

25.4. OpenSource Modelling Platform **flowR**

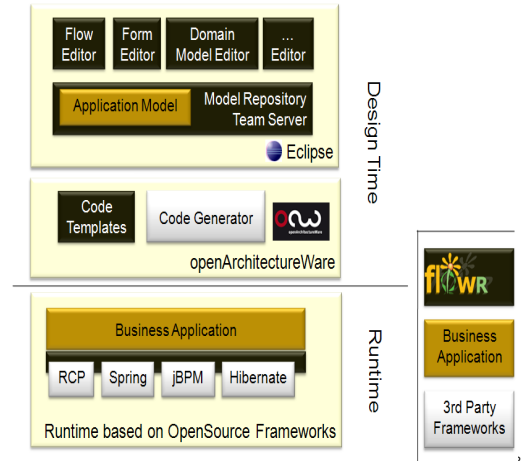


Achievo Deutschland AG
Tom Herrmann

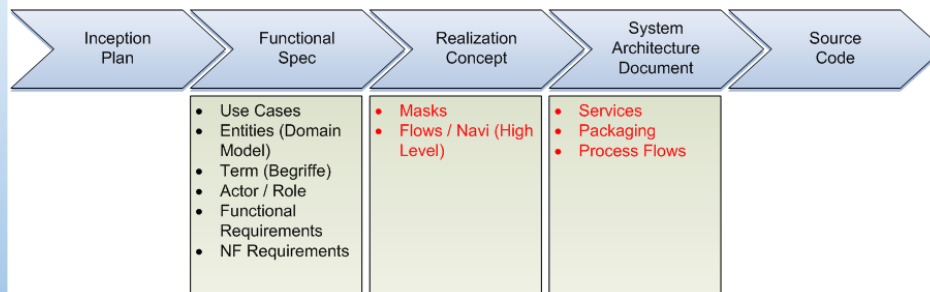
Achievo Confidential

25.4 Flowr von Achievo (Dresden), www.flowr.org

- Ein **Screen Flow** ist eine Abfolge von Masken oder Formularen in Informationssystemen (z.B. SAP-Anwendungen) (Screen/Seiten-Folgen)
 - Webshops
 - Versicherungsfälle
 - Internet Banking
- Flowr ist eine MetaCase-Umgebung zur Generierung von Screen-Flow-Anwendungen
- Metamodellierung in EMF
- Codegenerierung in OAW
- Ursprung in Dresden

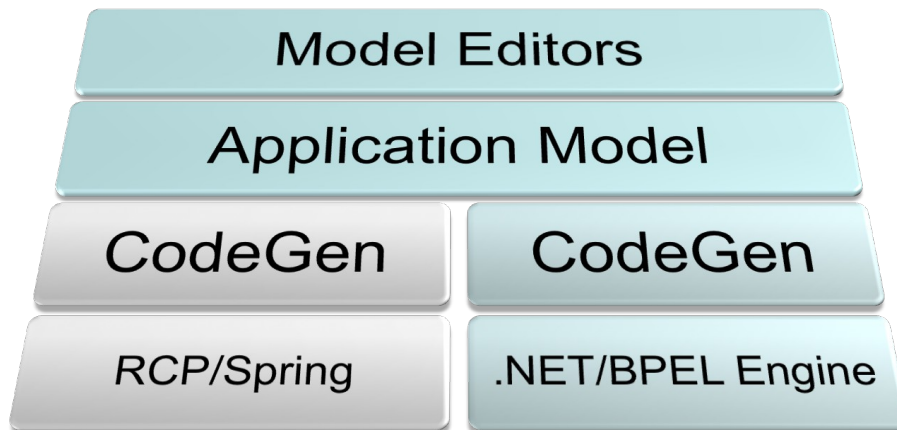
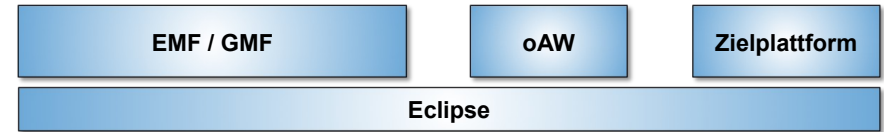
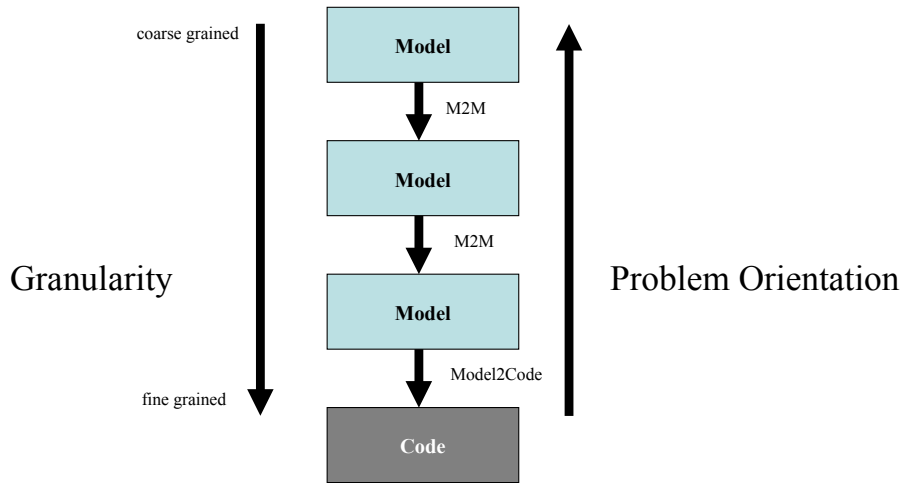


