Open-Access Interoperability Standards for Simulation and Test of Autonomous Systems

Martin Krammer

S1: Hitting the safety spot – but secured

www.v2c2.at

Overview

**Open-Access Interoperability Standards
for Simulation and Test of Autonomous Systems**

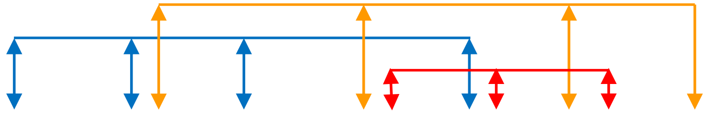
- Introduction
- Motivation & Need for Interoperability Standards
- Functional Mockup Interface (FMI)
- Distributed Co-Simulation Protocol (DCP)
- Modelica Association
- Conclusion

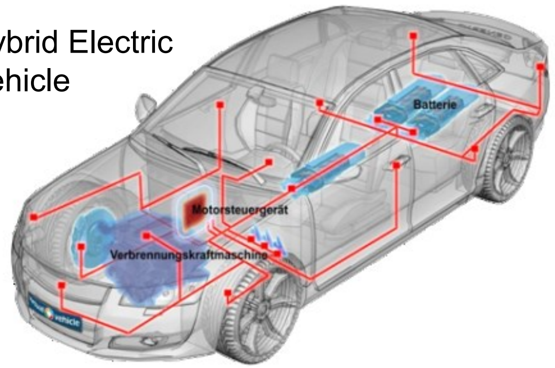
Motivation for Interoperability Standards

Multi-domain development

- Electric, electronic, thermal, mechanical,…
- Environment, infrastructure, traffic

Hybrid Electric Vehicle





Driver | Engine | Drivetrain | Hybrid Control Unit | E-motor | Battery | Cooling System

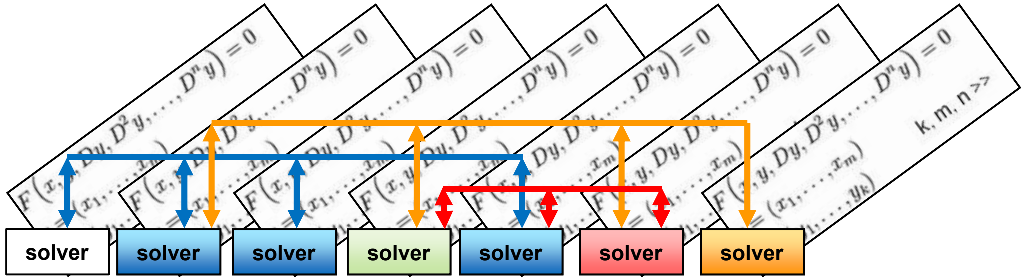
Multi-platform

- ASIC, ECU, PC, RT-system, cloud based computing, …

High diversity!

Multi-data

- Solvers, rates, steps, exchange formats and data structures

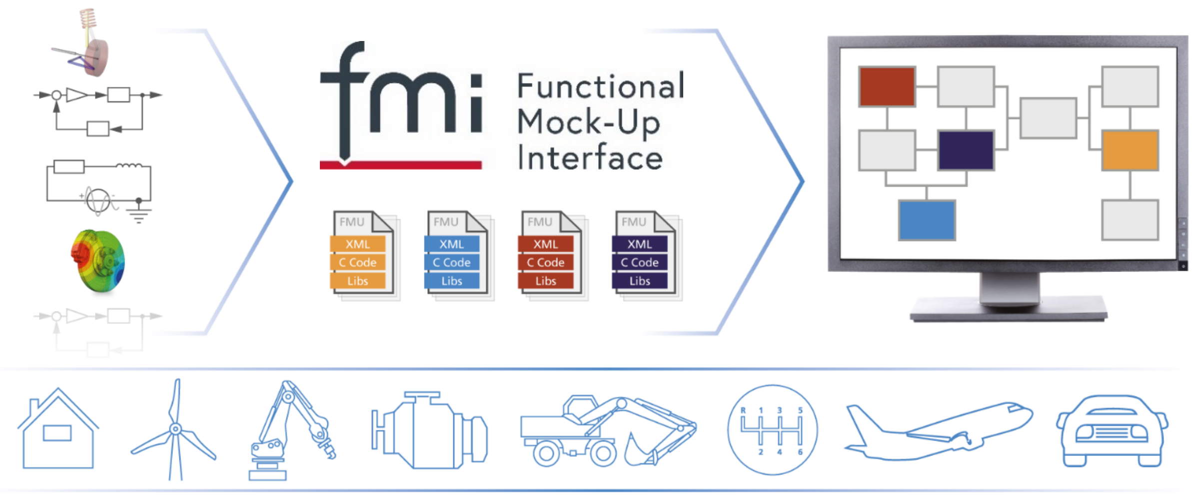


Multi-tool, multi-vendor

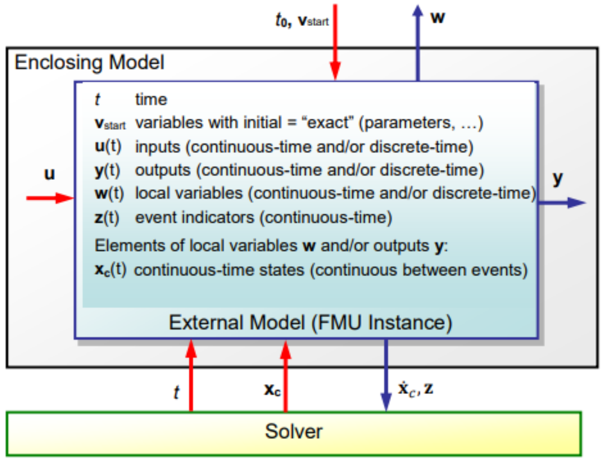
dSPACE, ETAS, BOSCH Technik fürs Leben, SIEMENS, esi get it right®, AVL

What is the Functional Mock-Up Interface?

