



RobMoSys

Model-Driven Robot Software Engineering:

From ROS-specific coding to framework-agnostic modeling

- *Composable Models and Software for Robotics Systems*
- *Towards an EU Digital Industrial Platform for Robotics*

Tutorial

<https://robmosys.eu>
<https://robmosys.eu/wiki/open-call-2>
<https://discourse.robmosys.eu>
<https://robmosys.eu/wiki>



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 732410.



Schedule

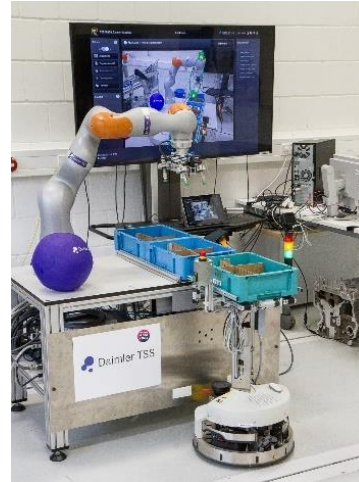
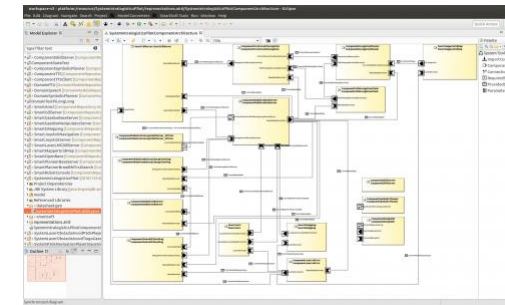
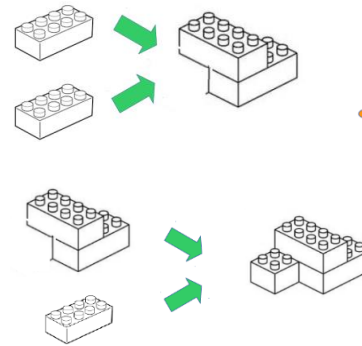
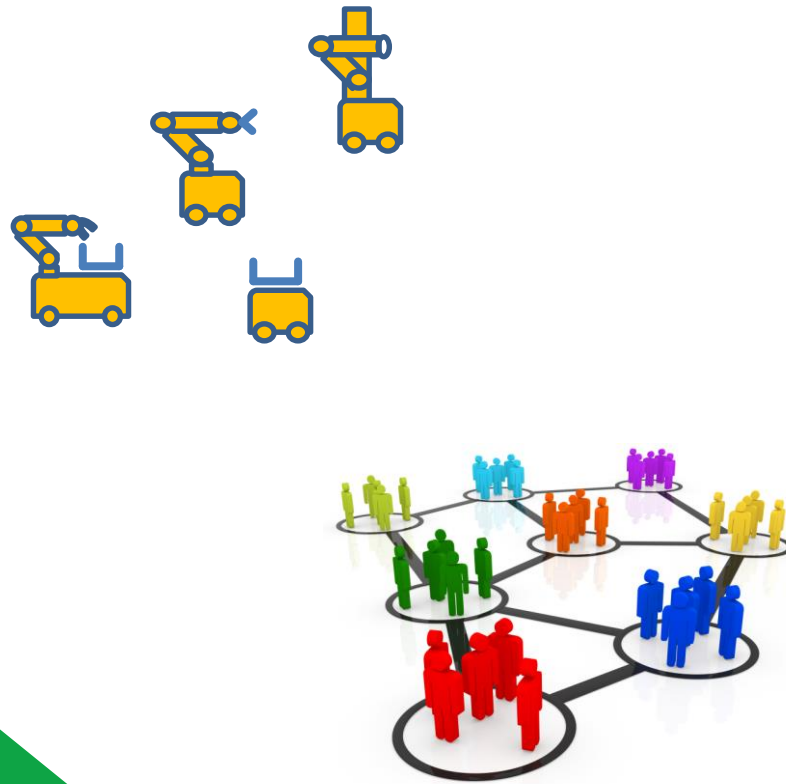


RobMoSys

09:00 – 09:20	Towards an Open and Industry-Grade European Robotics Software Ecosystem Christian Schlegel, Technische Hochschule Ulm
09:20 – 10:00	Interactive Tool Demo: Open Source Eclipse-based Tooling for System Composition: Piecing together software components to pilot applications Alex Lotz, Technische Hochschule Ulm
10:00 – 10:45	Interactive Tool Demo: Open Source Eclipse-based Tooling for Component Builders: Middleware-agnostic robotics software components Alex Lotz, Technische Hochschule Ulm
10:45 – 11:15	Coffee Break
11:15 – 11:55	Interactive Tool Demo: Safety-analysis by Model-Driven Tooling Selma Kchir, CEA List
11:55 – 12:35	Interactive Tool Demo: Stepwise Migration to Model-Driven Development: Linking Legacy Systems via the Mixed-Port Component, Linking ROS-Systems, OPC UA Systems, etc. Alex Lotz, Technische Hochschule Ulm
12:35 – 12:55	The Role of Higher-order Models in Robotics and its Reasoning Challenges Herman Bruyninckx, KU Leuven
12:55 – 13:00	Wrap Up: Summary of what to find where, how to get involved etc. Christian Schlegel, Technische Hochschule Ulm

From ROS-specific coding to framework-agnostic modeling

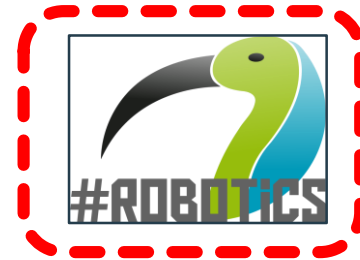
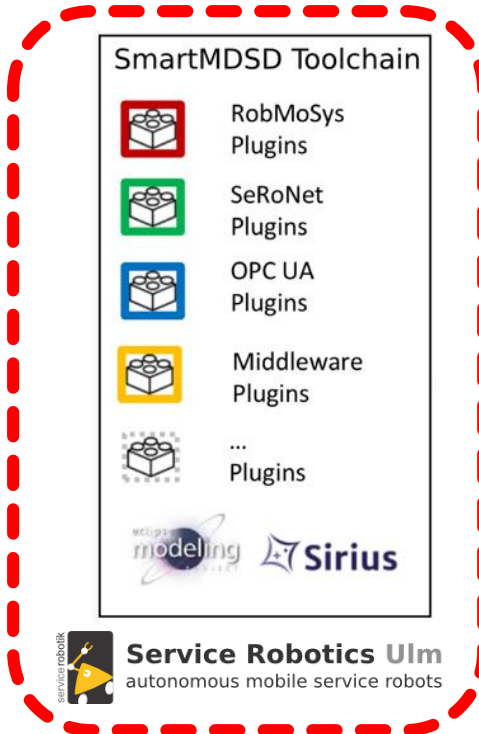
- Can we think of complex robotic systems *before* we build them?
- Can we answer „*what if*“ questions and can we find *adequate* solutions?
- Can we put systems together out of *configurable „as is“ building blocks*?
- Can we keep the behavior when we e.g. exchange the middleware?