OpenSource Project **flowR** Where in the Development Process is flowR?



OpenSource Project **flowR** Levels Of Abstraction

Activity	Example
Application Specific Modelling	
Implementation Specific Modelling	
Activity	<pre> Example public interface OidFactory { Oid newOid (String type); } public final static OidFactoryReference ref = new OidFactoryReference (); public final static class OidFactoryReference implements OidFactory { public Oid newOid (String type) { return ref ().newOid (type); } private OidFactory ref () { return (OidFactory) SingletonManager.getSingleton (OidFactory.class); } } // OidFactoryDelegator </pre>



- ▶ flowR is a generic design environment to build multi channel solutions using MDSD
- ▶ flowR is a Modelling Framework that can be used with any Generator Framework with EMF capabilities
- ▶ flowR is not a Generator Framework itself (i.e. oAW or AndroMDA)
- ▶ flowR is not a Runtime Framework

What does it provide?

- ▶ An example integration of a full MDSD stack using other OpenSource Frameworks
- ▶ A team server to share models
- ▶ A component model to organize reuse of models
- ▶ A set of standard editors (e.g. screen flows, domain model)
- ▶ A set of APIs to extend the existing editors and write your own editors

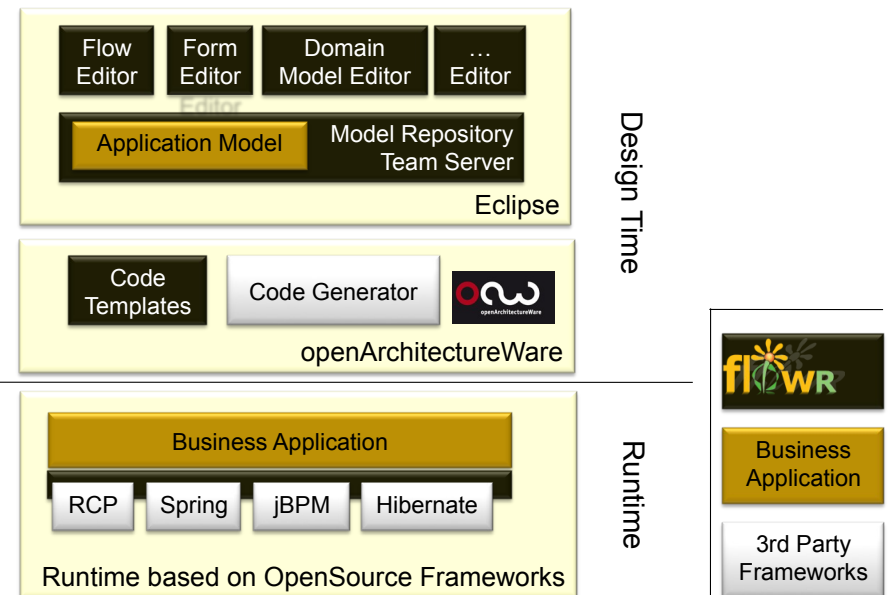
How does it do it?

