Modeling in a Team

on problems not yet (completely) solved

carsten.pitz@gmx.de

Stuttgart

Automotive

- Bosch / ETAS
- Daimler Trucks
- Mercedes-Benz
- Porsche
- Vector Informatik
- ...

Automotive

- 60+ million requirements
- ...
- 100+ ECUs
- 150.000+ signals (message types)
- 100+ million lines of C/C++ code
- …
- 20.000+ engineers involved
- many companies involved

Modeling in a team – shared model

shared model (CDO)

- really handy for small teams

but

- 100+ concurrent modelers
  - scaleability issues arise
- 10+ companies evolved
  - Who is allowed to see what?

Current State in Automotive

- overall AUTOSAR model for E/E-Architecture
  - PreeVision by Vector Informatik
- individual UML/SysML models for each ECU
  - Sparx EA, IBM Rational Rhapsody, NoMagic Cameo, ...

!!! no overall traceability !!!

Modeling in a team – how?

A car is a system of systems

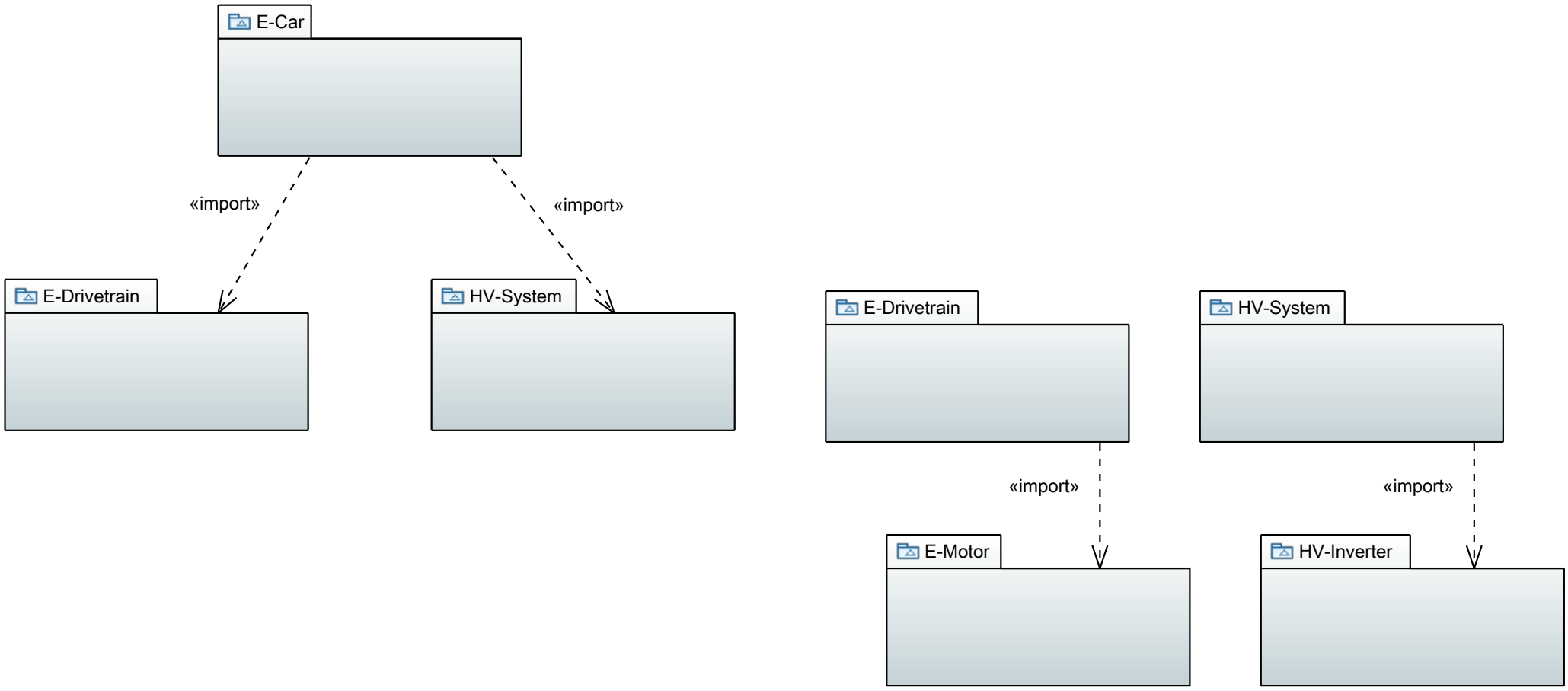
can be modeled as a model of models

My current experiment

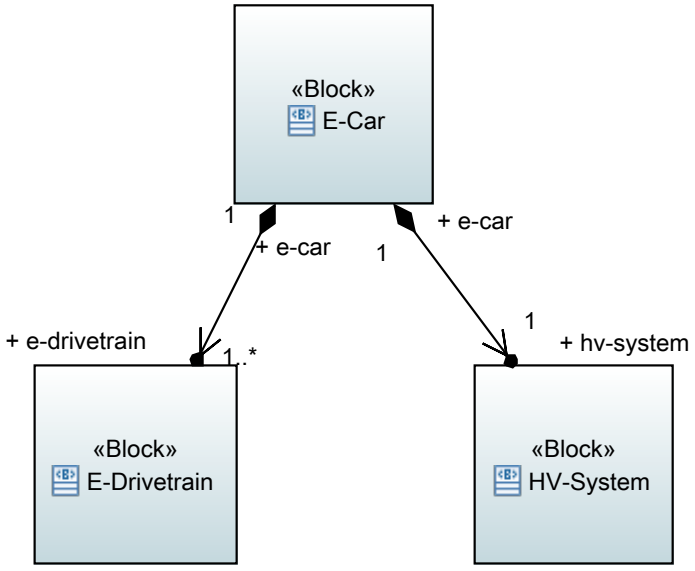
- each model is a separate Papyrus project
  - can be created seperately
  - can be reviewed seperately
  - can be released seperately

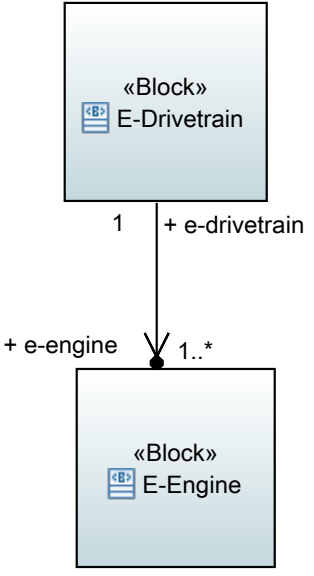
- models connected via UML:PackageImport

