



Automatic Generation of IEC 61499 Applications based on Workflow Models

M. Plasch, **G. Ebenhofer**, M. Hofmann
ETFA 2012 – 4DIAC User's Workshop

LEADING
INNOVATIONS

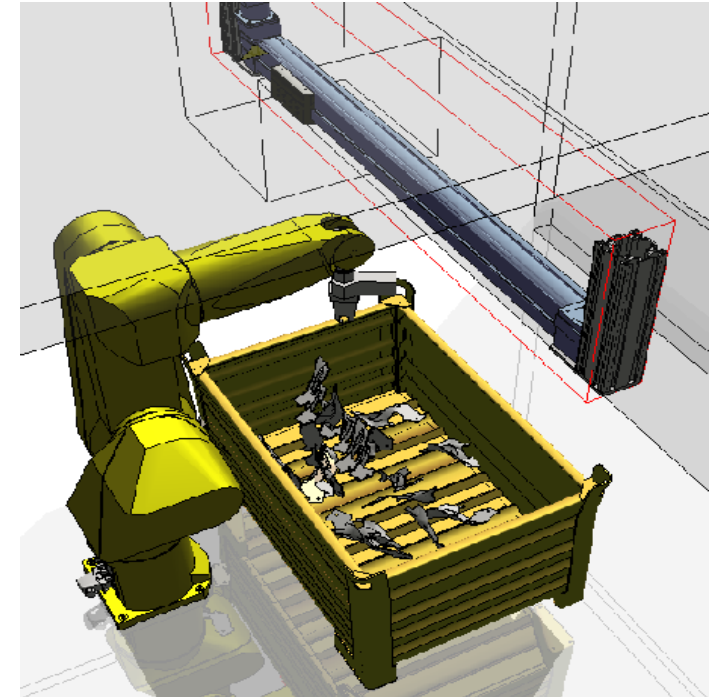
WWW.PROFACTOR.AT

Overview

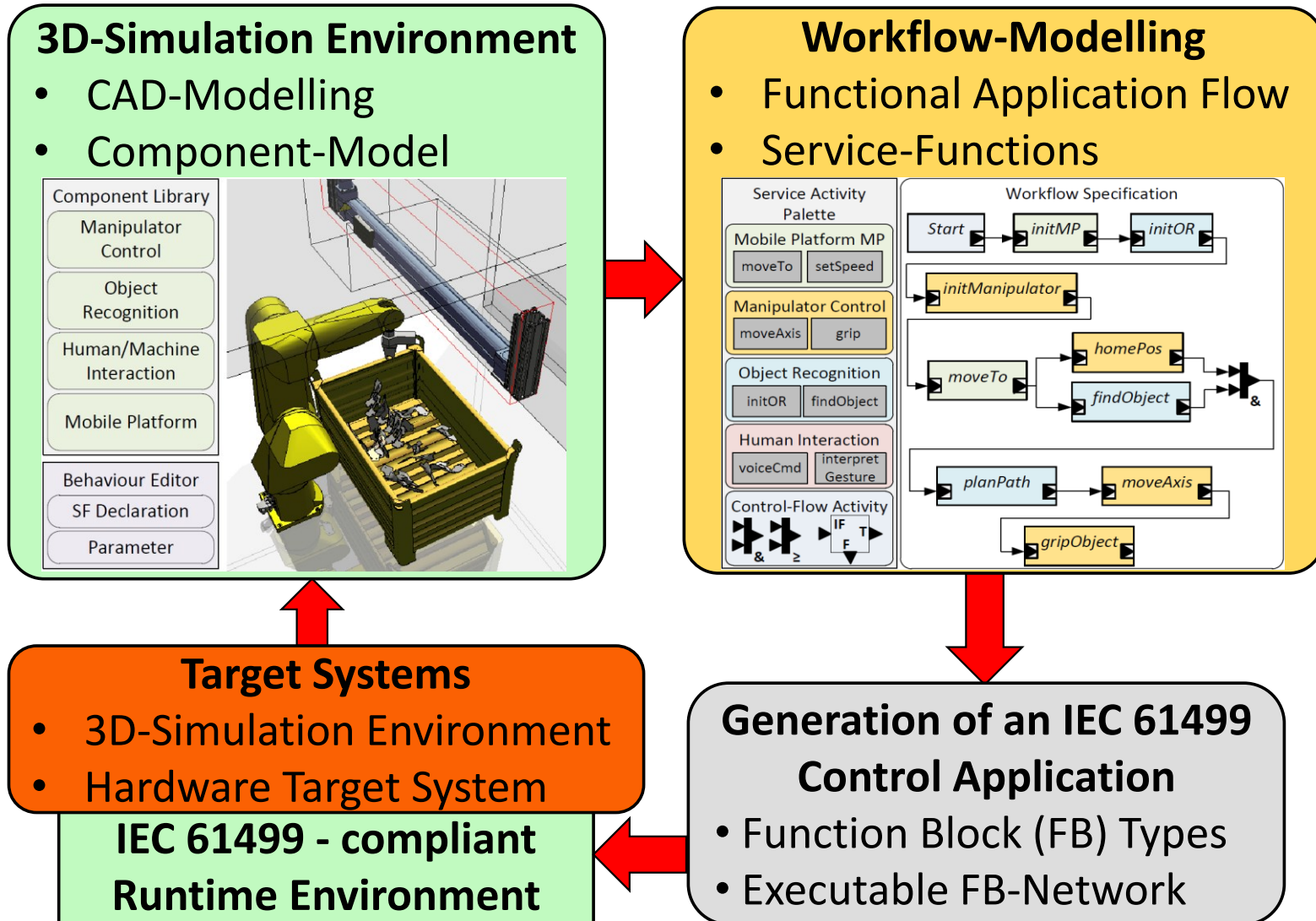
- Motivation and Aim of the Work
- Overall Concept
- Workflow Modeling
- Generation of the Supervisory Control Application (IEC 61499)
- Communication Concepts
- Resulting System Architecture
- Example Application