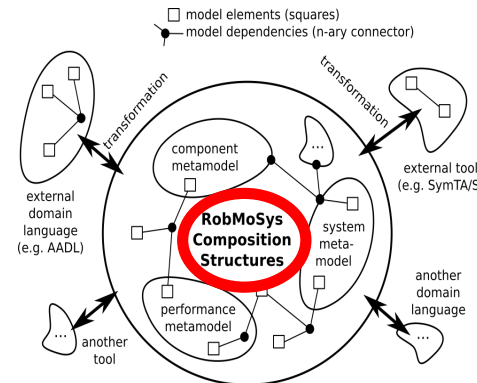
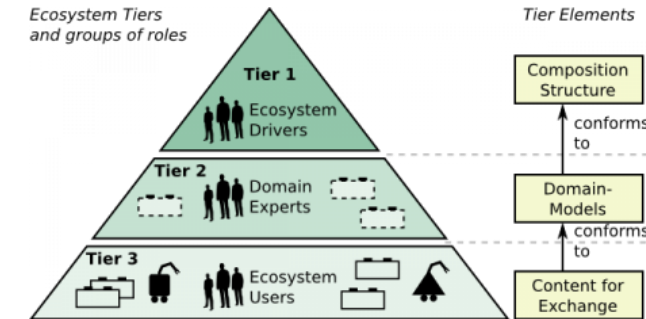
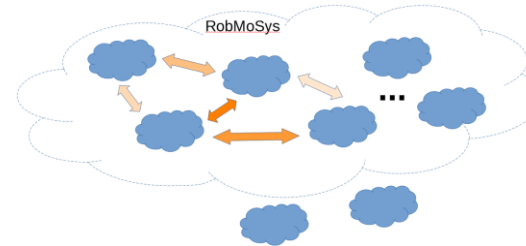
RobMoSys Conformant Toolings



RobMoSys



eITUS
Safety View for
Papyrus4Robotics



RoQME
Plugins for the
SmartMDSD
Toolchain

EG-IPC
Meta-Models
Models

CARVE
YARP Mixed Port
Component with
SmartMDSD

Plug & Bench
Benchmark Engineering
Tool for Skill Level,
links with **SmartMDSD**

Mood2Be
BehaviorTree.CPP:
Execution engine for
behavior trees

Mood2Be
Groot, an IDE to
create, modify and
monitor BehaviorTrees

<https://robmosys.eu/wiki/baseline:start>

<https://robmosys.eu/wiki/jumppage>

<https://robmosys.eu/wiki/open-call-2>

RobMoSys is more than just another project ...



RobMoSys

- is a community ***moderator***, is ***most inclusive*** and generates a „***movement***“
- ***joins forces*** to address the most challenging questions together
- builds a ***positive atmosphere*** and ***team spirit*** in the ***community***
- builds trust by openness and accessibility: ***transparency has priority*** over completeness
- *provides a **methodology** to organize the „robotics body-of-knowledge“*
- *provides consistent, agreed and resilient **reference structures** for making robotics accessible and usable*
- *underpins its benefits by selected and concrete **success stories** (tooling, pilots, applications, community, ...)*

Explication and management of the

- relationship between different *approaches*,
- interfaces between different *roles*,
- interfaces between different *building blocks*,
- interfaces to different implementation technologies

Make all accessible and usable in the form of

- better models,
- better tools,
- better systems

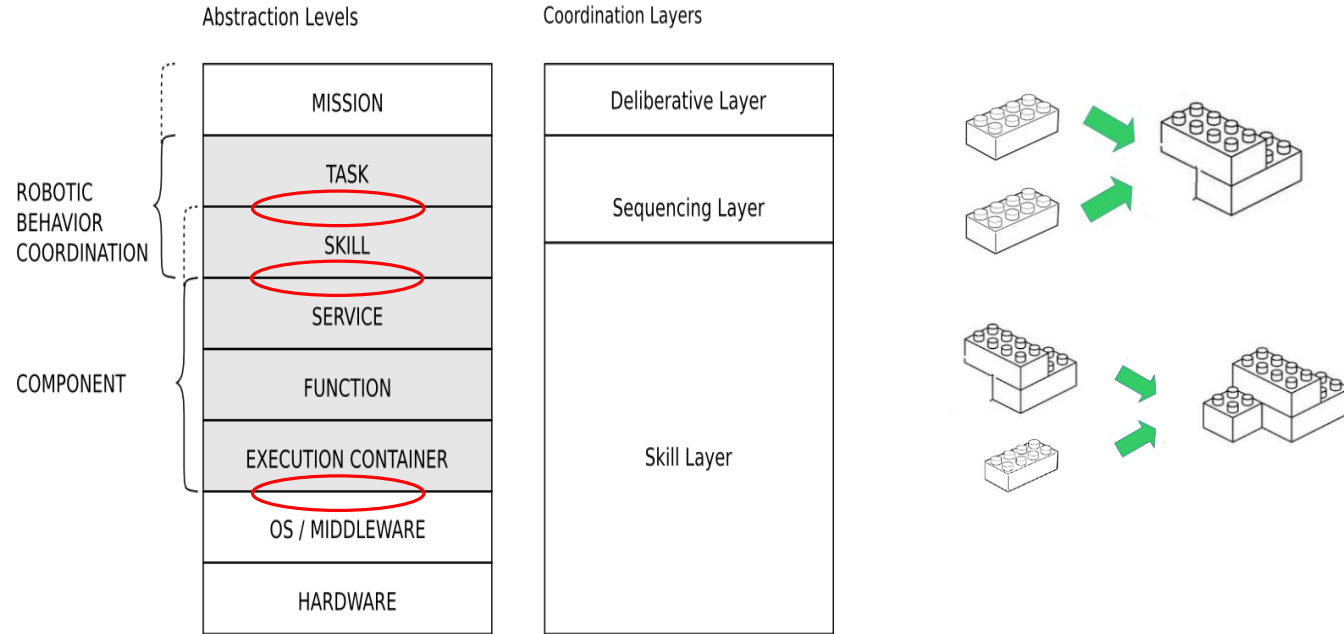
Definition of the RobMoSys Ecosystem:

The collection of assets (tools, models, software components, application pilots, guidance documents) and services (e.g. for adoption, coaching) issued by RobMoSys, which are developed, maintained and evolved by the RobMoSys Community.

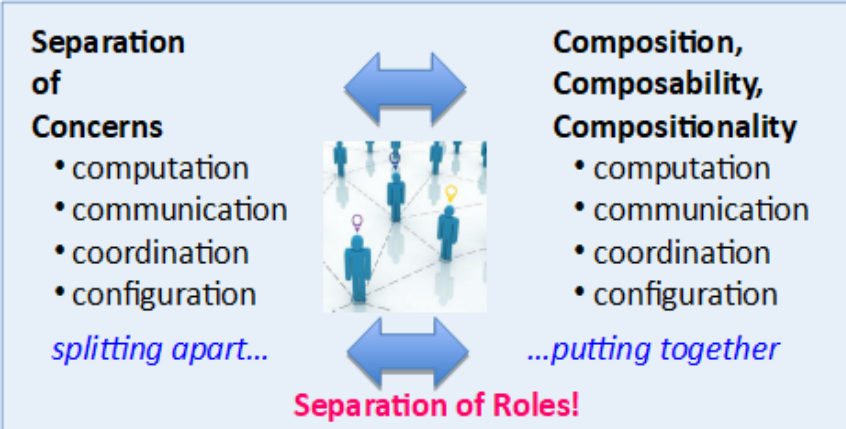
Composition, Blocks, Ports, Connectors, Data Sheets, Models