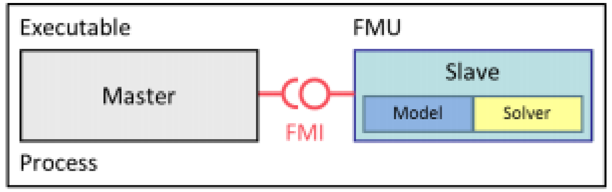
- Emerged out of ITEA 2 Modelisar project (2002)
- Developed as a *Modelica Association Project*
- https://fmi-standard.org
- FMI 2.0 released in 2014
- Supported by 100+ tools
- Focuses on positive system operation, intended functionality at the interface



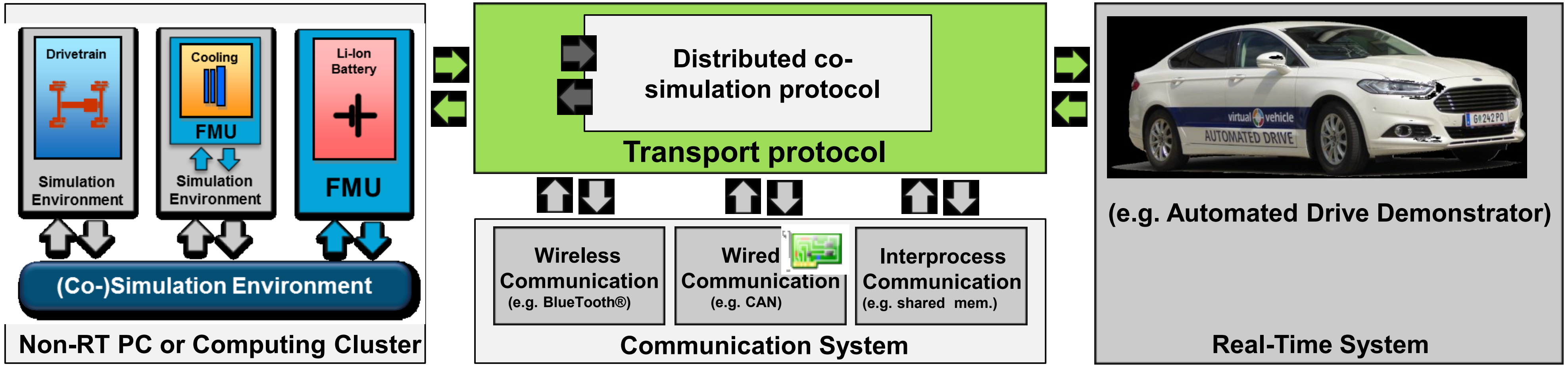
FMI for Model Exchange



FMI for Co-Simulation



FMI Technology Gap



FMI 2.0 specification p.93

In its most general form, a tool coupling based co-simulation is implemented on distributed hardware with subsystems being handled by different computers with maybe different OS (cluster computer, computer farm, computers at different locations). The data exchange and communication between the subsystems is typically done using one of the network communication technologies (for example MPI, TCP/IP). The definition of this communication layer is not part of the FMI standard. However distributed co-simulation scenarios can be implemented using FMI as shown in Figure 9.

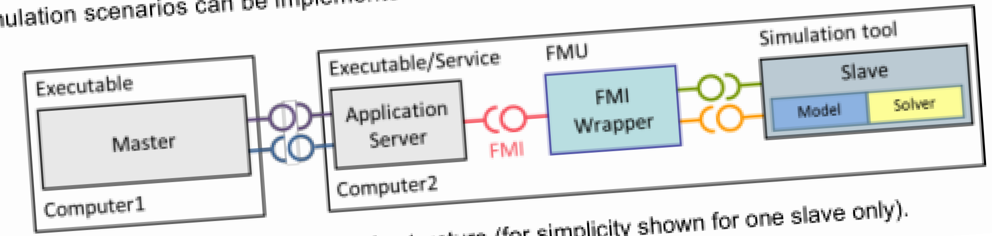
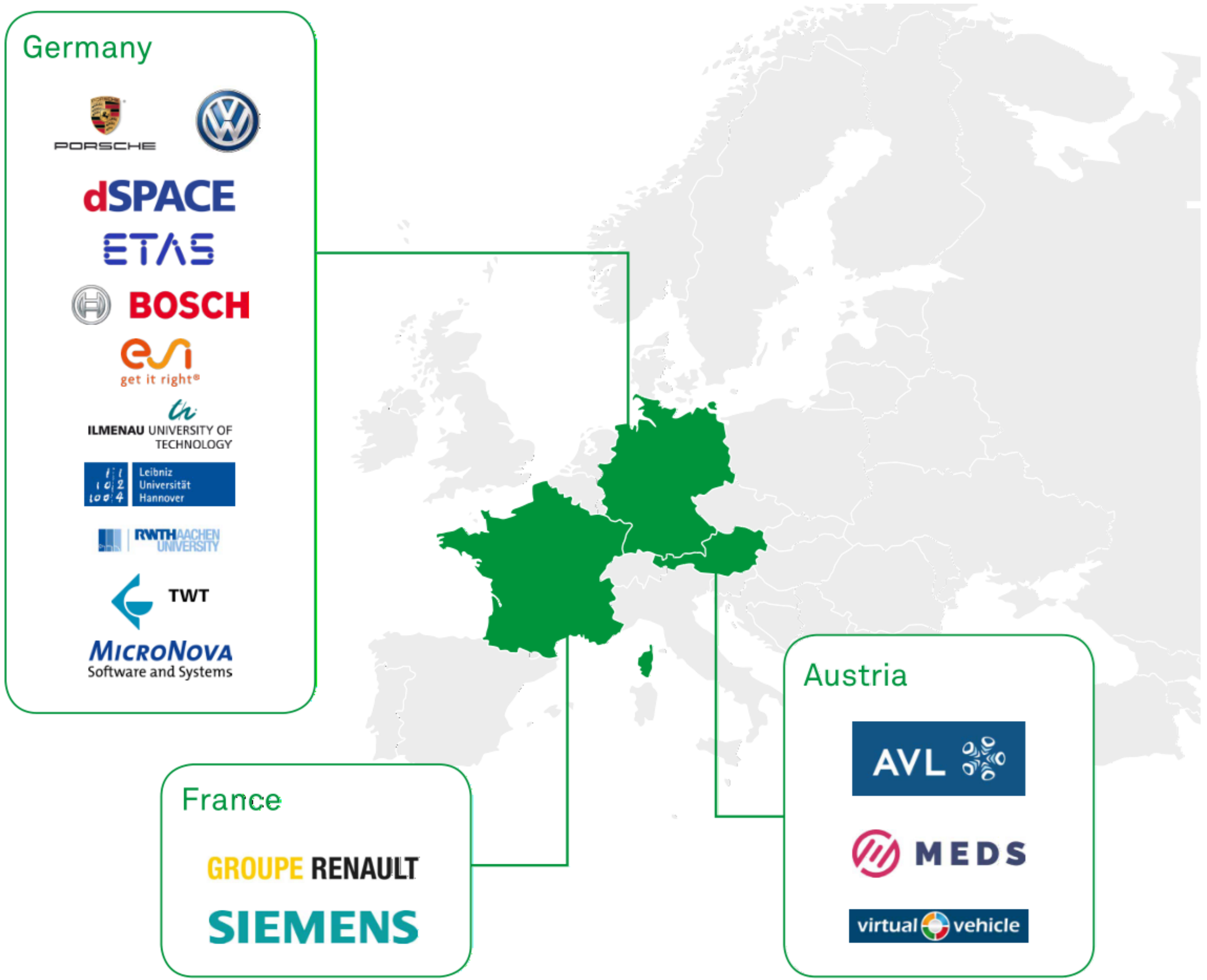


