

The eTrice Eclipse Project Proposal

Dipl.-Ing. Thomas Schütz, Protos Software GmbH

Eclipse Embedded Day 2010, Stuttgart

Agenda

- Motivation
- Scope of eTrice
- ROOM Language
- Codegenerators
- Middleware
- Realization
- Project Plan
- Conclusion

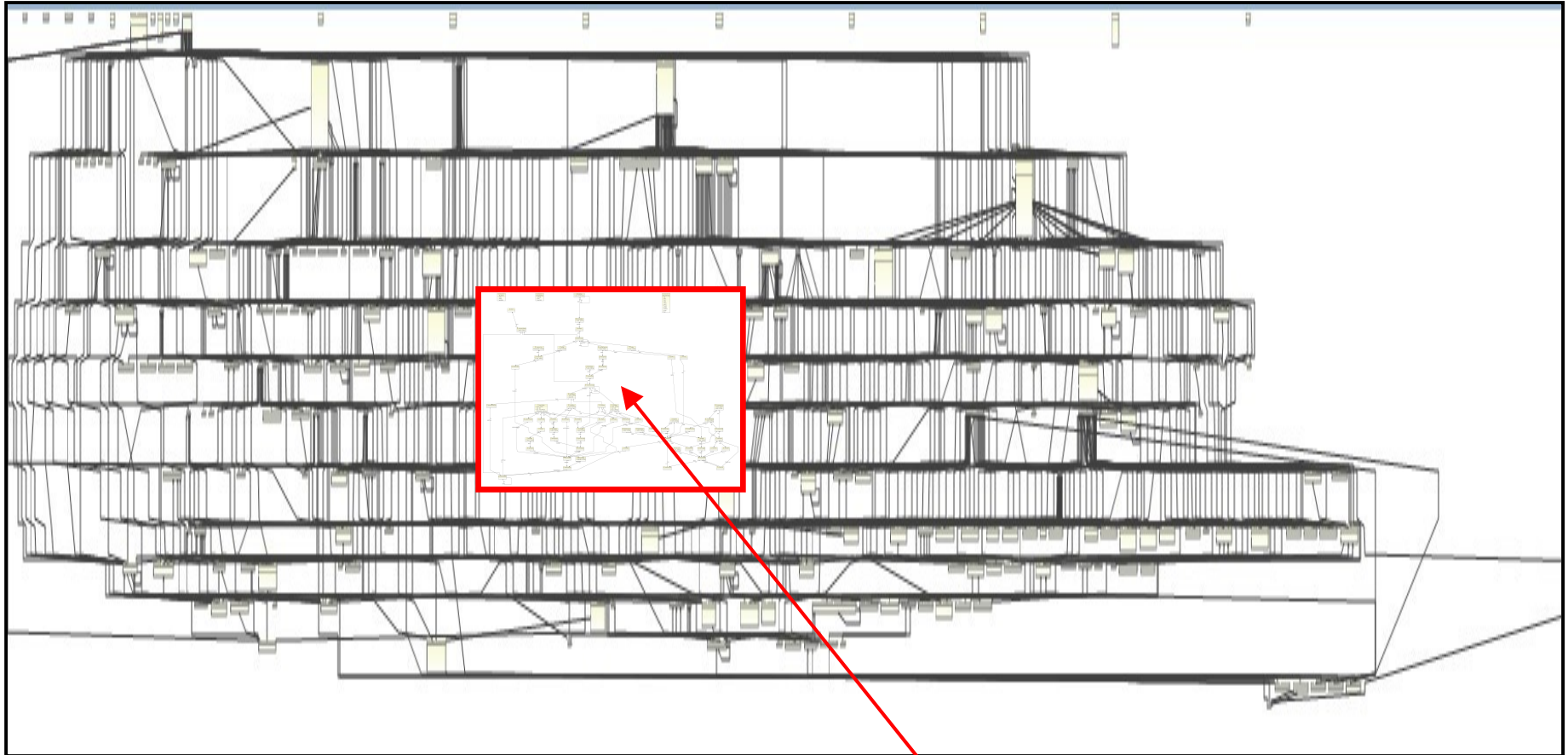
Motivation: Why?

