



CENTRE FOR  
**CONCEPTUAL MODELLING  
AND IMPLEMENTATIONS**



ccmi.fit.cvut.cz

## DynaCASE

software and business modeling platform

Peter Uhnák,  
Jan Blizničenko,  
Robert Pergl

## DynaCASE

### Dynamic Tool For Software And Business Modeling

#### Specialized modeling tools

- Tailored for specific notations and needs
- User-friendlier
- Limited extensibility

#### Meta-modeling tools

- Versatile
- Crude UI
- Requires domain experts

# Project context

We want unified modeling platform for:

- teaching
- student projects
- research

- 2009 – 2014 — OpenCABE on Eclipse EMF/GMF platform
  - after five years just one notation, dependent on a single programmer
- 2014 — experimental student project
  - TRY to rebuild OpenCABE in Pharo from scratch
  - two student programmers
  - two semesters
  - equivalent to two years of development in Eclipse

# Key factors of success

Key factors of success:

- dynamic features of Pharo
- Roassal2 library
- excellent community

Enough of this story...

# DynaCASE - Spec UI

The screenshot displays the DynaCASE software interface for a 'Pizza Delivery' project. The main window is titled 'Pizza Delivery - DynaCASE'. On the left, a tree view shows the project structure, including participants like Customer, Operator, Kitchen, and Delivery Boy, and their associated activities and states. The central area shows a UML activity diagram with three swimlanes: Customer, Operator, and Kitchen. The Customer swimlane includes activities like 'wants pizza', 'ordering pizza via phone', 'waiting', 'accepts pizza and pays', and 'purchase complete'. The Operator swimlane includes 'processing order', 'order accepted', 'passes order', and 'order processed'. The Kitchen swimlane includes 'accepts order', 'cooking pizza', 'hands over pizza', and 'order completed'. The Delivery Boy swimlane includes 'takes pizza', 'driving to customer', 'delivers pizza', and 'driving back'. Interactions between swimlanes are shown with arrows and data objects like 'pizza description', 'order', and 'pizza'. A 'money' data flow is also shown from the Customer to the Operator. The right side of the interface features a 'Tools palette' with various modeling tools and a 'Properties editor' for the selected element, 'Customer', showing its type as 'Person'. The bottom of the interface has a toolbar with zoom and simulation controls.