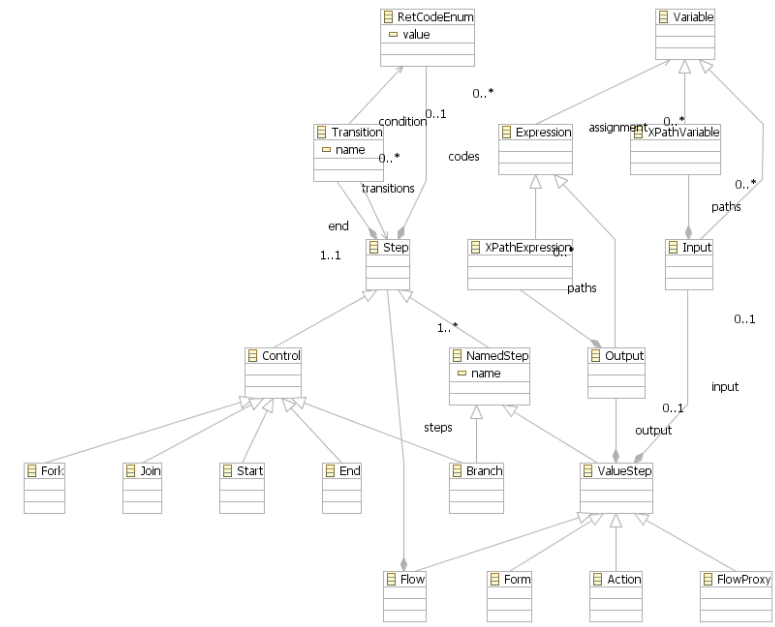
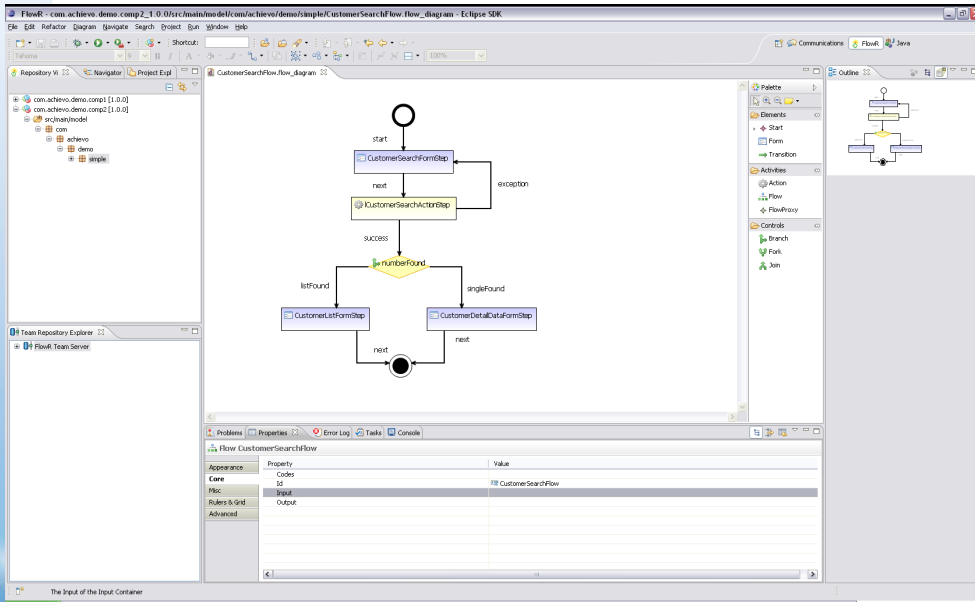
- ▶ Based on Eclipse EMF/GMF
- ▶ Uses openArchitectureware to generate code for different technology stacks
- ▶ Comes with a predefined example OpenSource runtime stack based on RCP and Spring

Which challenges does flowR address?

- ▶ How can I write an editor for my specific DSL and integrate it into my tooling environment?
- ▶ How can I manage my entire application model?
 - ▶ How do I organize team collaboration?
 - ▶ How can I ensure model consistency?
 - ▶ How can I divide my mode into reusable model components?
 - ▶ How can I provide model refactoring support?

