

OPC DAY
FINLAND 2024
21.11.2024

Digital Twin Implementation With FMI for Automation System

Tuomas Laine (Master of Science work)
Dr. Mika Karaila (thesis supervisor)



FINNISH SOCIETY OF AUTOMATION
SUOMEN AUTOMAATIOSEURA RY

SPONSORS:



BECKHOFF



Agenda

Use the FMI-standard to connect simulation models with an automation system

- The **FMI** (Functional Mock-up Interface)-standard is used to share dynamic simulation models of physical systems
- The goal was to create a **POC** to show how the standard can be used together with an automation system
- The simulation models were used to interact with **PLC-applications** running on the automation system, for example, the models were controlled by the PLC-applications and the model outputs were used as feedback for the applications
- The FMI-standard was used in combination with **OPC UA**, which provided bidirectional, real-time communication between the PLC-applications and simulations

Functional Mock-up Interface

- **What is FMI?**
 - **FMI** (Functional Mock-up Interface) FMI-standard defines an interface for sharing dynamic simulation models and is currently supported by 220+ simulation tools
 - Developed for **interoperability** between different simulation tools
 - Ensures models are **tool-agnostic**, portable, and re-usable
- **Key Features of FMI**
 - **FMUs** (Functional Mock-up Units): Model files compliant with the FMI standard
 - Can represent **individual components** or **entire systems**
 - **APIs** to enable seamless integration with simulation environments
 - Support for different interface types (**Model Exchange** and **Co-Simulation**)



Source: <https://fmi-standard.org/>

Functional Mock-up Interface

- **Key Interface Types**

- **Model Exchange (ME):** Provides equations and derivatives, relies on external solvers
- **Co-Simulation (CS):** Includes its own solver, runs simulations with time synchronization

- **Why Use FMI?**

- Facilitates **cross-tool compatibility** in system modeling and simulation
- Reduces **integration effort** in multi-tool environments
- Enhances **model reusability and portability** across platforms



Model Exchange

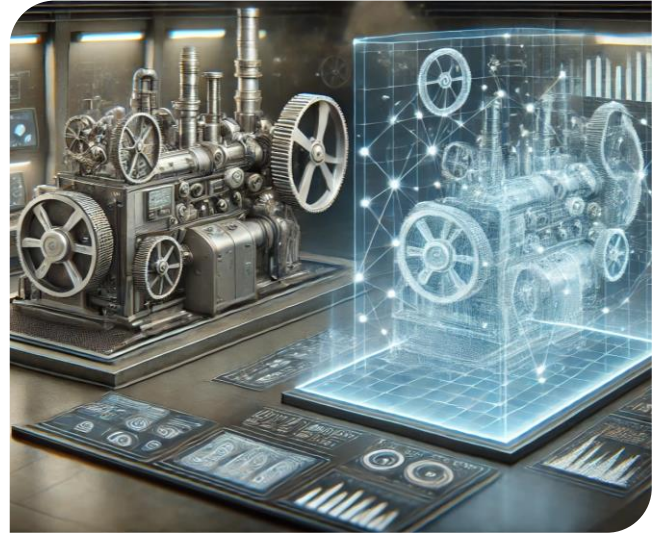


Co-Simulation

Source: <https://modelon.com/blog/fmi-functional-mock-up-unit-types/>

Digital Twins

- **What is a Digital Twin?**
 - A **virtual replica** of a physical system, process, or asset
 - Synchronizes with the **real-world system in real-time** using data from the physical system and simulations
 - Enables **monitoring, analysis, and optimization** of real systems
- **Key Components of a Digital Twin**
 - Digital model which models the physical system
 - Options for modelling:
 - **Machine Learning model**
 - **Physical model**
 - Bidirectional, **real-time communication** between physical and digital systems



Proposed implementation

- **The goal**

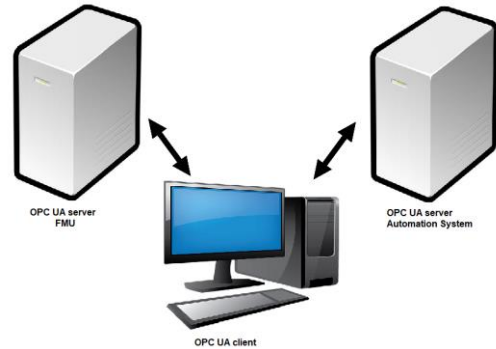
- Use FMUs as digital twins for PLC-applications running on an automation system
- Have bidirectional, real-time communication between the automation system and the FMU-simulations