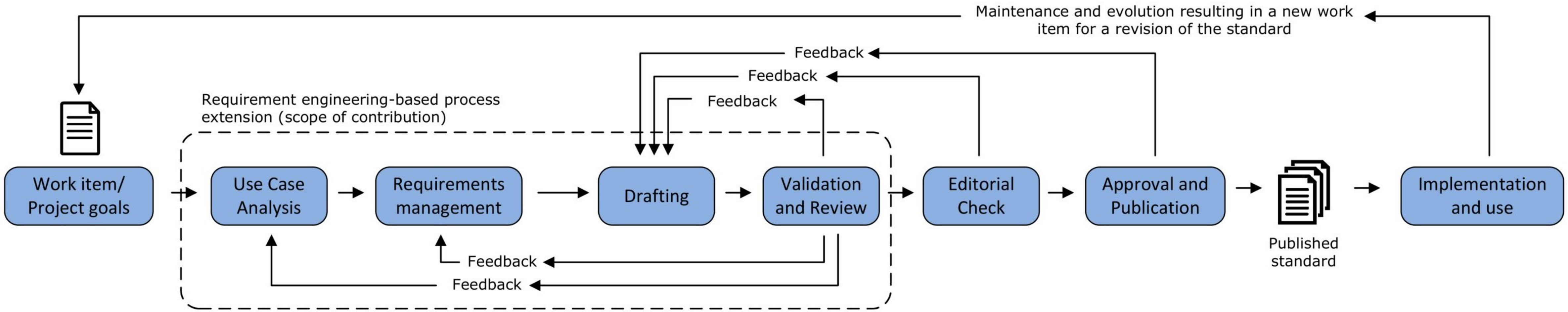
Figure 9: Distributed co-simulation infrastructure (for simplicity shown for one slave only).

The ACOSAR project

- *Advanced Co-Simulation Open System Architecture*
- ITEA 3
- Duration: 09/2015-08/2018
- Costs: 8,123k€
- Effort: 60 PY

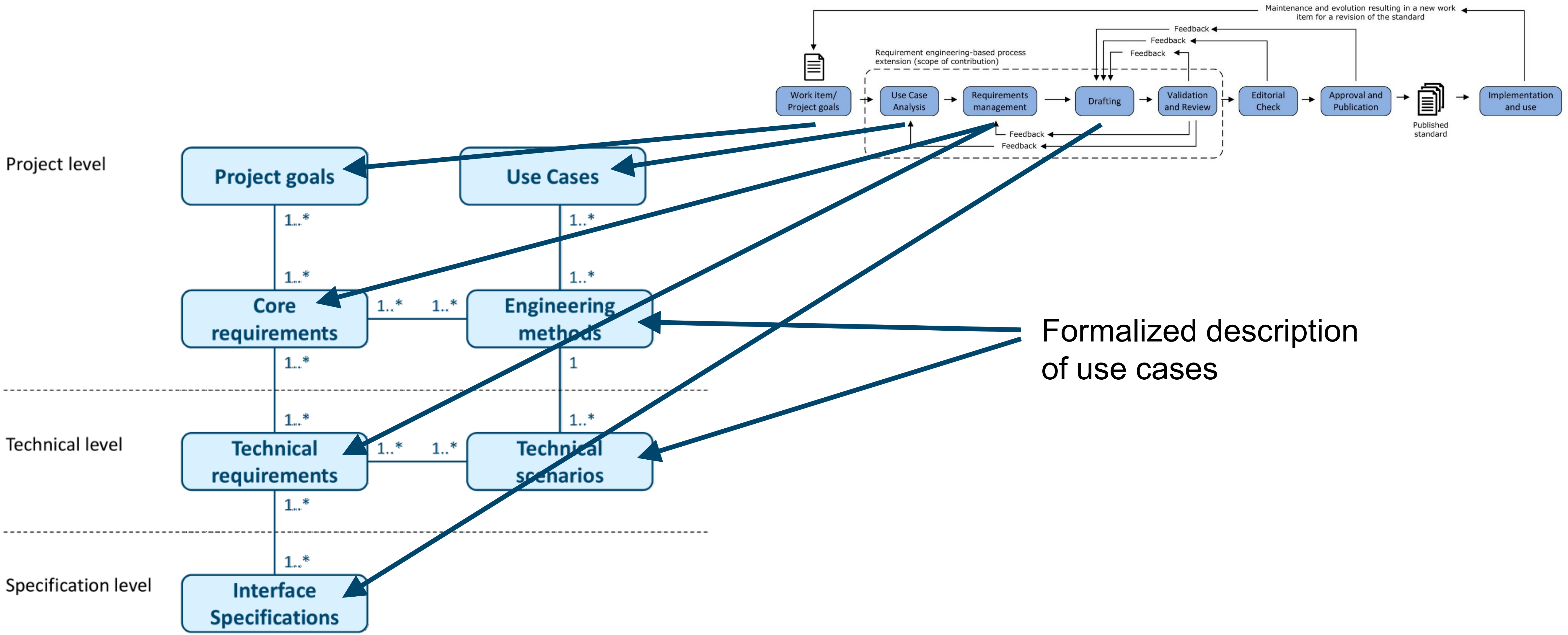


How to Develop Requirements for the Intended Standard?



