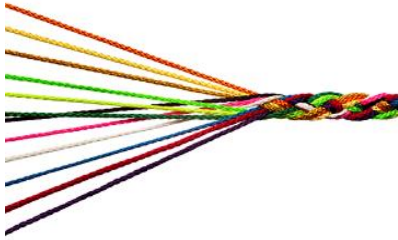
Composition, Separation of Roles and Model-Driven Approaches as Enabler of a Robotics Software Ecosystem

Towards an EU Digital Industrial Platform for Robotics

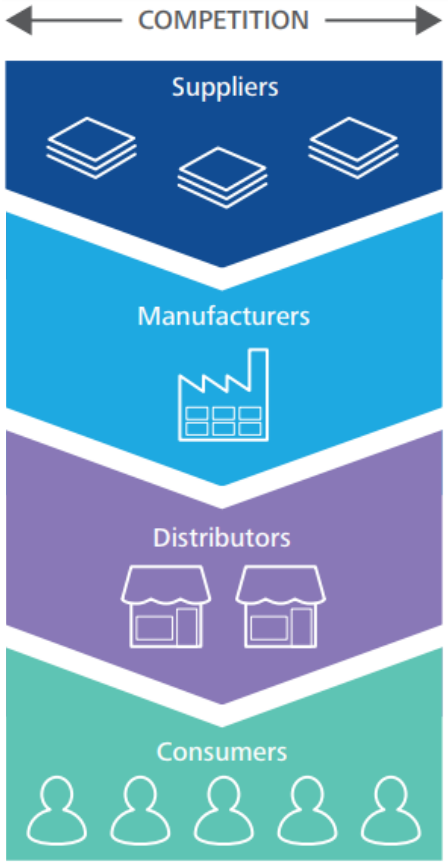
Prof. Dr. Christian Schlegel, Technische Hochschule Ulm, www.servicerobotik-ulm.de, christian.schlegel@thu.de

- Technical Lead of EU H2020 Project **RobMoSys** – Better Models, Better Tools, Better Systems
- Responsible for the Model-Driven Tooling in the BMWi PAiCE Project **SeRoNet**
- Coordinator of the euRobotics aisbl **Topic Group** on Software Engineering, System Integration, System Engineering
- Team is maintainer of the **SmartSoft** world including the Eclipse-based **SmartMDSD** Tooling



Towards an EU Digital Industrial Platform for Robotics

- The world is entering an era in which ideas and insights come from everywhere, and crowds, clouds, collaborators, competitions, and co-creators can fundamentally help define our shared future. The business environment is being permanently altered as a result.
- Ecosystems are dynamic and co-evolving communities of diverse actors who create and capture new value through both collaboration and competition.



Business ecosystems come of age, Deloitte University Press, 2015

changeability / flexibility of robotics not yet economically exploitable to face pressure of lot size 1

robots are not yet a tool which I can adjust myself to match my daily changing needs

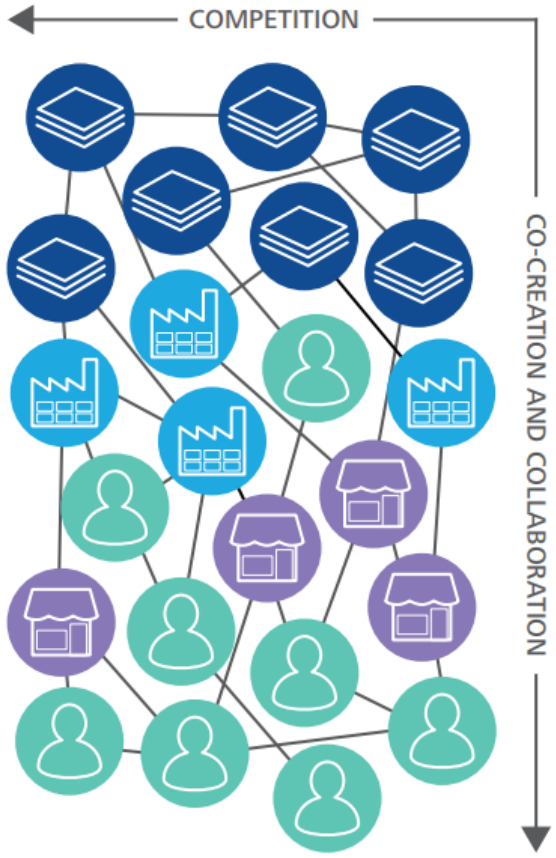
in robotics, costs of change are not in proper relation to similarity of solution

expectations of many domains to use robots cannot be fulfilled due to too high effort / costs

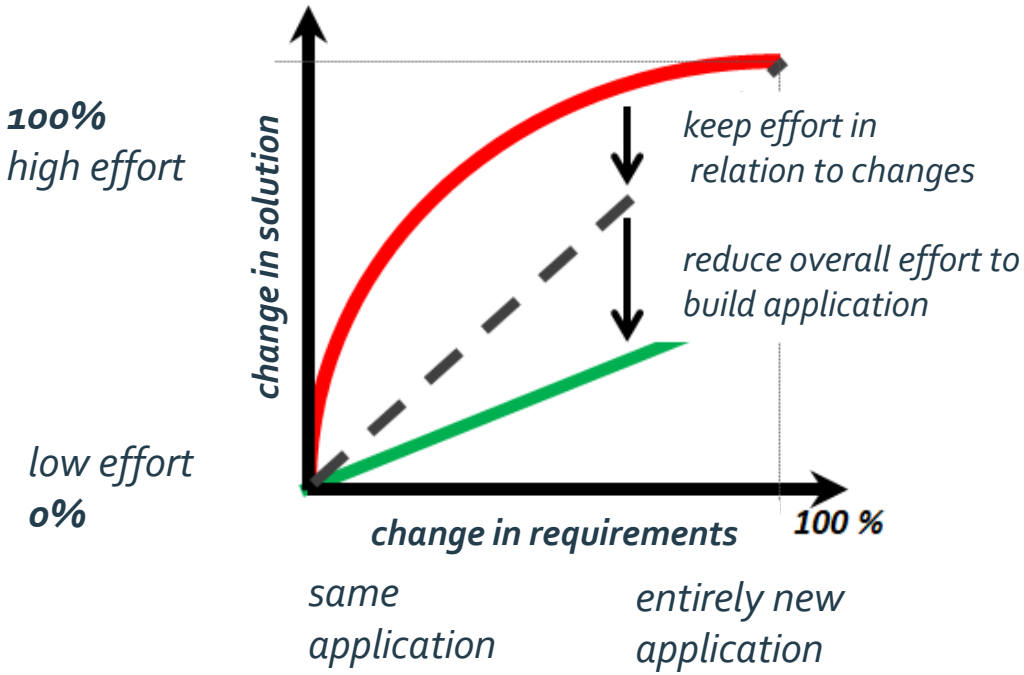
...

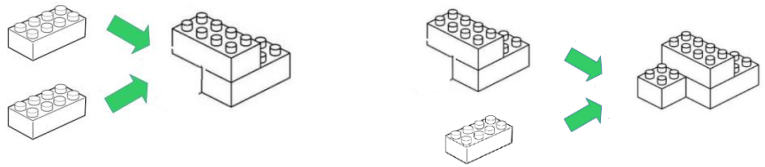
For the most part, supply chain functions of large businesses weren't set up to deal with a world of thousands of partners. Now they must adjust.