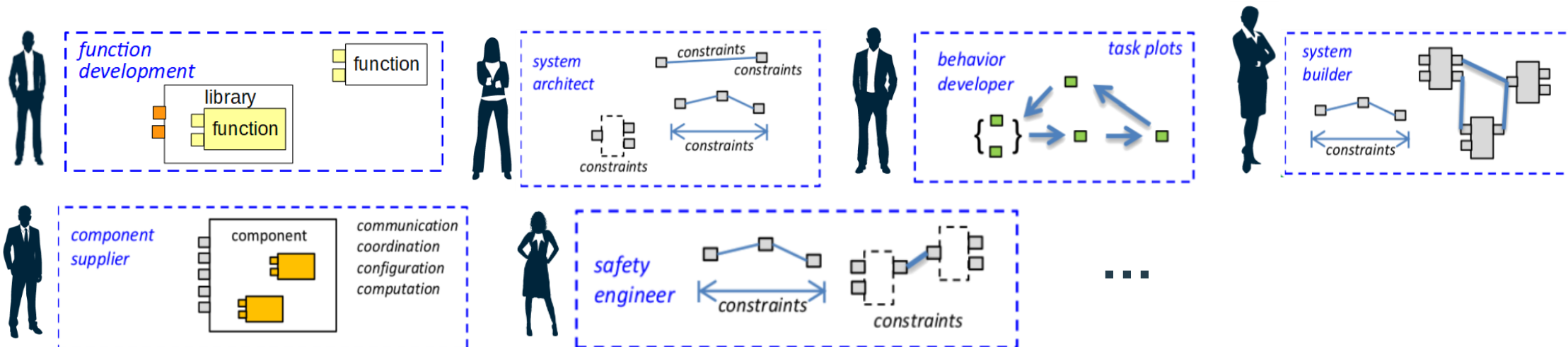
RobMoSys



Support as much freedom as possible while still ensuring **composability** despite **separation of roles**



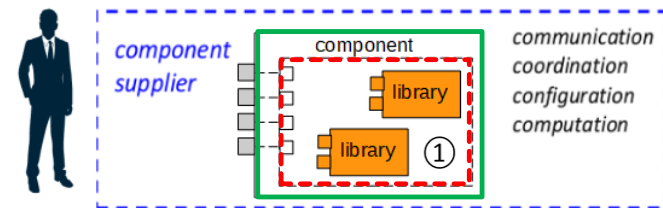
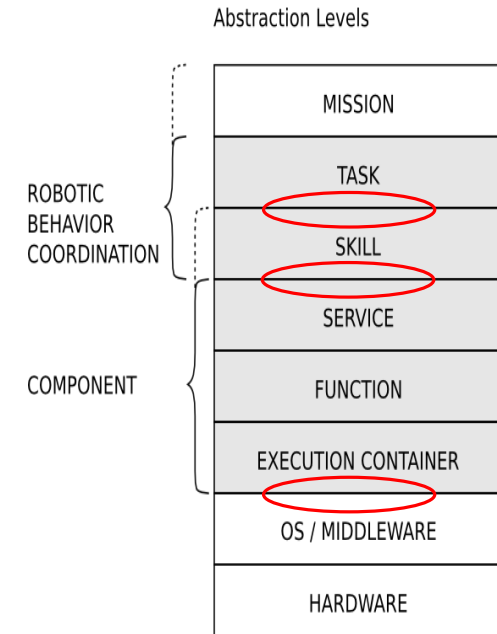
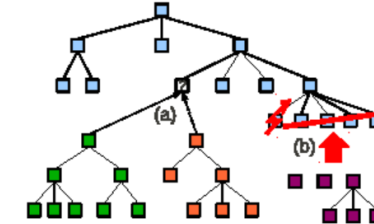
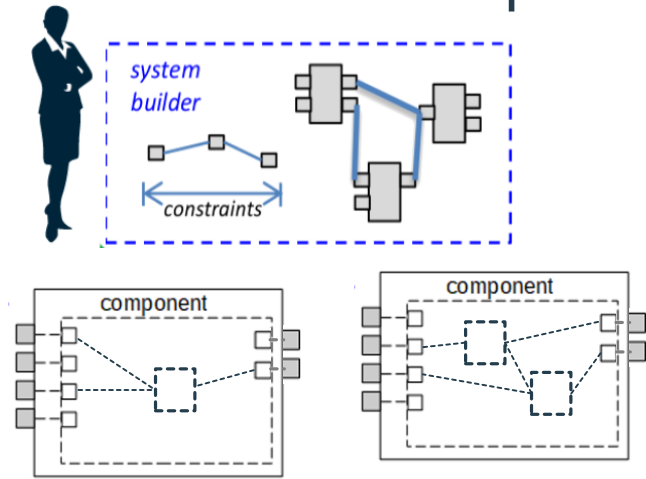
Which patterns and structures form the **sweet spot** between **freedom of choice** and **freedom from choice**?



Composition, Blocks, Ports, Connectors, Data Sheets, Models

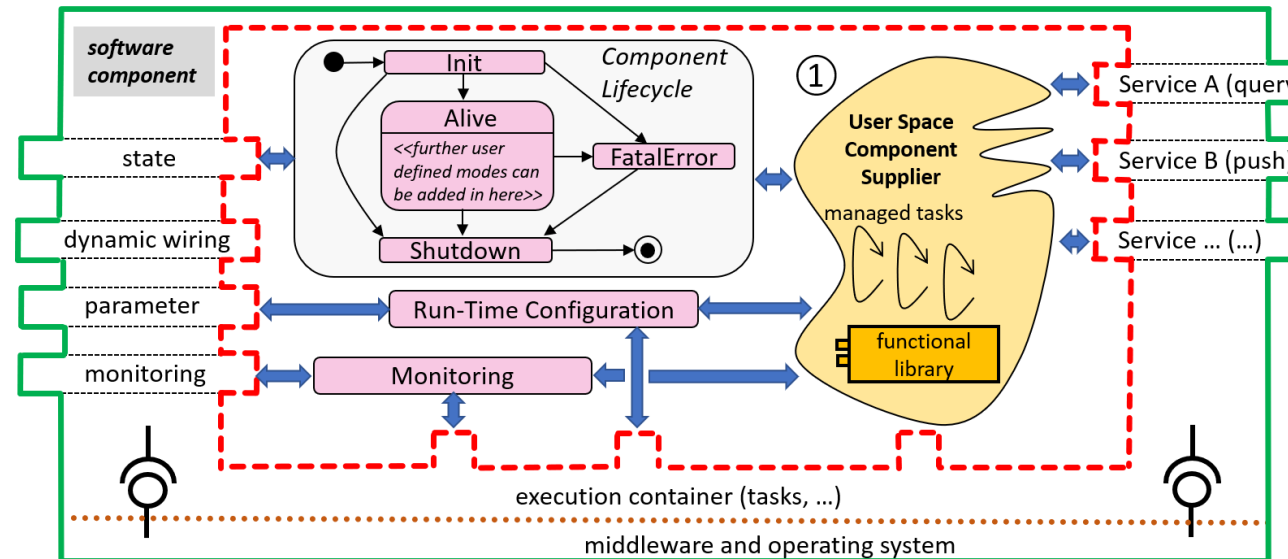


RobMoSys



S/W component with

- communication (service-oriented ports),
- configuration (resources, parameters),
- coordination (modes, lifecycle),
- computation