Motivation and Aim of the Work

- Rising need for flexible, modular robotic systems in production industry
- Fast (re)-configuration; preferably user-friendly programming methods and tools
- Example: Bin-Picking application
- **Aim:** Simplified programming of a supervisory control application (according to IEC 61499)
 - Graphical Workflow Modeling
 - Generation of the control application through a code generator



Overall Concept



Background: Workflow Modeling

➤ Workflow Definition

- Activities in order to describe a process
- Workflow Management System coordinates execution
- Hardware/Software Resources

➤ Workflow Modeling Languages

- Business Process Execution Language BPEL – textual language
- Petri-Net based languages – graphical language
- Unified Modeling Language UML – Activity Diagrams

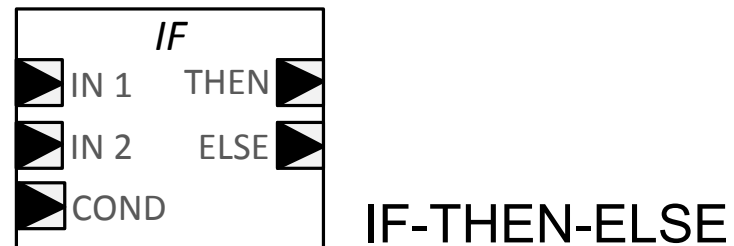
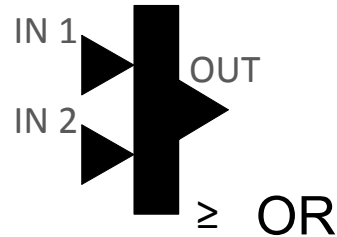
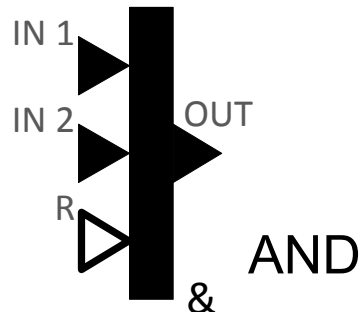
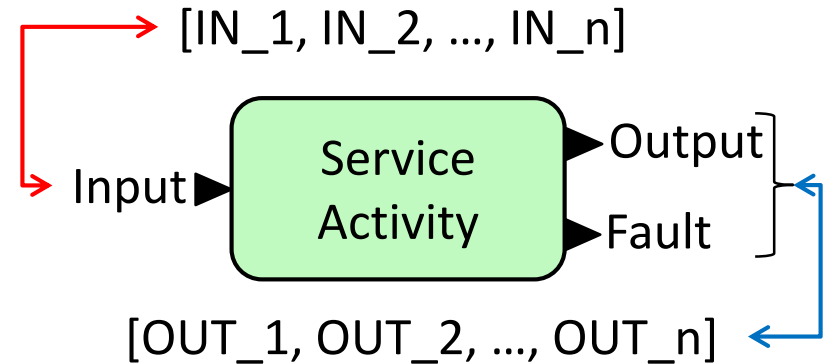
➤ Workflow Patterns

