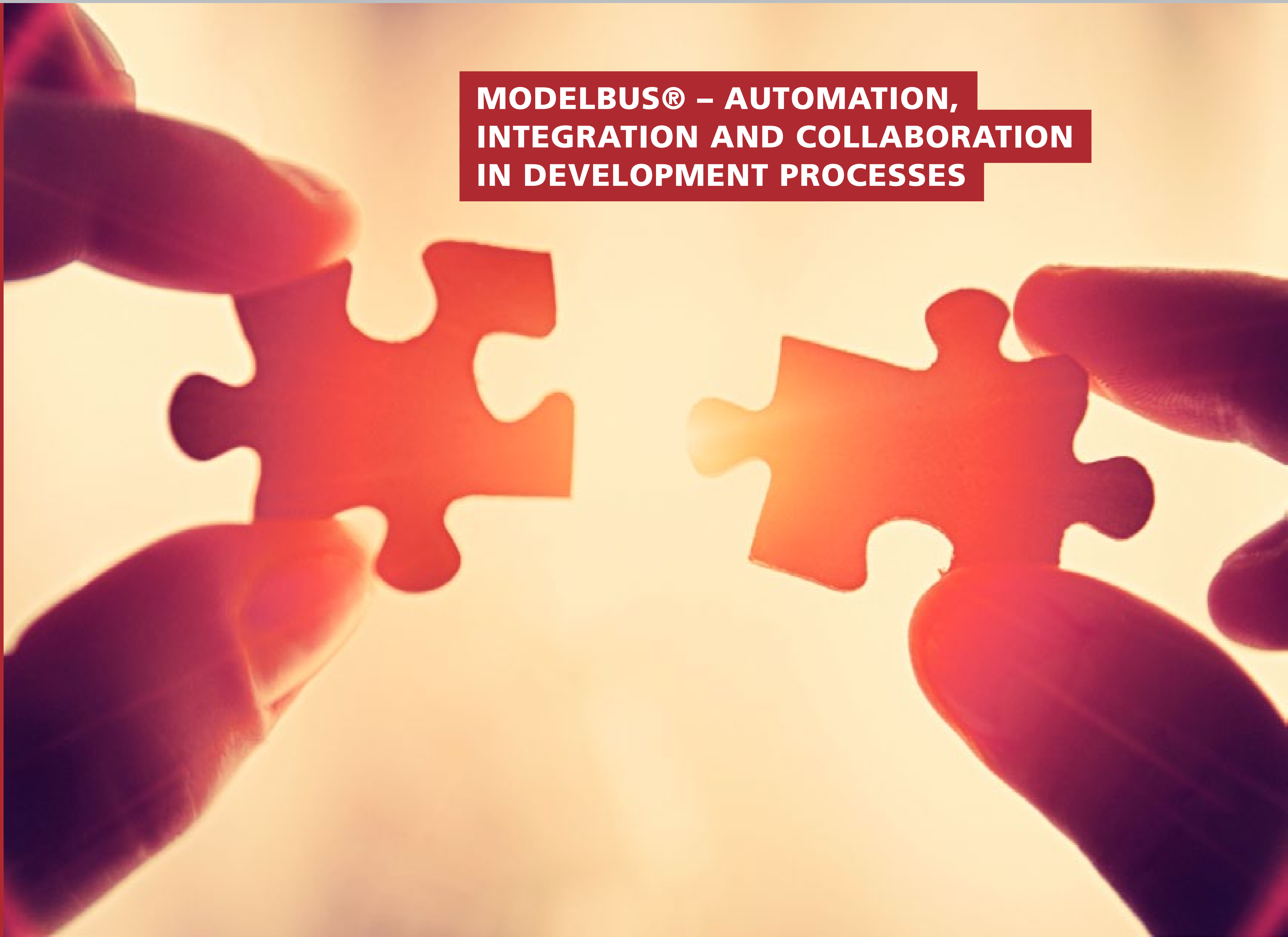
making
cities smart

Fraunhofer
FOKUS

FRAUNHOFER INSTITUTE FOR OPEN COMMUNICATION SYSTEMS FOKUS



MODELBUS® – AUTOMATION, INTEGRATION AND COLLABORATION IN DEVELOPMENT PROCESSES

MODELBUS® IMPROVES DEVELOPMENT AND TEST PROCESSES BY INJECTING AUTOMATION TO THE HIGHEST POSSIBLE DEGREE

INTRODUCTION

Since its formation in 2006, the ModelBus® Team at the System Quality Center at Fraunhofer Institute FOKUS has provided tools and services for improving system engineering processes. Being part of Fraunhofer - Europe's largest application-oriented research organization - our work focuses on increasing efficiency in system development processes. One of the major outcomes of this effort is ModelBus®, a framework automating system development, achieving tool interoperability and facilitating collaboration between engineers.

