Distributed
Co-Simulation
Protocol

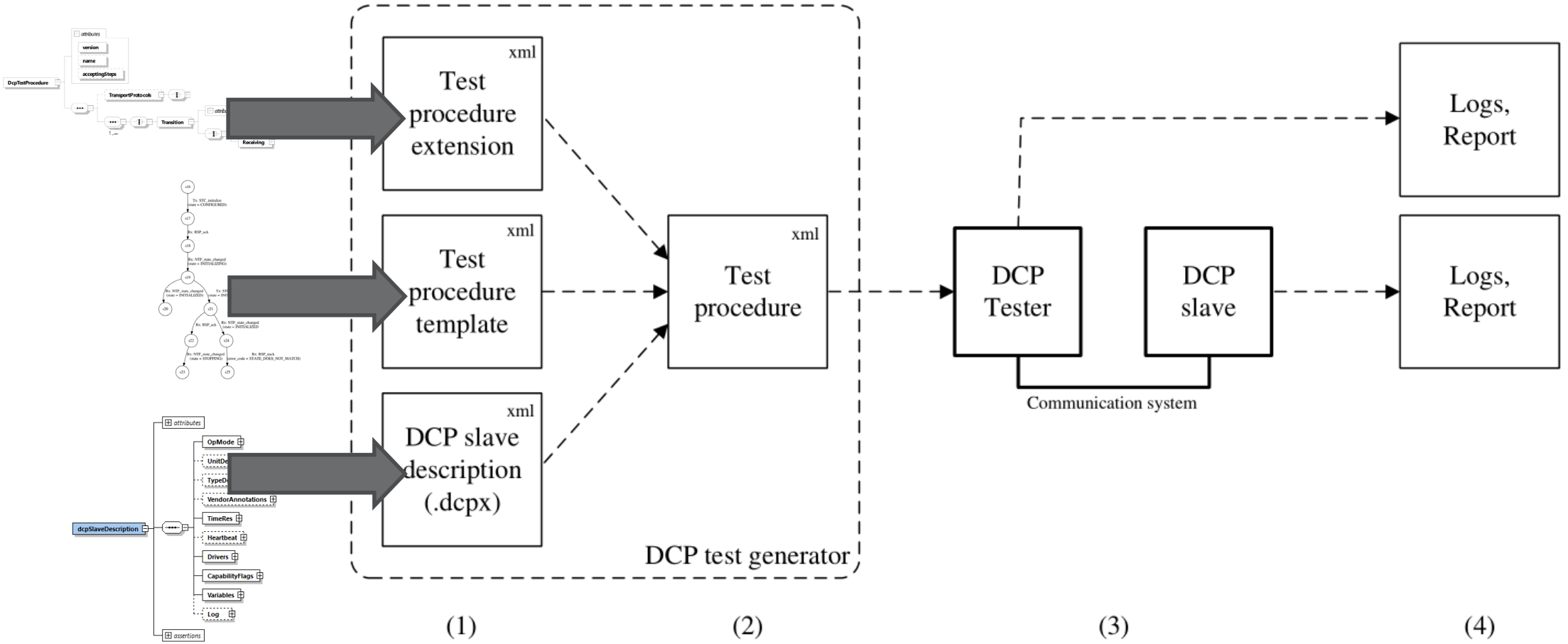
Modelica Association Project
“Distributed Co-Simulation Protocol”

Martin Krammer
DCP MAP Leader
martin.krammer@v2c2.at

@MAP_DCP


www.dcp-standard.org

Testing DCP



Krammer, M., Kater, C., Schiffer, C., & Benedikt, M. (2020). A Protocol-Based Verification Approach for Standard-Compliant Distributed Co-Simulation. *Proceedings of Asian Modelica Conference 2020, Tokyo, Japan, October 08-09, 2020*, *174*, 133–142. https://doi.org/10.3384/ecp2020174133

Protocol-based Verification

Materials published early 2021

Available materials:

- Whitepaper on Testing
- Code:
  - https://github.com/modelica/dcptester
  - https://github.com/modelica/dcptestgenerator
- Docker Container

Who is it for?

