



ModelBus

Automation, Integration and Collaboration in Development Processes

Application Lifecycle Management

Mastering the Application Lifecycle is the key aspect of successful software and system development.

Challenges in the development process:

- Collaboration
- Traceability
- Reporting and analytics
- Process automation

Features of ModelBus

- Integration of software tools
- Construction of integrated and automated tool chains
- Support of collaboration of developers
- Based on Service-oriented Architecture

Increasing Complexity in Software Development

Computer based systems are becoming more and more complex as they provide increasing functionality. They have to fulfill ever rising demands on availability and stability. It is getting harder for companies to deliver high quality complex systems quickly. New and strict quality and safety regulations are putting additional pressure on software makers and solution providers. Development experts need more specialized tools than ever before to cope with all aspects of today's complex systems.

In addition, the globalization of software development leads to separate and remote supply chains, which may involve nearshore and offshore companies. Software and system development processes involve multiple development teams at various locations and need efficient coordination and control.

Application Lifecycle Management (ALM) is becoming the main factor for mastering the whole development process. Time and cost efficient processes combined with high quality software are the key points to winning the fierce competition on the software production market.

Mastering Complexity

ModelBus is a framework for managing complex development processes and integrating heterogeneous tools. It makes it possible to integrate tools from different vendors serving different purposes. This integration creates a virtual bus-like tool environment, where data can be seamlessly exchanged between tools. This avoids the manual export and import of tool specific data, which is usually accompanied by manually executed data alignment steps.

ModelBus tool integration contributes to the collaboration of engineers and developers involved in the software and system development process. The virtual bus architecture is used to exchange information between tools and developers. Thus it supports the coordinated simultaneous work. ModelBus automation is key to increasing the efficiency in a software and system development environment. ModelBus facilitates the automatic and semi-automatic execution of process steps throughout the complete software development process.

Technologies

ModelBus uses only well-established and accepted standards, protocols and software packages.

Transportation:

- HTTP, CXF, JMS, SOAP

Orchestration:

- BPMN, BPEL, ODE

Core Technologies:

- DOSGi, SVN, EMF

Model-Driven Data Management and Service Execution

The key concept of ModelBus is the virtual bus-like architecture and the way it processes the data transmitted via this bus. ModelBus can work on traditional artifacts like source code or binaries, but its full potential lies in the handling of models. Tool data can be transmitted via the ModelBus as well-defined MOF/EMF based models, which enables the full power of model-driven engineering practices to the ModelBus data management. This includes the application of model-transformation techniques, consistency checks and full traceability across multiple process steps ranging from requirements to code for example.

With that approach every piece of information created during the development process is accessible and usable for the process and its control. Tools connected to ModelBus can offer or consume services acting on these data. In that way functionality – provided by individual tools – becomes available for the whole development process and can be used in automated process steps.

Selection of Adapters

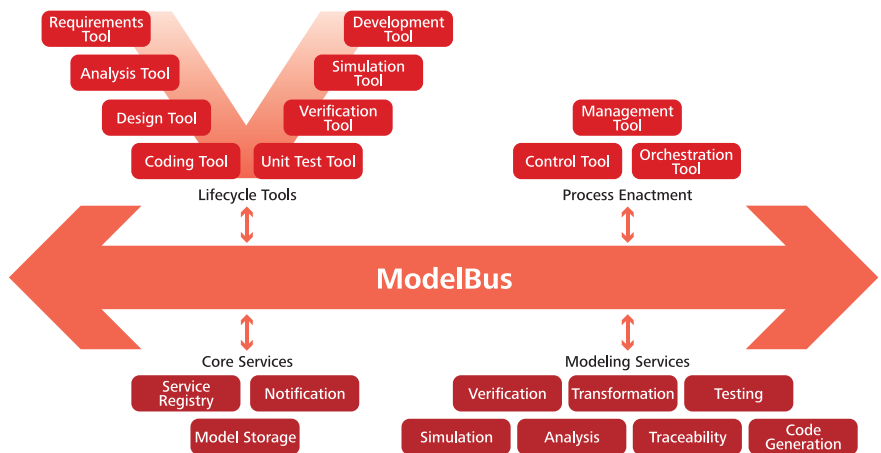
Tools:

- Eclipse: Topcased, Papyrus etc.
- IBM: DOORS, RSA
- Sparx Enterprise Architect
- Matlab: Simulink, Stateflow
- Microsoft Office
- TRAC

Services:

- Transformation: ATL, QVT etc.
- Verification: OCL, Metrino etc.
- Testing: FOKUS!MBT etc.
- Code and document generation: MOFscript, M2T etc.

The ModelBus Architecture



ModelBus realises a bus-like architecture for connecting tools and services.

Contact

Tom Ritter

tom.ritter@fokus.fraunhofer.de

+ 49 30 3463 7278

www.modelbus.org

Fraunhofer Institute for Open
Communication Systems FOKUS
Kaiserin-Augusta-Allee 31
10589 Berlin
www.fokus.fraunhofer.de

Application and Benefits of ModelBus

ModelBus is applicable in various domains including embedded systems design, IT-Business, automotive and avionics for example. The ModelBus framework makes it possible to create flexible development solutions adapted to the customer's needs. New tool adapters can be built upon request. It shows its full benefit in medium or large development processes but ModelBus can be used for small solutions as well.

Using ModelBus will help to improve performance of the development and test processes by injecting automation to the highest possible degree. ModelBus helps keep the existing processes and tools unchanged. Therefore it helps save licensing costs and training of developers.

The basic set of ModelBus is Open Source and free software, which makes it easy to use. Consultancy, support and maintenance services are available for establishing a ModelBus based development scenario fitting to individual needs.