Over the past years, my team and I have witnessed an ever-growing demand for integration and automation solutions for system development processes. As a result, the tools and services ModelBus® provides have constantly grown. We have not only extended the list of tool adapters available for ModelBus®, but we also included technologies for traceability, metric computation, requirements engineering and testing. Consequently, ModelBus® has evolved into an advanced tool integration platform, which serves as the basis for innovative, efficient and highly integrated development solutions.

The next pages offer an overview of ModelBus®, its components and how the framwork can improve specific engineering processes. By analyzing current development environments, examining the interactions within the teams and assessing data exchanged between tools, the ModelBus® team can help companies reach higher efficiency in their respective engineering process, either in small steps or through a substantial re-architecting.

We invent new system engineering approaches and we realize them with passion and precision.

I wish you an exciting and informative reading.

Dr. Tom Ritter

Head of ModelBus® Team
Director of System Quality Center

CONTENT

AUTOMATION, INTEGRATION AND COLLABORATION IN DEVELOPMENT PROCESSES

MODELBUS®

"ModelBus® enables our clients to reach a much higher consistency throughout the entire development process and bridges the gap between proprietary data formats and application programming interface."

Dr. Tom Ritter, Fraunhofer FOKUS

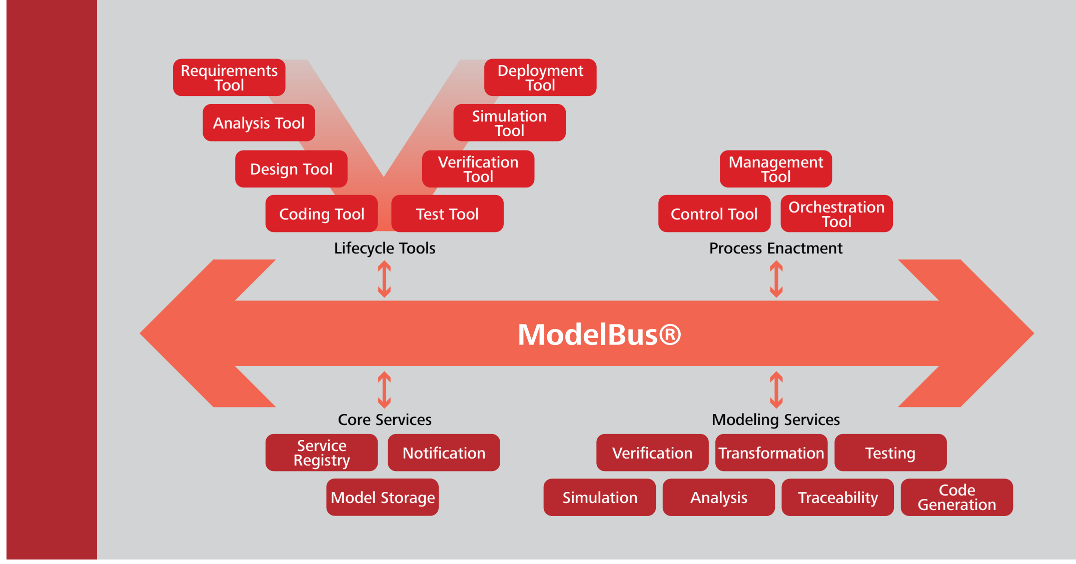
Computer based systems are becoming more and more complex due to the increasing functionality they provide. The systems have to fulfill ever rising demands on availability and stability, making it harder for companies to deliver high quality, complex systems in time. New and strict quality and safety regulations are putting additional pressure on system producers 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 system development has led to separate and remote supply chains, resulting in multiple development teams at various locations that require efficient coordination and control.

