
Design of Adaptive Systems Using a Functional Digital Mock-up Approach

Olaf Enge-Rosenblatt

P. Schneider, C. Clauß, A. Schneider
Fraunhofer-Institut für Integrierte Schaltungen
Institutsteil Entwurfsautomatisierung
Zeunerstraße 38, 01069 Dresden

Email: Olaf.Enger@eas.iis.fraunhofer.de
www.eas.iis.fraunhofer.de

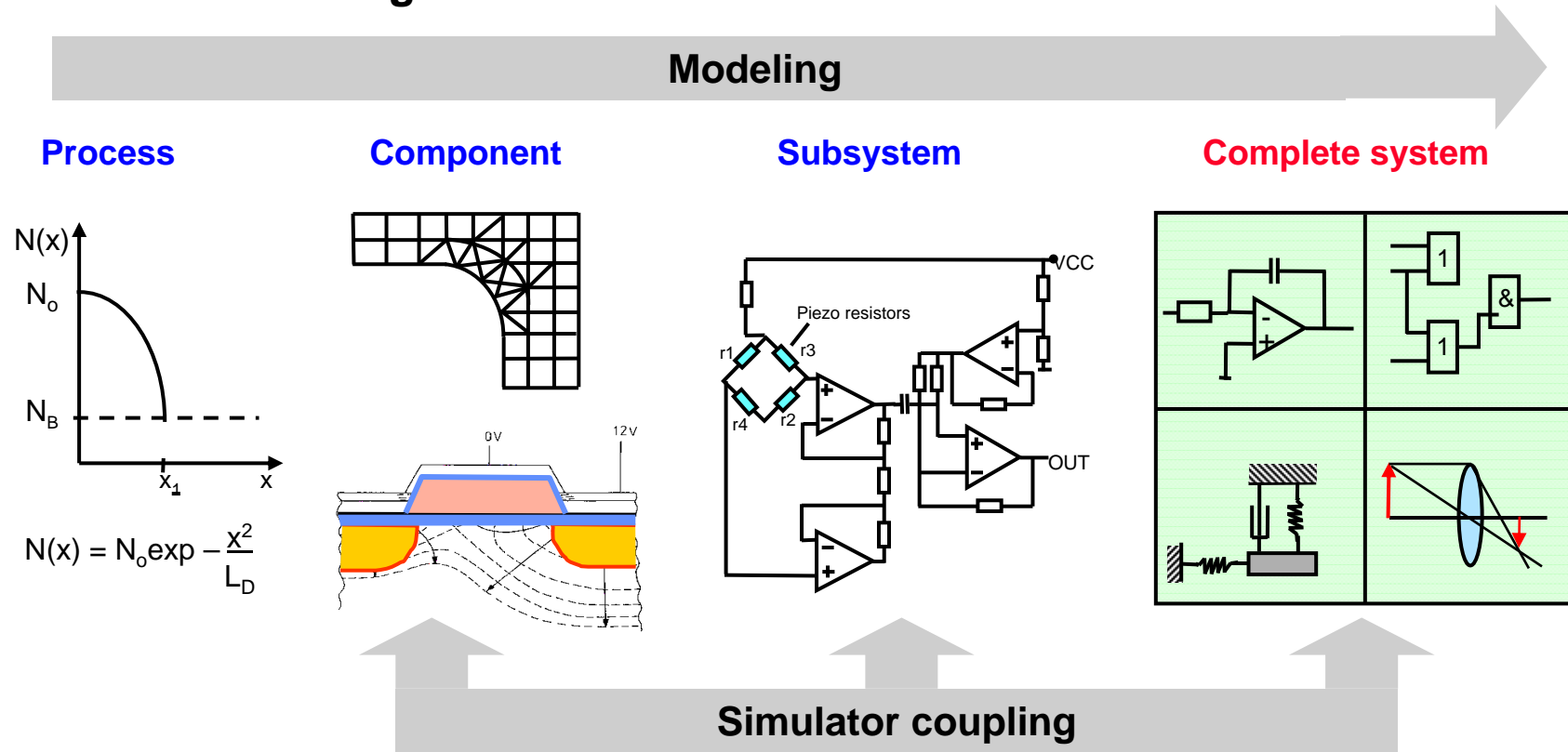
ECCM 2010, Paris, 16-21 May, 2010

Content

1. Motivation
2. Adaptive systems
 - Object-oriented modeling
 - Variable structure systems
3. Functional Digital Mock-up
 - From DMU to FDMU
 - Geometric and Behavioral Models
 - Coupling of Simulation Domains
 - Example
4. Adaptive Co-simulation
 - Exchanging subsystems
 - Adding subsystems
5. Summary

1. Motivation

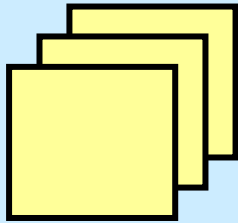
Model-based design



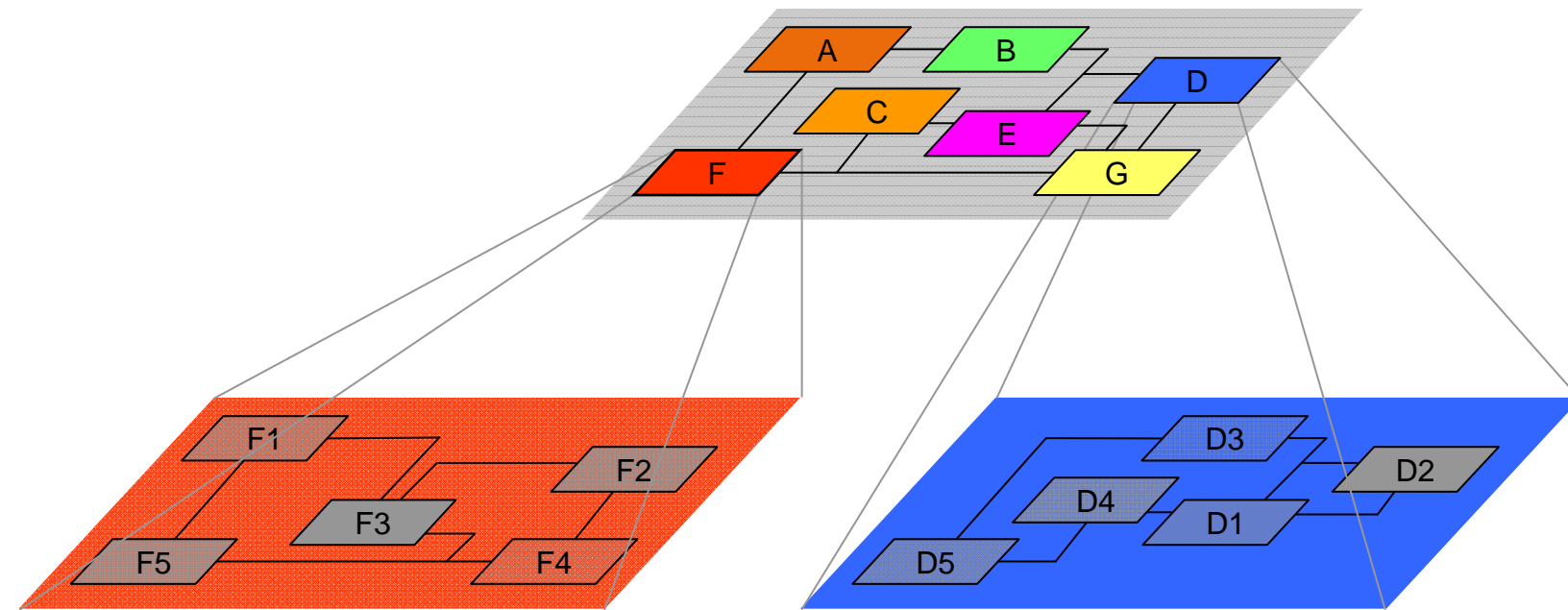
- Complex modeling of heterogeneous systems
- Different levels of detail
- ➔ **Holistic approach is highly desirable!**