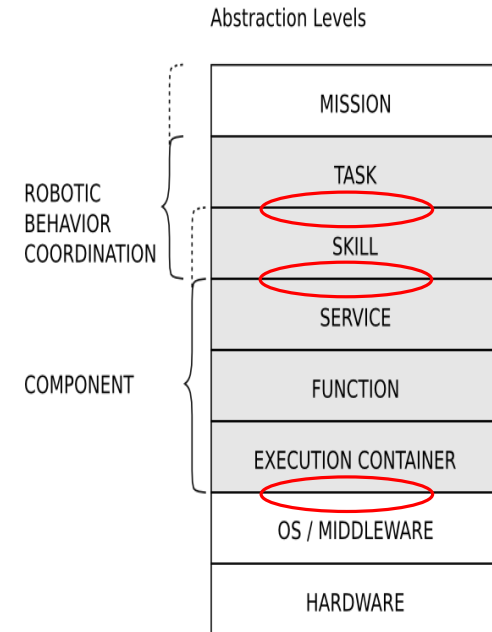
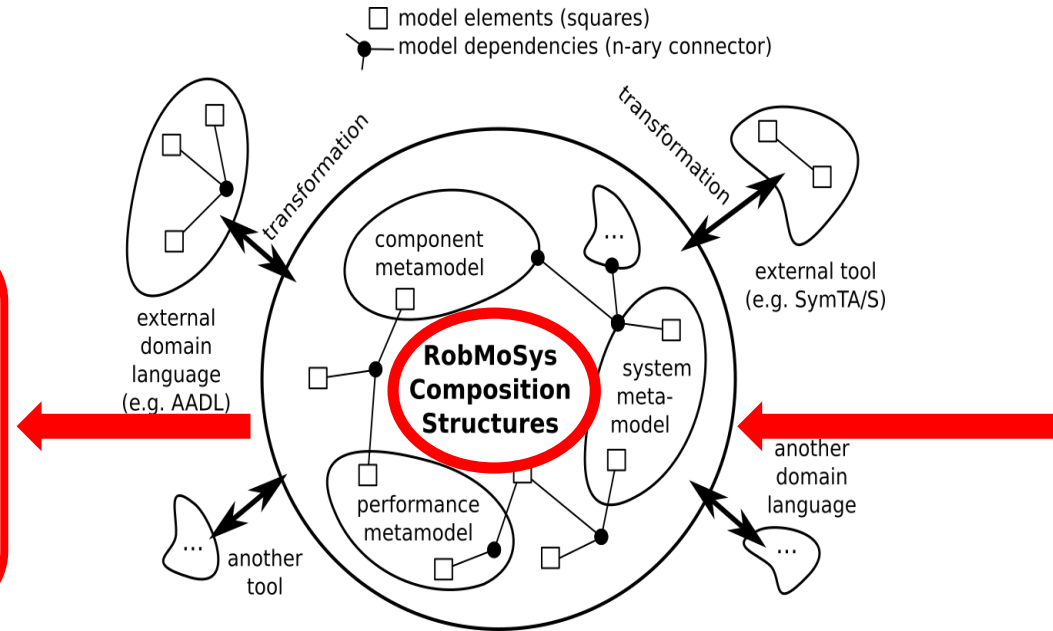
Composition, Blocks, Ports, Connectors, Data Sheets, Models



RobMoSys



Architectural Pattern for Communication
 Architectural Pattern for Component Coordination
 Architectural Pattern for Software Components
 Architectural Pattern for Managing Transition of System States
 Architectural Pattern for Task-Plot Coordination (Robotic Behaviors)
 Architectural Pattern for Service Definitions
 Architectural Pattern for Stepwise Management of Extra-Functional Properties
 Architectural Pattern for Coordinate-Frame Transformation
 Architectural Pattern for Reservation Based Resource Management
 ...

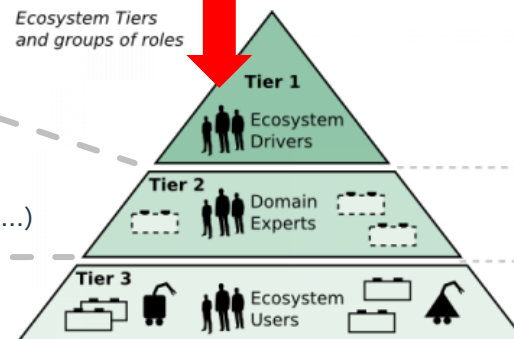


Meta-Model:
domain-independent

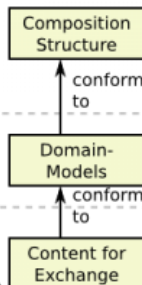
Model:
domain-specific
(mobile robots, intralogistics, manipulation, ...)

Implementation:
components and systems

Ecosystem Tiers and groups of roles



Tier Elements



Toolings

Tier 1 content:
Modeling Foundations &
Composition Structures

Tier 2 content:
Domain Models & Stacks

Tier 3 content:
Components & Systems

- Methodology
- Meta Models
- Models
- Implementation Technologies
- Toolings
- Building Blocks
- Pilot Applications
- Repositories
- Processes

Example I4.0:



Companion
Specification

Conformant
Devices

Composition, Blocks, Ports, Connectors, Data Sheets, Models



RobMoSys

PICTURE QUALITY

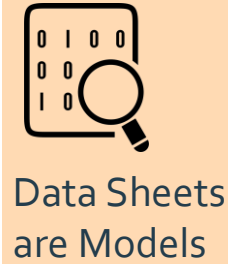
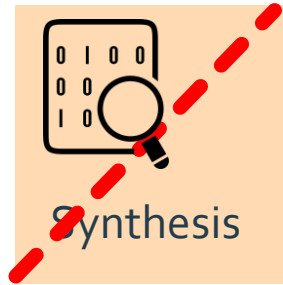
- Full HD 1080p
- Motion Rate 60
- Wide Color Enhancer

SMART

- Smart TV
- Full Web Browser

CONNECTIONS

- 2 HDMI® Connections
- 2 USB Connections
- 802.11n Wi-Fi Built In
- 1 Component in
- 1 Composite In (Shared with AV Component input)



Describes outside view, including internals only as far as you need to know them for using the asset and for becoming able to predict its fit (behavior, structure) for your context

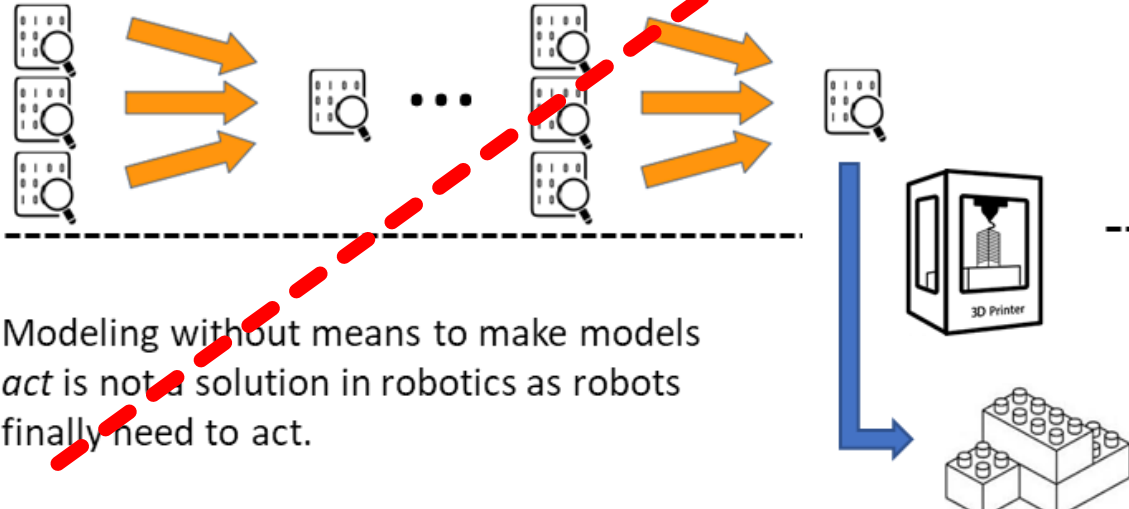
Composing different models for a full-fledged model for synthesis as the last step in the workflow so far only works in selected use-cases of 3D-printing.

from models to models

enrich, combine, analyze, predict, ...