- Design guidelines for Workflow Modeling Languages
- Benchmark criteria

Development of a Workflow Modeling Language

➤ Language elements

- Service Activities
- Activity Connections
- Control-Flow Activities

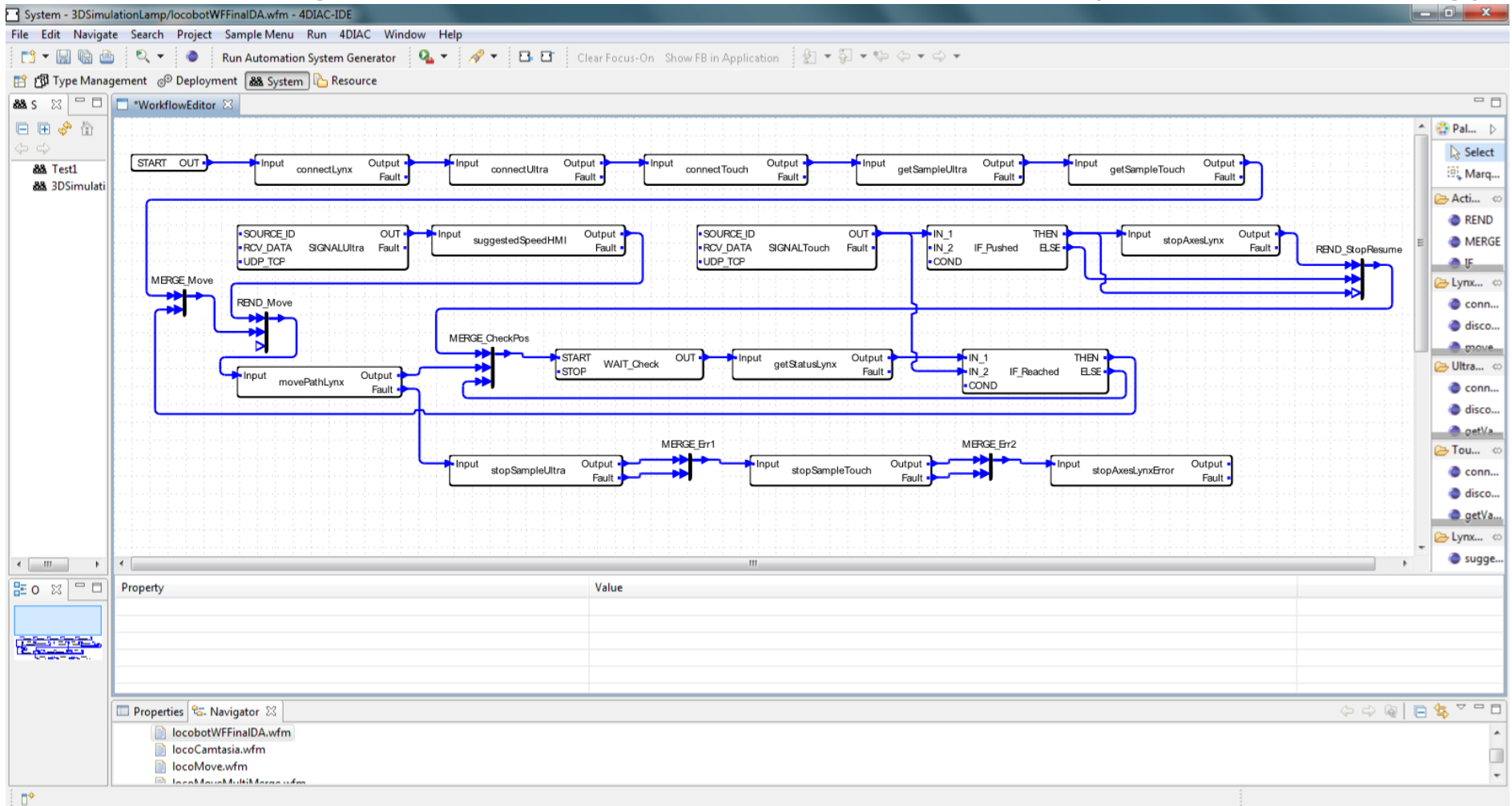


➤ Language Characteristics

- Combined control- and data-flow modeled by Activity Connections
- Parameter Mapping-Tables to model data passing
- Embedded Exception Handling

