Dr. Tom Ritter
Eclipse Con Europe, Papyrus Summit, 2014
COMMON CHALLENGES IN SOFTWARE ENGINEERING

- Inconsistency, low degree of automation, insufficient common terminology
- Complexity and costs
- Decoupled software tools
- Produced data remains proprietary and depends on specific tools
ModelBus® is a model-driven tool integration framework which allows you to build seamlessly integrated tool environments for your development process.

- ModelBus® connects your tools – commercial off the shelf or in-house tools
- ModelBus® helps automating your development processes
- ModelBus® uses SOA principles and well established standards
General Concept

- Lifecycle Tools are needed for creative work
- Process Enactment controls the development process
- Core services are needed to operate the ModelBus
- Modeling Services provides back-end functionality for automation
Architecture

- Tools are connected via a ModelBus Adapter
- Slight modification of existing tools instead of introducing new tools
- Services can be easily created via ModelBus service interfaces supported by the ModelBus Development Kit
- Workflows can be orchestrated and will be triggered by ModelBus events
- Repositories store relevant artefacts (e.g. models, code, documents) via standardised interfaces
Selection of Connected Tools

- Eclipse-based Tools
  - Papyrus, ProR, …
- Rational Software Architect
- Doors
- Rhapsody
- Simulink
- Microsoft Office (Word, Excel)
- Sparx Enterprise Architect
- AVL InMotion
- Modelling Services
  - QVT, ATL, OCL, Metric Computation, Report Generation, Model Repository
- Traceability with Traceino
- Requirements Engineering with Requino
- OSLC-Compatibility
- SVN, Git
Avoiding merge conflicts

- Locking of parts of the model
- Interactive highlighting of locked model parts
- Interactive modelling
FOKUS!MBT – Customizing Papyrus for MBT
Git Integration

ModelBus

ModelBus Server

ModelBus Server Application

ModelBus Git Repository

ModelBus Git Repository

ModelBus Git Repository

ModelBus Git Repository

ModelBus Git Repository

ModelBus Git Repository

Conflict Resolution in ModelBus Clients

ModelBus Client

“Central” Repository

Developer Git Repository 1

Developer Git Repository 2

Developer Git Repository N

Read/write

Sync

Sync

Sync

Sync
As product and consulting company for the automotive domain B&M is specialized in systems engineering, development and testing of complex electronic and mechanical systems. Within the case study B&M will appear as developer of automotive driver assistance software.

Traffic Sign Recognition System
- Different hardware and operating systems
- Different countries with variety of laws and regulations
- Different directions of traffic
- Different sets of customer specific functionalities

http://www.varies.eu/
VARIES CASE STUDY - EXISTING PROCESS

- Requirements Management using Doors
- Architecture using VISIO
- Design using VISIO
- Implementation using Eclipse
- Testing using CTE XL Professional and Messina
VARIES CASE STUDY - PROCESS AIMED AT

- Requirements Management using Doors & Meran
- Architecture using Artisan Studio (or Papyrus)
- Analysis (QFD, FMEA, Risk) using Requino
- Design using Artisan Studio (or Papyrus)
- Implementation using Eclipse
- Variability Management using pure::variants
- Testing using CTE XL Professional and Messina
VARIES CASE STUDY - PROPOSED REALIZATION

OSLC RM & Meran VM via OSLC REST

CVL

pure::variants VM via ModelBus

UML & OVM via ModelBus

CTE XL & Files Via ModelBus

Analysis (QFD, FMEA, Risk) using Requino

UML & OVM via ModelBus

UML & OVM via ModelBus

Files (p::v) via ModelBus

transformations

traces
CONSISTENT VARIABILITY MANAGEMENT HANDLING

- Requirements Management using Doors & Meran
- OSLC RM & Meran VM via OSLC REST
- UML & OVM via ModelBus
- CVL
- Analysis (QFD, FMEA, Risk) using Requino
- UML & OVM via ModelBus
- Pure::variants VM via ModelBus
- CTE XL & Files Via ModelBus
- Files (p::v) via ModelBus

Transformations:

Traces:
CONSISTENT VARIABILITY MANAGEMENT HANDLING

Meran → OVM → pure::variants

traces and bidirectional conservative transformations

CVL
CROSS FRAMEWORK TRACEABILITY

OSLC RM & Meran VM via OSLC REST

pure::variants VM via ModelBus

CVL

Analysis (QFD, FMEA) using Requino

UML & OVM via ModelBus

CTE XL & Files Via ModelBus

UML & OVM via ModelBus

Files (p::v) via ModelBus

transitions

traces
CROSS FRAMEWORK TRACEABILITY
Example Workflow

**MODELBUS®**

Eclipse/Papyrus

UML concept space

Sparx Enterprise Architect

Simulink concept space

mapping of the concepts

Simulink

UML

Block

Class

Port

Port

Line

Information Flow

Matlab Simulink

Fraunhofer FOKUS
Benefits

- ModelBus®
  - connects tools and data coming from different teams
  - improves consistency in the development process
  - makes you independent of tool providers and prevents vendor lock-in
  - automates tedious tasks in your development

- ModelBus®
  - is extensible and customisable
  - Basic functionality is free to use and Open Source
  - A number of pre-existing tool adapters are available

ModelBus®: [http://www.modelbus.org](http://www.modelbus.org)
YouTube Channel: [http://www.youtube.com/user/ModelBusOrg](http://www.youtube.com/user/ModelBusOrg)

We want to move ModelBus Core and TeamProvider to Eclipse
Dr. Tom Ritter
Director System Quality Center

Fraunhofer Institute
for Open Communication Systems FOKUS
Kaiserin-Augusta-Allee 31, 10589 Berlin, Germany
Tel: +49 (30) 34 63 – 7278
Fax: +49 (30) 34 63 – 99 7278

tom.ritter@fokus.fraunhofer.de
SQC: http://www.fokus.fraunhofer.de/go/sqc
ModelBus®: http://www.modelbus.org
YouTube Channel: http://www.youtube.com/user/ModelBusOrg