Tool interoperability and Application Lifecycle Management (ALM) are becoming the main factors for mastering the entire development process. Time- and cost-efficient processes combined with high quality software are essential to win the fierce competition on the software production market. However, the challenges in the development process are numerous, ranging from tool interoperability, collaboration and traceability to reporting and analytics, as well as process automation.

»The ModelBus® framework makes it possible to create flexible development solutions adapted to customers' needs«

ModelBus® is a framework for managing complex development processes and integrating heterogeneous tools. It allows to integrate tools serving different purposes from different vendors. This integration creates a virtual bus-like tool environment, where data can be seamlessly exchanged between tools, avoiding the manual import and export of tool-specific data, which is usually accompanied by manually executed data alignment steps. The ModelBus® framework also allows to link data, thereby establishing traceability of the work products created during the development process. ModelBus® interoperable tool integration contributes to the collaboration of engineers and developers involved in the software and system development process. Thus, it supports coordinated, simultaneous work. ModelBus® automation is the key to increasing efficiency in a software and system development environment. ModelBus® facilitates the automatic and semiautomatic execution of process steps throughout the complete software development process.

The key concept of ModelBus® for tool interoperability is the virtual bus-like service-oriented 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 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. Due to 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.

APPLICATION AND BENEFITS OF MODELBUS®

ModelBus® is applicable in various domains including embedded systems design, IT-Business, automotive and avionics. Since new tool adapters can be built upon request, the ModelBus® framework makes it possible to create flexible development solutions adapted to the customers' needs. While ModelBus® shows its full benefit in medium or large development processes, it 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® aims at keeping the existing processes and tools unchanged, thereby saving costs on licensing or the training of developers. The basic set of ModelBus® is open source and free software. Tool adapters, consultancy, support and maintenance services are available for establishing a ModelBus®-based development scenario adjusted to individual needs.

Features of ModelBus®

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

Technologies

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

Transportation

- HTTP, HTTPS, XMPP, CXF, JMS, SOAP, REST, OSLC
- DOSGi, EMF, SVN, GIT

Orchestration

- BPMN, BPEL, ODE

Core Technologies

- Distributed DOSGi, SVN, EMF

MODELBUS® ADAPTERS

ModelBus® adapters utilize the capabilities of the respective tools for a seamless integration into the tool workflow and for an unconstraint user experience. Each tool is connected to ModelBus® via a specific ModelBus® adapter and can optionally act as a service by providing its functionality to other tools.

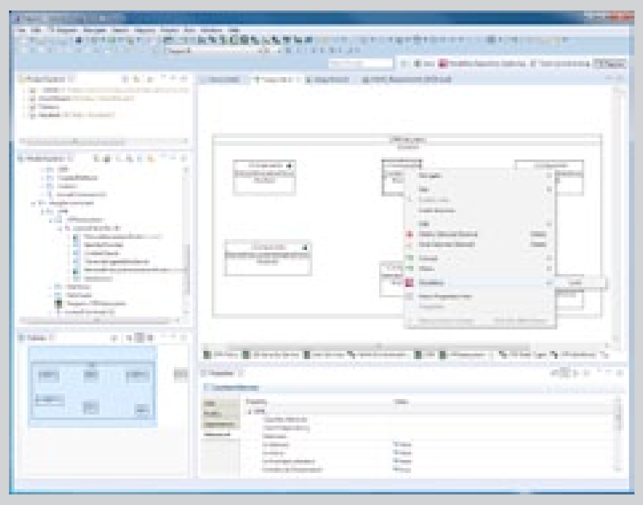
CONNECTING TOOLS TO MODELBUS®
IMPROVES EFFICIENCY IN SYSTEM
ENGINEERING PROCESSES

With ModelBus®, Fraunhofer FOKUS offers a model-driven tool integration framework which supports the integration of heterogeneous development and engineering tools as well as the automation of otherwise error-prone tasks. The engineer's activities automatically trigger subsequent actions, like model transformation, code generation and quality checks, so that the exchange of data in distributed teams with their respective tools is ensured. That way, ModelBus® can substantially increase the productivity of creating complex software-based systems. An adapter typically bridges between a tool's internal representation of data and a tool's external format, which is then the basis for further processing within the development process.

ECLIPSE PAPYRUS

Today more and more tools are moving towards the Eclipse Integrated Development Environment IDE as it is a very powerful and extensible framework. As new, tools are likely to be created on top of the Eclipse IDE. The ModelBus® Adapter allows to connect every Eclipse-based tool to the ModelBus® independently, whether it works on models or on files.