2. Adaptive systems

Object-oriented modeling

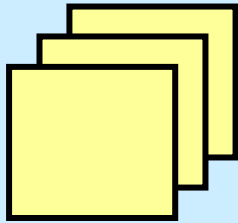


- Hierarchical design
- **Modular structure** (→ easily exchangeable subsystems)

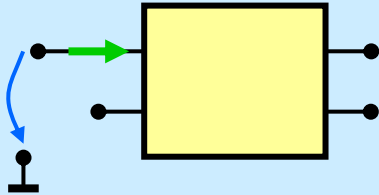


2. Adaptive systems

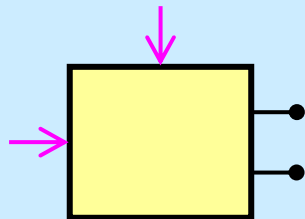
Object-oriented modeling



- Hierarchical design
- **Modular structure** (→ easily exchangeable subsystems)



- Conservative blocks using connectors with a **potential quantity** and a **flow quantity**



- Non-conservative blocks using **unidirectional signals** (inputs/outputs)

2. Adaptive systems

Variable structure systems

Model-based design

- Heterogeneous systems
- Different levels of detail

Product's environmental influences

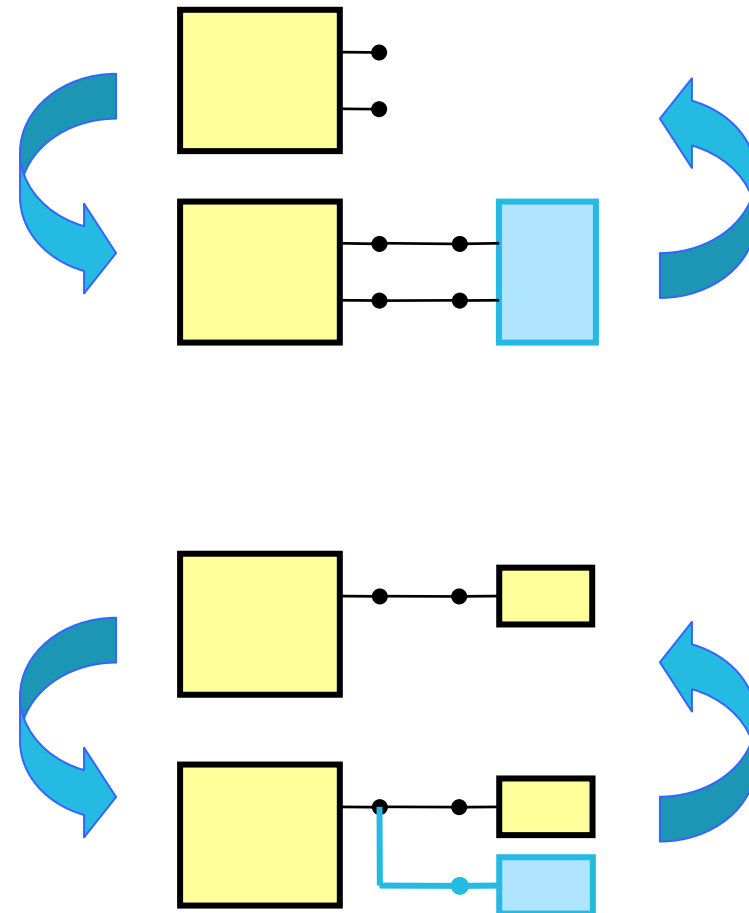
- Changing boundary conditions
- Functional restrictions
- Functional disorders

→ Adaptive systems are to be considered!

- adaptive controllers
- add / eliminate subsystems
- additional functionality



→ Adding/removing blocks and connections



2. Adaptive systems

Variable structure systems

Model-based design

- Heterogeneous systems
- Different levels of detail

Product's environmental influences

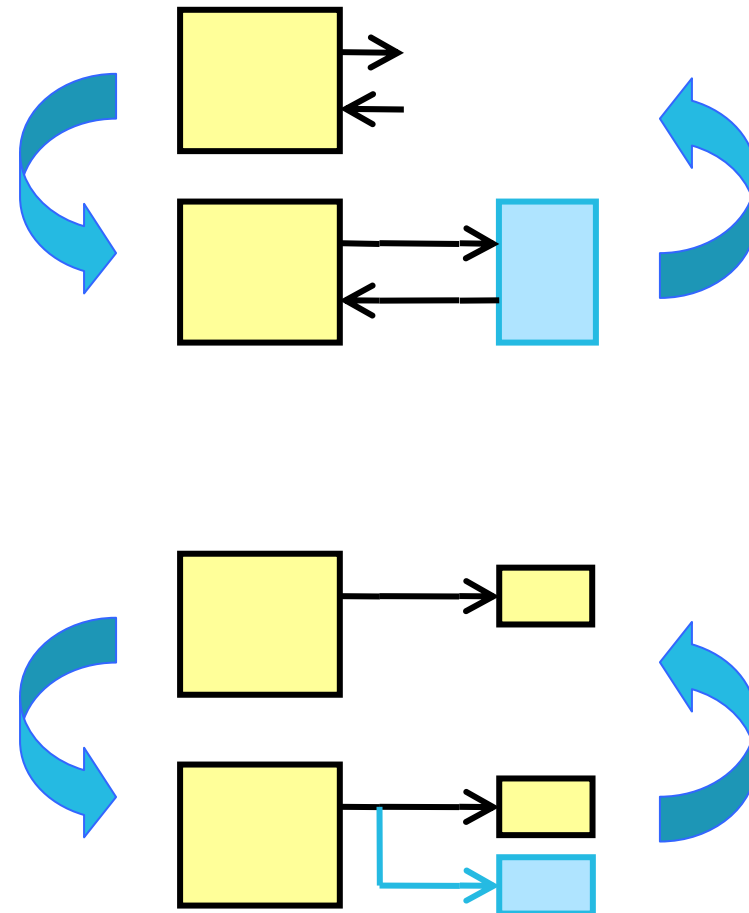
- Changing boundary conditions
- Functional restrictions
- Functional disorders

→ Adaptive systems are to be considered!

- adaptive controllers
- add / eliminate subsystems
- additional functionality

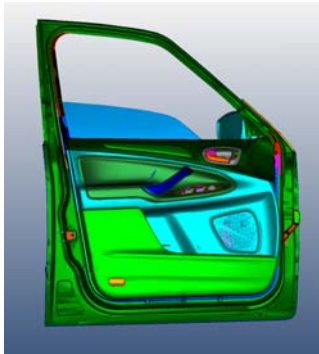


→ Adding/removing blocks and connections



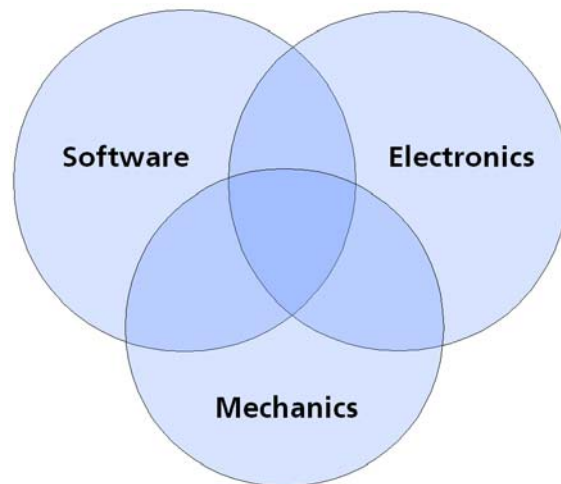
3. Functional DMU

DMU approach – state of the art



Objectives:

- Substitution of physical models
- Providing different views of shape and function
- Collision testing, assembly testing