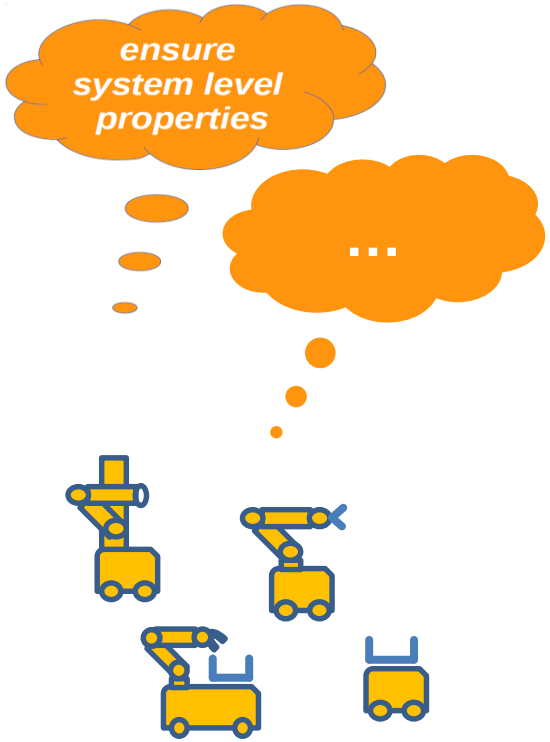
A distinctive characteristic of many ecosystems is that they form to achieve something together that lies beyond the effective scope and capabilities of any individual actor (or even group of broadly similar actors).



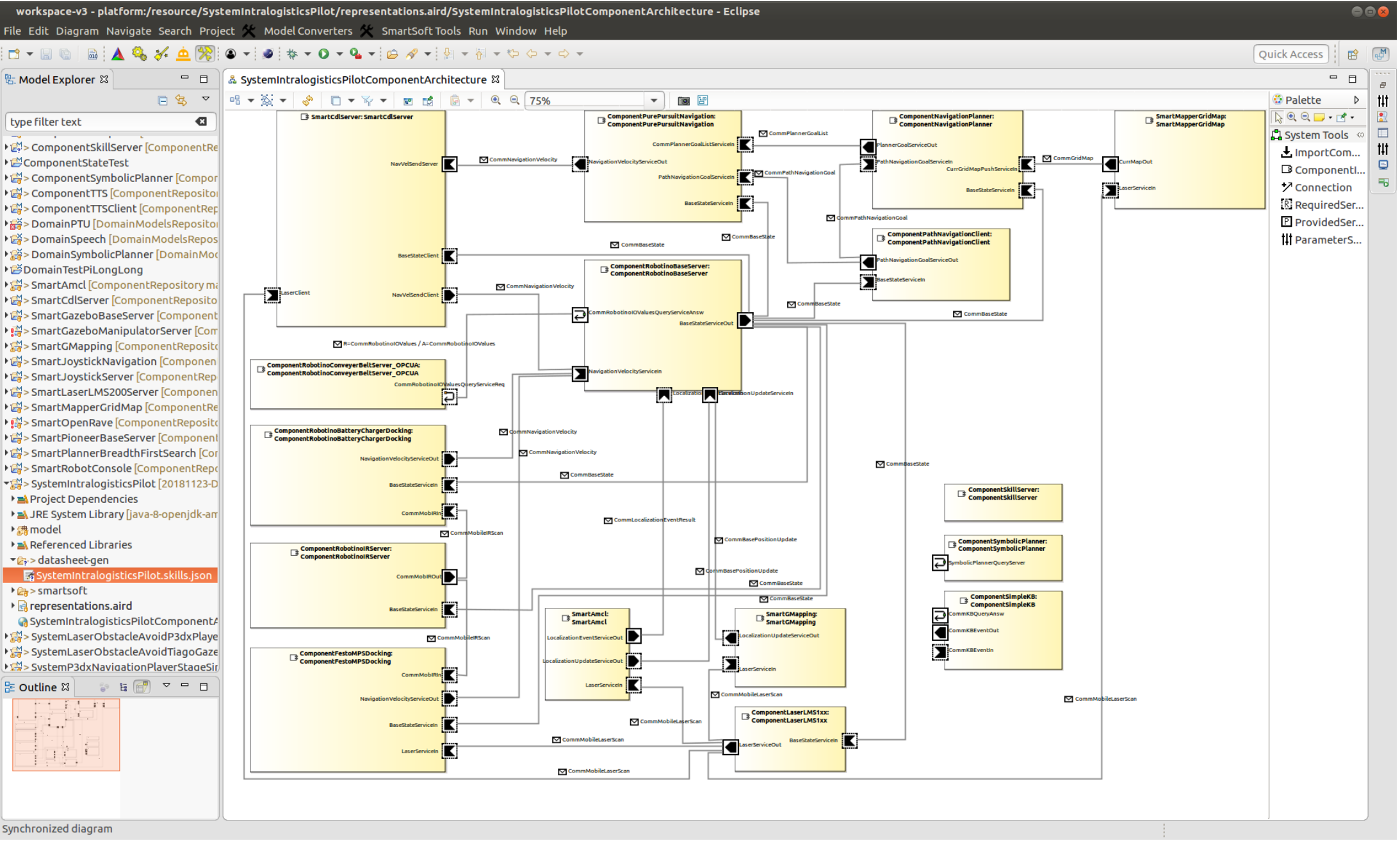
Towards an EU Digital Industrial Platform for Robotics