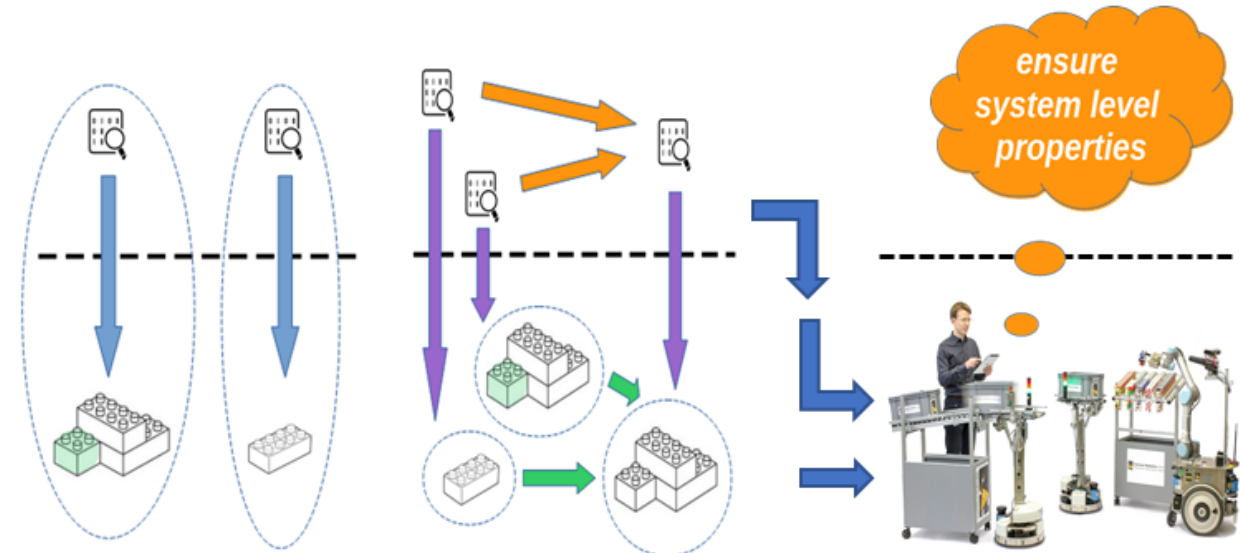
from models to models

enrich, combine, analyze, predict, ...



Modeling without means to make models *act* is not a solution in robotics as robots finally need to act.

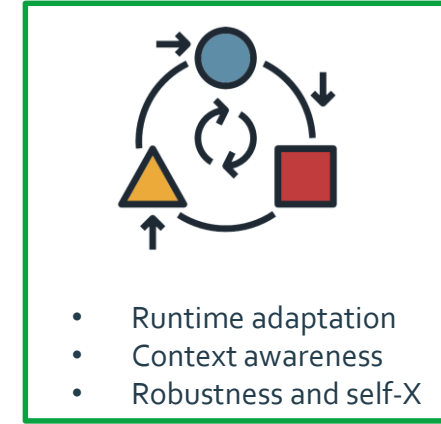
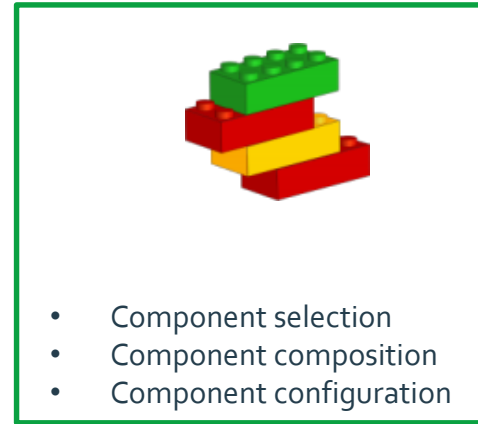
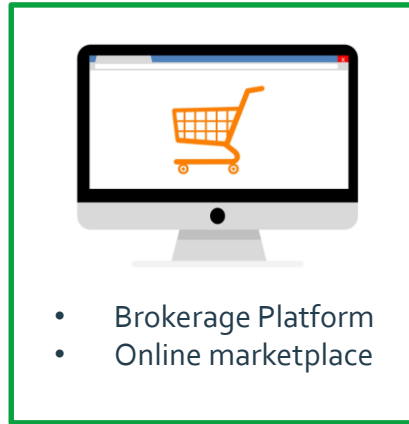
Data sheets (models of artefacts that act) *represent* components, sub-systems, task-plots etc. Suitability, traceability, simulation, etc. of system properties all via *composed data sheets*. When all is fine, then *compose* (put together and accordingly configure) the real artefacts to get the real system with properties as expected.



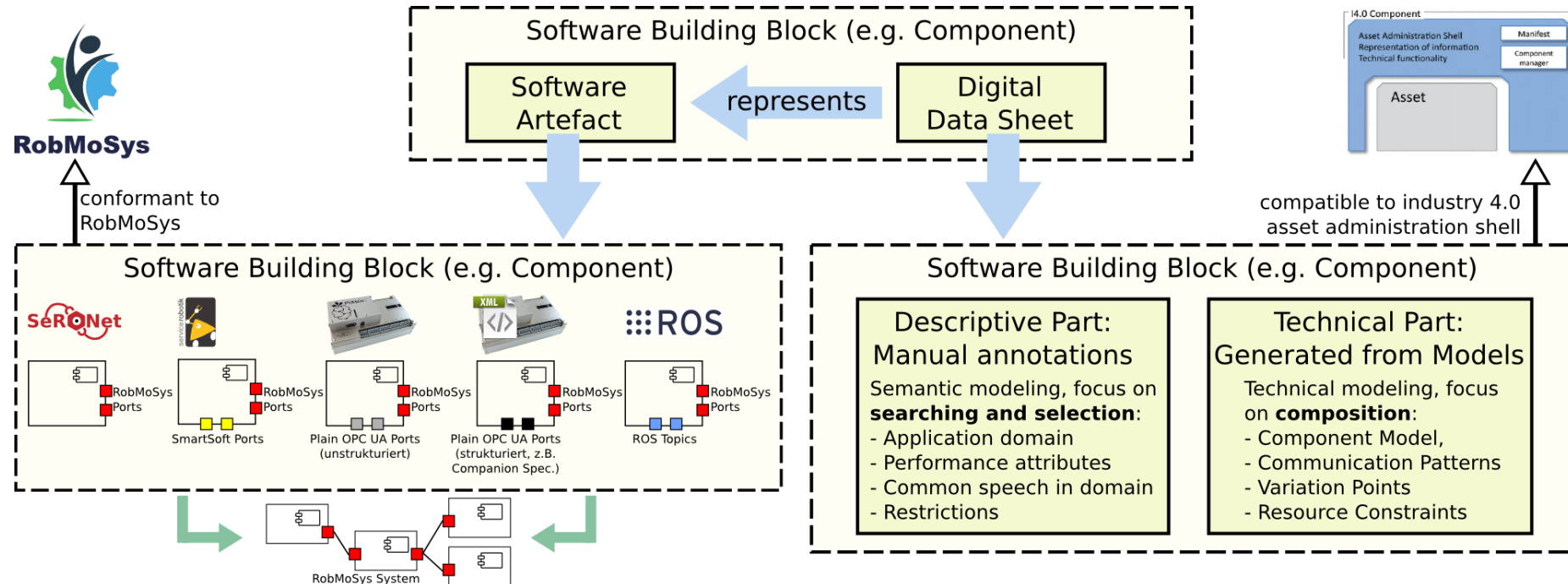
Composition, Blocks, Ports, Connectors, Data Sheets, Models