ModelBus® Adpater for Papyrus



It is realized as an extension of the Eclipse Team Provider interface. The workflow is very similar to the one that developers are used to when working with Eclipse Subversion clients for example. The adapter provides several different views, which show the content of the ModelBus® Repository or the ModelBus® Notifications for example. The adapter also tracks dependencies of a model to other models including meta-models or profiles. In that way they can be synchronized automatically.

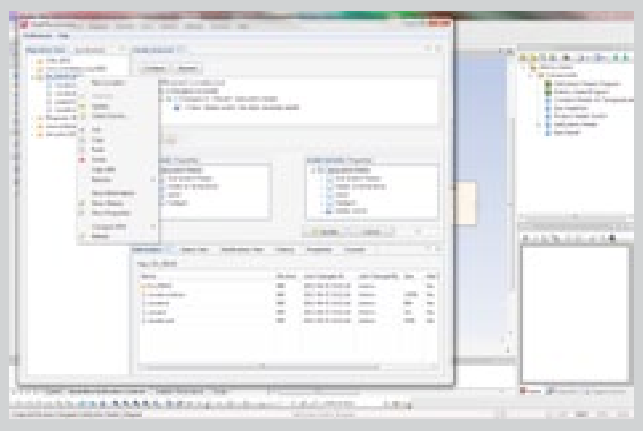
Particular emphasis has been given to the Papyrus UML Editor, which provides special features and is convenient to use. When working with UML models in Papyrus it is possible to lock parts of the model. This is signaled to those developers, who work on the same model and attempt to modify that model part which is blocked for those users until the lock has been released.

SPARX ENTERPRISE ARCHITECT

Besides the extended use of textual and graphical domain specific languages the Unified Modeling Language (UML) and its profiles play an important role throughout the complete development lifecycle. This effect is supported by the availability of affordable, customizable and high quality UML tools. The Sparx Enterprise Architect is a well-known UML tool used in various domains. The ModelBus® Adapter for Sparx Enterprise Architect connects this tool to other UML tools or to a complete ModelBus® based engineering processes and is capable of importing and exporting UML models. This way the tool can be used to edit UML models in Enterprise Architect which might be created by other UML tools or which are the result of a model transformation.

The adapter supports the simultaneous work on UML models by providing specific dialogs for identifying changes and potential conflicts. Thus, developers can keep track of the changes applied to the model, either by team members or by themselves. Additionally, the adapter supports all UML diagrams offered by Sparx Enterprise Architect and exports them to or imports them from the Eclipse MDT Papyrus format, so graphical information captured in the diagrams will be preserved. Using the ModelBus® Adapter for Sparx Enterprise Architect offers a new way of interoperability between tools. It is now possible to seamlessly exchange UML models with other tools. There is no need to bridge between Sparx Enterprise Architect and the various existing XMI dialects or tool proprietary formats supported by other tools anymore.

ModelBus® Adpater for Sparx



At the same time, it offers great opportunities for automation of development processes benefiting from the ModelBus® tool integration framework. Thus, an automatic processing like code generation or model transformation can be performed additionally. A complete tool environment, chaining various different tools together, can be created with the help of the other available ModelBus® tool adapters and by creating new ones, e.g. for in-house tools. The ModelBus® adapter for Sparx Enterprise Architect offers great user experience as it performs very fast imports and exports of models and comes with custom-made user interface extensions for iterating on changes in the model and for presenting ModelBus® notifications.

MATLAB SIMULINK

MATLAB Simulink is a widely used tool for modeling, simulating and verifying multi-domain dynamic systems. It provides a platform for graphical modeling from a customizable set of block libraries, and provides mechanisms for model simulation and analysis. The ModelBus® adapter for Simulink enables MATLAB Simulink to connect with other tools by allowing seamless exchange of its native models as EMF models. Thus, through the adapter models developed in Simulink can be exchanged, transformed and visualized in other modeling tools and vice-versa. Furthermore, the adapter equips a modeler to synchronize, compare and merge local MATLAB Simulink models with models remotely created and/or modified. Additionally, the adapter furnishes Simulink to participate in the ModelBus®-based systems engineering process and provides access to ModelBus® services such as traceability, model-to-model transformations and process orchestration.