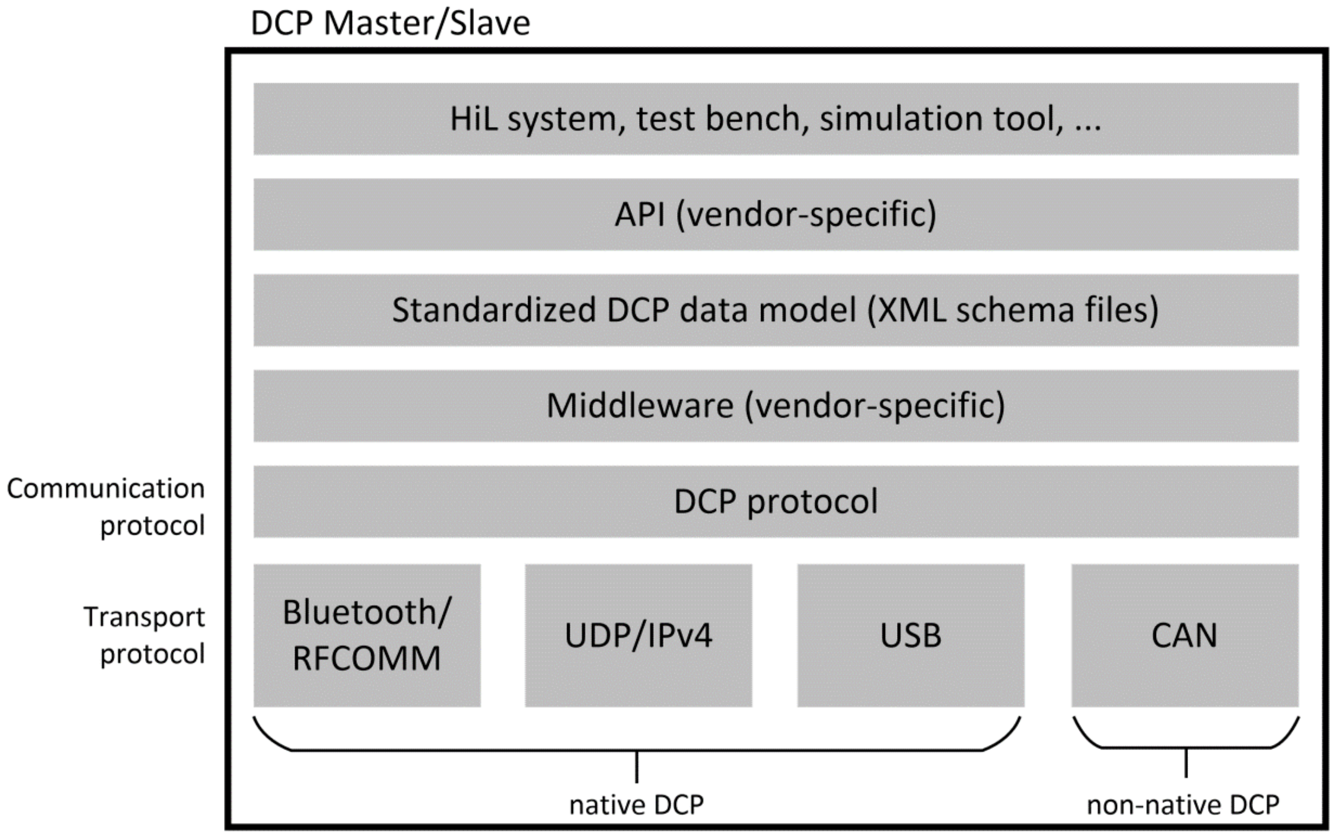
Krammer, Martin, Nadja Marko, and Martin Benedikt. 2018. "Requirements Engineering for Consensus-Oriented Technical Specifications." In *Proceedings - 2018 IEEE 26th International Requirements Engineering Conference, RE 2018*, 315–24. Banff, Alberta, Canada. https://doi.org/10.1109/RE.2018.00039.

Process Artifacts

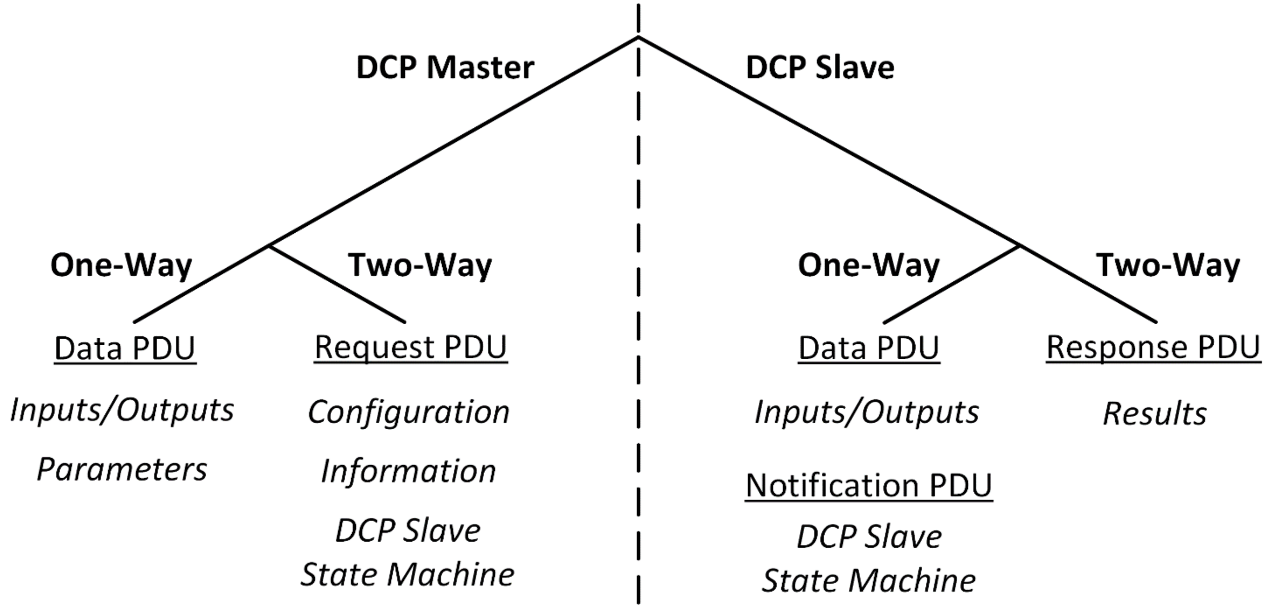


Krammer, Martin, Nadja Marko, and Martin Benedikt. 2016. “Interfacing Real-Time Systems for Advanced Co-Simulation - The ACOSAR Approach.” In *CEUR Workshop Proceedings*, edited by Catherine Dubois, Francesco Parisi-Presicce, Dimitris Kolovos, and Nicholas Matragkas, 1675:32–39. Vienna, Austria: Dubois, Catherine Parisi-Presicce, Francesco Kolovos, Dimitris Matragkas, Nicholas.