Implementation of a Graphical Workflow Editor

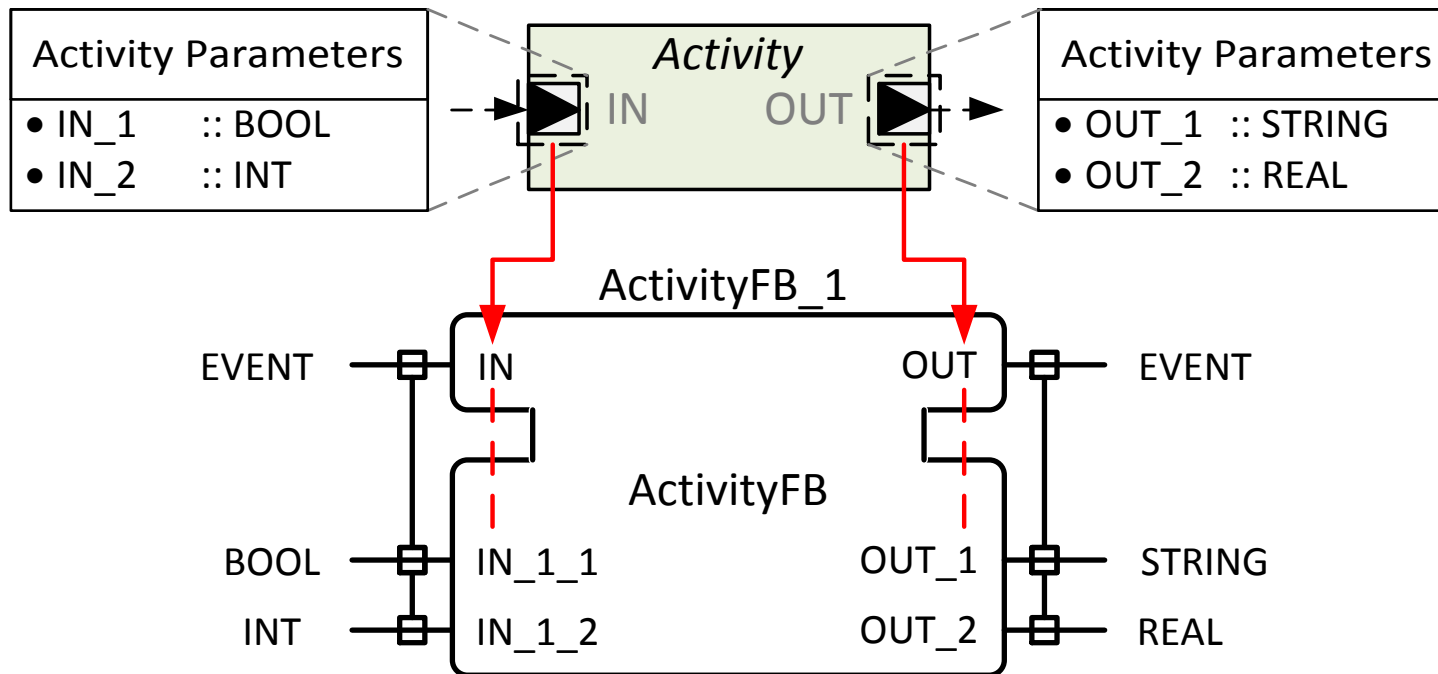
➤ 4DIAC Integrated Development Environment (www.fordiac.org)



Generation of the Supervisory Control Application (1)

➤ Step 1: Generation of a Function Block (FB) type library

- Represent Control-Flow- and Service-Activities
- Activity Interfaces \leftrightarrow FB WITH-Constructs
- Behavior according to FB base types