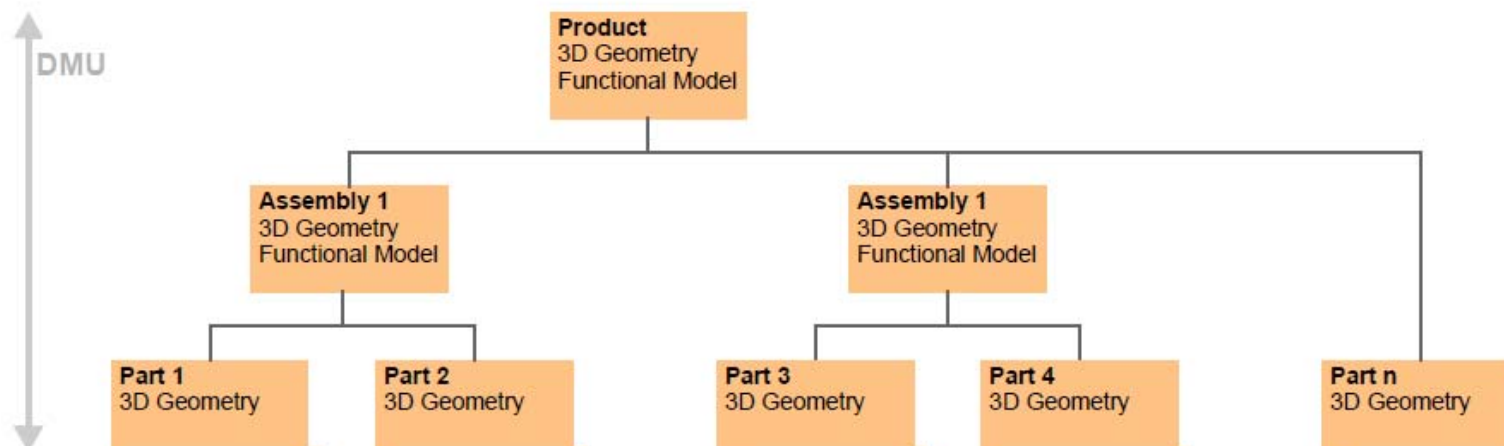


Limits:

- Simulation of mechanical constraints
- Other domains not included (e.g. electronics)
- For comprehensive validation of DMU a Physical Mock-up is needed

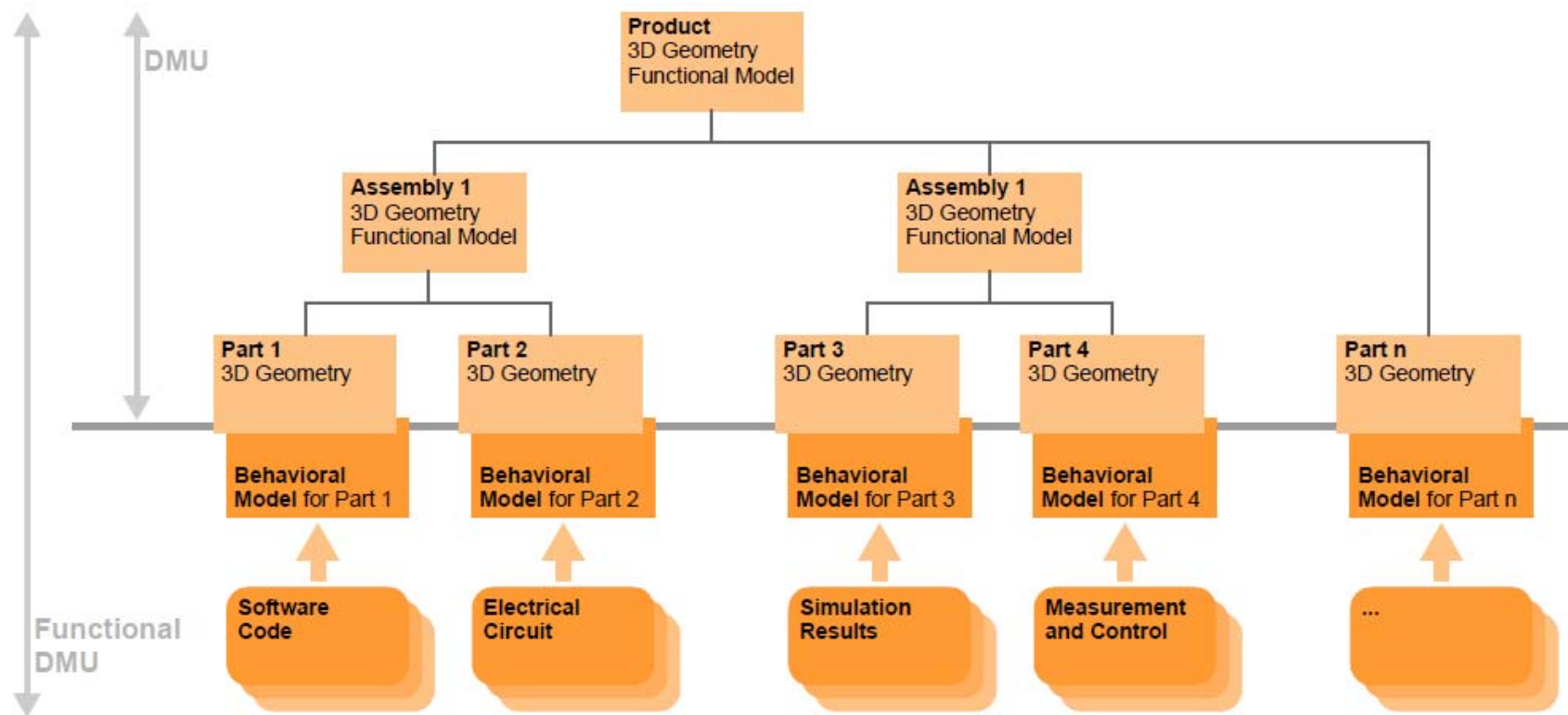
3. Functional DMU

From DMU to Functional DMU



3. Functional DMU

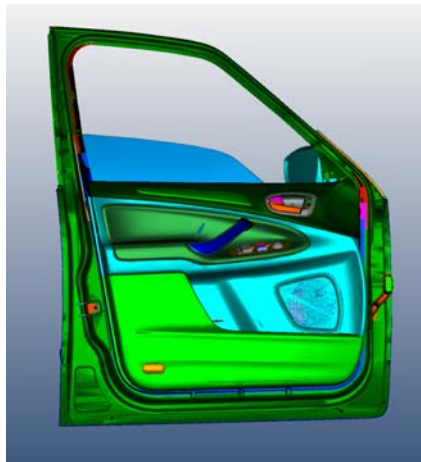
From DMU to Functional DMU



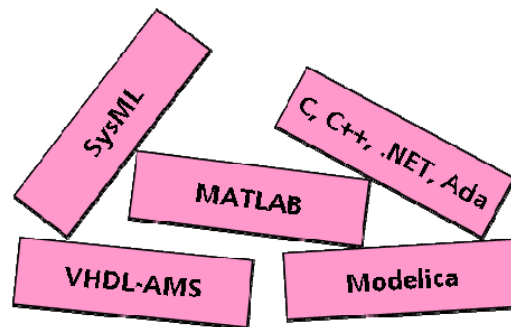
3. Functional DMU

FDMU integration platform

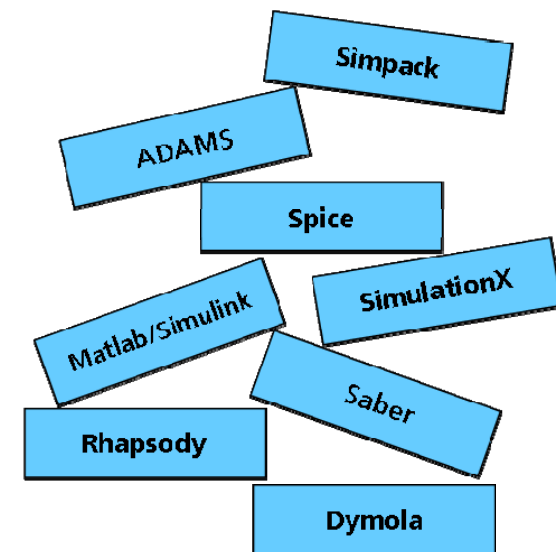
- Geometric information added by functional properties
- Many modeling languages in use
- Many simulator tools available, commonly highly specialized
- Integration platform needed



CAD model



behavioral modeling languages



simulator tools

3. Functional DMU

FDMU integration platform

Software platform:

- Web-Service-based platform
- Interactions between simulation tools
- Integration of different simulator tools and modeling languages

But: Holistic approach from user's point of view!