DCP Communication Architecture



Protocol Data Units



					DCP Frame																			
					type_id	pdu_seq_id	resp_seq_id	sender	receiver	param_id	data_id	pos	target_vr	source_vr	source_data_type	transport_protocol	state_id	numerator	denominator	steps	op_mode	error_code	log_category	
Protocol Data Unit (PDU)	Control	Request	Configuration (CFG)	CFG_set_time_res	0x20	y			y									y	y					
				CFG_set_steps	0x21	y			y		y									y				
				CFG_config_input	0x22	y			y		y	y	y		y									
				CFG_config_output	0x23	y			y		y	y		y										
				CFG_config_clear	0x24	y			y															
				CFG_set_target_network_information	0x25	y			y		y					y								
				CFG_set_source_network_information	0x26	y			y		y					y								
				CFG_set_parameter	0x27	y			y						y									
				CFG_set_config_tunable_parameter	0x28	y			y	y		y			y									
				CFG_set_param_network_information	0x29	y			y	y						y								
				CFG_set_logging	0x2A	y			y														y	y
				CFG_set_scope	0x2B	y			y		y													
			State change (STC)	STC_register	0x01	y			y								y				y			
				STC_unregister	0x02	y			y								y							
				STC_configure	0x03	y			y								y							
				STC_initialize	0x04	y			y								y							
				STC_run	0x05	y			y								y							
				STC_reinitialize	0x06	y			y								y							
				STC_do_step	0x07	y			y								y			y				
				STC_send_outputs	0x08	y			y								y							
				STC_stop	0x09	y			y								y							
				STC_reset	0x0A	y			y								y							
			Information (INF)	INF_state	0x80	y			y															
				INF_error	0x81	y			y															
				INF_log	0x82	y			y														y	
		Response (RSP)		RSP_ack	0xB0		y	y																
				RSP_nack	0xB1		y	y														y		
				RSP_state_ack	0xB2		y	y									y							
				RSP_error_ack	0xB3		y	y														y		
				RSP_log_ack	0xB4		y	y																
	Notification (NTF)			NTF_state_changed	0xE0			y									y							
				NTF_log	0xE1			y																
	Data (DAT)			DAT_input_output	0xF0	y					y													
				DAT_parameter	0xF1	y				y														

Complex Data Types

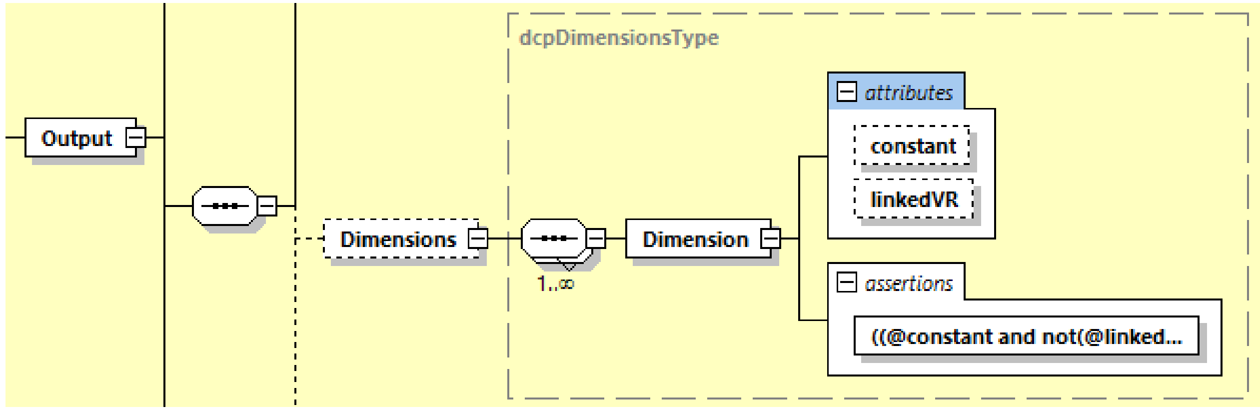
Binary/String

Position	n	n+1	n+2	n+3
PDU_{Bin}	0 0 0 0 0 1 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
PDU_{Hex}	0x04	0x00	0x00	0x00

Position	n+2	n+3	n+4	n+5
PDU_{Bin}	0 0 1 1 1 0 0 1	1 1 1 0 0 1 1 0	0 0 1 0 1 0 0 1	1 1 0 1 0 0 1 0
PDU_{Hex}	0x39	0xE6	0x29	0xD2

Arrays – data structure containing collection of variables of same type

Variables may have multiple dimensions



Operating Modes – Covering Different Time Domains

Operating mode	Description
Soft real-time (SRT)	Synchronous to absolute time, tolerant to RT violations
Hard real-time (HRT)	Synchronous to absolute time, intolerant to RT violations
Non-real-time (NRT)	Independent of absolute time

State Machine for Simulation Control

A typical simulation cycle

- Phase 1: Registration
- Phase 2: Configuration
- Phase 3: Initialization
- Phase 4: Running
- Phase 5: Computation/Sending (NRT)
- Phase 6: Stopping
- Error handling