- Increasing size and complexity of software in embedded systems
- Increasing requirements for quality (steer by wire)
- Modeling can help by
 - raising the level of abstraction
 - raising the degree of automation
- Open Source Modeling Tool for event driven embedded systems with complete codegenerator and middleware is missing

➔ The ROOM based Tool eTrice could fill in

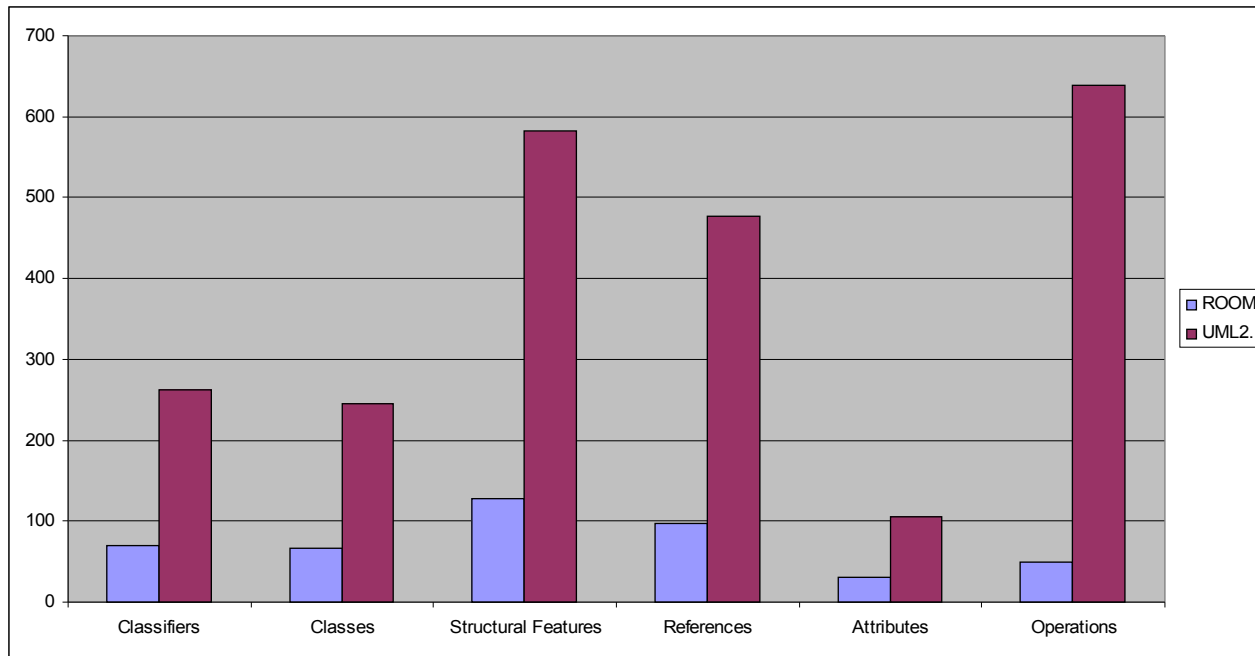
Motivation: Why ROOM and not UML2?

UML2 Meta Model



ROOM Meta Model

Motivation: Why not UML2?