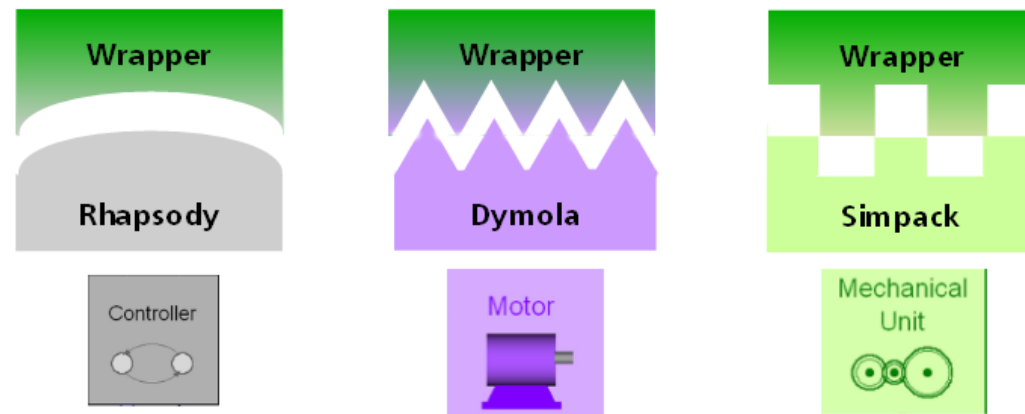
Additional features:

- Multi-domain simulation models
- Methodology for integration of geometrical and behavioral data
- Interactive visualization and control of simulation

3. Functional DMU

Coupling of simulation domains

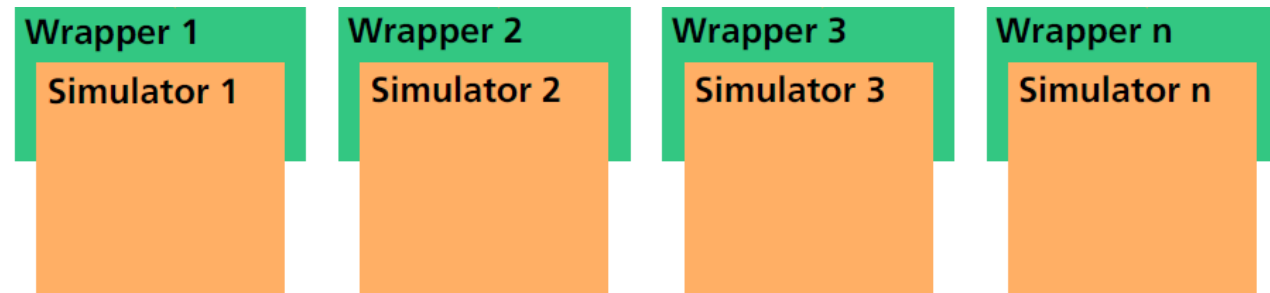
- Many simulator tools
- Different interfaces
- Integration via Wrappers
 - ➔ Encapsulation of simulator tools (external C functions)
 - ➔ Unification of interfaces



3. Functional DMU

Coupling of simulation domains

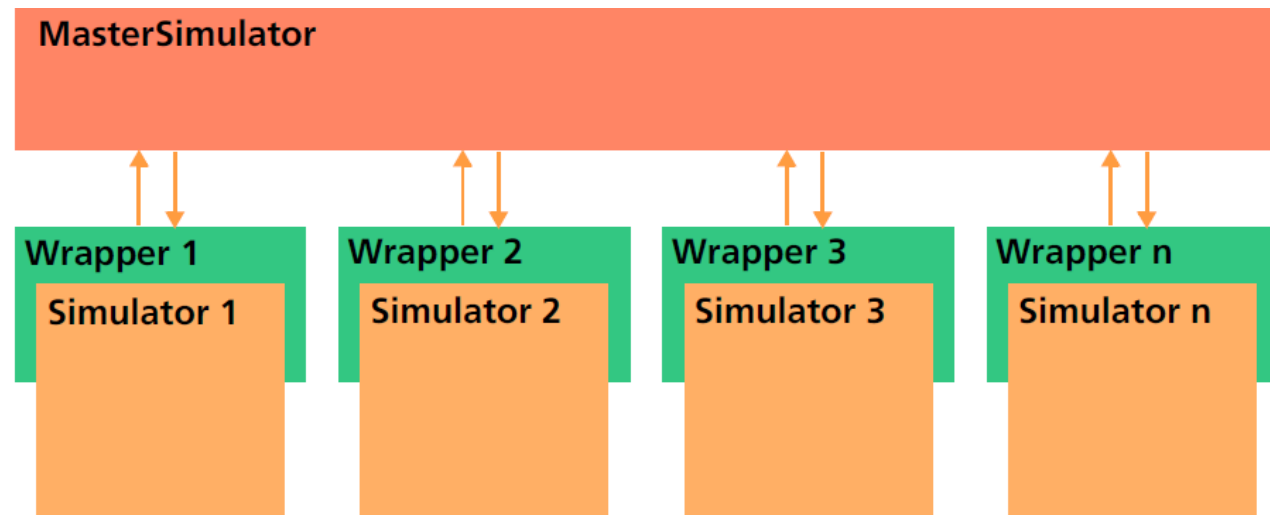
- Many simulator tools
- Wrappers enable co-simulation



3. Functional DMU

Coupling of simulation domains

- Many simulator tools
- Wrappers enable co-simulation
- Data exchange via MasterSimulator (resampling, protocol conversion)