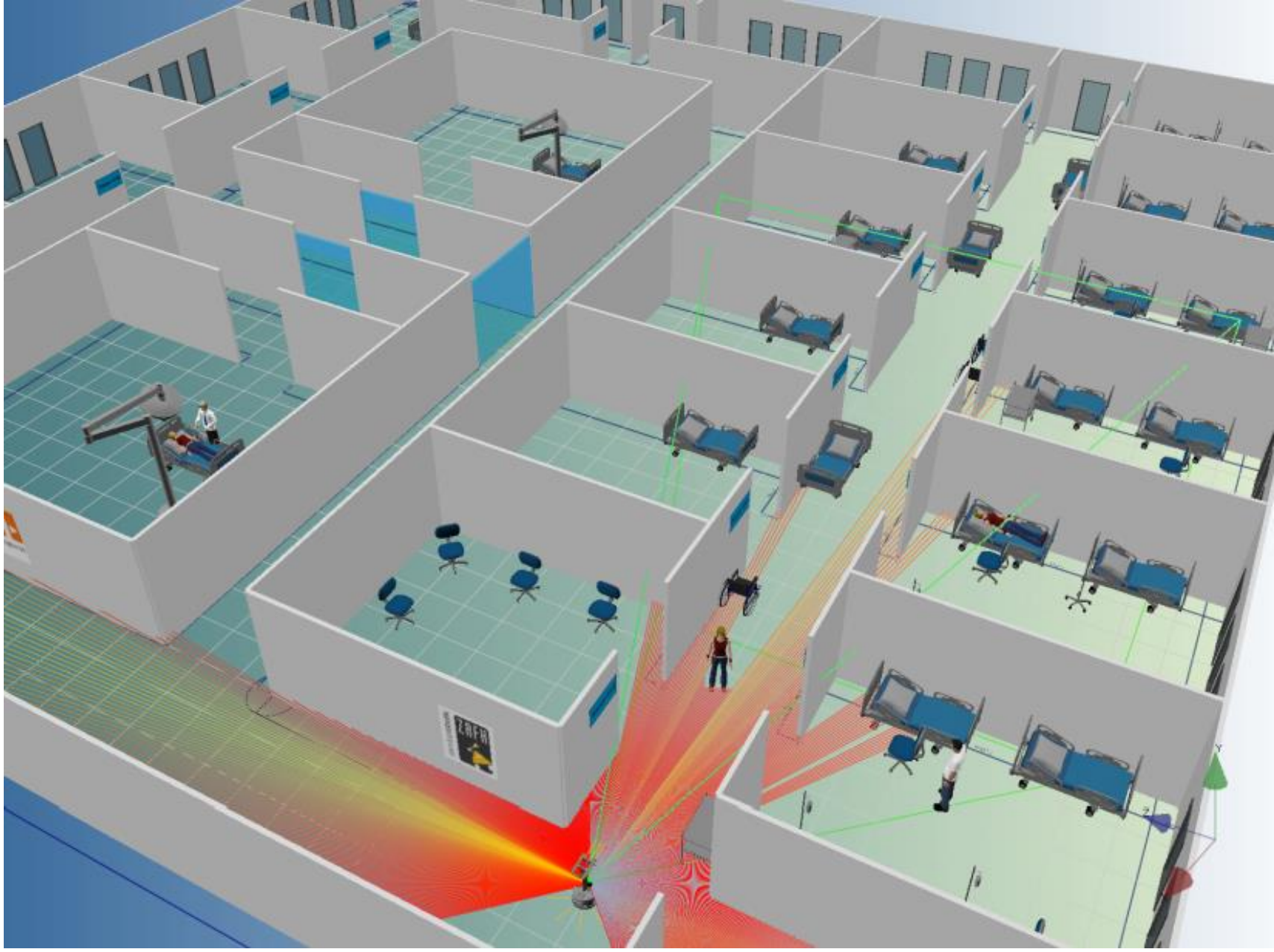
- Can we think of complex robotic systems ***before*** we build them?
- Can we answer ***„what if"*** questions and can we find ***adequate*** solutions?
- Can we put systems together out of ***configurable „as is" building blocks***?
- Can we keep the behavior when we e.g. exchange the middleware?
- **Can we bring effort & costs in relation to similarity of an application?**
- Can we build **adequate** solutions with adequate effort?
- Can we **explain** why the system does what?
- Can we generate enough **trust** into the systems – and how and by what means?


workspace-v3 - platform:/resource/SystemIntralogisticsPilot/representations.aird/SystemIntralogisticsPilotComponentArchitecture - Eclipse
File Edit Diagram Navigate Search Project Model Converters SmartSoft Tools Run Window Help
Quick Access
Model Explorer
type filter text
ComponentSkillServer [ComponentRe
ComponentStateTest
ComponentSymbolicPlanner [Compor
ComponentTTS [ComponentReposito
ComponentTTSClient [ComponentRep
DomainPTU [DomainModelsReposito
DomainSpeech [DomainModelsRepos
DomainSymbolicPlanner [DomainMod
DomainTestPiLongLong
SmartAmcl [ComponentRepository ma
SmartCdlServer [ComponentReposito
SmartGazeboBaseServer [Component
SmartGazeboManipulatorServer [Com
SmartGMapping [ComponentReposito
SmartJoystickNavigation [Componen
SmartJoystickServer [ComponentRep
SmartLaserLMS200Server [Componen
SmartMapperGridMap [ComponentRe
SmartOpenRave [ComponentReposito
SmartPioneerBaseServer [Component
SmartPlannerBreadthFirstSearch [Cor
SmartRobotConsole [ComponentRepo
SystemIntralogisticsPilot [20181123-D
Project Dependencies
JRE System Library [java-8-openjdk-am
model
Referenced Libraries
datasheet-gen
SystemIntralogisticsPilot.skills.json
smartsoft
representations.aird
SystemIntralogisticsPilotComponentA
SystemLaserObstacleAvoidP3dxPlaye
SystemLaserObstacleAvoidTiagoGaze
SystemP3dxNavigationPlayerStageSi
Outline
SystemIntralogisticsPilotComponentArchitecture
75%
SmartCdlServer: SmartCdlServer
ComponentPurePursuitNavigation: ComponentPurePursuitNavigation
ComponentNavigationPlanner: ComponentNavigationPlanner
SmartMapperGridMap: SmartMapperGridMap
ComponentPathNavigationClient: ComponentPathNavigationClient
ComponentRobotinoBaseServer: ComponentRobotinoBaseServer
ComponentRobotinoConveyerBeltServer_OPCUA: ComponentRobotinoConveyerBeltServer_OPCUA
ComponentRobotinoBatteryChargerDocking: ComponentRobotinoBatteryChargerDocking
ComponentRobotinoIRServer: ComponentRobotinoIRServer
ComponentFestoMPSDocking: ComponentFestoMPSDocking
SmartAmcl: SmartAmcl
SmartGMapping: SmartGMapping
ComponentLaserLMS1xx: ComponentLaserLMS1xx
ComponentSkillServer: ComponentSkillServer
ComponentSymbolicPlanner: ComponentSymbolicPlanner
ComponentSimpleKB: ComponentSimpleKB
Palette
System Tools
ImportCom...
ComponentI...
Connection
RequiredSer...
ProvidedSer...
ParameterS...
Synchronized diagram

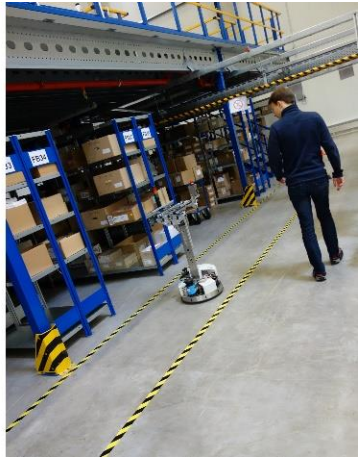
Servicerobotics Research Group / Technische Hochschule Ulm

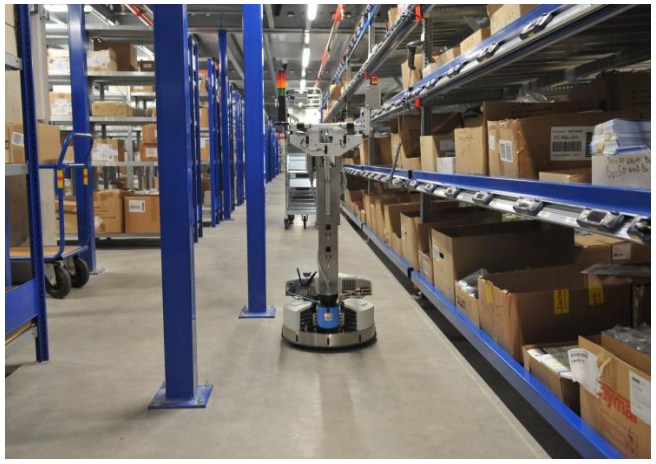
https://www.youtube.com/user/RoboticsAtHsUlm