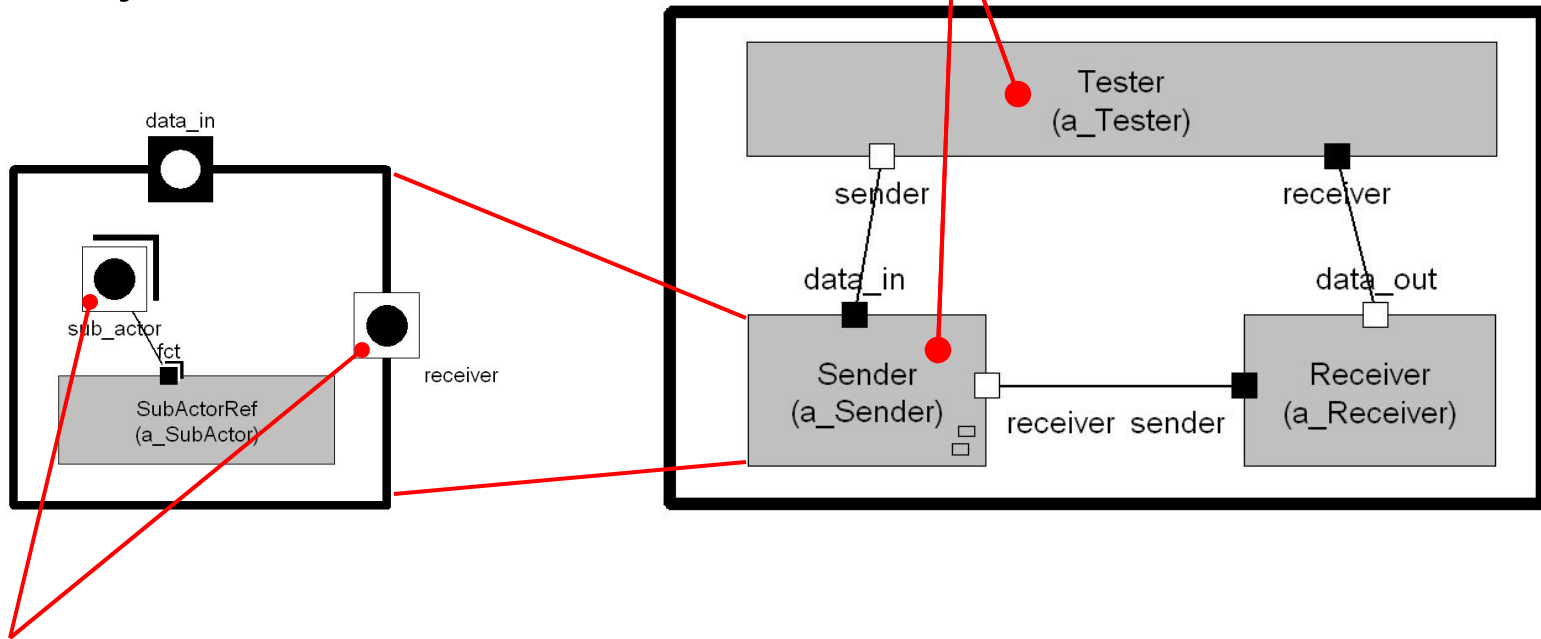
- UML2 is too complex for some/most projects
- UML2 tools are very costly to build and maintain
- UML2 was not designed for embedded systems
- UML2 is not very specific about semantics

Scope of eTrice

- provide an implementation of the modeling language Real Time Object Oriented Modeling (ROOM)
 - build ready to use editors for ROOM models (textual and graphical)
 - create code generators and portable target runtime libraries for Java and C++, later also for ANSI-C
 - provide built-in support for modeling level debugging of the running target software: state machine animation, data inspection and manipulation and message injection
 - provide built-in possibilities for sequence diagram creation from the running software
 - support heterogenous distributed systems out of the box
- ➔ eTrice is a Modeling Toolset for eventdriven, distributed embedded systems

ROOM Language: Actors & Ports

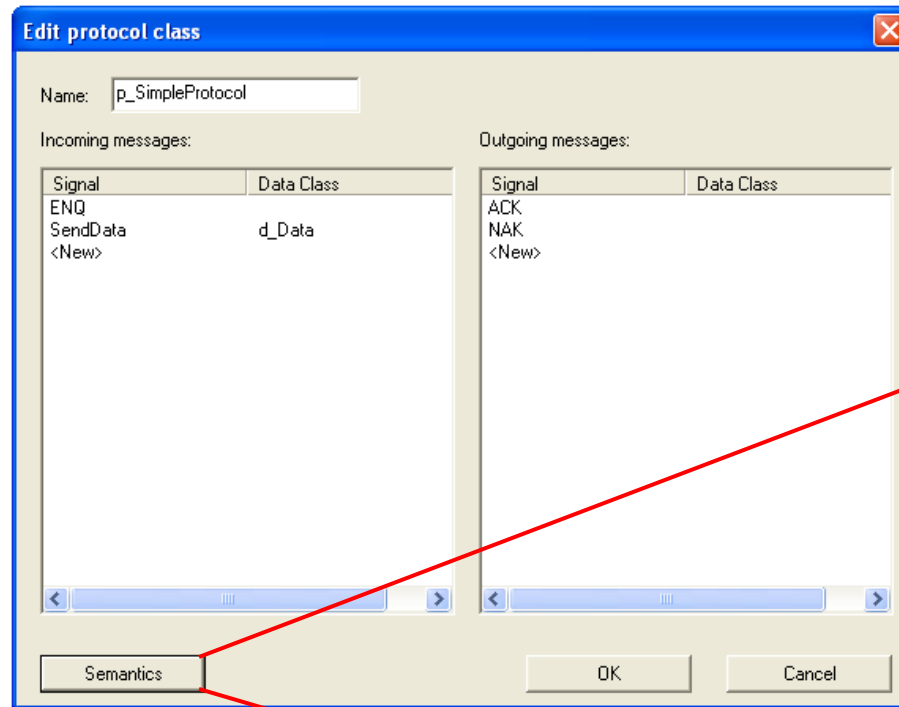
Hierarchical Components called **Actors** define the Structure of a System