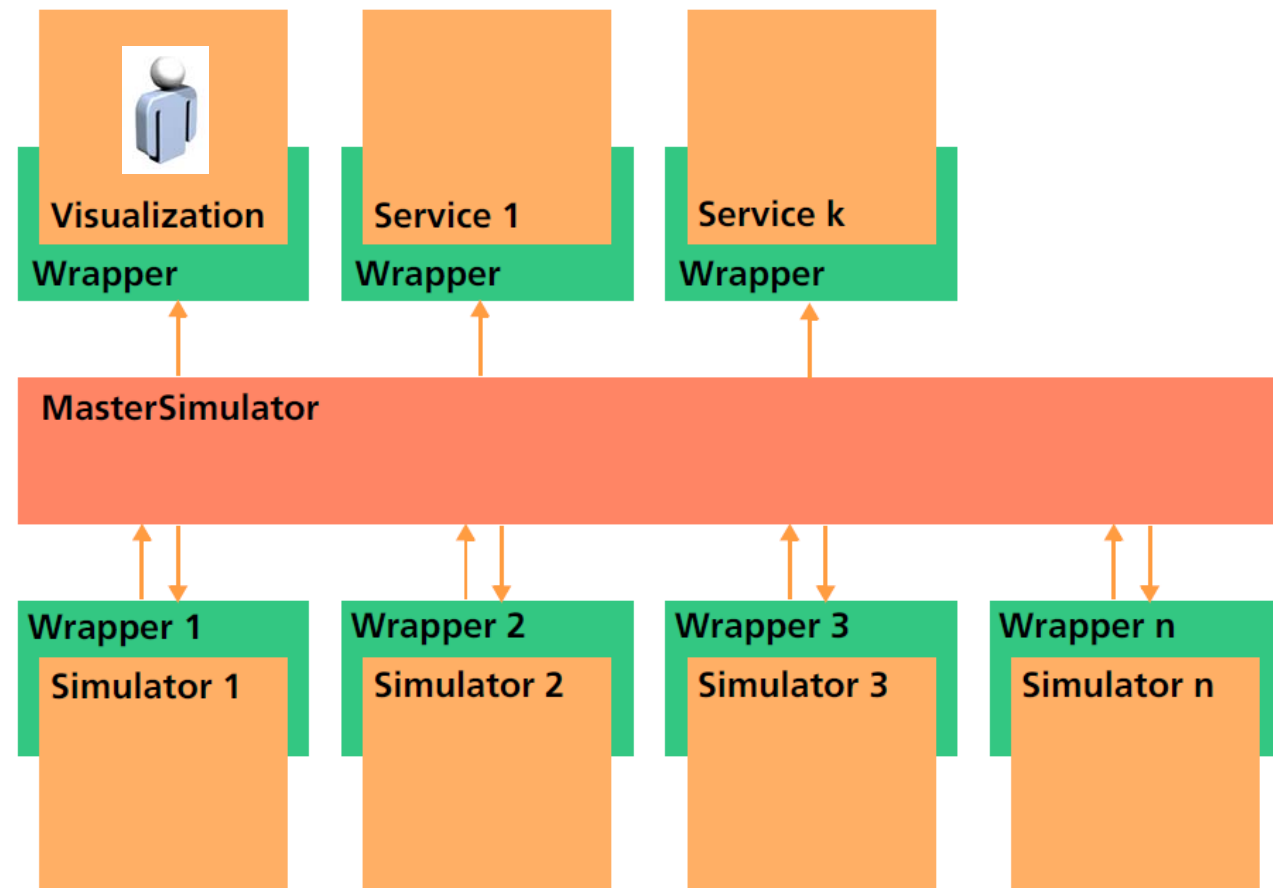


3. Functional DMU

Coupling of simulation domains

Additionally:

- Visualization
- Web services
FBB library
model deployment
...



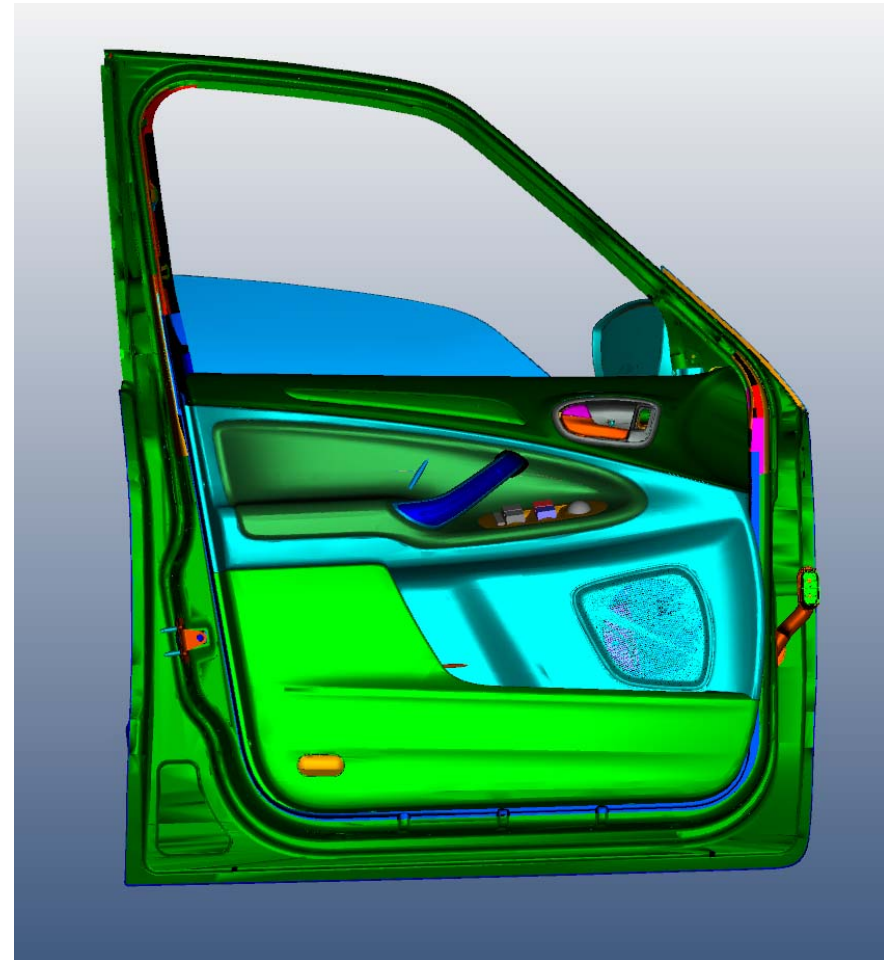
3. Functional DMU

Example: Car window regulator

Model:

- CAD model
 - frame
 - door accessories
 - window pane
 - up/down buttons

- Functional model
 - moving window (direction)
 - window driven by electric motor
 - motor regulated by switches (using a controller for safety)

