

# Modular EMF/GMF Customization with ObjectTeams/Java

Marco Mosconi

Technische Universität Berlin

[mosconi@cs.tu-berlin.de](mailto:mosconi@cs.tu-berlin.de)

Eclipse DemoCamp Hamburg `09



## Case Study:

- ▶ Refactoring of the Eclipse UML2 Tools class diagram editor using ObjectTeams/Java



## Goals:

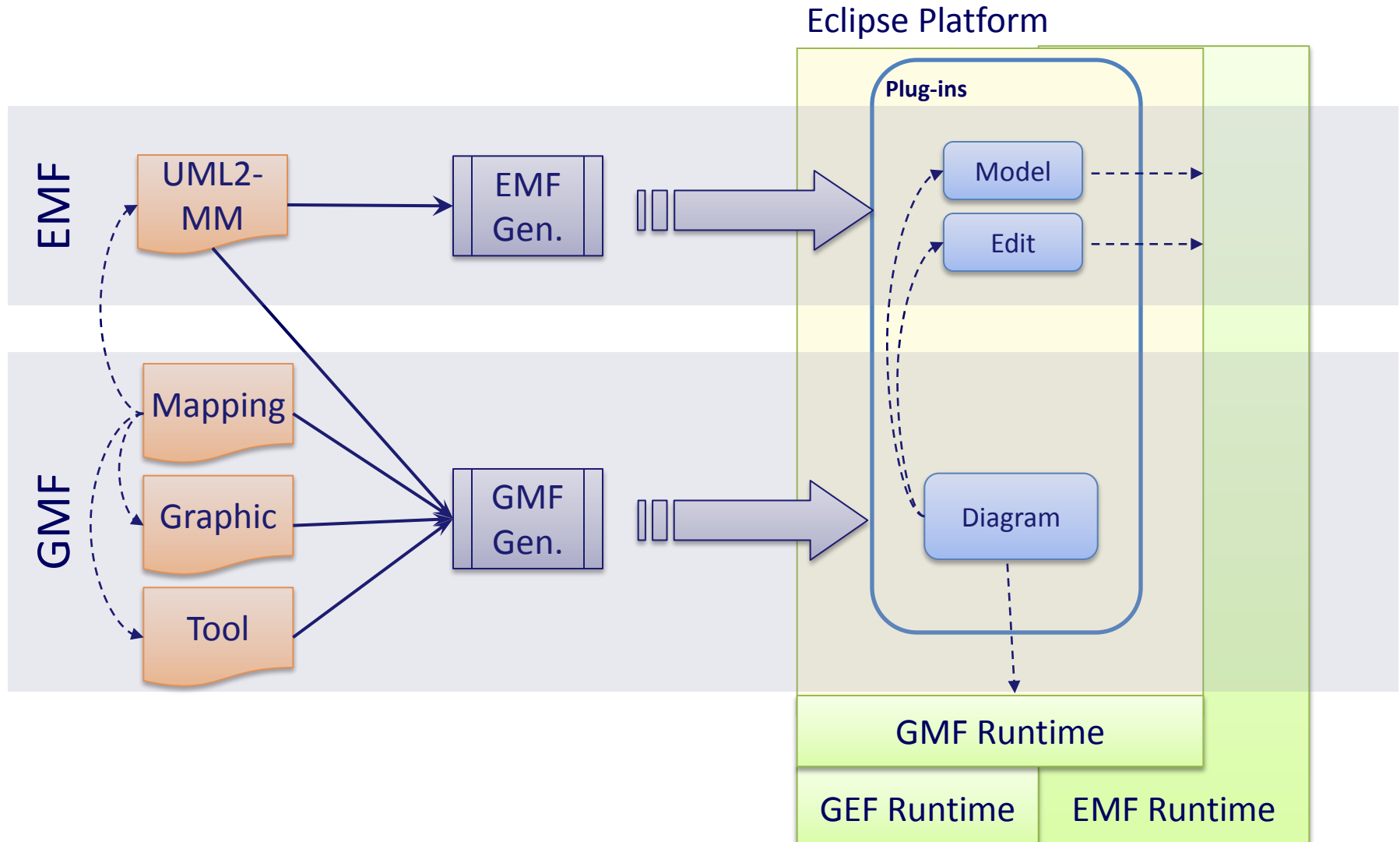
- ▶ 100% separation of generated and custom code
- ▶ modular, feature-oriented architecture for better maintainability of customizations