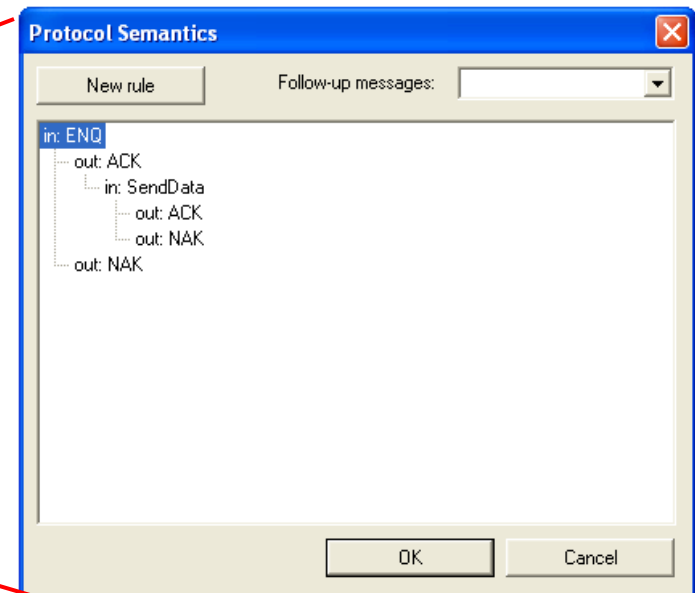
Ports are the only Interfaces of an actor and define a specific role in its environment. They also make Actors always deployable by decoupling them.

ROOM Language: Protocols

Protocols define the Syntax and Semantics of incoming and outgoing messages between Ports.