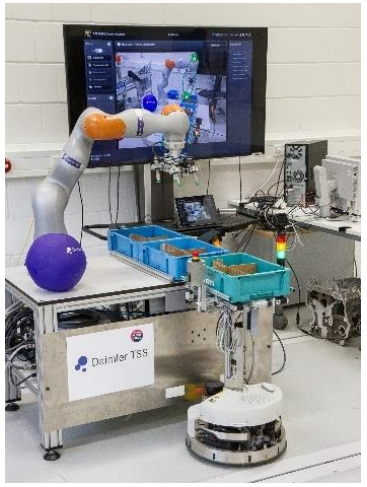
https://www.fendt.com/de/xaver



https://doi.org/10.2314/GBV:87332112X







https://www.youtube.com/user/RoboticsAtHsUlm

Service Robotics Ulm

autonomous mobile service robots

www.servicerobotik-ulm.de

Towards an EU Digital Industrial Platform for Robotics
Projects EU H2020 RobMoSys and BMWi PAiCE SeRoNet

RobMoSys

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

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

cea · Hochschule Ulm University of Applied Sciences · KU LEUVEN · TUM Technische Universität München · COMAU · PAL ROBOTICS · SIEMENS · EUnited Robotics European Robotics Association · eclipse

conforms to

SeRoNet

Gefördert durch:
Bundesministerium für Wirtschaft und Energie
aufgrund eines Beschlusses des Deutschen Bundestages

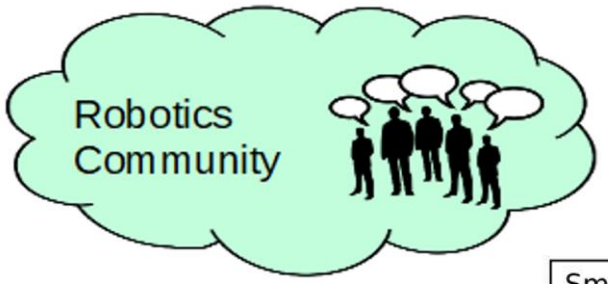
PAiCE

https://www.seronet-projekt.de/plattform/tooling.html

Fraunhofer IPA · ISW · THU Technische Hochschule Ulm · RUHRBOTICS We Program Your Success · TRANSPHARM competence.connection.courage

KUKA · FZI · Universität Paderborn · Daimler TSS · KMD Klinikum Mannheim Dienstleistungsgesellschaft mbH

RobMoSys is more than just another project…



RobMoSys

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.

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

https://robmosys.eu/wiki/jumppage

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

#ROBOTICS

Service Robotics Ulm
autonomous mobile service robots

SmartMDSD Toolchain
- RobMoSys Plugins
- SeRoNet Plugins
- OPC UA Plugins
- Middleware Plugins
- ... Plugins

modeling Sirius

eITUS

Safety View for **Papyrus4Robotics**

RoQME
Plugins for the **SmartMDSD** Toolchain

Mood2Be
BehaviorTree.CPP: Execution engine for behavior trees

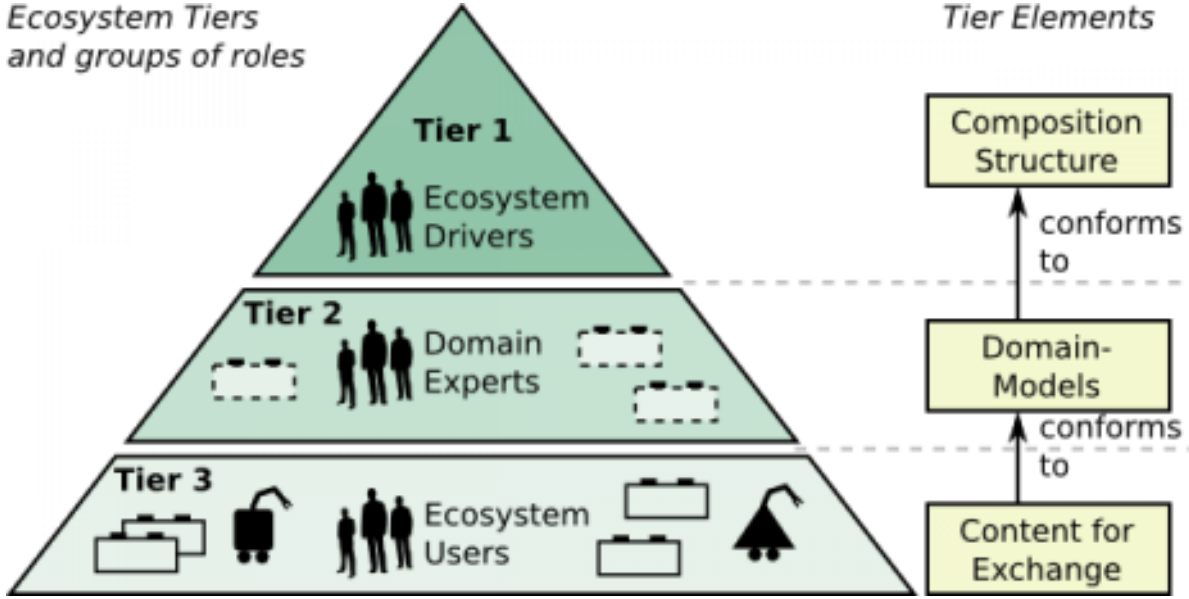
Mood2Be
Groot, an IDE to create, modify and monitor BehaviorTrees

CARVE
YARP Mixed Port Component with **SmartMDSD**

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

EG-IPC

Meta-Models
Models



The SmartMDSD Tooling: Conformant to RobMoSys and SeRoNet

RobMoSys

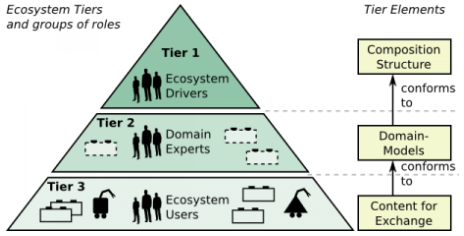
EU H2020
Composable Models and Software for Robotics Systems

conform-to

SeRoNet

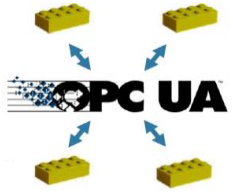
BMWi PAiCE SeRoNet

Eine Plattform zur arbeitsteiligen Entwicklung von Serviceroboter-Lösungen



technology-agnostic
generic structures
enabling composition
allow separation of roles

Brokerage platform
OPC UA as base technology



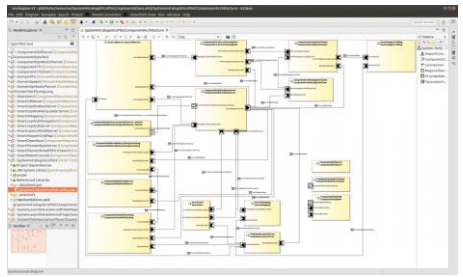
made accessible by tooling

conforms to

made accessible by tooling

conforms to

SmartMDSD Tooling



RobMoSys Plugins

SeRoNet Plugins

OPC UA Plugins

Middleware Plugins

… Plugins

eclipse modeling PROJECT

Sirius

DDS OPC UA …

ubuntu Windows 10 RTAI … x86 ARM

supports

separation of roles

supports

building blocks with data sheets:
- selection
- composition
- configuration
- implementing
- modification