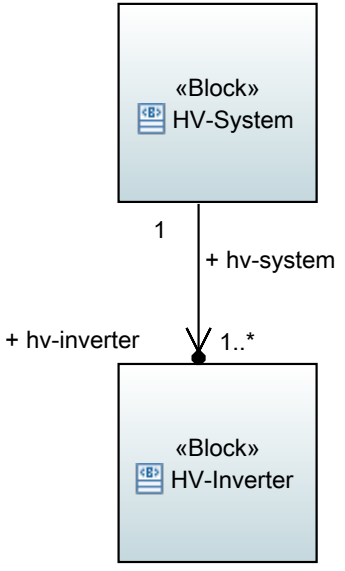
Down the tree - import



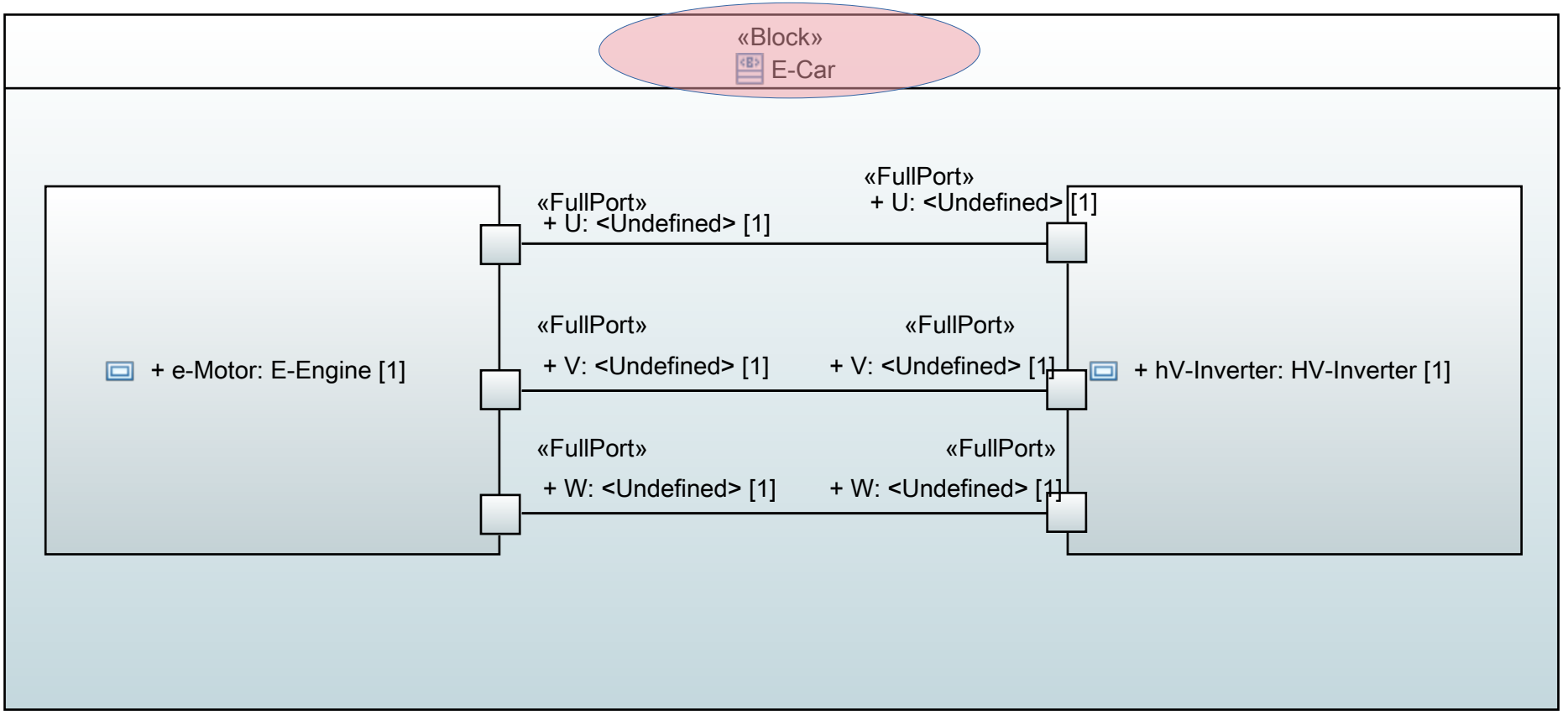
Down the tree - breakdown







Interaction between Components



Interaction between Components

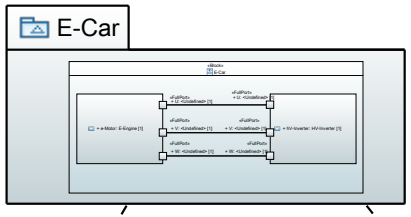


Diagram is located in model E-Car

«import»

«import»

E-Drivetrain

HV-System

E-Drivetrain

HV-System

«import»