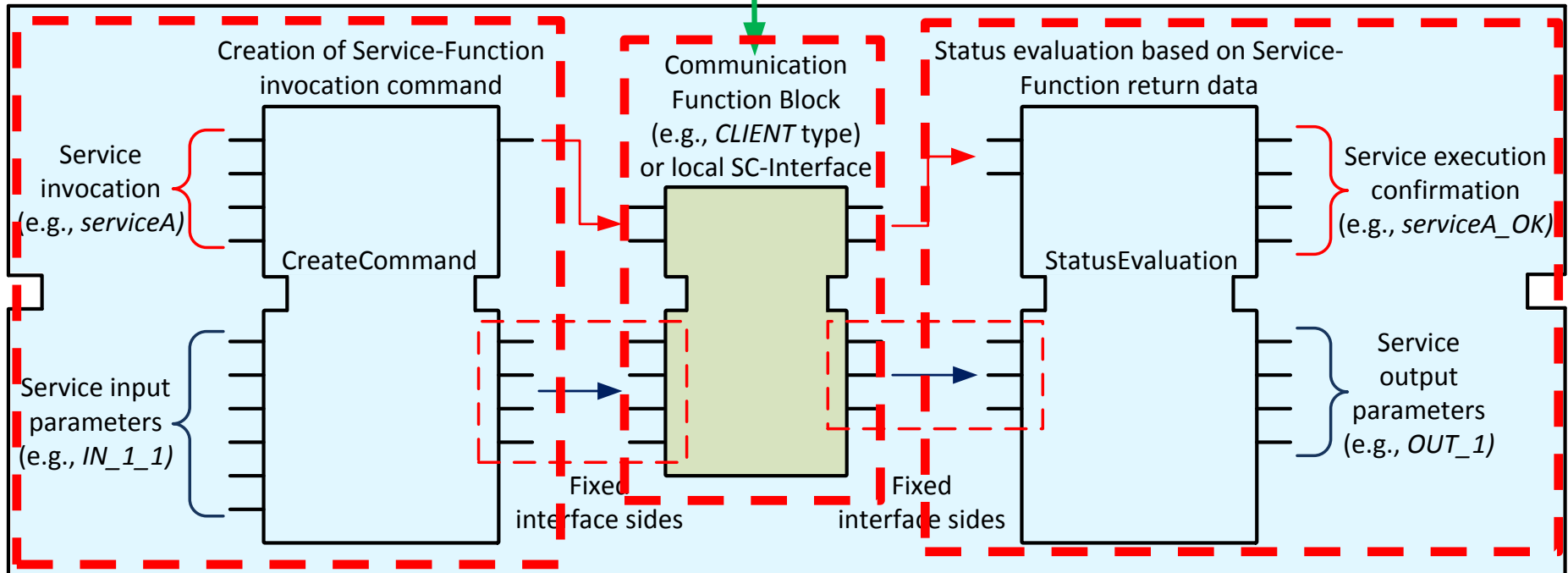
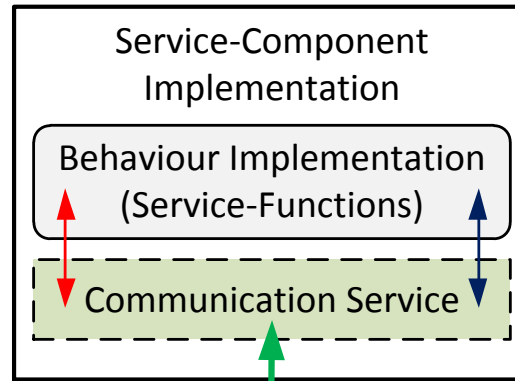
Generation of the Supervisory Control Application (2)

- **Step 2: Creation of FB instances**
 - Required instances according to Workflow Model