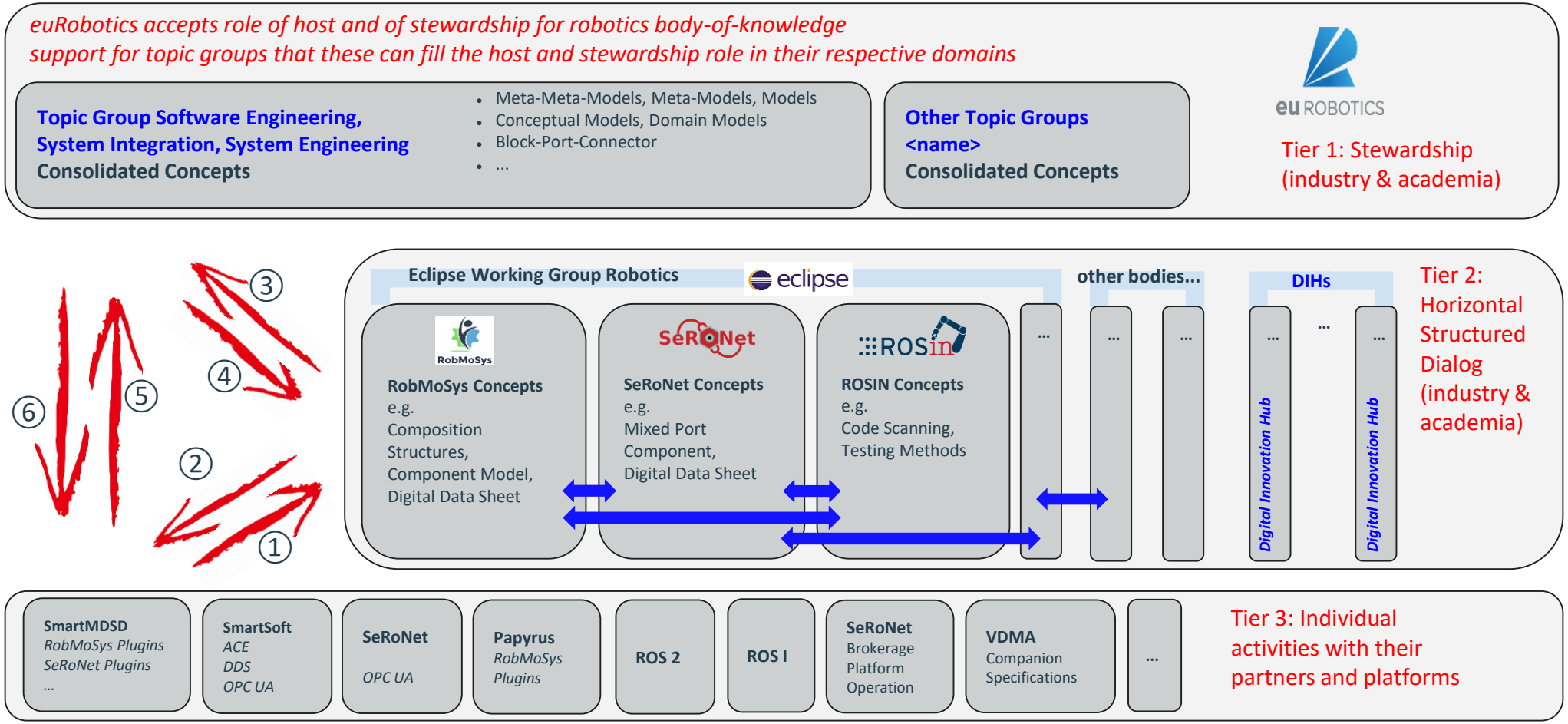
supports

import / export of
- service-definitions
- S/W-components
- robot systems
- applications

Process: Sustainability

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, …



What is the approach? What is the way of thinking?

Blocks, Ports, Connectors

Separation of Concerns

Technology Agnostic

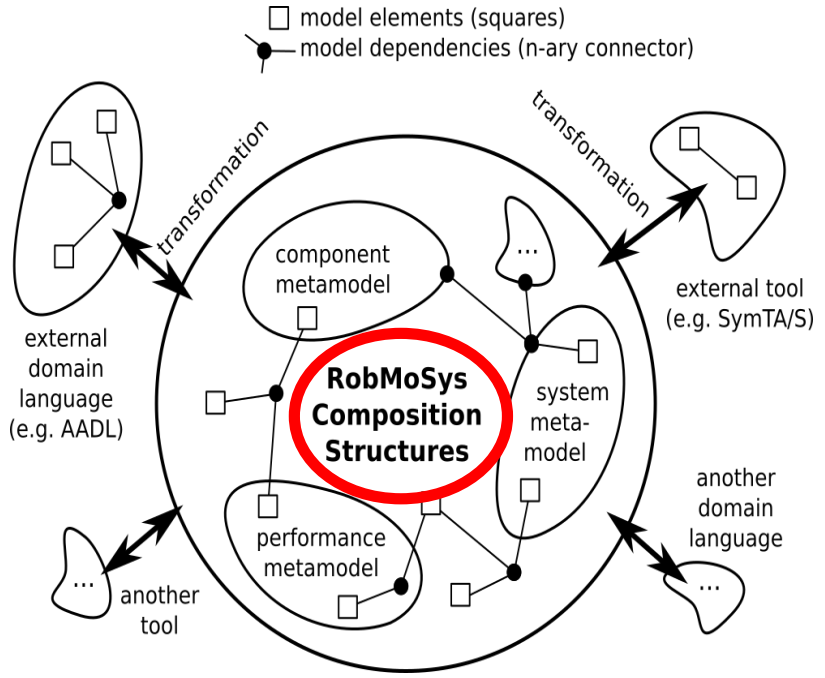
Separation of Roles

Component Based

Model-Driven

Service Oriented

Composition
Composability
Compositionality



Hooks for formal methods

Quality of Service
Resource Awareness
Adequateness

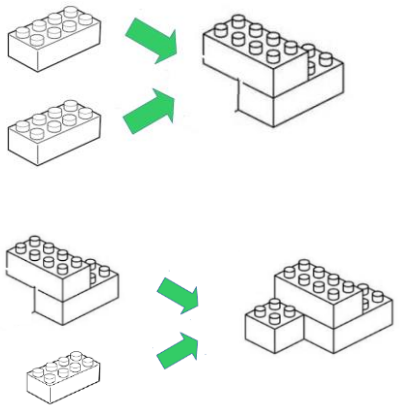
Run-time Variability for Adequateness, Robustness, ...

Digital Data Sheets

Business Ecosystem

...

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



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

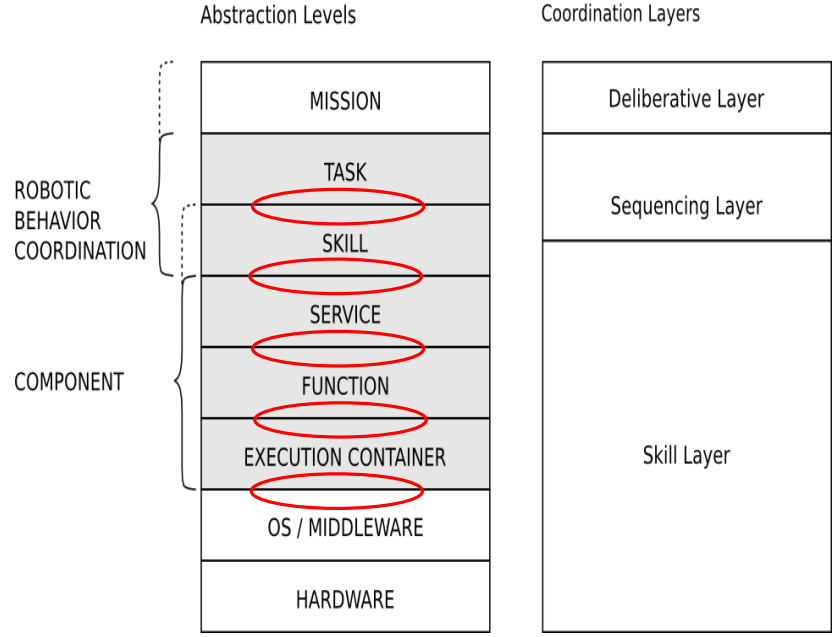
Separation of Concerns
- computation
- communication
- coordination
- configuration

splitting apart...