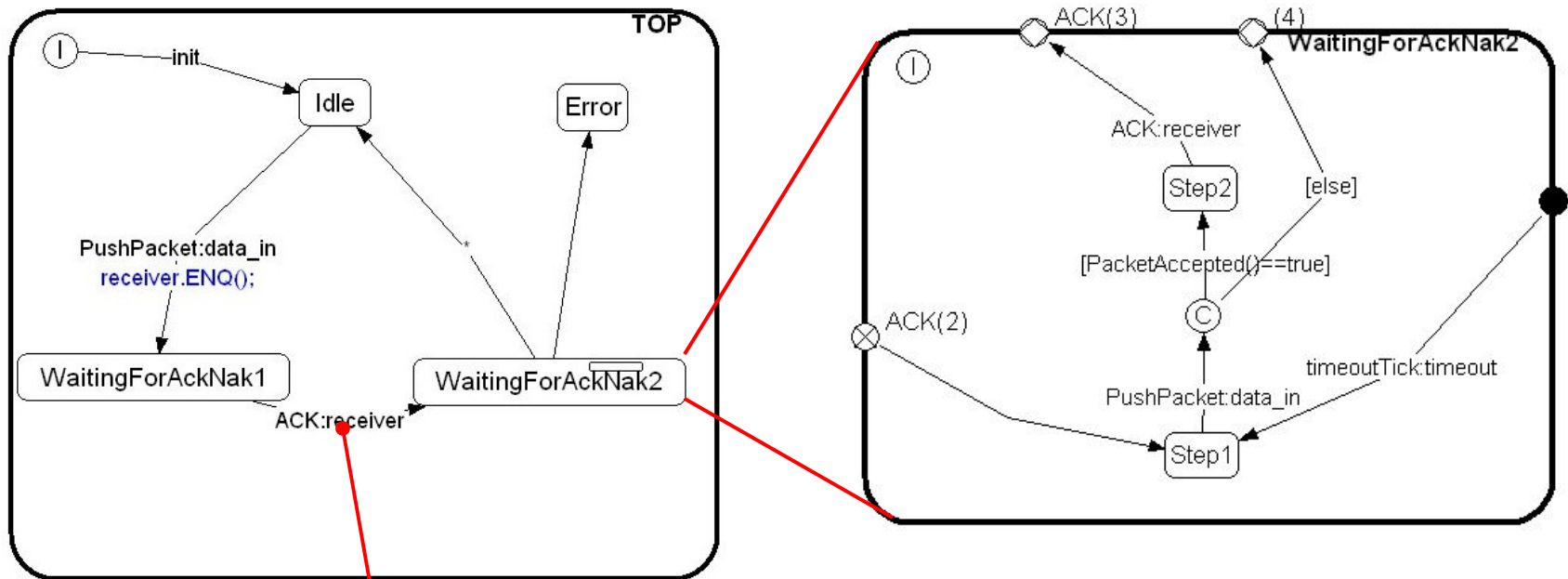


Model checking can proof the correct implementation of semantics



ROOM Language: Statemachines

hierarchical **Statemachines** define the dynamical behaviour of Actors

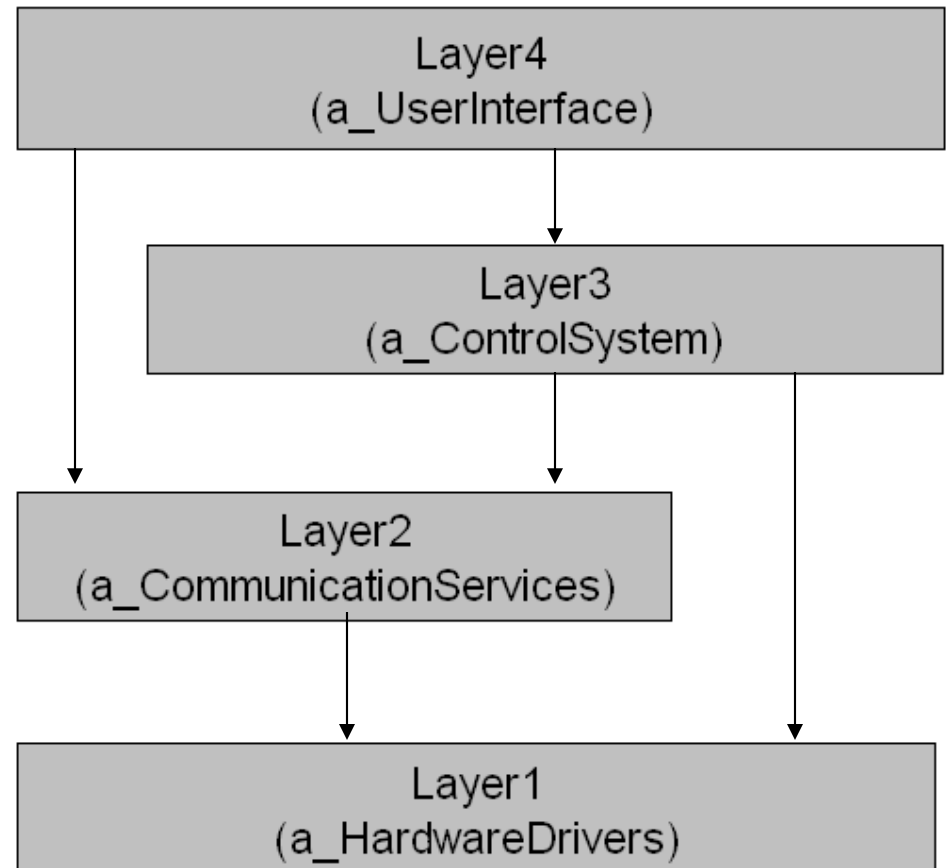


Incoming messages from the Ports trigger transitions

ROOM Language: Layering

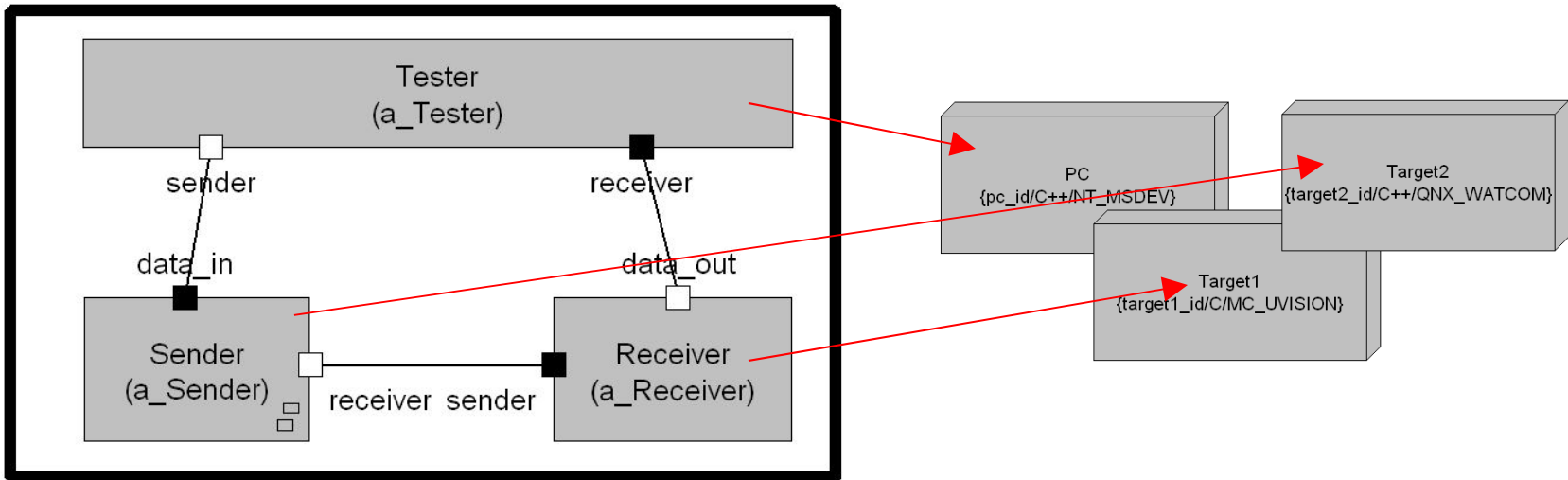
Layering enables the explicit modeling of layered architectures

Layering is a powerful element in ROOM to master complexity



ROOM Language: Deployment

a set of actors can be deployed to a physical node



one or several actors can be assigned to an execution thread

Codegenerators

- the high degree of formalization enables the complete generation of structure and event driven behavior of the model in high level languages
- manual code can be added at various points in the model to add more detailed behaviour
- codegenerators for Java, C++ and C will be implemented
- codegenerators for other languages can easily be implemented

Middleware

the generated code needs a runtime library (middleware) to close the abstraction gap

- platform abstraction / portability
- communication (asynchronous messaging)
- debugging / tracing on model level
- invariant part of modeling elements
- framework for generated statemachines
- deployment / lifecycle
- error handling

Realization

Modeling

- ROOM metamodel with EMF
- initial editors with XText