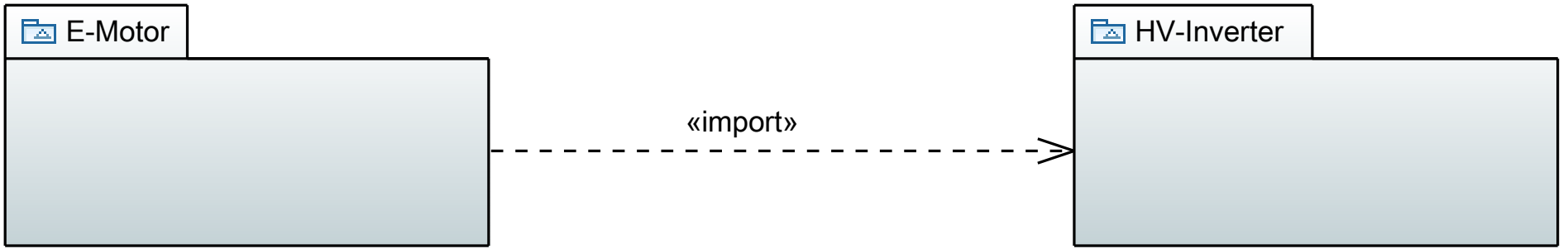
«import»

E-Motor

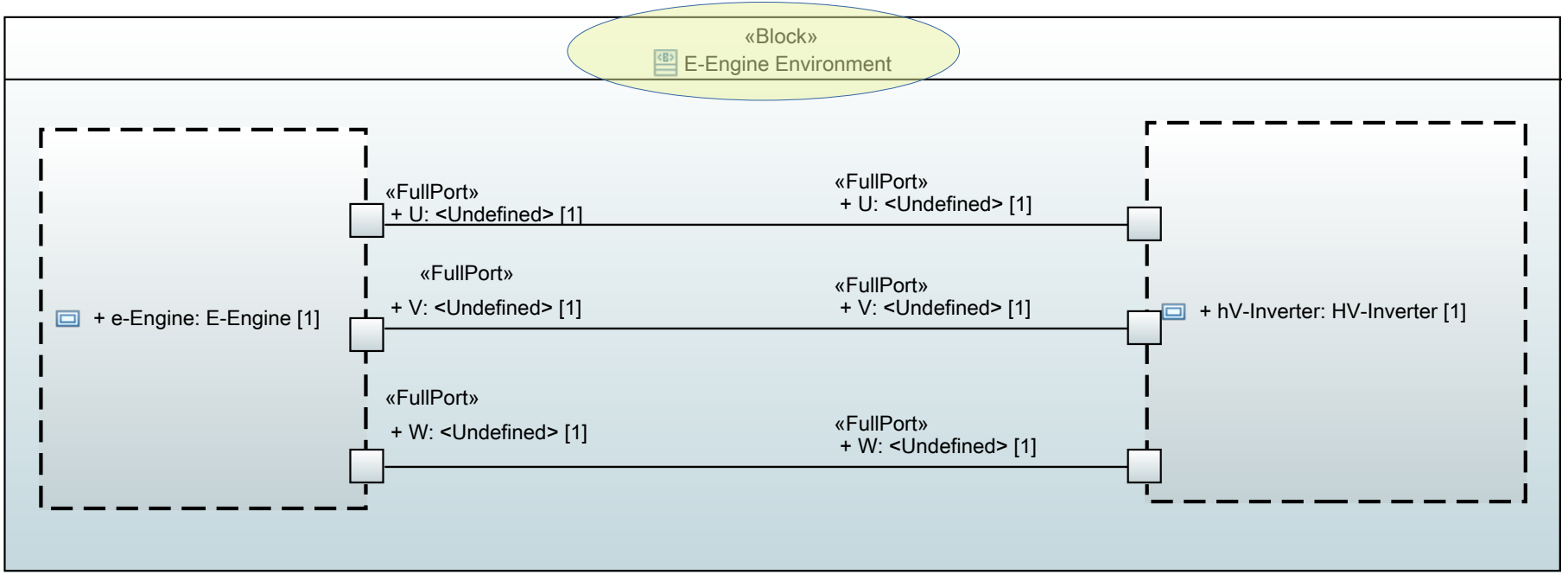
HV-Inverter

Diagram is not visible in model E-Motor

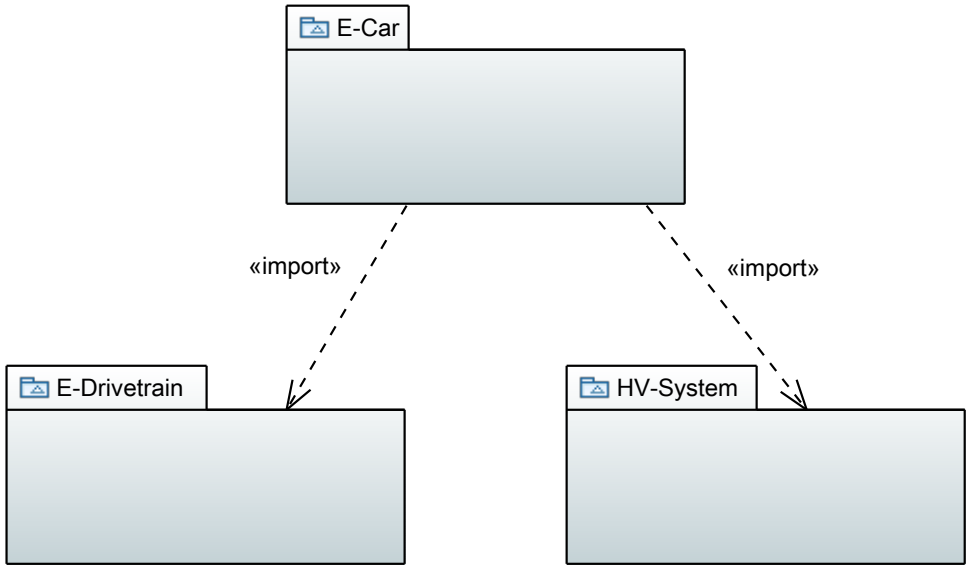
Interaction between Components



Interaction between Components



Interaction between Components



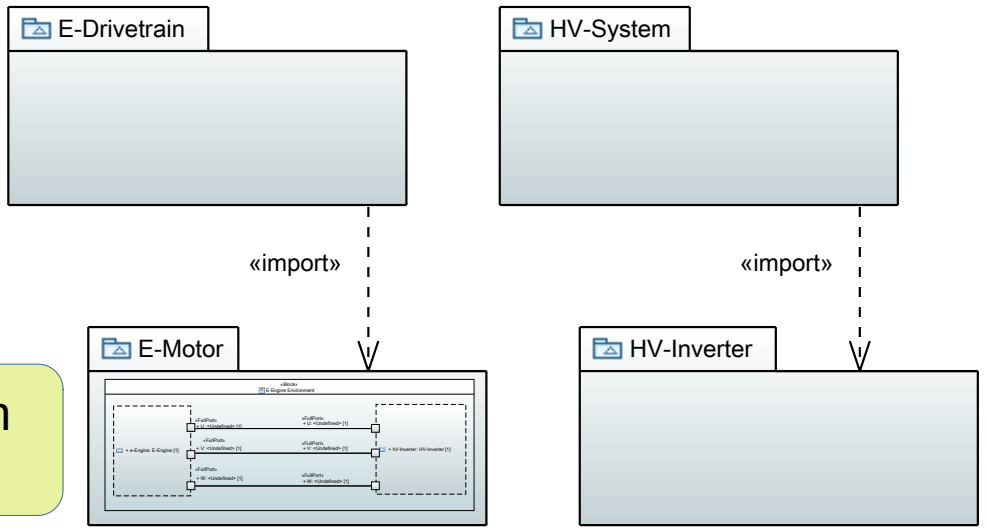


Diagram is visible in model E-Motor

Conclusions (so far)

- Papyrus supports methodology really well
  - even details like the ownership of association ends are handled correctly
- «import»-structure backed by Eclipse e4 project dependencies
  - EGit does all the git magic by itself
  - imported models should be strictly read only, though
- Papyrus handles cyclic imports

Modeling in a Team

many thanks

carsten.pitz@gmx.de