The ModelBus® Adapter for Simulink offers a new way for interoperability among tools. It is now possible to seamlessly exchange native MATLAB Simulink models with other tools without the need for a specialized bridge between Simulink and the proprietary model formats of the various tools. The adapter also offers the possibility for the automation of the development process by providing access to code generation, model transformation and the process orchestration services and capabilities of the ModelBus® tool integration framework. By assembling various different tools, a complete tool environment can be created with the help of the other available ModelBus® tool adapters.

The ModelBus® Adapter for Simulink performs imports and exports of models quickly and comes with a user interface that allows the modeler to synchronize, compare and merge local and remote models. These, along with the notification mechanism of the ModelBus® framework, allows for a collaborative and interactive model development experience.

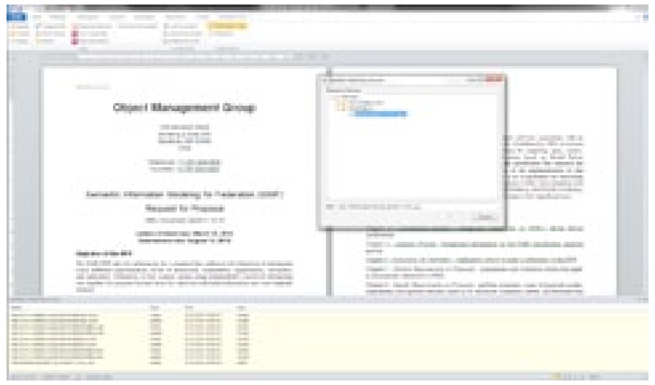
MODELBUS® ADAPTERS

GENERIC TOOL ADAPTERS CAN BE TAILORED TO MEET INTEGRATION AND PROCESS REQUIREMENTS

MICROSOFT OFFICE

It is a well-known secret that Microsoft Office tools are extensively used in software and systems engineering. Word documents are for example used to write specifications and Excel sheets to capture requirements. Unfortunately, the integration with the rest of the development tools is poor and mostly targeted on the generation of documents. A similar situation exists with Microsoft Visio. It is often used to draw system design diagrams because of its straight user interface which does not require the user to be an UML expert in order to simply draw some boxes and lines.

ModelBus® Adpater for Microsoft Word



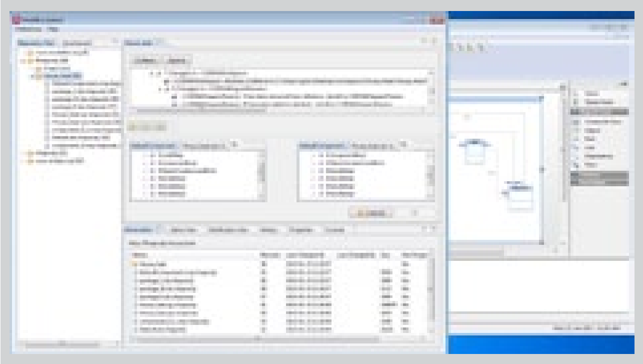
The ModelBus® adapter for Microsoft Office extracts knowledge from existing documents and tables into respective models and allows the processing of those models in ModelBus® and other tools respectively. Of course, this becomes easier when the input documents and tables are well structured. On the other side, the generation of tables and documents is supported as well. The integration into the user interface of Microsoft Office is achieved by using the Add-in mechanism. The ModelBus® adapter for Microsoft Visio allows the import and export of Visio files to ModelBus®. In particular, by using only the simple UML stencil of Visio, the adapter for example creates respective Papyrus/UML models, which also preserves the diagram information. In this way, Visio files can be perfectly used as early sketches of system design and refined by experts in respective tools later on.

IBM RATIONAL RHAPSODY

Systems engineering typically involves a high number of specialized tools ranging from quality control to safety design. Furthermore, participating engineers are located in different organizational units or companies. To overcome these interoperability issues, languages like UML and SysML were built. However, in reality exchanging models with team members often is a major problem. The ModelBus® adapter for IBM Rational Rhapsody overcomes this challenge as it allows you to share UML and SysML models - based on the ModelBus® infrastructure - with other team members who might be working with other tools. The adapter is capable of exchanging models. The ModelBus® adapter for IBM Rational Rhapsody integrates the system engineer's work, experience, and knowledge into a model-driven development environment. The adapter helps developers to benefit from functionalities offered by Rhapsody at various locations in the development process even if a different tool regime is already in place. Seamless exchange of Rhapsody models with other tools used for different jobs increases the productivity of the development teams. In this way, IBM Rational Rhapsody can easily complement other UML/SysML tools including open source tools.