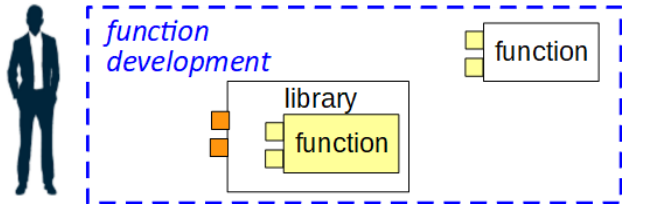
Composition, Composability, Compositionality
- computation
- communication
- coordination
- configuration

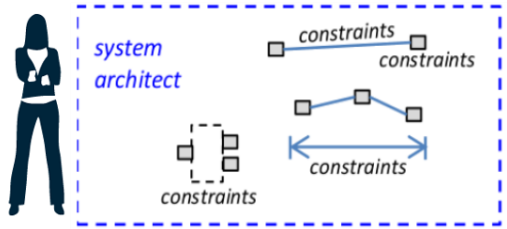
...putting together

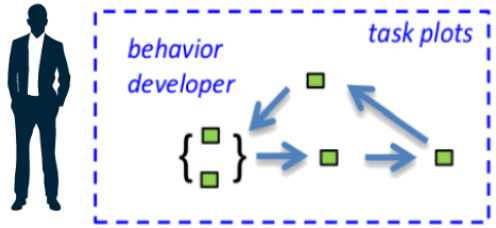
Separation of Roles!

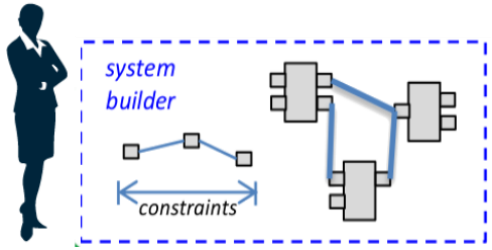


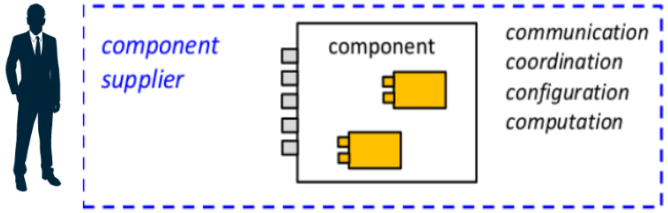
Which patterns and structures form the ***sweet spot*** between ***freedom of choice*** and ***freedom from choice***?

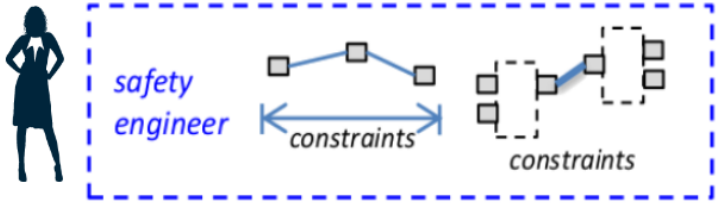






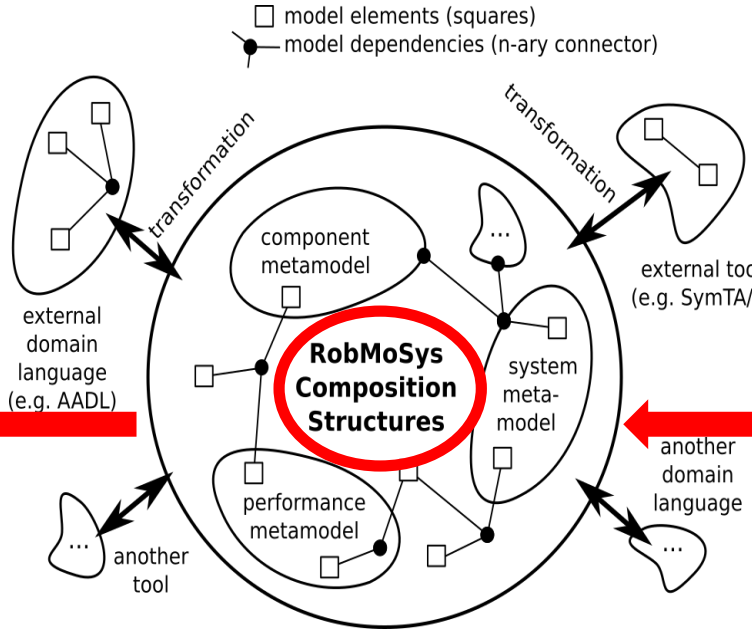


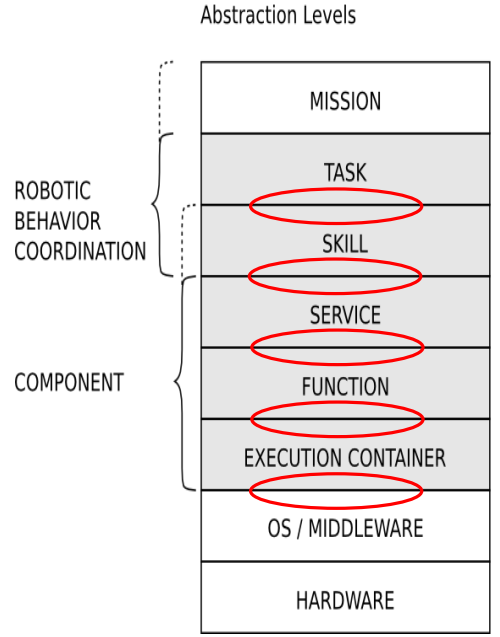




...

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



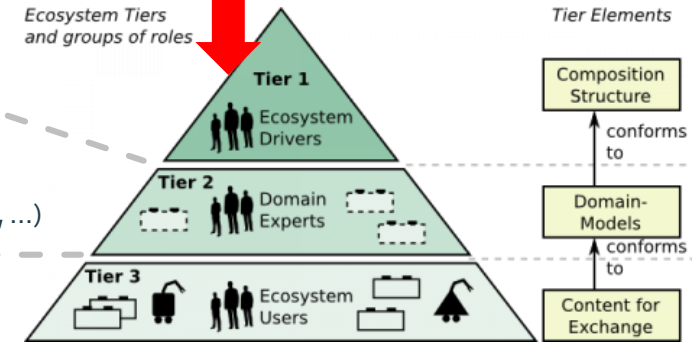


Architectural Pattern for Communication
Architectural Pattern for Component Coordination
Architectural Pattern for Software Components
Architectural Pattern for Managing Transition of System States
Architectural Pattern for Task-Plot Coordination (Robotic Behaviors)
Architectural Pattern for Service 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