- Primarily DCP developers, implementing the protocol

Updated DCP Version

We collected community inputs

Decided to aim for a version update, include non-breaking changes

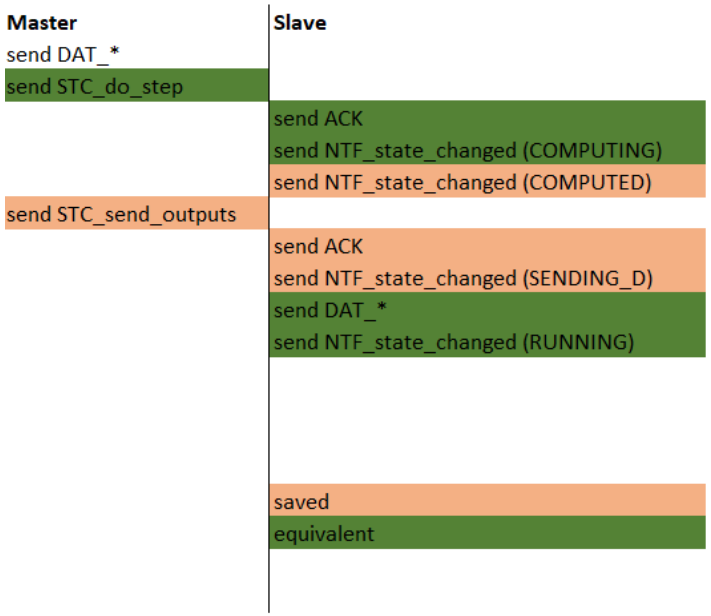
Version number 1.0.1

- Corrected some typos
- Improved somewhat blurry definitions with respect to error handling
- We aim for replacement of master/slave terminology, probably „DCP unit“
- Additional operating mode for non-real-time (NRT) operation
  - Fixes unnecessary overhead when no slave-to-slave data transfer is configured by the master

Additional NRT Mode for Non-Slave-to-Slave Communication

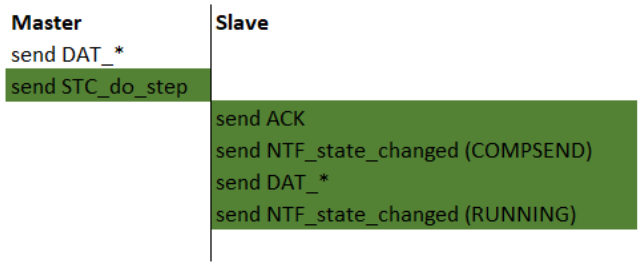
Current version

DCP 1.0



Variant 1: Reduced-Effort NRT

DCP 1.0.x Assumption: 1 State for Compute & Send

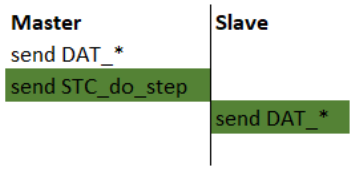


Advantages:
Reduced communication effort
Master-to-Slave scheme
Saved PDUs: 1 (M), 3 (S)
Full control maintained
e.g. introduce NRT2 mode

Disadvantages:

Variant 2: SRT with triggered steps

DCP 1.0.x Assumption: Interpret DAT_* as NTF/ACK



Advantages:
Massively reduced communication effort
Master-to-Slave scheme
Saved PDUs: 1 (M), 6 (S)
Behaves like function

Disadvantages:
Breaks DCP conventions
No transition after STC
No ACK after STC
interpret DAT_* as Ctrl-PDU
Violates PDU taxonomy
Slaves with no output will not return anything

Backlog

We‘ve built up a backlog with topics to discuss or improve

Most of them would qualify for a major update (break the protocol in some way or the other)

Some points:

- Value conversion based on unit definitions
- Setting internal states
- Multicast
- Automation issues, like auto discovery

Supporting Tools

Tools

Name	License	Platform	Master	Slave	Transport
DCPLib	Open Source	Linux/Windows	yes	yes	UDP/IPv4, TCP/IPv4
Model.CONNECT	Commercial	Windows	yes	yes	UDP/IPv4, TCP/IPv4
SimulationX	Commercial	Windows	planned	yes	UDP/IPv4, TCP/IPv4
xMOD	Commercial	Windows	yes	no	UDP/IPv4, TCP/IPv4

Any questions?

For all enquiries about DCP:

contact@dcp-standard.org