- **Method used**

- Create FMUs to represent the physical system that could be controlled by the automation system
- Use OPC UA to connect the FMU-simulations with the PLC-applications
- The FMUs execute simulations according to real-time data received from the PLC-applications and simulation outputs are used to control the system

System Architecture

- **System consists of:**
 - 2 OPC UA servers
 - One for handling FMUs
 - One on the automation system
 - OPC UA client(s)
 - Used for connecting the servers
- **FMU-server:**
 - Reads the FMU-file, provides simulation data
- **AS-server:**
 - Included with the automation system
 - Provides data about the PLC-applications
- **OPC UA client(s)**
 - Transfers simulation and control data between the two servers



OPC UA methods

- FMU-server uses methods for handling the FMUs
- The methods can be invoked by the clients that connect to the server
- **Methods:**
 - Import FMU
 - Reads the FMU-file, generates an information model to the Address Space according to it
 - Instantiate model
 - Creates an instance of the previously generated information model
 - Allows the user to create multiple instances, which can be executed simultaneously
 - Execute Simulation
 - Executes the FMU-simulation according to parameters given by the user

Address space and instance method call to “Execute simulation”

The screenshot shows the OPC UA Client interface. The 'Data Access View' window displays a table of data points:

#	Server	Node Id	Display Name	Value	Datatype	Server Timestamp	Statuscode
1	OPC UA FMU ...	NS2String1_In_k	k	1	Double	12.19	Good
2	OPC UA FMU ...	NS2String1_Param_const.k	const.k	-1	Double	12.19	Good
3	OPC UA FMU ...	NS2String1_Param_const1.k	const1.k	1	Double	12.19	Good
4	OPC UA FMU ...	NS2String1_Param_hysteresis_precy_start	hysteresis_precy...	false	Boolean	12.19	Good
5	OPC UA FMU ...	NS2String1_Param_hysteresis_ul_high	hysteresis_ul_high	1	Double	12.19	Good
6	OPC UA FMU ...	NS2String1_Param_hysteresis_ul_low	hysteresis_ul_low	-1	Double	12.19	Good
7	OPC UA FMU ...	NS2String1_Param_integrator.k	integrator.k	1	Double	12.19	Good
8	OPC UA FMU ...	NS2String1_Param_integrator_start	integrator_start	0	Double	12.19	Good
9	OPC UA FMU ...	NS2String1_Out_den	den	0	Double	12.19	Good
10	OPC UA FMU ...	NS2String1_Out_x	x	0	Double	12.19	Good

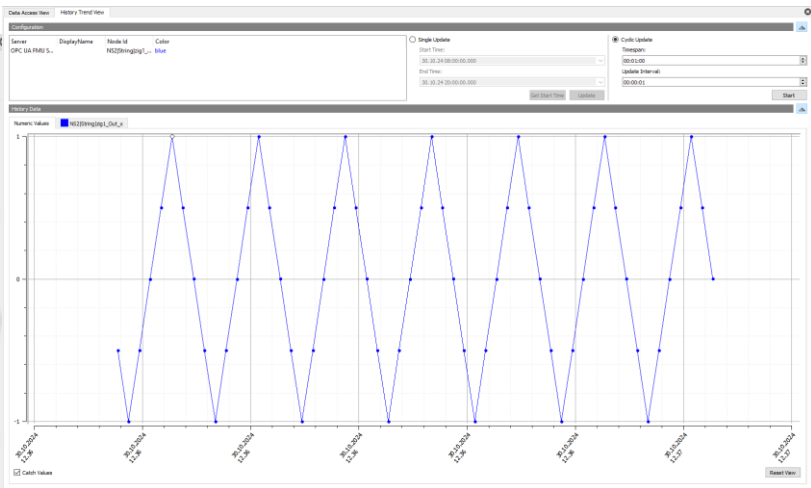
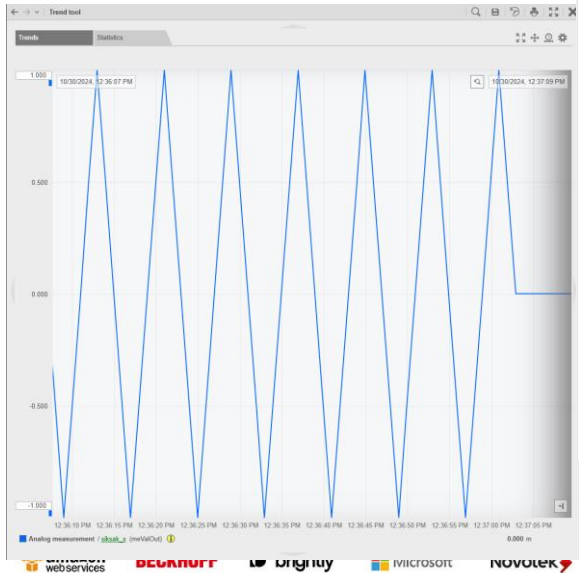
The 'Address Space' tree on the left shows the hierarchy: Root > Objects > FMU-Models > zig1 > Model-Instances > zig1 > Execute simulation. A red arrow points from this path to the 'Call Execute simulation on zig1' dialog box.