## Partners:

- ▶ TU Berlin (customer, technology provider)
  - ▶ Stephan Herrmann, Marco Mosconi
- ▶ Nordakademie Elmshorn (realization)
  - ▶ student project
  - ▶ supervisor: Prof. Frank Zimmermann



# Graphical Modeling Framework



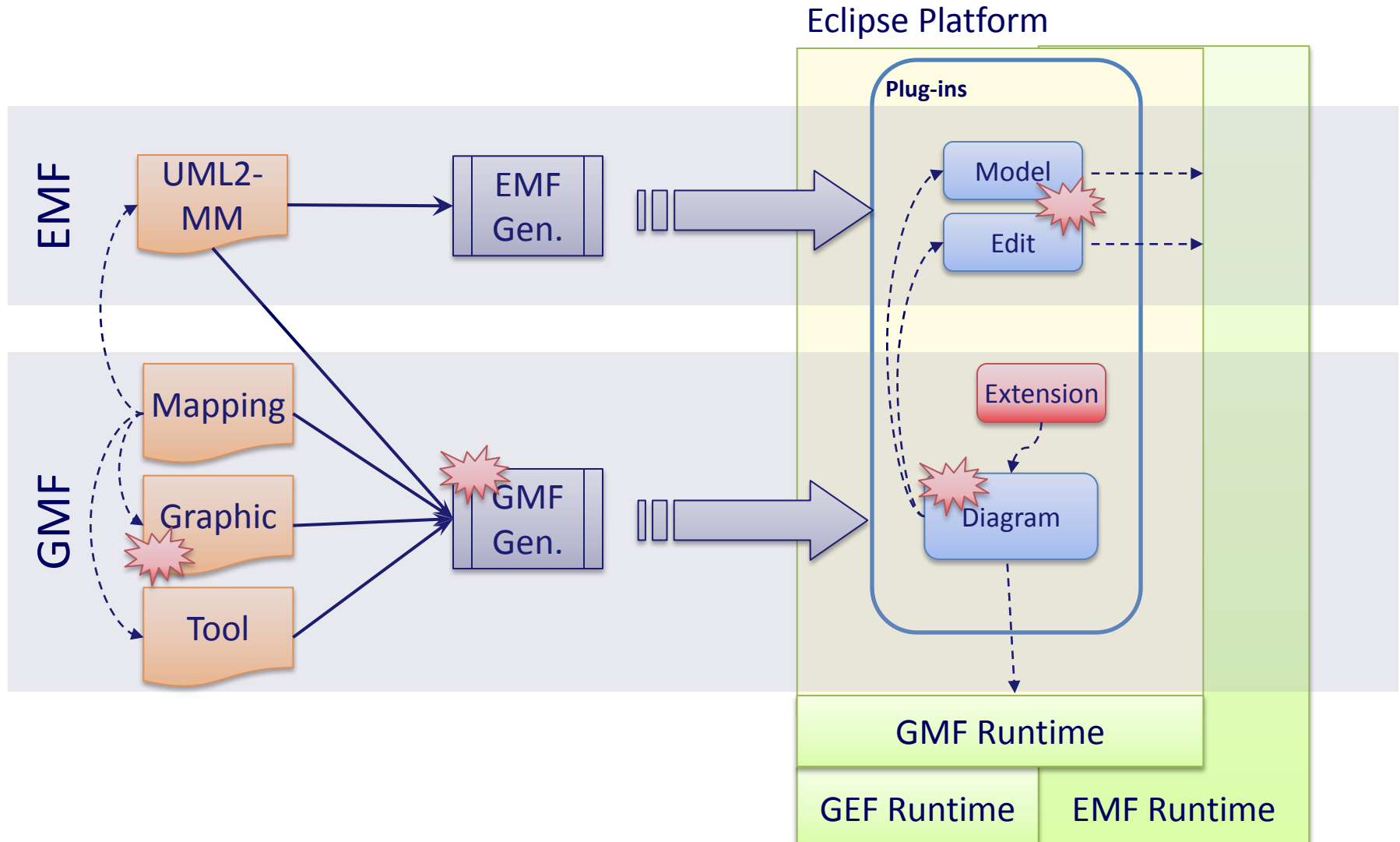
# UML2 Tools Facts

- Eclipse UML2 Tools Project
  - ▶ Suite of graphical editors for UML 2
  - ▶ Developed model-driven with GMF
  - ▶ Uses EMF-based Eclipse UML2 implementation
- Lots of generated code (*that's good news!*)
  - ▶ A bunch of declarative models generate a (fully?) functional graphical editor
  - ▶ *~670 classes, ~82 KLOC* in one plug-in
  - ▶ *But: it doesn't meet all of the requirements*