This ModelBus® adapter is perfectly integrated into the tools' user interfaces and provides a fast import and export of models including diagram information. While using the ModelBus® infrastructure the engineers can benefit from the rich set of functionality offered by the ModelBus®, including traceability, consistency checks, model transformation or document generation. The other way round the ModelBus® adapter for IBM Rational Rhapsody enables you to seamlessly add the specific features provided by Rhapsody (e.g. simulation of UML models, generation of code) in any existing development process.

ModelBus® Adpater for Rhapsody



IBM RATIONAL DOORS

The elicitation and management of requirements become more and more important and highly specialized as well as customizable tools are being used. IBM Rational Doors is nowadays a wide-spread and often used tool for managing the vast amount of requirements gathered by building complex systems. By investing much effort in creating a huge range of requirements, specification is integral in order to use this knowledge in later development phases. Working on requirements in other tools than the original one and, particularly, automating their processing is critical for the improvement of the productivity of development teams. To achieve this goal, it is vital to retrieve information from Doors database and, then, to provide this information to engineers and system engineers in a way they can make use of it. For example, the requirements could be mapped to test models being processed by respective test tools and where test engineers can derive new test requirements or test objectives.

The ModelBus® Adapter for IBM Rational Doors allows to extract requirements and to store them in the ModelBus® repository. There, they can be analyzed with specific tools and traceability links can be established. Those requirements can also be transformed into different formats like the SysML requirements type. Alternatively, the ReqIF requirements format can be used. The inclusion of Doors into a ModeBus® based development process can be realized in different ways which utilizes the user interface, the requirements analysis (via scripts) or the requirements database capabilities of Doors.

TRAC – ISSUE TRACKER

An important asset used in development teams for keeping track of defects or change requests is an issue tracker like Trac. Trac provides an issue database with a web-based user interface and a Wiki in advance. A change request can be filed (e.g. by end-users) to track and, in the course of such a change, to request one or more developers to apply changes to the system design or system implementation and to document their changes in Trac. This typically involves the usage of specific tools, especially in big development projects. The variety of tools used for that purpose increases when different organizations work together. Exchanging information about the change request and its management among all stakeholders and developers is crucial for an immediate and appropriate feedback and, moreover, for quality and project control.

The ModelBus® Adapter for Trac extracts the data and transmitted into the ModelBus®. From there the extracted issues can be further processed by different tools. Sometimes, Trac is also used for gathering requirements. In this case, the Trac Adapter is paired with a transformation which converts Trac issues into the ReqIF format that can then be pushed into other tools requirements like ProR. A second option is to treat the Trac issues as change requests and to allow further processing in other tools. In this case, the ModelBus® adapter for Trac can be paired with a transformation and the OSLC-CM bridge which allows to push these issues in the IBM Rational Team Concert for example.

MODELBUS® SERVICES

The true power of automation lies in the ModelBus services. These services are automatically executed when needed and undertake otherwise tedious tasks. The ModelBus® framework manages the appropriate execution of ModelBus® services.

One of the promises of model-driven engineering is the automatic generation of relevant documents out of models. However, oftentimes legally binding information, such as requirements specification or system acceptance test reports, is kept solely in documents. Similarly, in many domains source code is still the primary artifact for actually implementing the system behavior. It is vital to have source code compiled in order to build executable binaries, which could be flashed on an embedded device for example. But similar to documents, the source code is actually representing the content of the models and can therefore be generated automatically. ModelBus® services allow the automatic generation of such documents via the models. Thereby, all relevant information regarding a system development project is accessible through a standard mechanism.

Both, the generation of documents and the generation of source code, can be achieved with ModelBus® generation services. There is a number of predefined templates which can be used in order to create a standardized document and they can be worked on with the ModelBus® Adapter for Microsoft Office. Manual changes, applied to the document, can be pushed back to the model in this way. Similarly, standardized code generators can be used e.g. code generators provided by standard tools connected by a Service Adapter to the ModelBus®. But of course custom made code generators can be used as well.