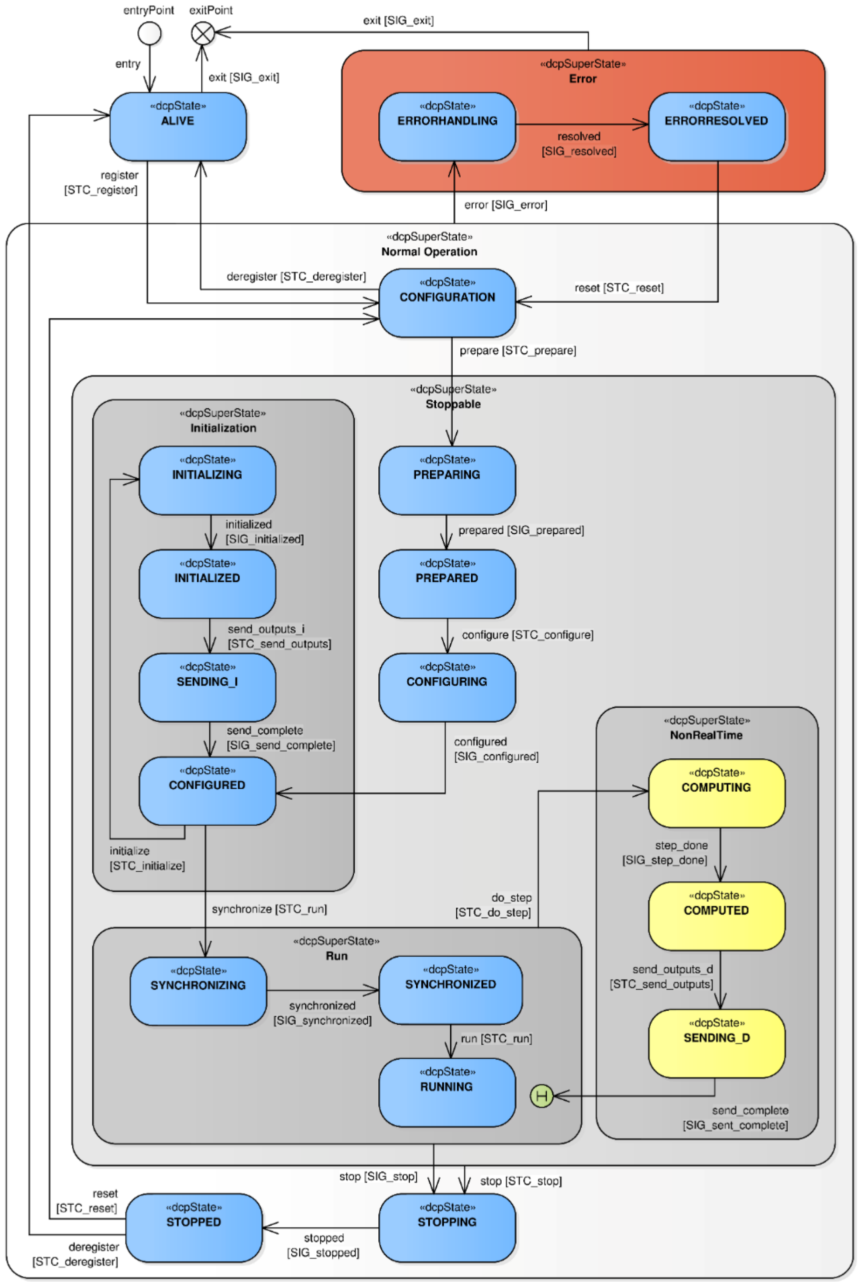
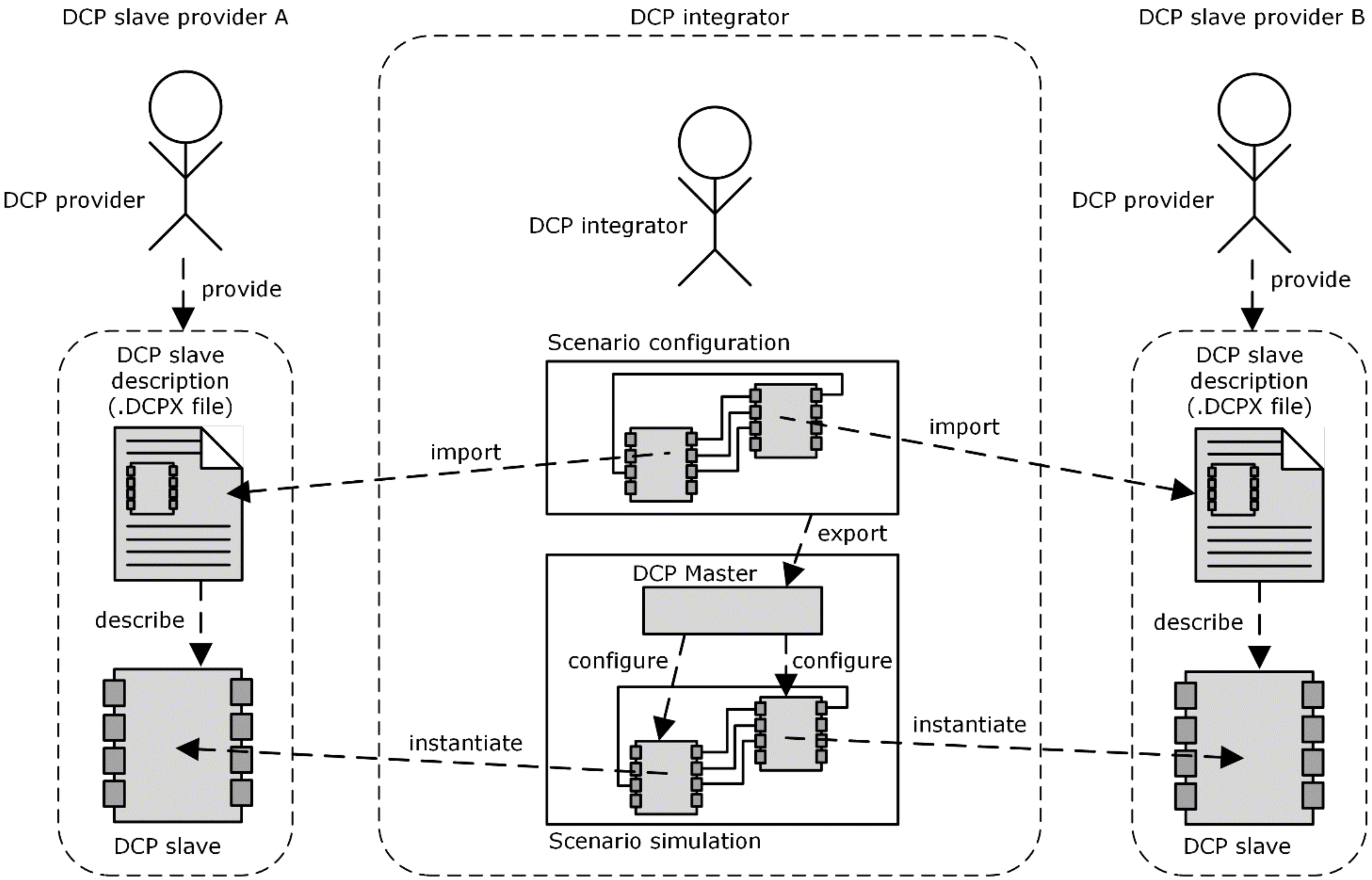
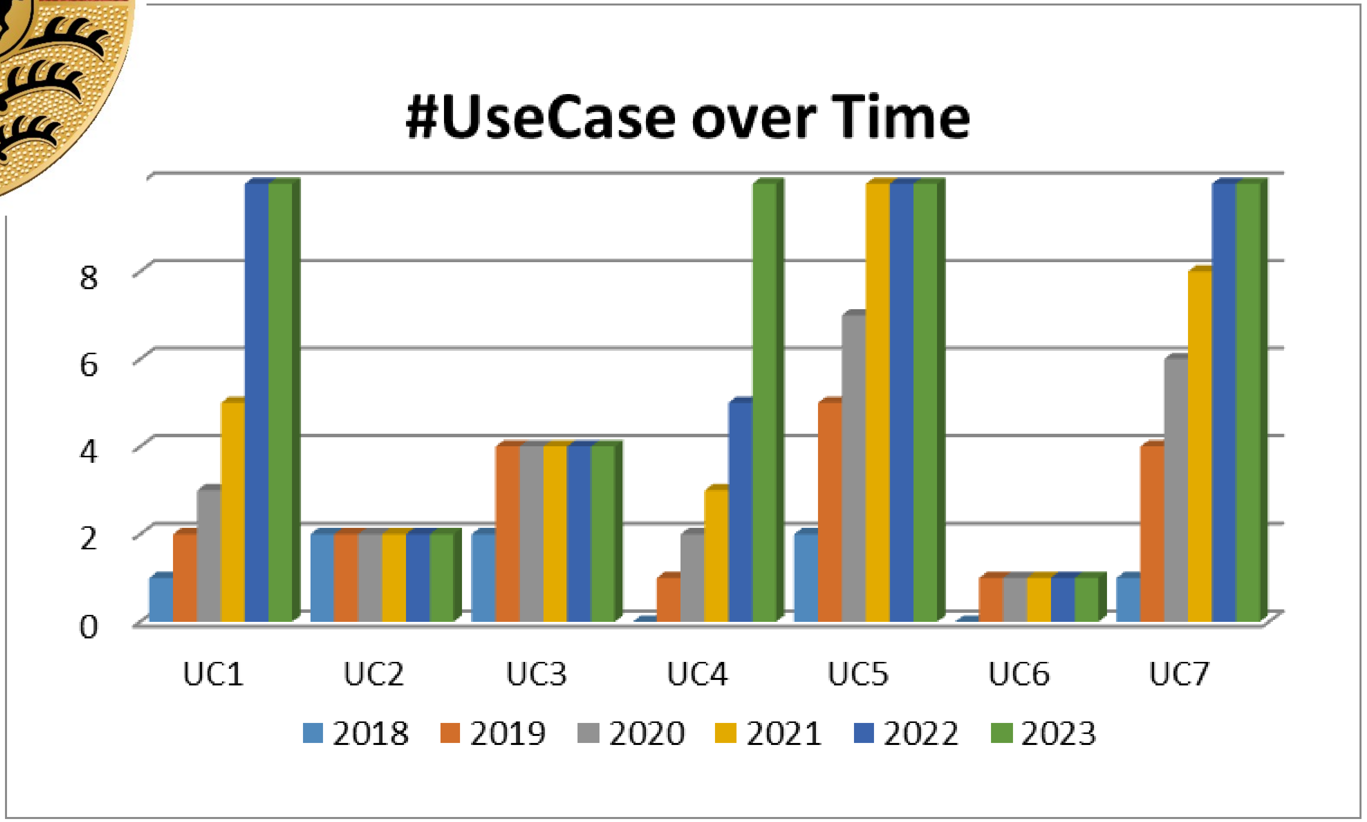