The dialog box 'Call Execute simulation on zig1' contains the following fields:

- Execute simulation (checked)
- Input Arguments table:

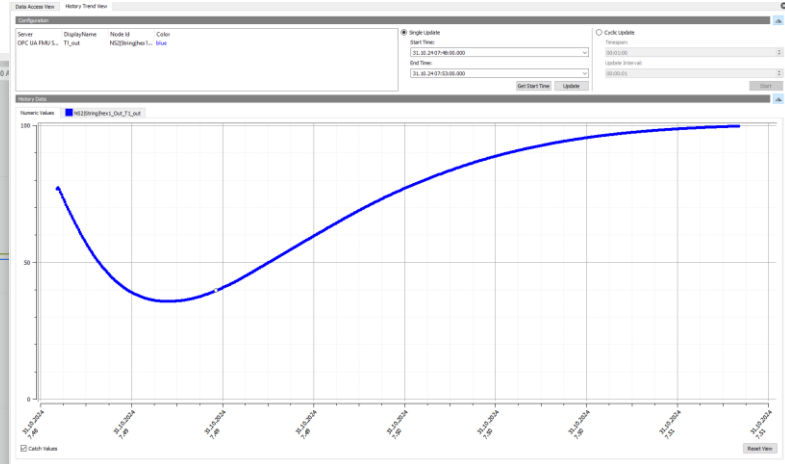
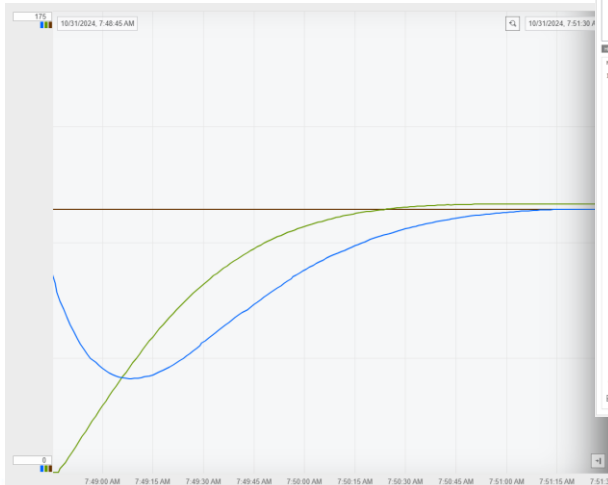
Name	Value	Data Type	Description
Stop time	[input field]	Float	Execution time(s), 0 infinite
Simulate Connectors?	<input type="checkbox"/>	Boolean	Checked = Simulate, Not checked = Don't Simulate
Step size	[input field]	Float	Give the step size of the simulation
Speed of Simulation	[input field]	Float	Give the multiplier for simulation speed
- Result field
- Call and Close buttons

Example: An alternating measurement level



Example: FMU-simulation controlled by PID

FMU-simulation result used as feedback for PID



Conclusion

- FMUs and their simulations can be used to interact with PLC-applications of an automation system
- The POC demonstrates how FMUs can use real-world data as inputs for the simulations
- The outputs of the simulations can be used to control a real-world system, providing bidirectional dataflow between the digital and physical world with OPC UA
- OPC UA acts as a suitable data transfer technology between the PLC-applications and FMU-simulations

Thank you!

Tuomas, please come to stage!

Detailed questions to actual man

Aalto publication archive: <https://urn.fi/URN:NBN:fi:aalto-202411217342>