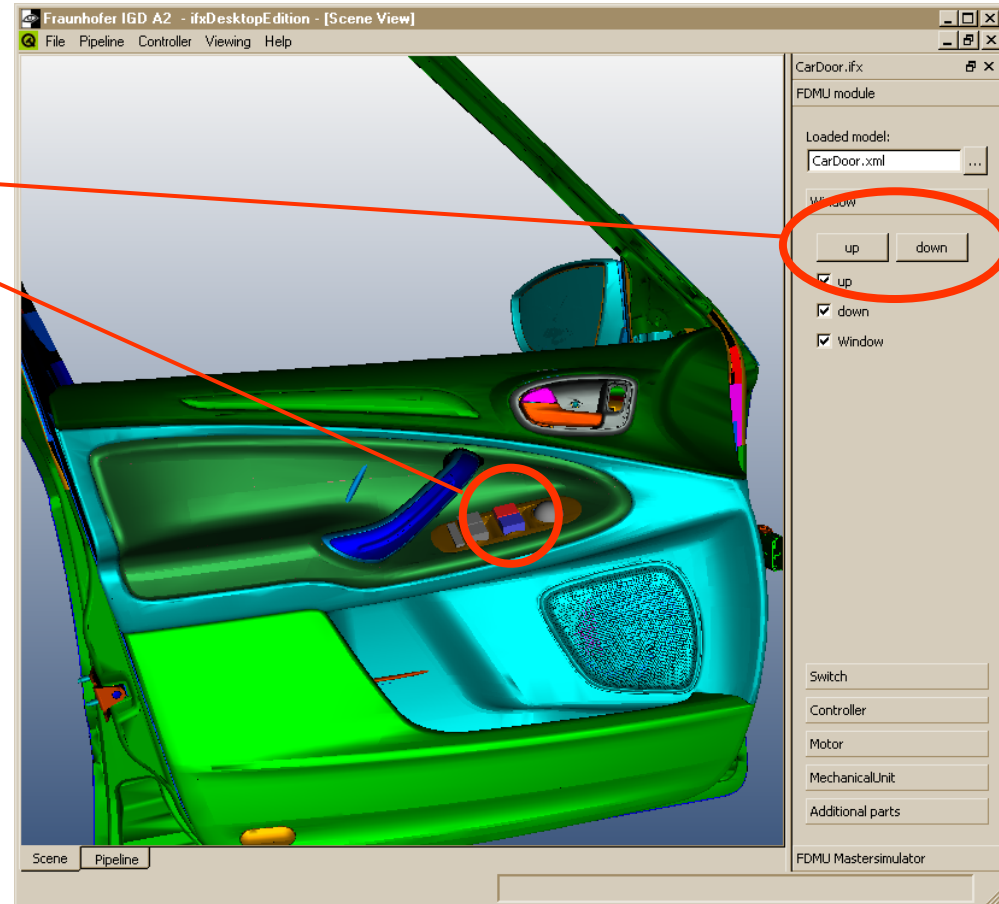


3. Functional DMU

Example: Car window regulator

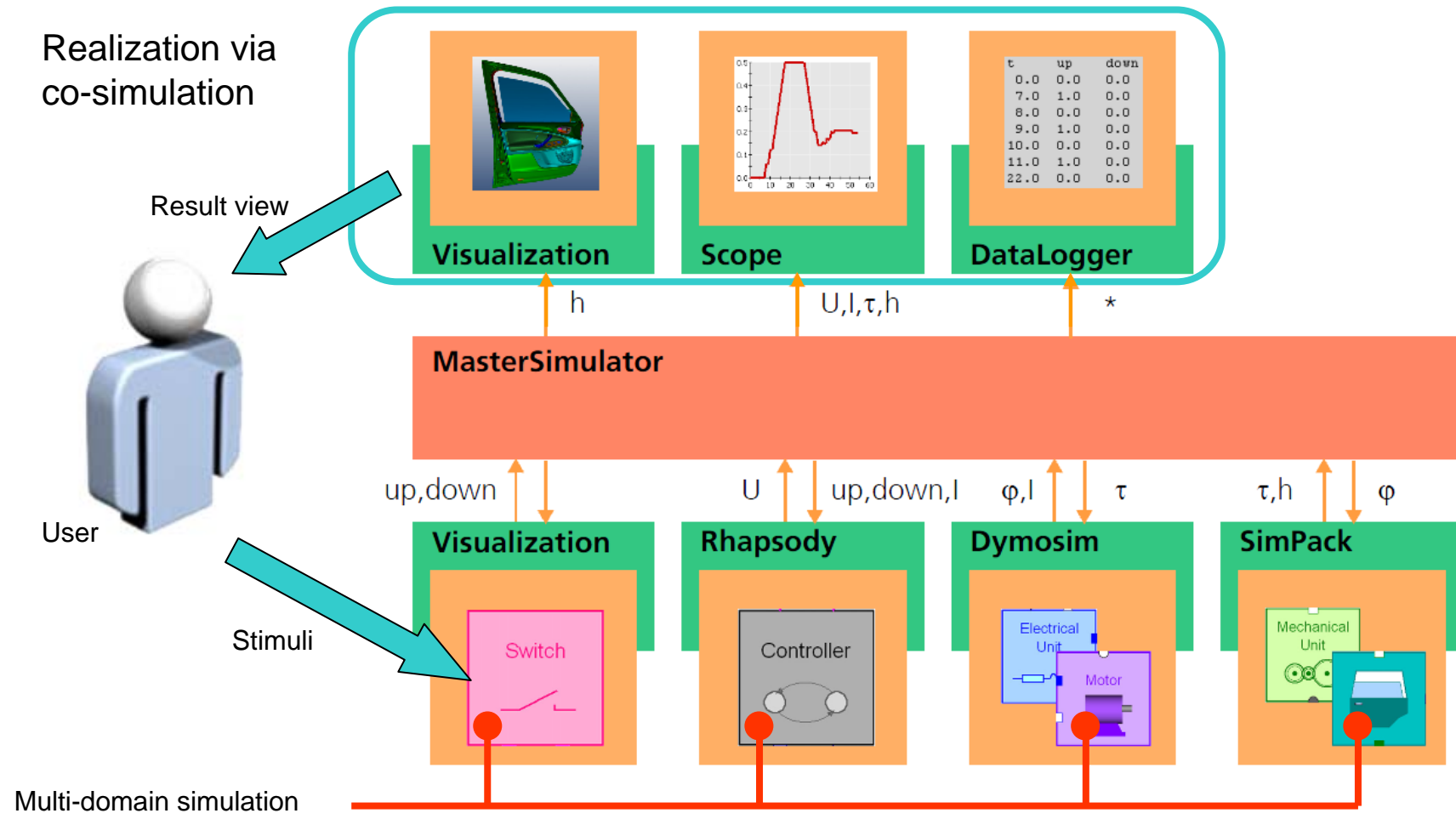
Interactive Visualization:

- Input
 - up/down buttons
- Output
 - moving window pane



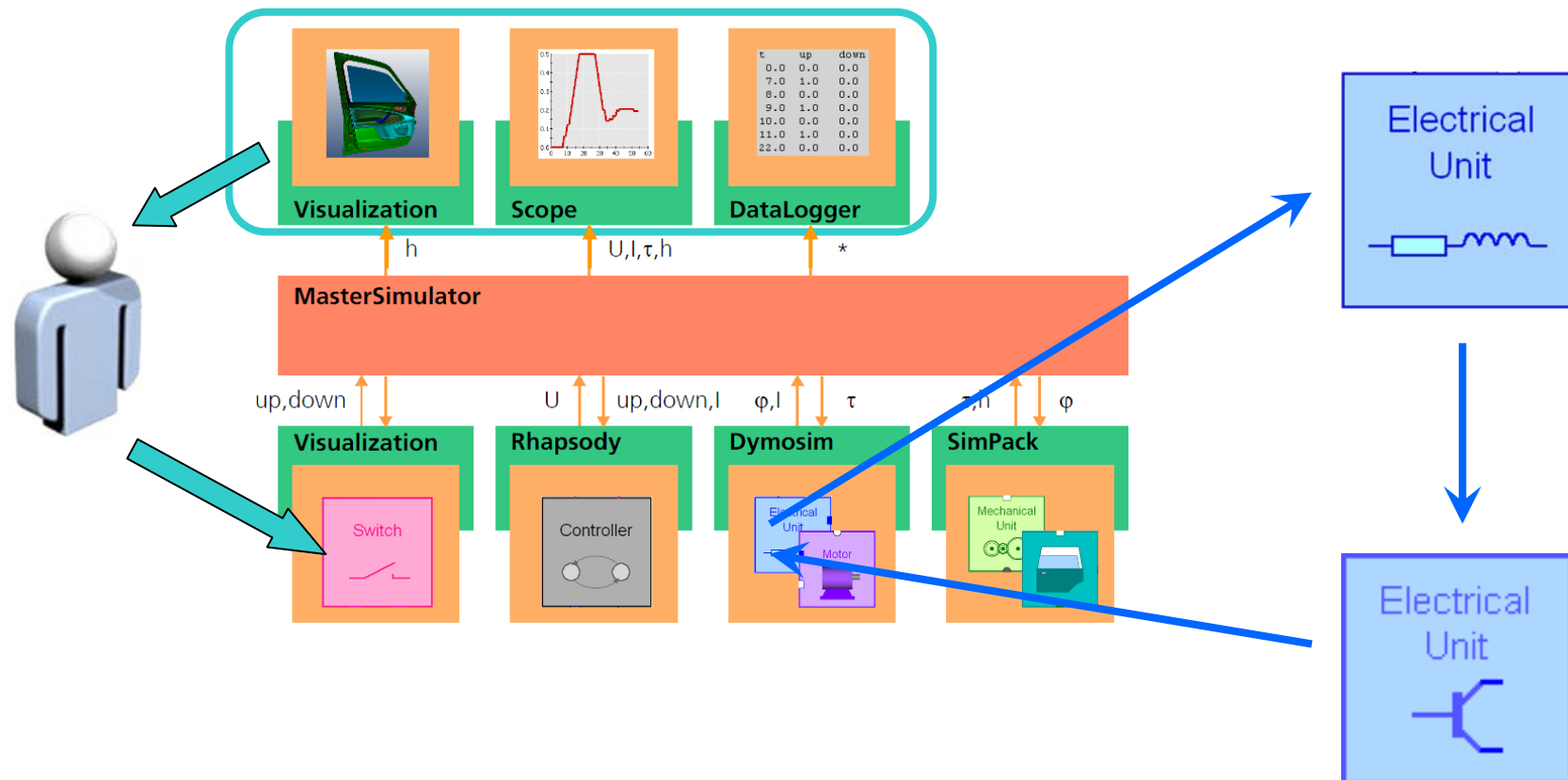
3. Functional DMU

Example: Car window regulator



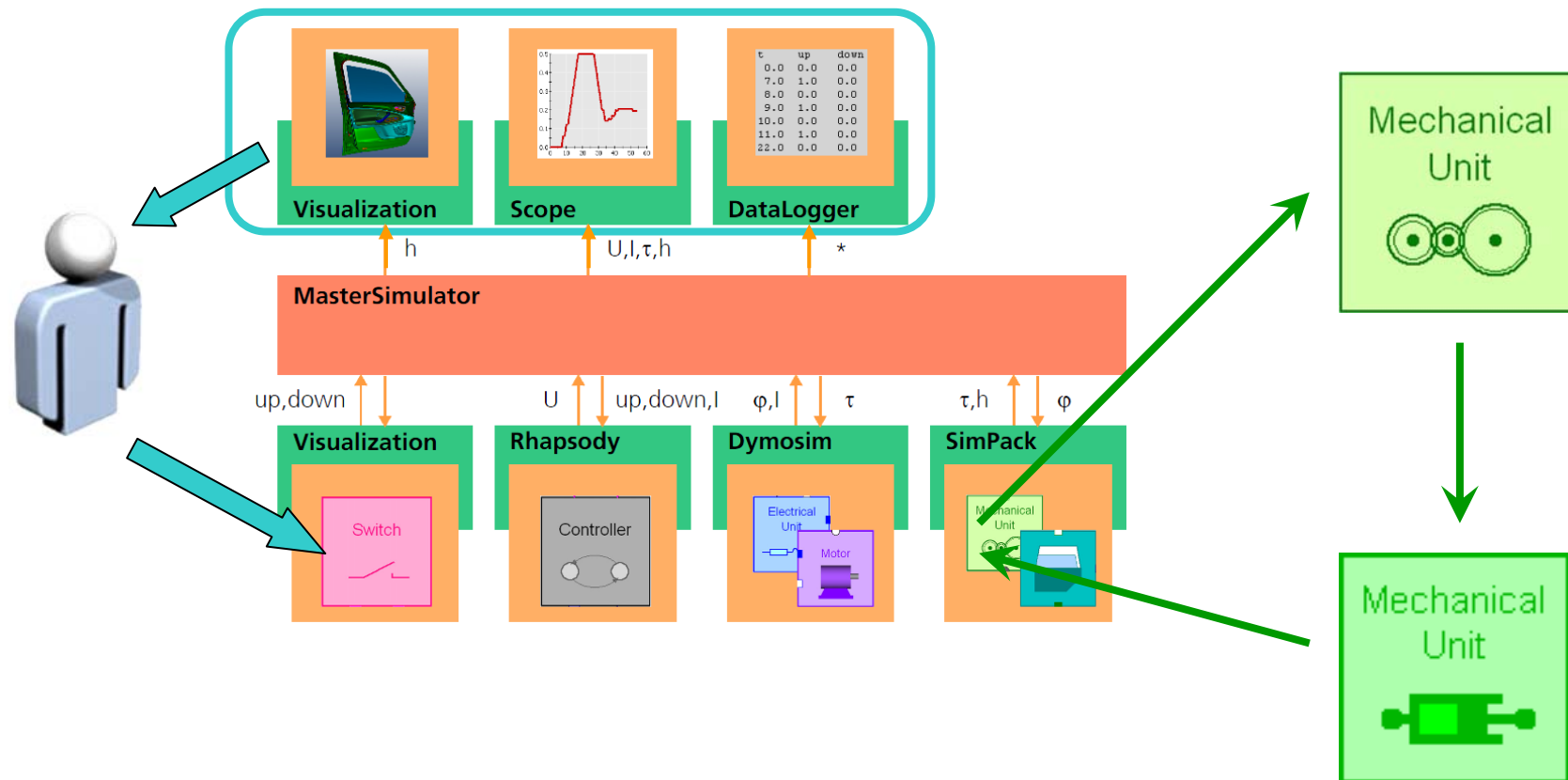
4. Adaptive Co-simulation

Exchanging a subsystem



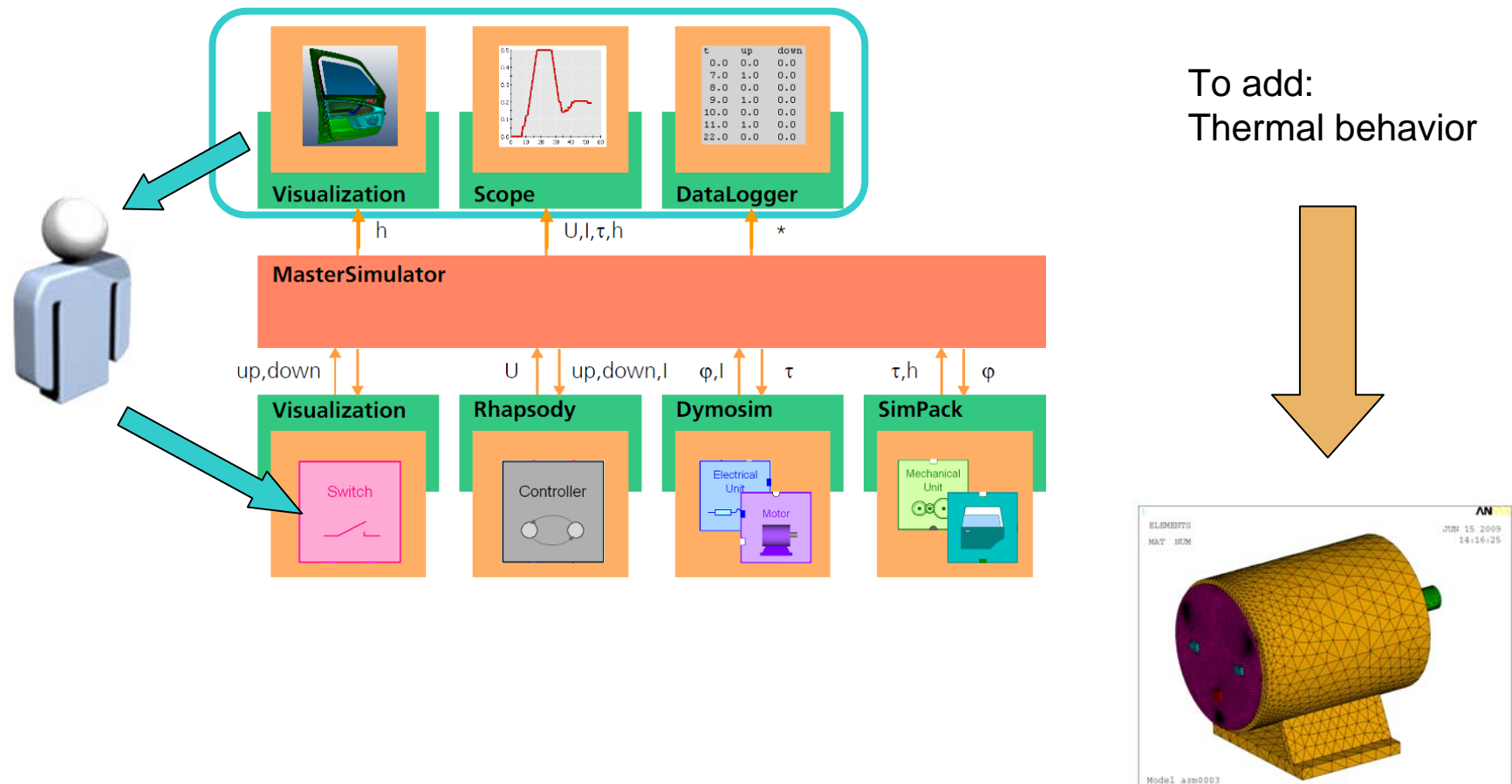
4. Adaptive Co-simulation

Exchanging a subsystem



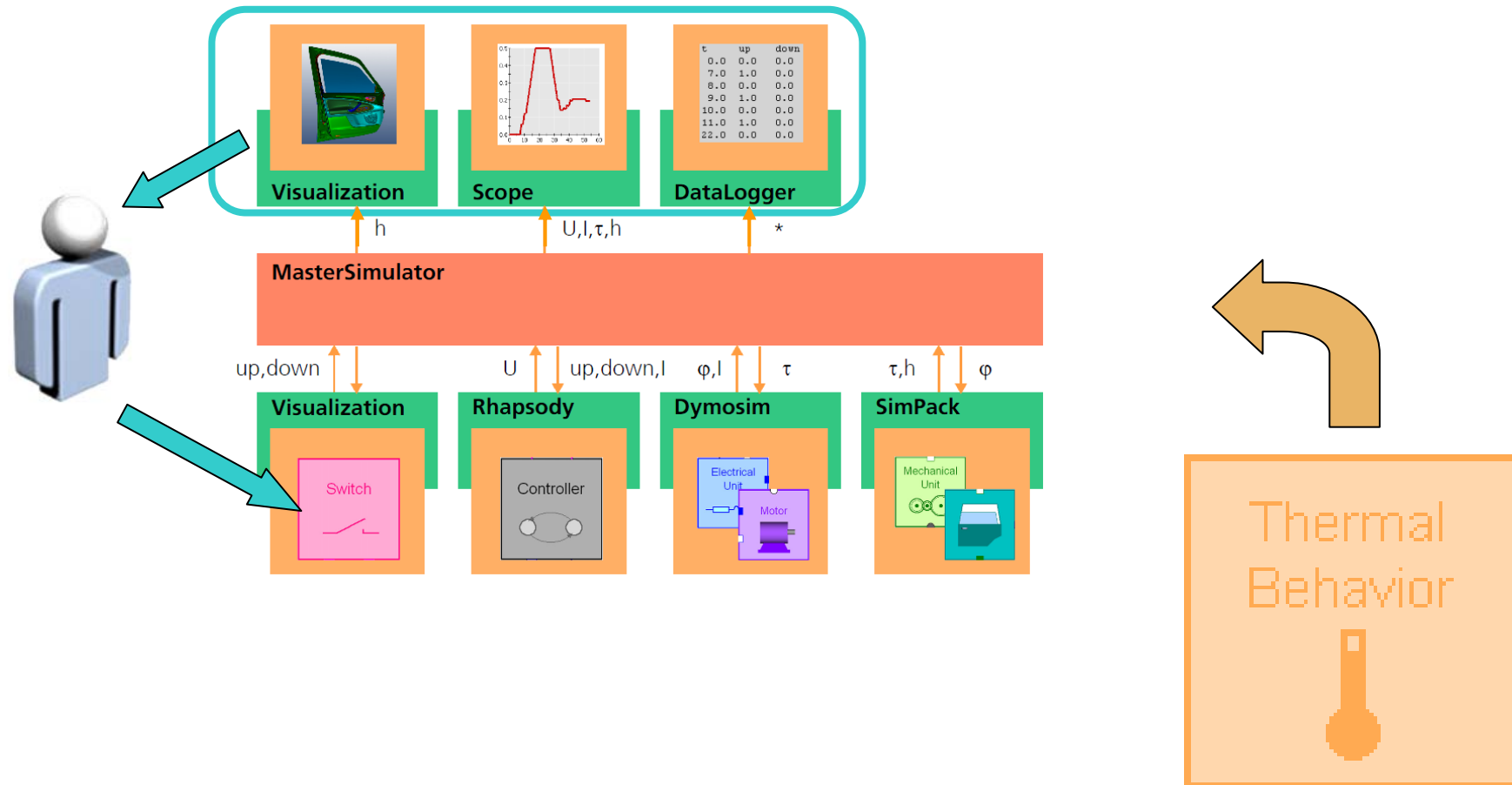
4. Adaptive Co-simulation

Adding a subsystem



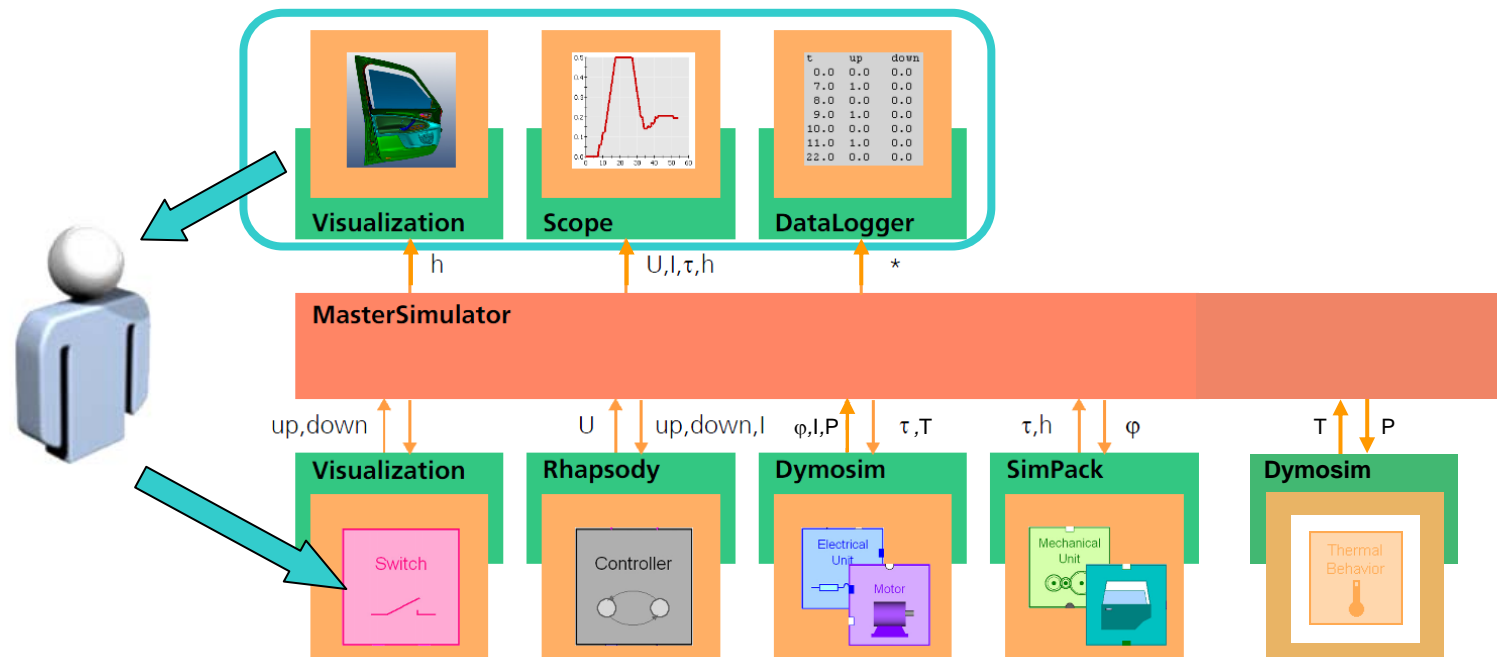
4. Adaptive Co-simulation

Adding a subsystem



4. Adaptive Co-simulation

Adding a subsystem



5. Summary

- Motivation for heterogeneous models and co-simulation from [model-based design](#)
- Idea of [modular](#) model [structure](#) (Hierarchy, object-oriented modeling)
- Consideration of adaptive systems as systems with [changing structure](#)
- Functional DMU stands for a [holistic approach](#) to modeling and simulation of multi-domain (e.g. mechatronic) systems.
- FDMU combines geometrical data and functional properties.
- Presentation of an FDMU-based integration platform
- Coupling of simulation domains/simulator tools
- Idea of adaptive co-simulation by exchanging subsystems or adding/removing subsystems

Thank you for your attention.