## UML2 Tools Facts

- Lots of modifications (*oh, that's bad!*)
  - 153x @generated NOT in 61 classes
  - 57x @NOT-generated
  - 41 additional classes, tight + cyclic coupling
  - Already using customized generator templates
  - Just class diagram (there are 12 more)!

# Graphical Modeling Framework



## UML2 Tools Facts

- ❏ Lots of modifications (*oh, that's bad!*)
  - 41 additional classes, tight + cyclic coupling
  - 153x `@generated` **NOT** in 61 classes
  - 57x `@NOT-generated`
  - Already using customized generator templates
  - Just class diagram (there are 12 more)!
- ❏ How to find them? How to maintain?
  - full-text search, EMFT Mint
  - dead code?
  - re-generation, versioning, diff ...

custom code in UML2 Tools editor:  
overview + associations

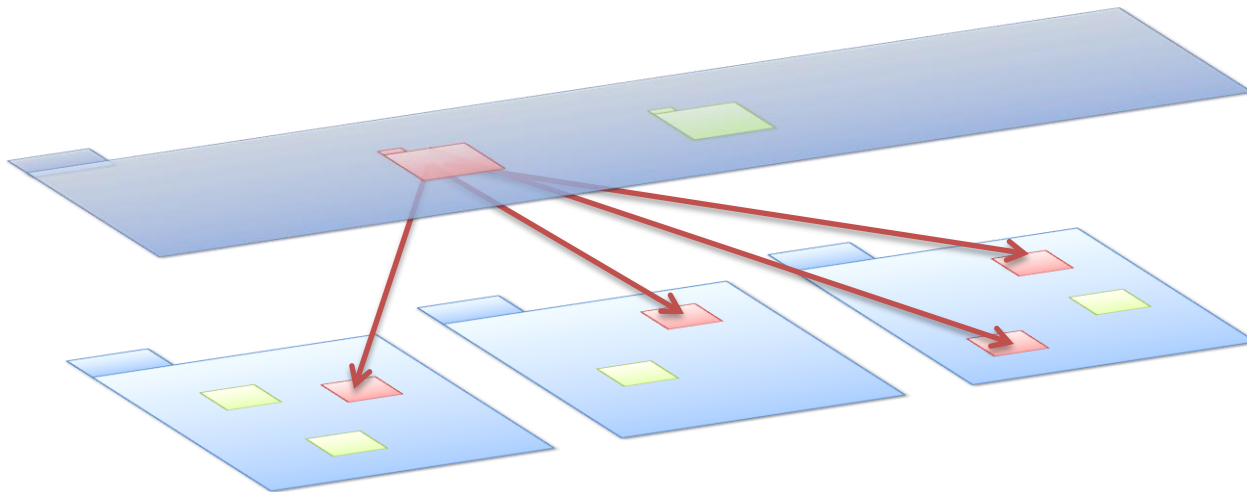
# LIVE DEMO

- Technical packaging of GMF
  - ▶ leads to *crosscutting* customizations for specific features/requirements

Feature	Packages	Classes	Methods
Classifier	1	1	21
Interface	2	3	3
Port	3	6	15
<b>Association</b>	<b>5</b>	<b>12</b>	<b>35</b>
Dependency	3	3	3
Generalization	6	11	18
Instance Specific.	3	3	10
Package	5	5	9
Template Binding	2	2	4