- graphical editors (Statemachines, Actor hierarchies, ...) with GMF or Graphiti

Codegenerators:

- Xpand/Xtend

Target Middleware:

- Java (JDT), C++, C (CDT)

Project Plan: Current Status

Organizational:

- pre-proposal phase
- gathering of community

Technical:

- current tool Trice since 1998
 -> new implementation with Eclipse
- first proof of concept running
- 50% of ROOM meta model
- XText editors
- simple codegenerator and middleware for Java

Project Plan: Next Steps

Organizational:

- proposal phase
- gathering community

Technical:

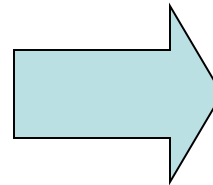
- prototype with textual syntax until November
- first industry pilot project until july 2011
- maturity 12/2011

Project Plan: Pilot Project

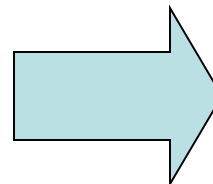
Pilot Customer PARItec



1. smaller production system for inhalers



2. bigger production system for compressors



Committers / Interested Parties

Initial Committers:

- Thomas Schütz, project lead (Protos)
- Henrik Rentz-Reichert, committer (Protos)

Interested Parties:

- Tieto
- Harman Automotive
- Infineon
- PARIttec

Conclusion

The eTrice project will create:

- A ROOM metamodel
 - Textual and graphical editors for ROOM models
 - Code generators for Java, C++, C, ...
 - portable target middleware
 - Model level debugging
- ➔ The eTrice project will create a development tool for eventdriven embedded and real time systems

Thank you for your attention

Thomas Schütz, protos Software GmbH

ts@protos.de