»All relevant information regarding a system development project is accessible through a standard mechanism«

ANALYSIS AND SIMULATION

Model-driven engineering and in particular the usage of ModelBus® has the big advantage that all relevant information regarding a system development project or even all information that has been created within the whole company is accessible through a standard mechanism. This allows generic analysis tools to work on all models in a standardized way. With ModelBus®, special services are applicable to all models and, in particular, can analyze dependencies between models and the traces between mode elements in order to collect and condense the information regarding a certain aspect. In this way, ModelBus® services can constantly create analysis reports including all relevant models in the development process.

The execution of simulation is another area where ModelBus® service can help to increase the degree of automation in a development process. Modeling services are in particular useful when working with long lasting tasks like simulation runs. By utilizing ModelBus® simulation services, the developer and his work stations can be relieved from executing certain simulation tasks which they can be triggered automatically as soon as new versions of simulation models are available.

MODEL TRANSFORMATION

One important aspect of model-driven engineering is the transformation of models. A couple of languages and transformation engines have been developed for that goal and general purpose languages like Java are being used in this context as well. ModelBus® transformation services allow the automated transformation of any model coming from any tool as soon as a new version arrives or a specific request has been made. This relieves the developer of executing the transformation by himself. The ModelBus® transformation service leverages different modeling languages like QVT and ATL, but it can be extended to work with any custom-made model transformation.

YOU CAN’T MANAGE
WHAT YOU DON’T MEASURE

MODELBUS® ENGINEERING SUPPORT

While ModelBus® is the platform for interoperable system engineering, the ModelBus® engineering support components extend the power of ModelBus®. They build upon basic ModelBus® features like model repository with versioning, dependency resolution or user management.

MODELBUS® ENGINEERING SUPPORT PROVIDES A SIGNIFICANT ECONOMIC BENEFIT BY DECREASING THE PRODUCT DEVELOPMENT COSTS.

METRINO

Metrino is a tool to support the validation and quality assurance of models and can be used as an independent tool or in combination with ModelBus®. It enables you to generate and manage the metrics for domain specific models and allows you to automatically derive metrics from a meta-model based on an extensible set of rules or to define custom metrics for it. The metrics can be applied to any model, which conforms to the meta-model they have been generated for.

Metrino analyzes and verifies the attributes of the artifacts including complexity, size and well-formedness. Furthermore, the tool offers different capabilities to present and visualize the metric's computational result, e.g. in tabular way and kiviat diagram. These results can be analyzed over time, since the tool stores results of multiple evaluations.

The overall goal of Metrino is to improve each individual artifact as well as the complete system information and to assure the quality of the final software-based system. Metrino can be employed in all process steps and can be applied to all models stored in ModelBus®.

»Requino allows the user to synchronize, compare, merge and control local and remote requirement models in an interactive or collaborative environment«

TRACEINO

A key functionality in achieving a high productivity with model-driven engineering is the traceability, which means to know exactly all the relationships, so called trace links, between the work products created during the development lifecycle. The more detailed the information about the links is the more benefit it can bring. Therefore, those trace links shall cover the model element level – either within or across different models. In addition to that, trace links by themselves can have different semantics and directions and they can be created automatically, e.g. during model transformation, or manually, e.g. during requirements coverage assignment.

Traceino allows the definition of custom trace types or to use predefined trace types. In this way, the level of traceability can be adjusted, which is needed in the development process, so that an adoption of trace semantics is not required, which is provided by other tools. Traceino comes with a nice integration in several of the ModelBus® adapters and in particular Eclipse-based tools (e.g. ProR) and gives visual feedback as tables and graphs about trace links between the elements in models.

REQUINO

In today's world, the complexity of engineering products and the need to get them "first time right" is immense. The best practices in systems engineering dictate that development of a new product starts with requirements engineering. However, an error-prone or incomplete requirements engineering process is not beneficial as it introduces design inconsistencies that require additional time, effort and money to debug, redesign, verify, validate and test. In short, what the customer wants is not what the customer gets.

Requino is a model-driven requirements engineering methodology and tooling that offers a new and pragmatic way for mechatronic product development for small and medium enterprises (SMEs) as well as large enterprises. Requino is an installation-free browser-based light-weight tool that goes beyond the traditional text based tools and allows a requirements engineer to structure, categorize, reference, trace, control, verify and reuse requirements.