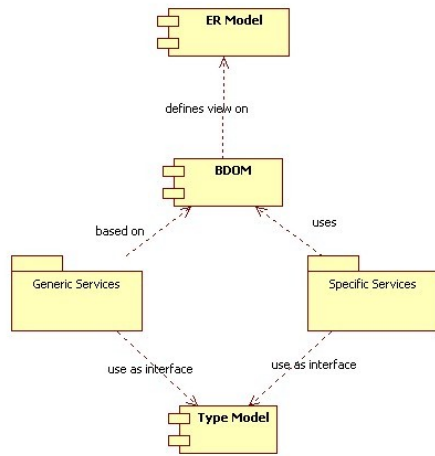
The Big Picture



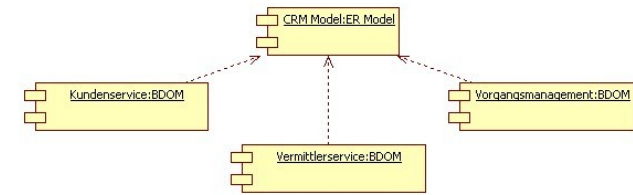


- ▶ Multi-user capability
- ▶ Check in/check out
- ▶ Remote References (shared ref)
- ▶ Central point of validation
- ▶ Central source for Continuous Build

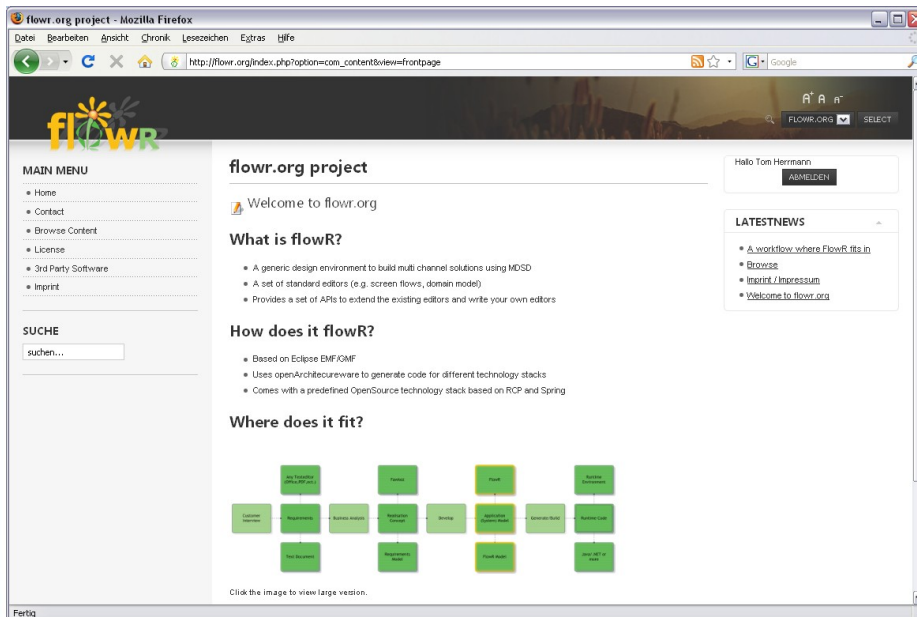
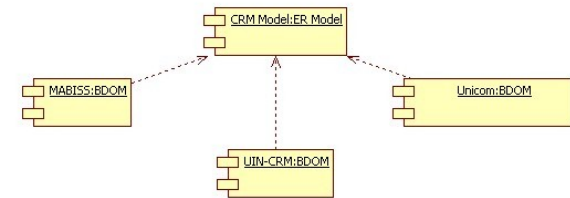
- ▶ Link between Application Model and chosen Generator
- ▶ Generator Model is kind of a Platform Specific Model (PSM)
- ▶ Different Generator Models for different runtime platforms but just a single Application Independent Model (PIM)
- ▶ Allows the embedding of platform specific tools (editors/wizards for manual code) in the tooling
- ▶ Platform specific tools can be deployed depending on the current runtime platform



▶ by functionality



▶ by application



Wir freuen uns auf die Zusammenarbeit mit Ihnen!



Tom Herrmann
Director
Software Development

Achievo Deutschland AG
Blasewitzer Str. 43
01307 Dresden
Germany

Phone +49 351 4403 66 202
Fax +49 351 4403 66 200
Mobile +49 176 1818 89 11
tom.herrmann@achievo.de
www.achievo.de

Global Software & IT Outsourcing