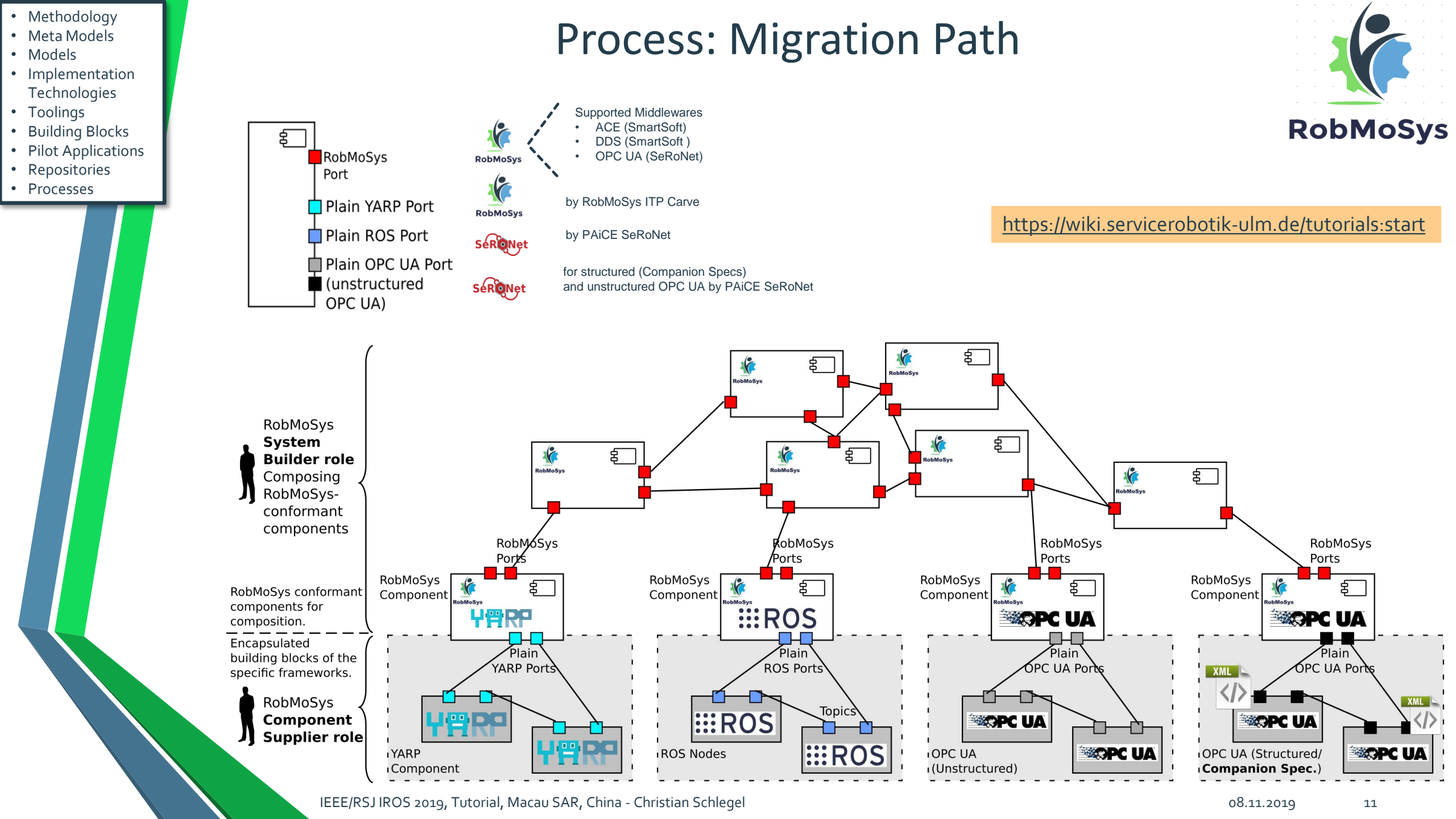


RobMoSys



**Digital Data Sheet
as Submodel of
Industry 4.0 Asset
Administration Shell**





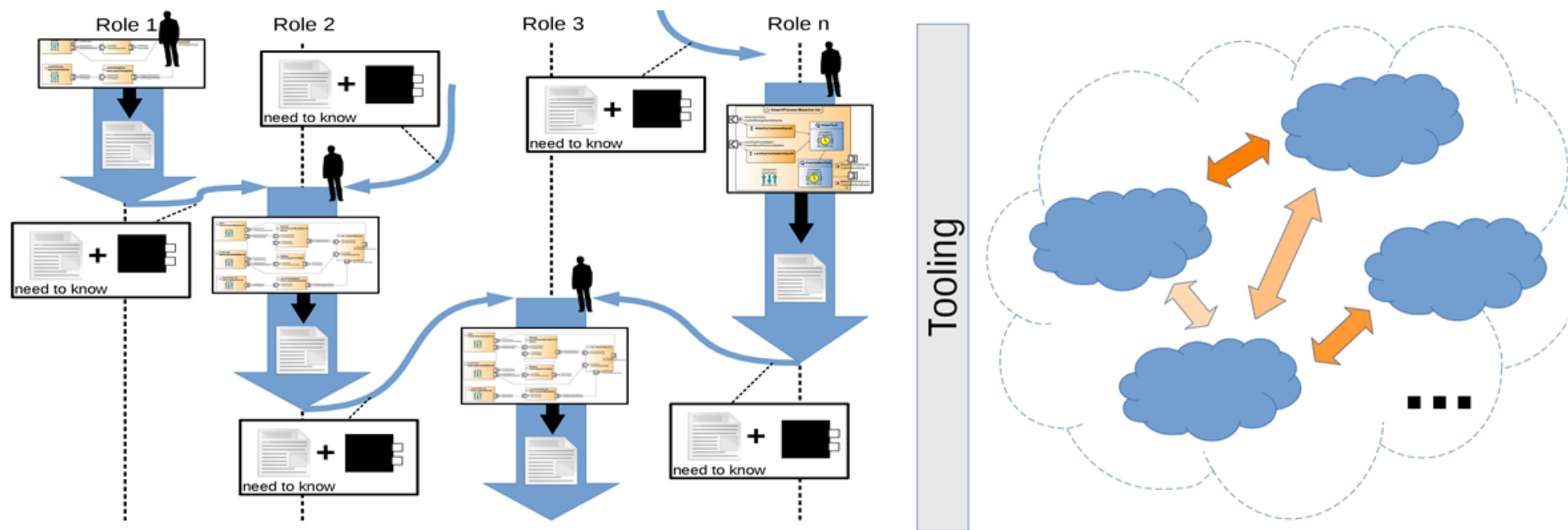
Process: Coverage and Conformance



RobMoSys

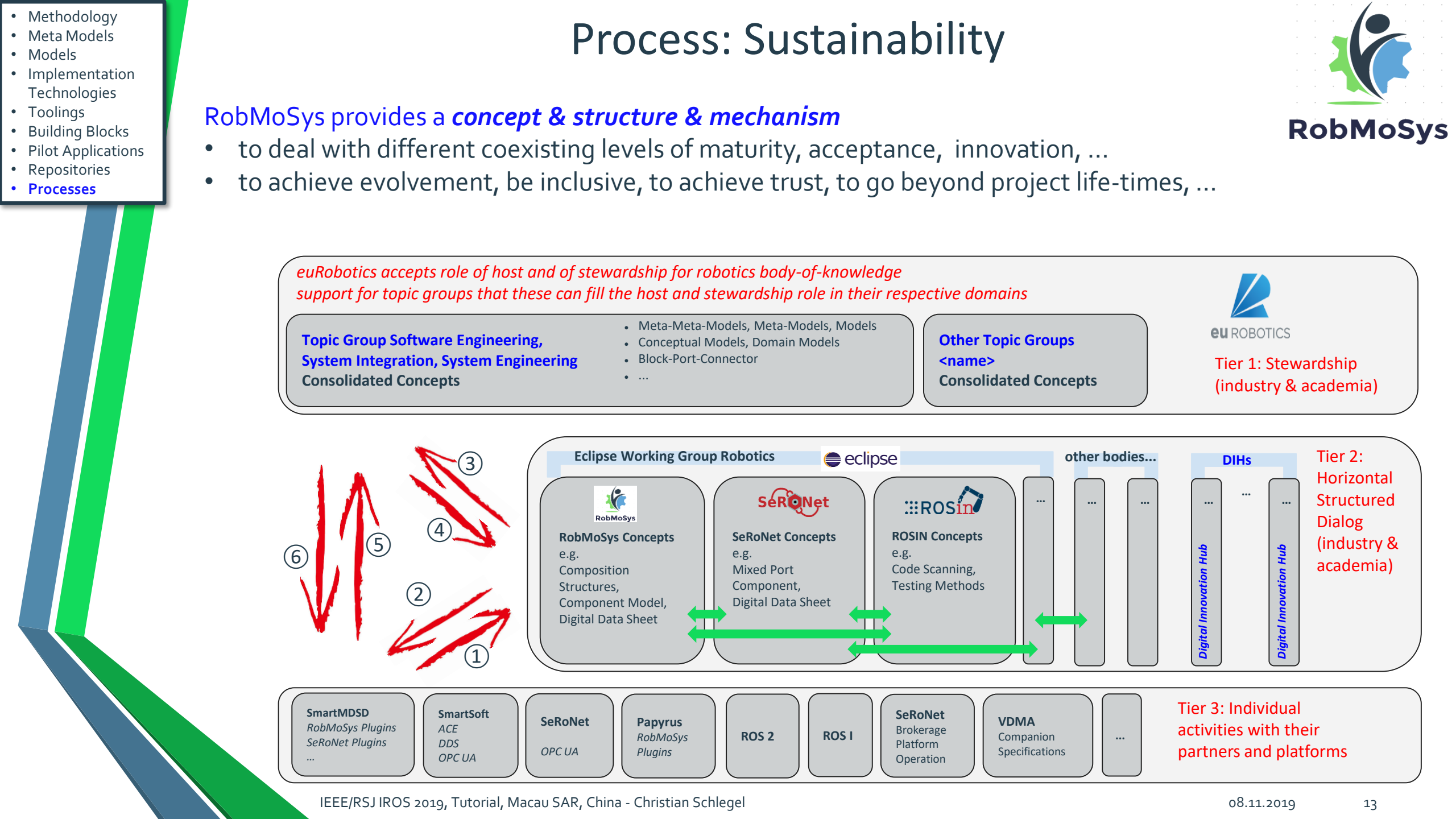
RobMoSys provides a **concept & structure & mechanism**

- to deal with different coexisting levels of maturity, acceptance, innovation, ...
- to achieve evolvement, be inclusive, to achieve trust, to go beyond project life-times, ...




Build upon the RobMoSys composition structures and align with them!

Wiki „Incubator“ => Wiki „Stable Body-of-Knowledge“





- Methodology
- Meta Models
- Models
- Implementation Technologies
- Toolings
- Building Blocks
- Pilot Applications
- Repositories
- Processes

RobMoSys Open Access Repositories




RobMoSys

Tier 3 Systems

Name	Description	Purpose	Vendor	Tooling	Status	Figure
 SystemTiagoNavigation	A pilot skeleton that covers the navigation aspect of the Intralogistics Industry 4.0 Robot Fleet Pilot and Assistive Mobile Manipulation Pilot. This system covers the TIAGo Robot in simulation/Gazebo.	Navigation	HSU	SmartMDSD Toolchain v3	Ready	
	ot skeleton that covers navigation aspect of intralogistics Industry	Navigation	HSU	SmartMDSD Toolchain v3	Ready	-

<https://robmosys.eu/wiki/model-directory:start>



RobMoSys Wiki

<http://www.robmosys.eu>







Recent Changes Media Manager Sitemap Mainpage Imprint

You are here: [RobMoSys Wiki](#) » [RobMoSys Model Directory](#)




RobMoSys Model Directory

A list of domain models, software components and systems for use with RobMoSys Tooling. Please see end of page for a legend.