Image source: DCP specification v1.0 RC 2

Integration Method



Efficiency Study



13.000 days / 60 PY
of integration effort saved

~5-7 Mio €
of integration effort saved

Billons € global
of integration effort saved

Standardization

The DCP 1.0-RC1 was submitted to Modelica Association

- Non-profit, non-governmental organization, since 1996
- Members from Europe, North America and Asia
- Open access standards
- Open source software
- Addressing cyber physical systems

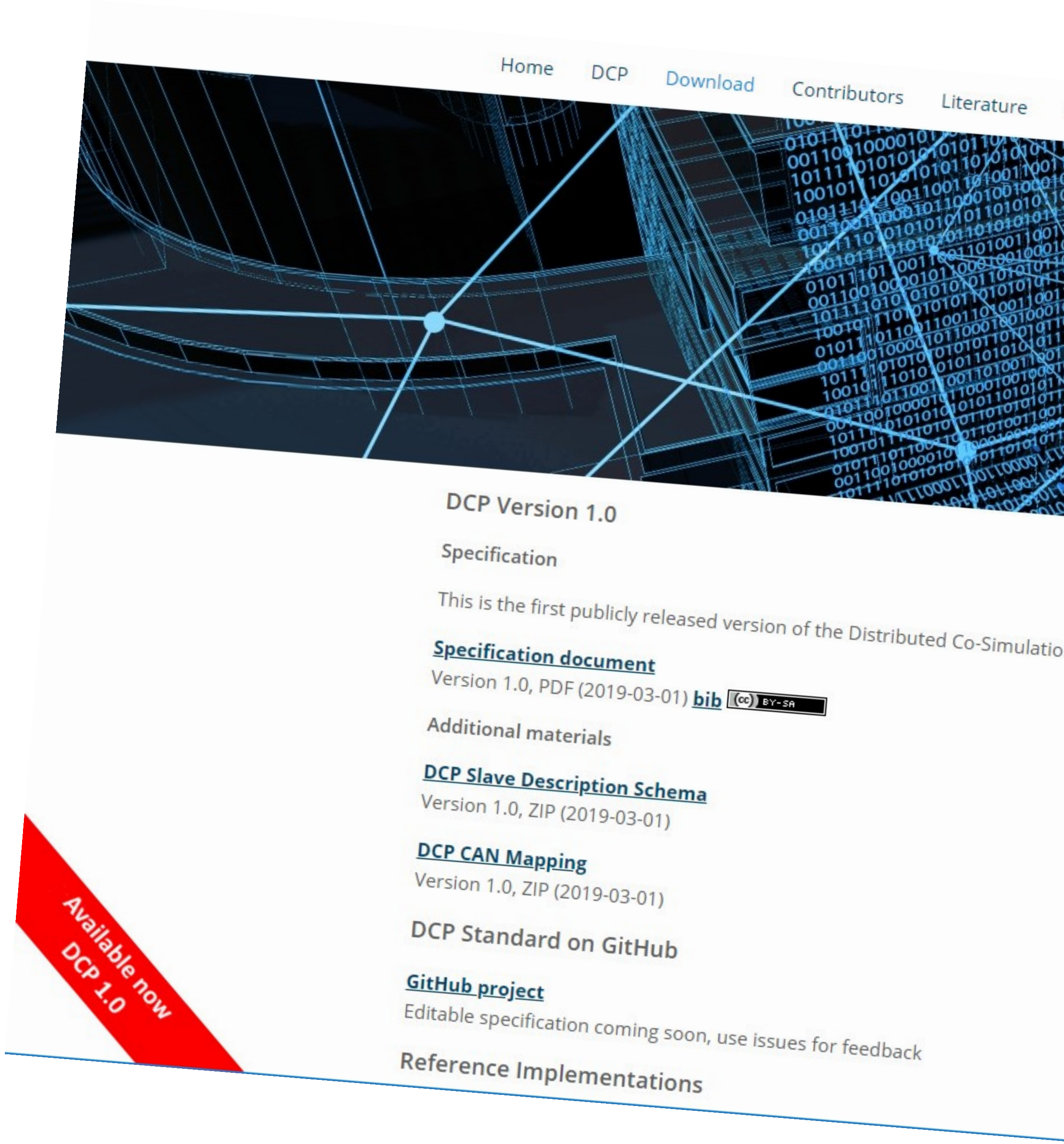
www.modelica.org

Accepted as Modelica Association Project (MAP) in July 2018

DCP Version 1.0 was released on March 4 2019

Open Access/Open Source

- **CC-BY-SA 4.0 license**
  - Open access **DCP 1.0 specification** document,
- **BSD-3 license**
  - **DCP slave description XML schema** files
  - DCP **CAN mapping**
  - DCP **reference implementation**,
  - **DCP Tester**
  - **DCP Test Generator**