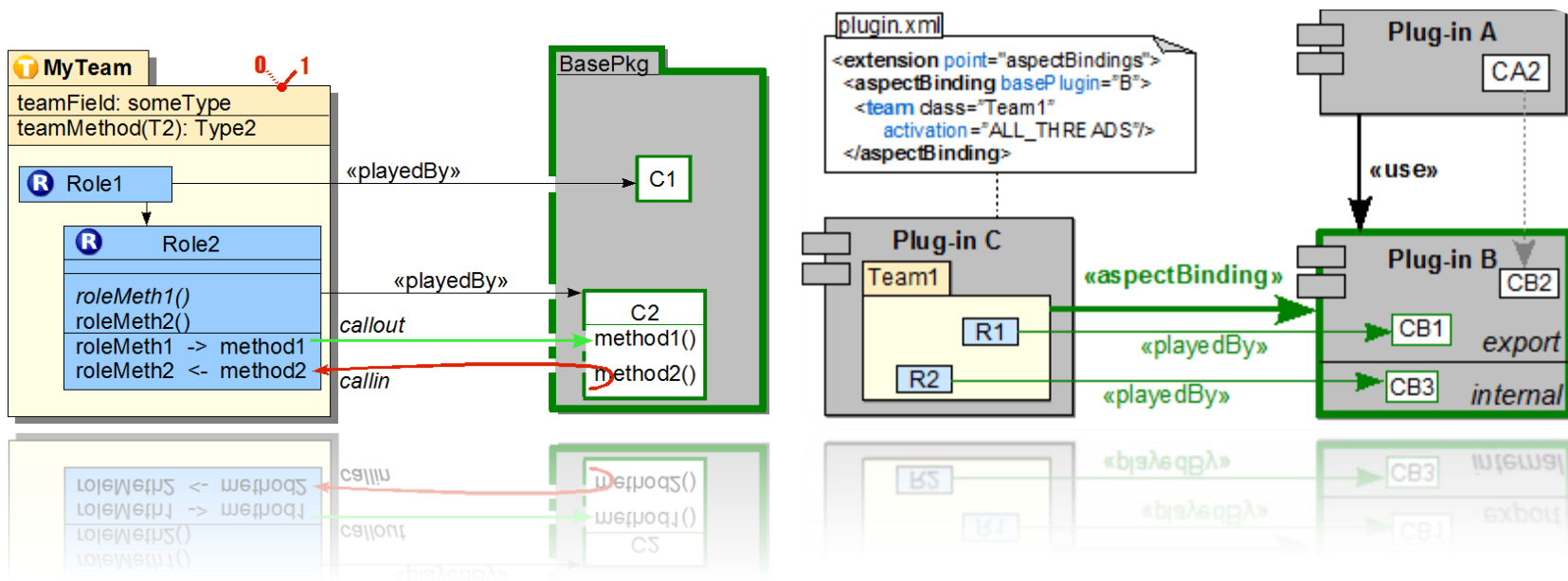
*(modified units per model element)*

- Goal:
  - feature-oriented modularization
  - support different structure



# Technology: Object Teams

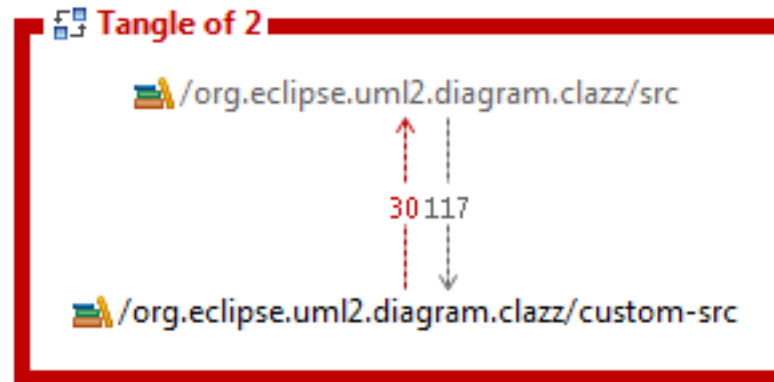
- **ObjectTeams/Java:**
  - ▶ **role-based** and **aspect-oriented** programming for Java
  - ▶ powerful + flexible module concept
  - ▶ non-invasive adaptation
  
- **OT/Equinox:**
  - ▶ ObjectTeams/Java for Eclipse plug-ins
  - ▶ component-level bindings: `aspectBinding`



customization refactoring with Object Teams:  
gen. NOT to callins - roles – inferred callouts - teams

# LIVE DEMO

- On the road to full separation ...
  - ▶ cyclic coupling between *src* and *custom-src* normally prevents separation