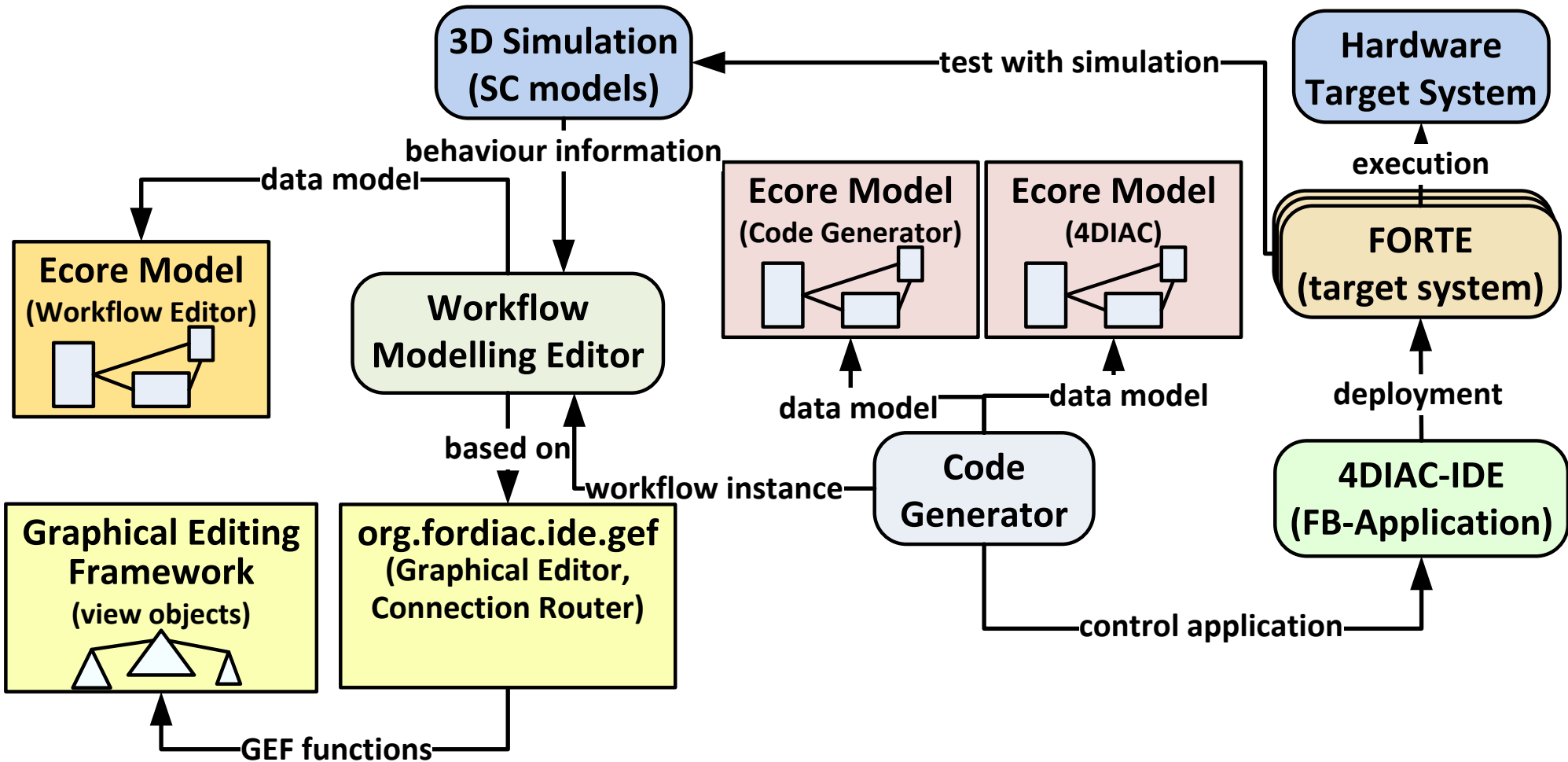
- **Step 3: Creation of Event- and Data connections**
 - Modeled Activity Connections and Parameter-Mappings
 - Data connection start- and end-points through WITH-Construct

- **Execution of the control application**
 - IEC 61499 compliant runtime environment
 - Alternatively combined with 3D-Simulation Environment

Interaction of Control Application and Service-Component



Resulting System Architecture



Example Application (Video)

- 5 Degrees-of-Freedom robotic arm
 - IEC 61499 Service Component
- Ultrasonic sensor, touch sensor from Mindstorms NXT
- Robot arm follows predefined path continuously
- Adaption of speed with ultrasonic sensor
- Interruption of movement with touch sensor
- Link: <http://www.locobot.eu/2012/09/12/model-based-application-programming/>

Results and Future Work

➤ Results

- Implemented graphical Workflow Editor
- Code Generator for IEC 61499 compliant control applications
- Appliance of the approach within the European funded research projects LOCOBOT (www.locobot.eu)

➤ Future Steps

- Hierarchical Workflow Structures
- Improved exception handling concepts
- Improved integration of Service-Components into the simulation model



Thanks for your attention!

Contact Speaker
Gerhard Ebenhofer
PROFACTOR GmbH
Im Stadtgut A2
4407 Steyr-Gleink, AUSTRIA
gerhard.ebenhofer@profactor.at
www.profactor.at