



Rascal-Ecore

Tijs van der Storm
storm@cwi.nl / [@tvdstorm](https://twitter.com/tvdstorm)

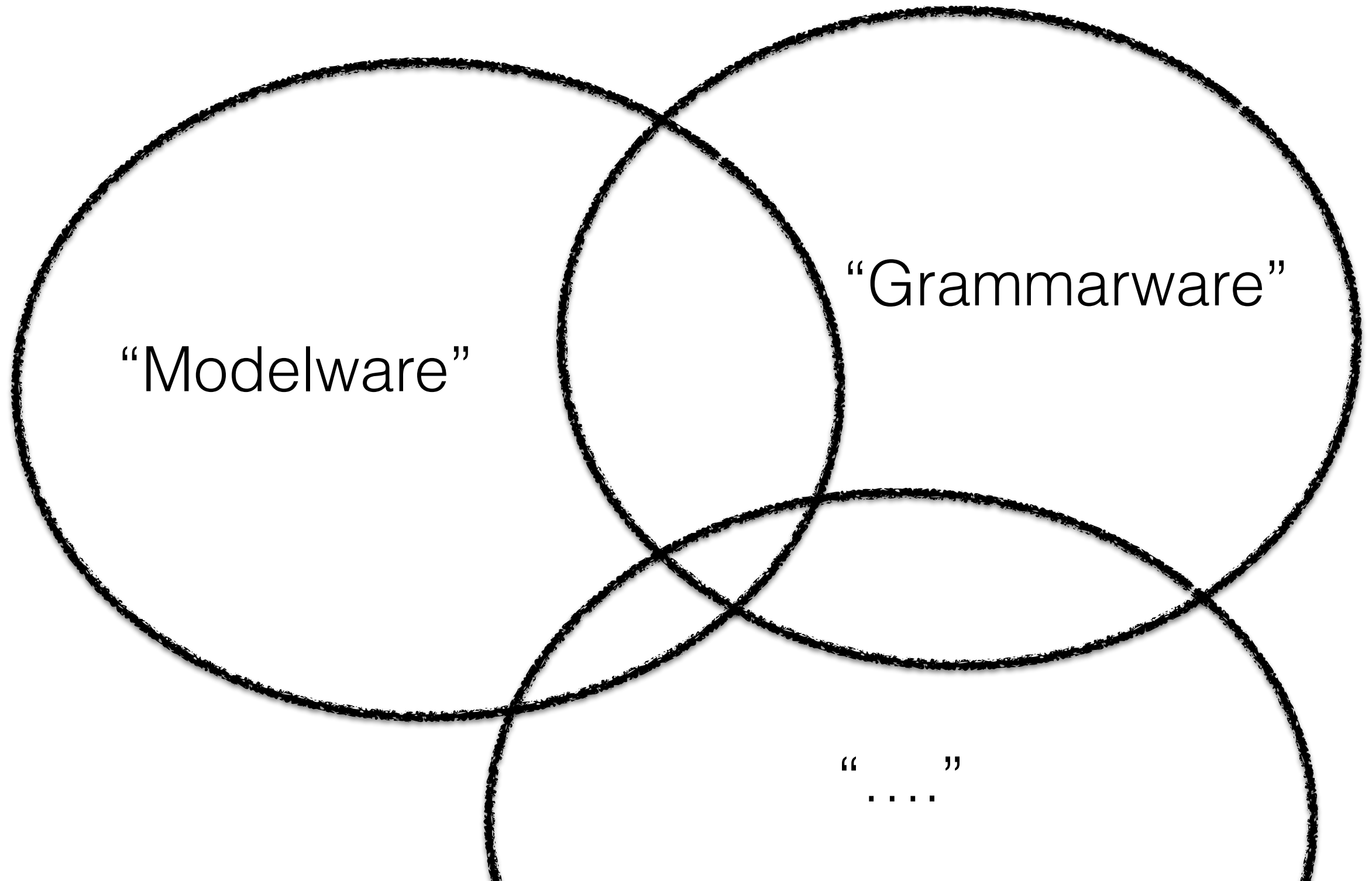


Centrum Wiskunde & Informatica

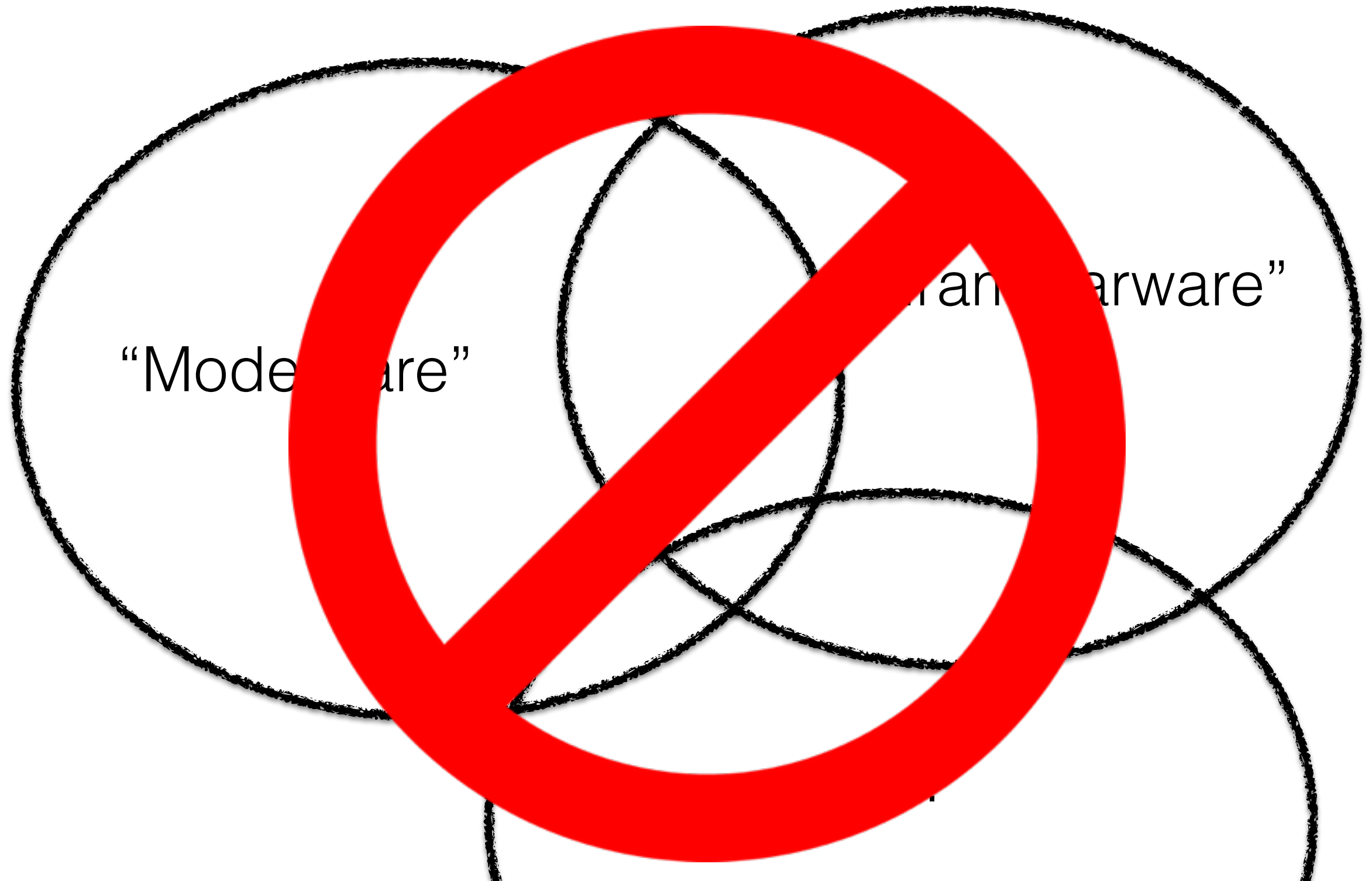


university of
 groningen

Software Language Engineering



Software Language Engineering



- Lots to reuse on both sides of the “divide”...
- *Modelware*: diagram editors, animators, persistence frameworks, etc.
- *Grammarware*: parsers, editor services, transformation languages, etc.
- Can we map and/or connect?
- Not a new idea!

Concretely

- Ecore *Meta models* as Rascal ***algebraic data types***
- Ecore *Models* as Rascal ***values***
- Why?
 - Make Rascal available to MDE scenarios (e.g., model trafo)
 - Reuse MDE tools in the context of Rascal (e.g., diagram editors, animators, etc.)

rascal-ecore

- import/export Ecore meta models and models
- generate ADTs from meta models
- map Rascal defined concrete syntax to models
- infer meta models from annotated Rascal grammars
- generate “HUTN” grammars from meta models
- diff and patch operations on models
- patch of concrete syntax source code

Live coding

rascal-ecore

- tools for processing EMF models in Rascal
- lightweight mapping of concrete syntax and models
- diff/patch supports online synch of *actual* models