www.dcp-standard.org

Modelica Association Standards Landscape

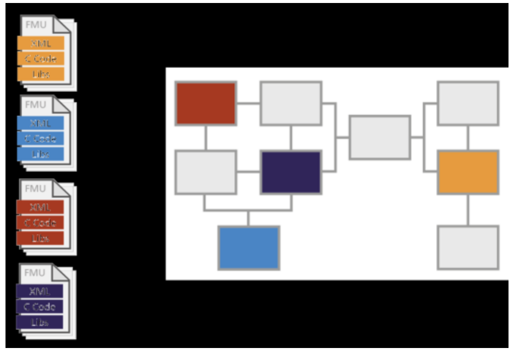
virtual vehicle

Modeling and Simulation
e.g. SimulationX, Modelon,…

Modelica Language

Modelica Libraries

Interoperability standards

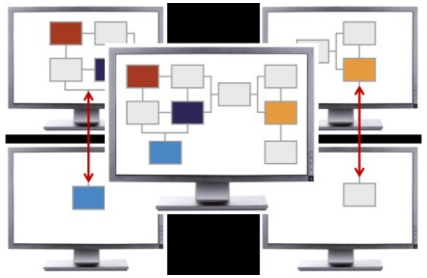


ssp System Structure & Parameterization

Version 1.0 released
Description of overall simulation scenarios

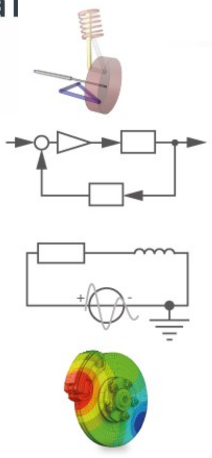
dcp Distributed Co-Simulation Protocol

Version 1.0 released
Distributed/RT co-simulation



fmi Functional Mock-Up Interface

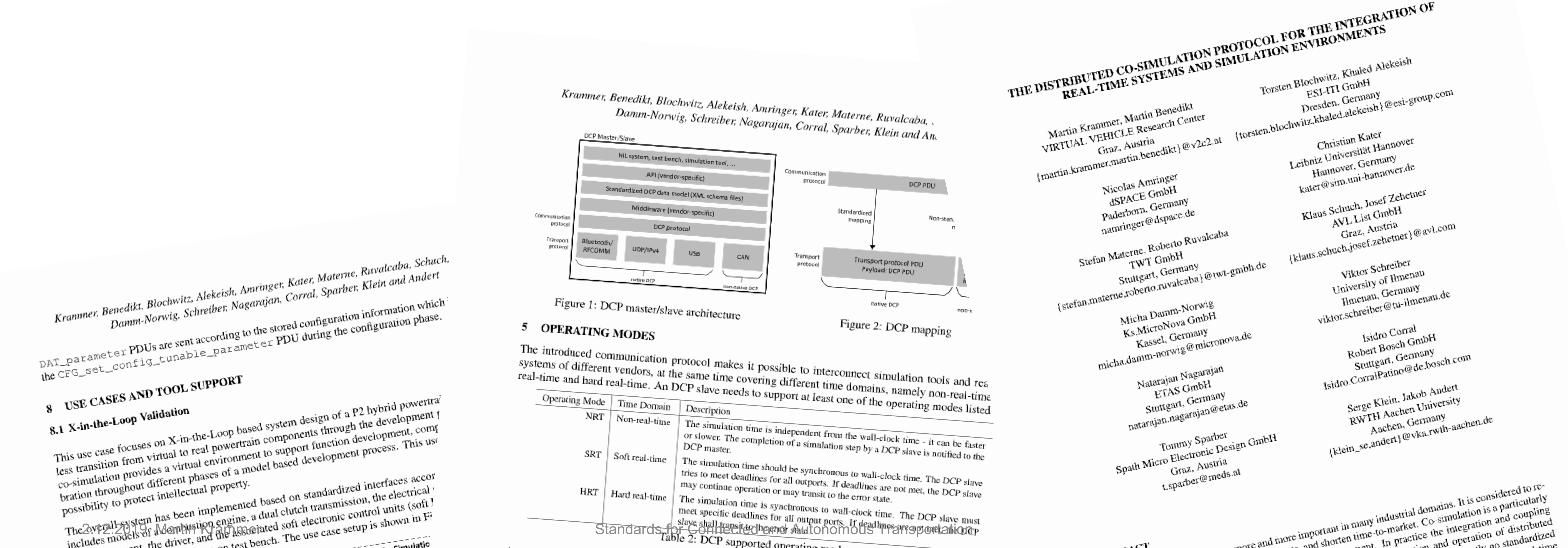
FMI 3.0 coming up
Ports and icons
Binary data
Intermediate outputs
Hybrid co-simulation



Publication

„The Distributed Co-Simulation Protocol for the Integration of Real-Time Systems and Simulation Environments“

Martin Krammer, Martin Benedikt, Torsten Blochwitz, Khaled Alekeish, Nicolas Amringer, Christian Kater, Stefan Materne, Roberto Ruvalcaba, Klaus Schuch, Josef Zehetner, Micha Damm-Norwig, Viktor Schreiber, Natarajan Nagarajan, Isidro Corral, Tommy Sparber, Serge Klein and Jakob Andert



Summary

Automated driving demands new approaches for test

Virtual, real and mixed approaches call for adequate standards

The DCP closes an FMI technology gap

Lightweight specification, efficient execution

Modelica Association maintains open access standards

Supplied by open source software

virtual vehicle

Enabling future vehicle technologies

QUESTIONS?

Martin Krammer

martin.krammer@v2c2.at

Bundesministerium Verkehr, Innovation und Technologie · Bundesministerium Digitalisierung und Wirtschaftsstandort · FFG Promoting Innovation. · Das Land Steiermark · SFG · TU Graz Graz University of Technology

Virtual Vehicle Research GmbH is funded within the COMET – Competence Centers for Excellent Technologies – programme by the Austrian Federal Ministry for Transport, Innovation and Technology (BMVIT), the Federal Ministry for Digital and Economic Affairs (BMDW), the Austrian Research Promotion Agency (FFG), the province of Styria and the Styrian Business Promotion Agency (SFG). The COMET programme is administrated by FFG.