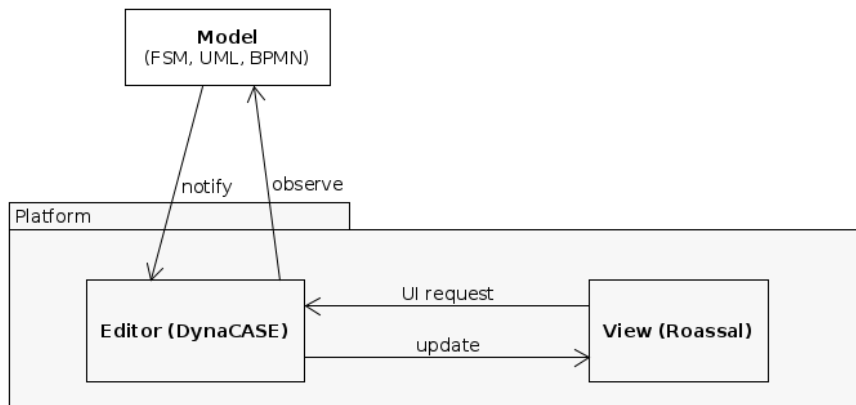
Models in tree view

Roassal View

Properties editor

Tools palette

# MVC



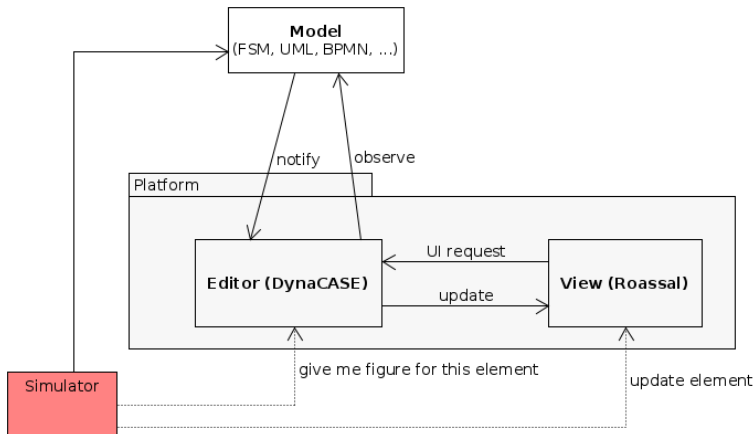


## Finite State Machines

# FSM Simulator

## For simulation we need

- access to the view to visualize it
- access to the model



Platform doesn't care what you will do with view

- ~~this is special interface for moving element~~
- here is Roassal element, do whatever you want
- integrator decides what to do and how
  - this is sci-fi in Eclipse

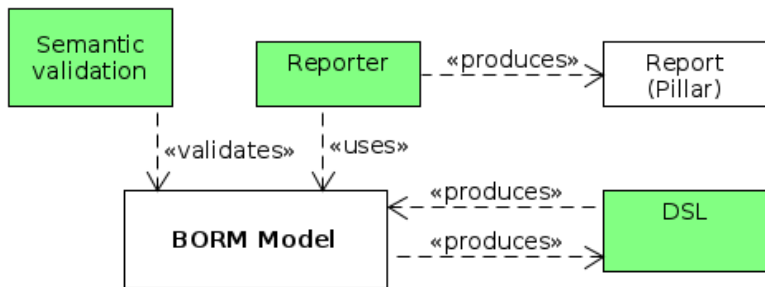
You control the behavior of the flow

- string matching, code execution, ...

## BORM — Business Objects Relation Modeling

# BORM

- DSL
- live validation
- semantic validation (WIP)
- reporting (WIP)

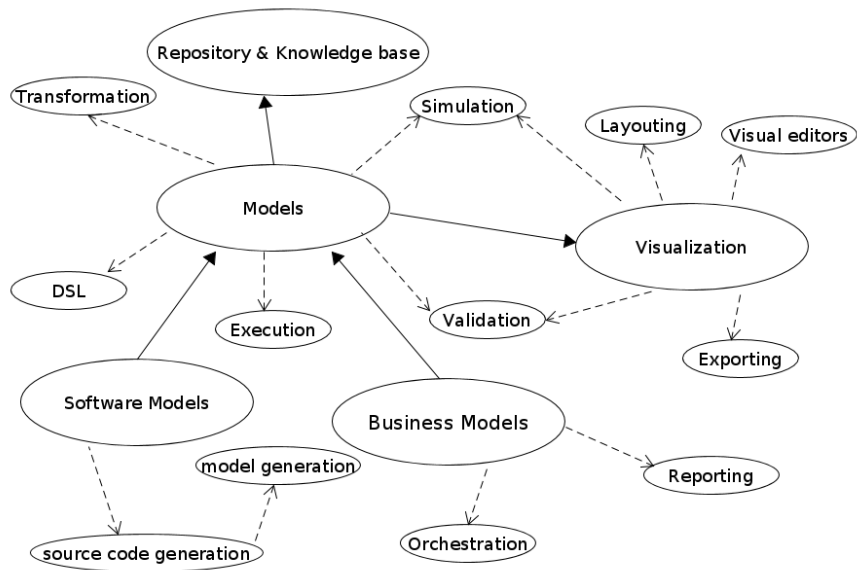


## OntoUML — Ontological Structural Modeling

- DynaCASE as a tool
- DynaCASE as a platform
- Pharo as a platform



# Future



# dynacase.github.io

 @peteruhnak

Peter Uhnák <uhnakpet@fit.cvut.cz>

Jan Blizničenko <bliznjan@fit.cvut.cz>

Robert Pergl <robert.pergl@fit.cvut.cz>



CENTRE FOR  
**CONCEPTUAL MODELLING  
AND IMPLEMENTATIONS**



ccmi.fit.cvut.cz