Tier 2 Domain Models

Name	Description	Purpose	Vendor	Tooling
 CommBasicObjects	A collection of very basic service definitions and communication objects for use in almost every robotics system.	Universal	HSU	Smart Toolchain
 CommNavigationObjects	A collection of domain models for wheeled robot navigation .	Navigation	HSU	Smart Toolchain
 CommRobotinoObjects	A collection of domain models for use with the FESTO Robotino robot.	Mobile-Base	HSU	Smart Toolchain
 CommLocalizationObjects	A collection of domain models for localization .	Localization	HSU	Smart Toolchain
 CommManipulationPlannerObjects	A collection of domain models for (mobile) manipulation.	Mobile Manipulation	HSU	Smart Toolchain
 CommManipulatorObjects	A collection of domain models for manipulators.	Manipulation	HSU	Smart Toolchain

Tier 3 Component Models

Name	Description	Purpose	Vendor	Tooling	Status
 SmartCdlServer	Implements the Curvature Distance Lookup (CDL) algorithm for fast local obstacle avoidance. It considers the dynamics and kinematics of the robot, as well as its polygonal shape.	Navigation	HSU	SmartMDSD Toolchain v3	Ready
 ComponentLaserObstacleAvoid	The SmartLaserObstacleAvoid component implements a simple reactive obstacle avoidance.	Navigation	HSU	SmartMDSD Toolchain v3	Ready
 ComponentPlayerStageSimulator	The SmartPlayerStageSimulator simulates a robot in a 2D bitmapped environment using Player/Stage. It offers several services for controlling the robot, such as sending navigation commands, providing access to the robot's odometry and laser scans.	Simulation	HSU	SmartMDSD Toolchain v3	Ready