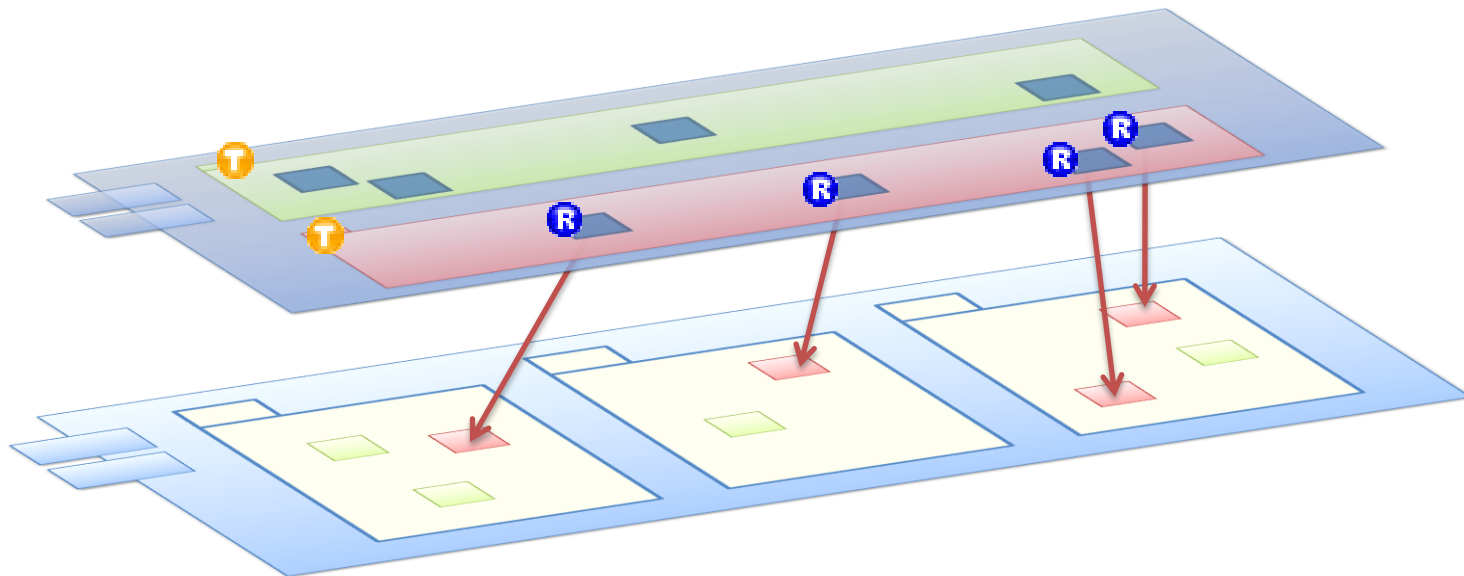
- ▶ not every package is exported by the generated plug-in

100% separation: aspectBinding, custom-src

# LIVE DEMO

## Resulting Architecture

- functional packaging
- separate plug-in



# Conclusions

## Incremental process



- invasive modifications
- improving design
- full separation

## Even more is possible

- optimized generators (generating less)
- implementation of customizations from scratch with OT/J
- using dynamic (de)activation
- re-usable customizations

# Conclusions

- Incremental adoption
  - ▶ low entry-level
  - ▶ advanced features for harder problems
  - ▶ scalability
- Diversity
  - ▶ adaptation of generated source + framework components
  - ▶ can replace multiple extension techniques
- Modular customization with Object Teams
  - ▶ general purpose concepts
  - ▶ type-safe, consistency
  - ▶ explicit adaptation interface

# Object Teams Facts

- ❏ Object Teams Development Tooling (OTDT)
  - since 2003
  - deep integration with Eclipse JDT/PDE
  - makes heavy use of OT/Equinox itself
  - current milestone available for Galileo
  - test suite (>40.000 tests)
- ❏ ObjectTeams/Java Language Definition
  - cross-media: html, online-help, pdf
- ❏ Wiki, Design Patterns, Bugtracking, ...
  - <http://trac.objectteams.org/ot>

# Vielen Dank für die Aufmerksamkeit!

*project team:*

Paul Maria Bartusch

Matthias Diehl

Felix Duus

Tobias Gurski

Christian Laakmann

Fabian Maack

Constantin Pagenkopp

Ruwen Schwedewsky



case study:

▶▶ <http://www.assembla.com/spaces/n-ot-e>



Object Teams:

▶▶ <http://www.objectteams.org>