Feedback