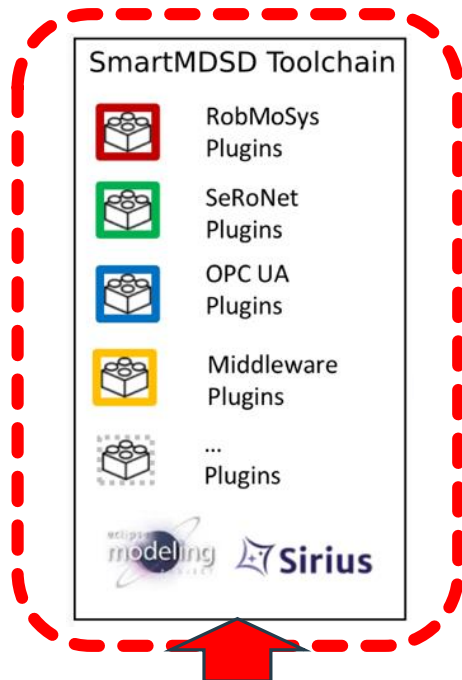
IEEE/RSJ IROS 2019, Tutorial, Macau SAR, China - Christian Schlegel

08.11.2019

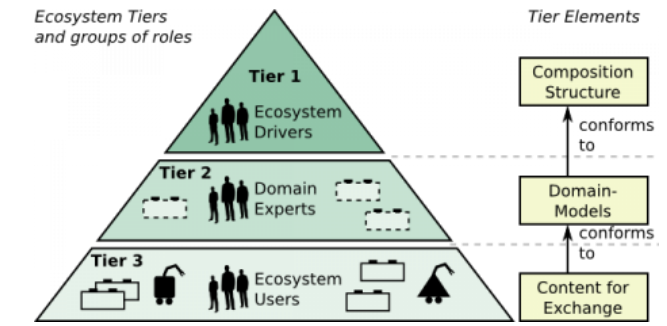
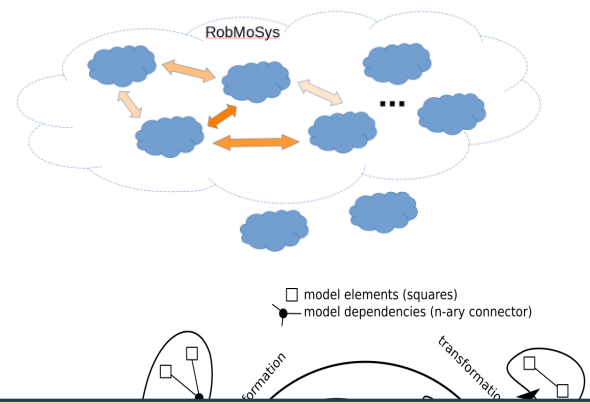
14

- Methodology
- Meta Models
- Models
- Implementation Technologies
- Toolings
- Building Blocks
- Pilot Applications
- Repositories
- Processes

RobMoSys Conformant Toolings



eITUS
Safety View for
Papyrus4Robotics



Composing a Robotics Application in a Day – A low code approach
We make Robotics Software Systems Engineering easier!



<https://wiki.servicerobotik-ulm.de/start>

<https://wiki.servicerobotik-ulm.de/smartmdsd-toolchain:start>

- **one-click download** of the full Open-Source Eclipse-based development environment
- **start development with zero installation effort**
- comes with Gazebo-Simulator and all kinds of components, stacks, pilot applications, tutorials, etc.
- skill-based engineering, task-level coordination, robot fleet coordination, graphical tools for end-users
- fully middleware-agnostic: ACE, DDS, OPC UA, etc.
- mixed-port component as migration path: link to ROS, I4.0 OPC UA, etc.

<https://robmosys.eu/wiki/baseline:start>

<https://robmosys.eu/wiki/jumppage>

<https://robmosys.eu/wiki/open-call-2>

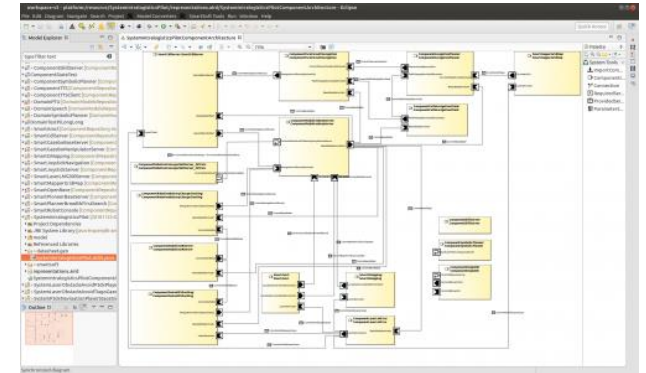
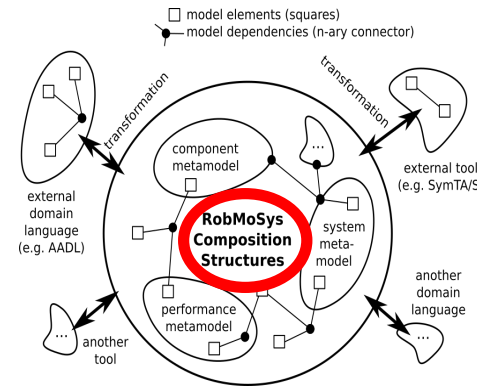
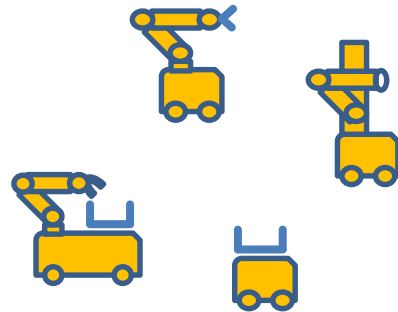
From ROS-specific coding to framework-agnostic modeling



RobMoSys

Models always have a purpose: overall purpose is consistency

- organize consistent abstraction for e.g. prediction
- better understanding in early phases avoids costs at later stages
- organize interfaces and ensure fits while decoupling roles, responsibilities, scopes, etc.
- ensure traceability of properties, conformance by design and not just by discipline, etc.



A model-driven approach allows to

- ...secure your design and solution efforts
- ...decouple different paces of evolution
- ...be technology-agnostic (**semantics: early binding / technology: late binding**)
- ...predict what you get before you build it
- ...exploit the power of combinatorics
- ...explicate otherwise hidden magic numbers
- ...benefit from low effort in modifications towards lot size 1
- ...achieve robust job fulfillment by context-aware run-time decisions

Schedule



RobMoSys

09:00 – 09:20 **Towards an Open and Industry-Grade European Robotics Software Ecosystem**
Christian Schlegel, Technische Hochschule Ulm

09:20 – 10:00 **Interactive Tool Demo: Open Source Eclipse-based Tooling for System Composition:
Piecing together software components to pilot applications**
Alex Lotz, Technische Hochschule Ulm



Service Robotics Ulm
autonomous mobile service robots

10:00 – 10:45 **Interactive Tool Demo: Open Source Eclipse-based Tooling for Component Builders:
Middleware-agnostic robotics software components**
Alex Lotz, Technische Hochschule Ulm

<https://wiki.servicerobotik-ulm.de/start>

<https://wiki.servicerobotik-ulm.de/smartmdsd-toolchain:start>

10:45 – 11:15 **Coffee Break**

11:15 – 11:55 **Interactive Tool Demo: Safety-analysis by Model-Driven Tooling**
Selma Kchir, CEA List

11:55 – 12:35 **Interactive Tool Demo: Stepwise Migration to Model-Driven Development:
Linking Legacy Systems via the Mixed-Port Component, Linking ROS-Systems, OPC UA Systems, etc.**
Alex Lotz, Technische Hochschule Ulm

12:35 – 12:55 **The Role of Higher-order Models in Robotics and its Reasoning Challenges**
Herman Bruyninckx, KU Leuven

12:55 – 13:00 **Wrap Up: Summary of what to find where, how to get involved etc.**
Christian Schlegel, Technische Hochschule Ulm