In addition to customer requirements, Requino allows the integration of requirements from directives and standards, ensuring the customer requirements are up-to-date in an ever-changing regulatory environment. In addition to requirements capturing, Requino also allows the user to perform systems analysis methods such as FMEA, QFD, FTA, complexity analysis and so on.

Furthermore, mechanisms for variability management, generation of new variants of mechatronic products, and import and export of requirements in the ReqIF standard are integrated into the tool. As part of the ModelBus® family of tools, Requino offers an array of repository services and a rich model development experience and environment.

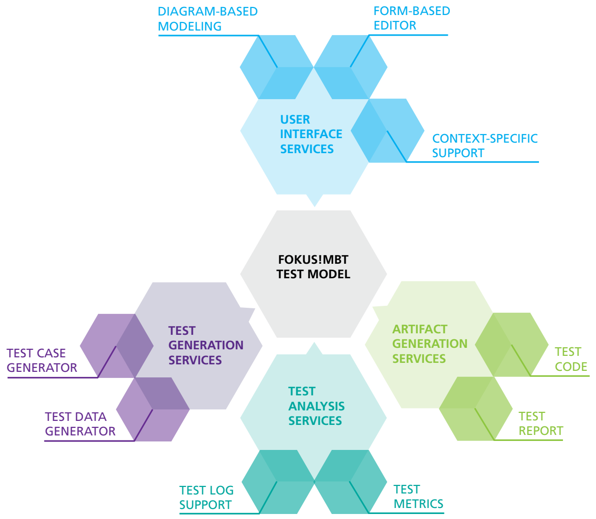
Thus, the user is able to synchronize, compare, merge and control local and remote requirement models in an interactive or collaborative environment. The application of the methodology and tooling is expected to provide a significant economic benefit by decreasing the product development costs. Requino will allow a product development team to develop more and better products with a shorter time-to-market while precisely addressing what their clients are asking for. In short, with Requino, what the customer wants is what the customer gets.

FOKUS!MBT

Fokus!MBT is an integrated test modeling environment that guides the user along the methodology of Fokus!MBT and thereby simplifies the creation of the underlying test model.

A test model includes test relevant structural, behavioral and methodical information. By formalization, the tester's knowledge can be machinably preserved as well as evaluated and exploited at any time – for instance to generate further test-specific artifacts, such as test cases and test scripts. Another benefit of the test model is the possibility to visualize and to document test specifications. The modeling notation used by Fokus!MBT is the UML Testing Profile (UTP) specified by the Object Management Group. It is a test-specific extension of the Unified Modeling Language (UML), which is prevalently used in the industry. This enables testers to rely on the same language concepts as system architects and requirement engineers and, therefore, thereby overcome communication problems and to support the mutual comprehension.

Fokus!MBT is based on the flexible Eclipse RCP platform, the Eclipse Modeling Framework (EMF) and Eclipse Papyrus and it perfectly works in the ModelBus® environment. The validation of the system under test concerning its requirements is the main target of all testing activities. Thereby, the consequent and continuous traceability among requirements and test artifacts – especially among requirements and test cases – is indispensable, but not sufficient. Fokus!MBT takes a major step forward by integrating the test execution results into the test model's inherent traceability network. This establishes a consistent traceability network between requirement, test case, test script and test execution results, making conclusions about the coverage of the particular requirement or the test progress itself immediately calculable. Furthermore, the visualization of the test execution results allows a detailed analysis of the test execution flow to preprocess and, ultimately, to evaluate the test results.



EDITORIAL NOTES

Fraunhofer Institute for
Open Communication Systems FOKUS
System Quality Center – SQC
Kaiserin-Augusta-Allee 31
10589 Berlin
Germany

Phone +49 30 3463-7278
tom.ritter@fokus.fraunhofer.de

www.fokus.fraunhofer.de/en/sqc
www.modelbus.org

Institute Director
Prof. Dr. Manfred Hauswirth

Editors-in-Chief
Dr. Tom Ritter, Ronny Meier, Laura Gertken

Editorial Team
Dr. Tom Ritter, Ronny Meier, Laura Gertken

Layout, Typesetting, Photo Editor
Benny Adler

Photo acknowledgement
iStockphoto.com, Fraunhofer FOKUS

Project Logos

MODELBUS

METRINO

REQUINO

TRACEINO

Fokus!
MBT