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

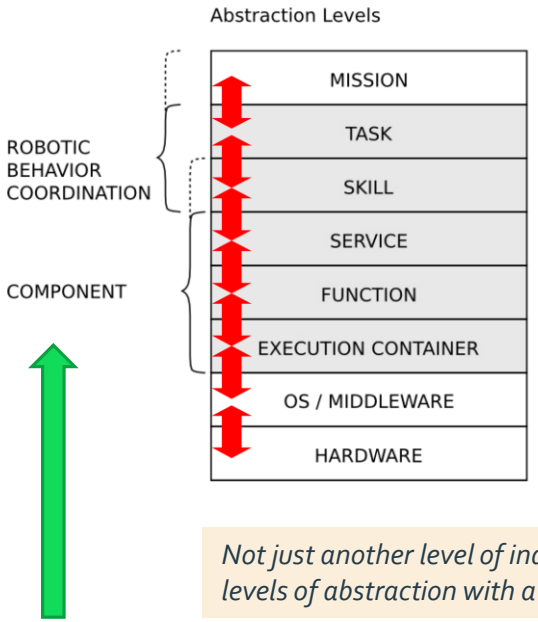
Example I4.0:

OPC UA

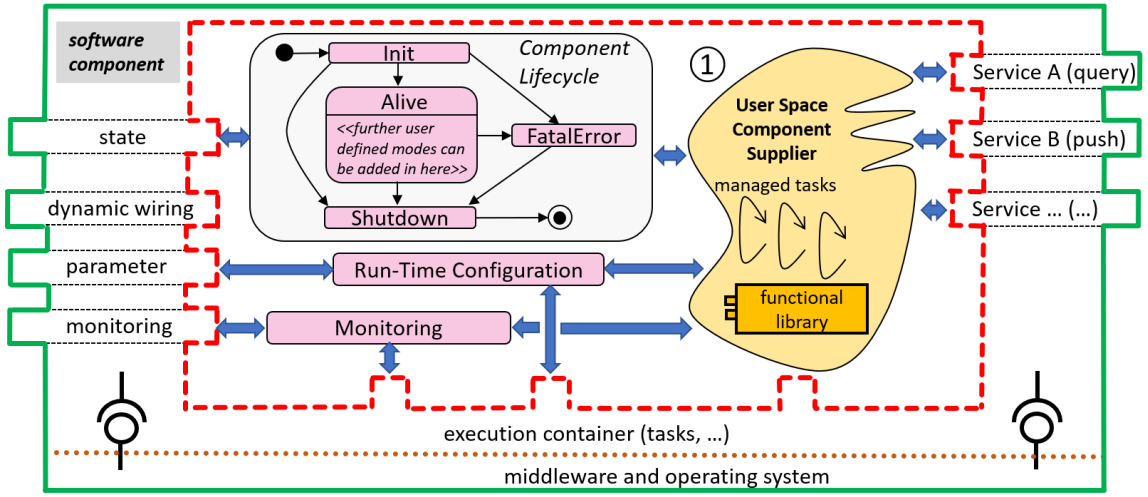
Companion Specification

Conformant Devices

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



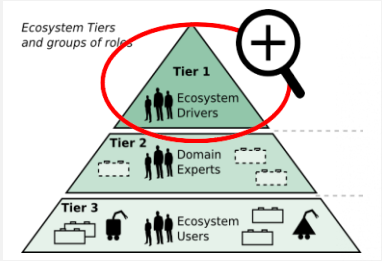
Not just another level of indirection, but levels of abstraction with a real benefit

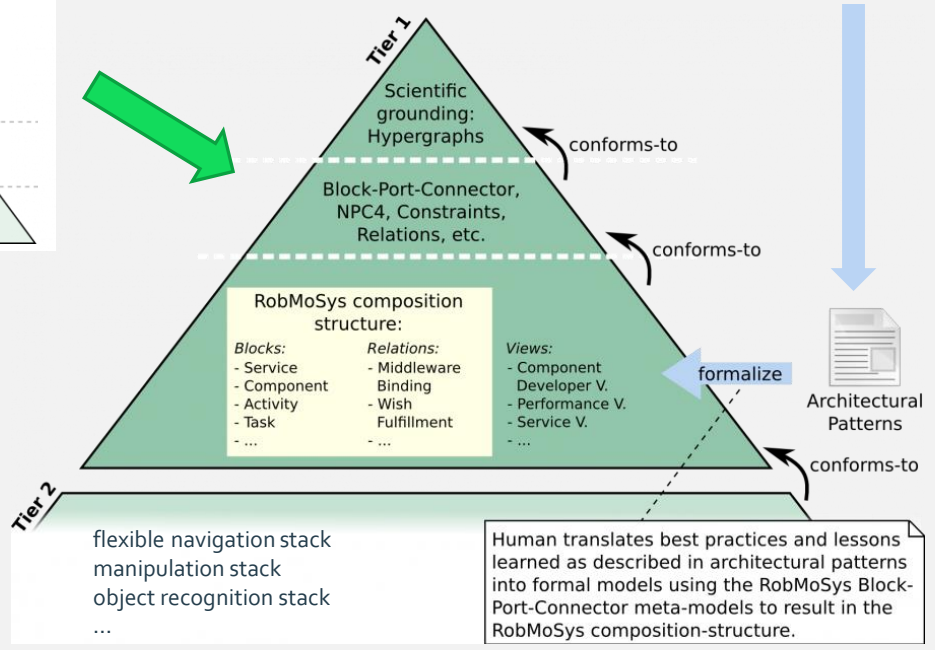


S/W component with communication (service-oriented ports), configuration (resources, parameters), coordination (modes, lifecycle), computation

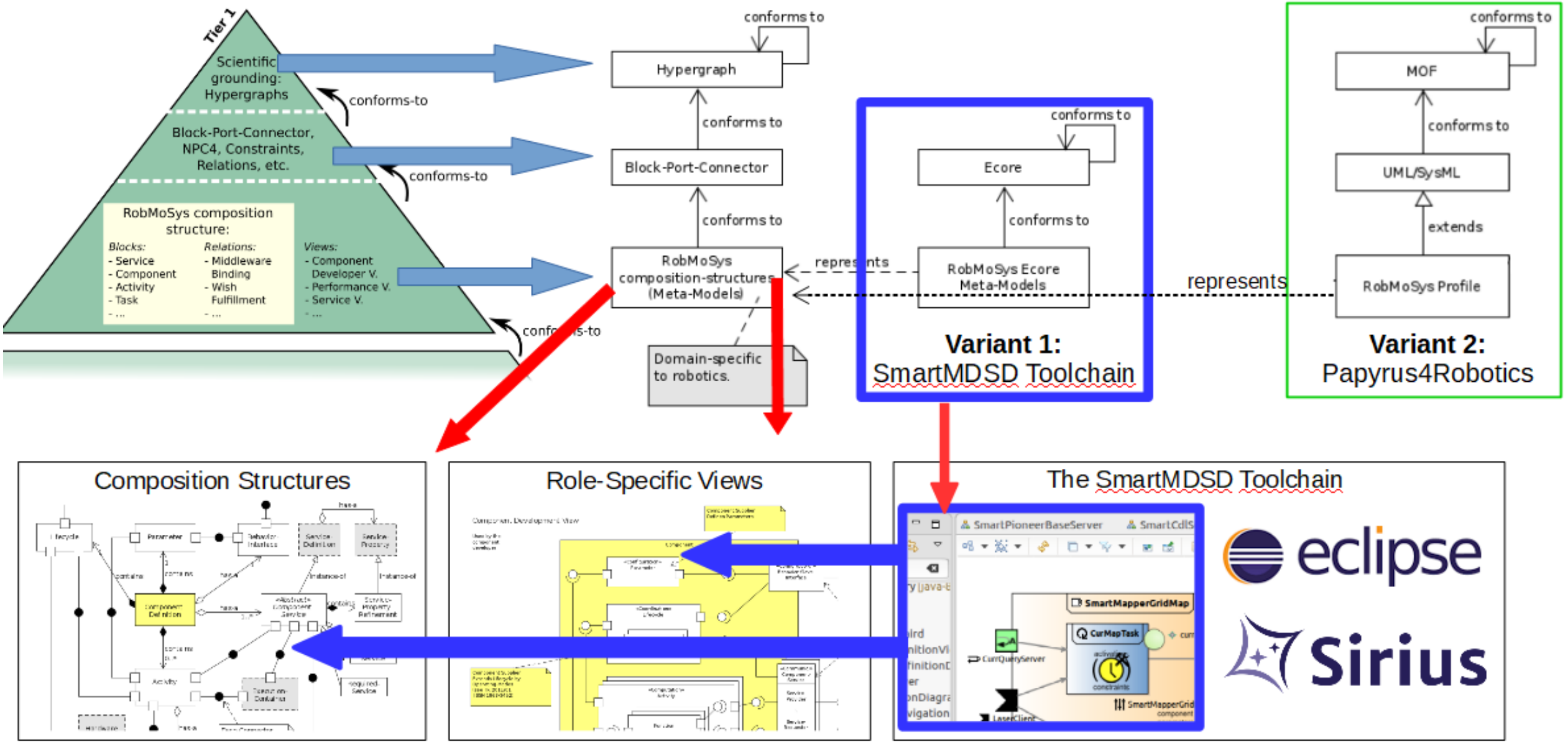
Tier 1 provides the **general structures for composition**.
Three levels can be distinguished:

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
...





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



The Concept of a Digital Data Sheet

- *Data sheets are models*
- *A data sheet describes an outside view of an asset, including its variation points*
- *A data sheet includes internals only as far as you need to know them for using the asset and for predicting its fit (behavior, structure) for your context*
- *Data sheets are, by purpose, not rich enough for synthesis of the artefact*

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

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

SMART
- Smart TV
- Full Web Browser

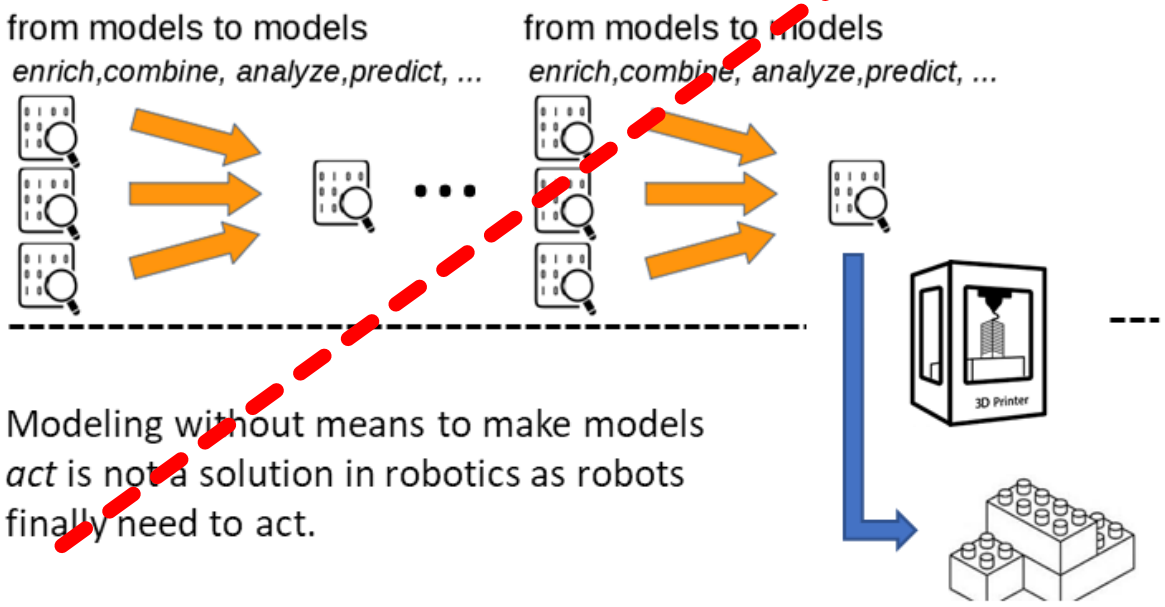
CONNECTIONS
- 2 HDMI® Connections
- 2 USB Connections
- 802.11n Wi-Fi Built In
- 1 Component in
- 1 Composite In (Shared with AV Component input)

Synthesis

Data Sheets are Models

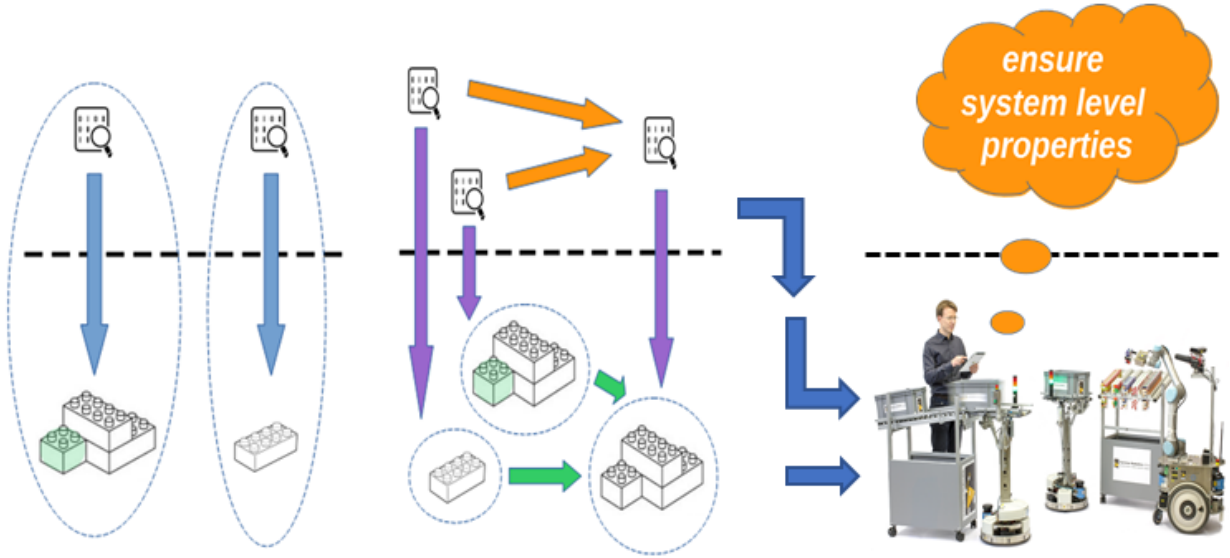
Describes outside view, including internals only as far as you need to know them for using the asset and for predicting its fit (behavior, structure) for your cóntext

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.



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

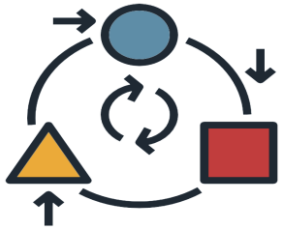
Data sheets (models of artefacs that act) *represent* components, sub-systems, task-plots etc. Suitability, traceability, simulation, etc. of system properties all via *composed data sheets*. When all is fine, then *compose* (put together and accordingly configure) the real artefacts to get the real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



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

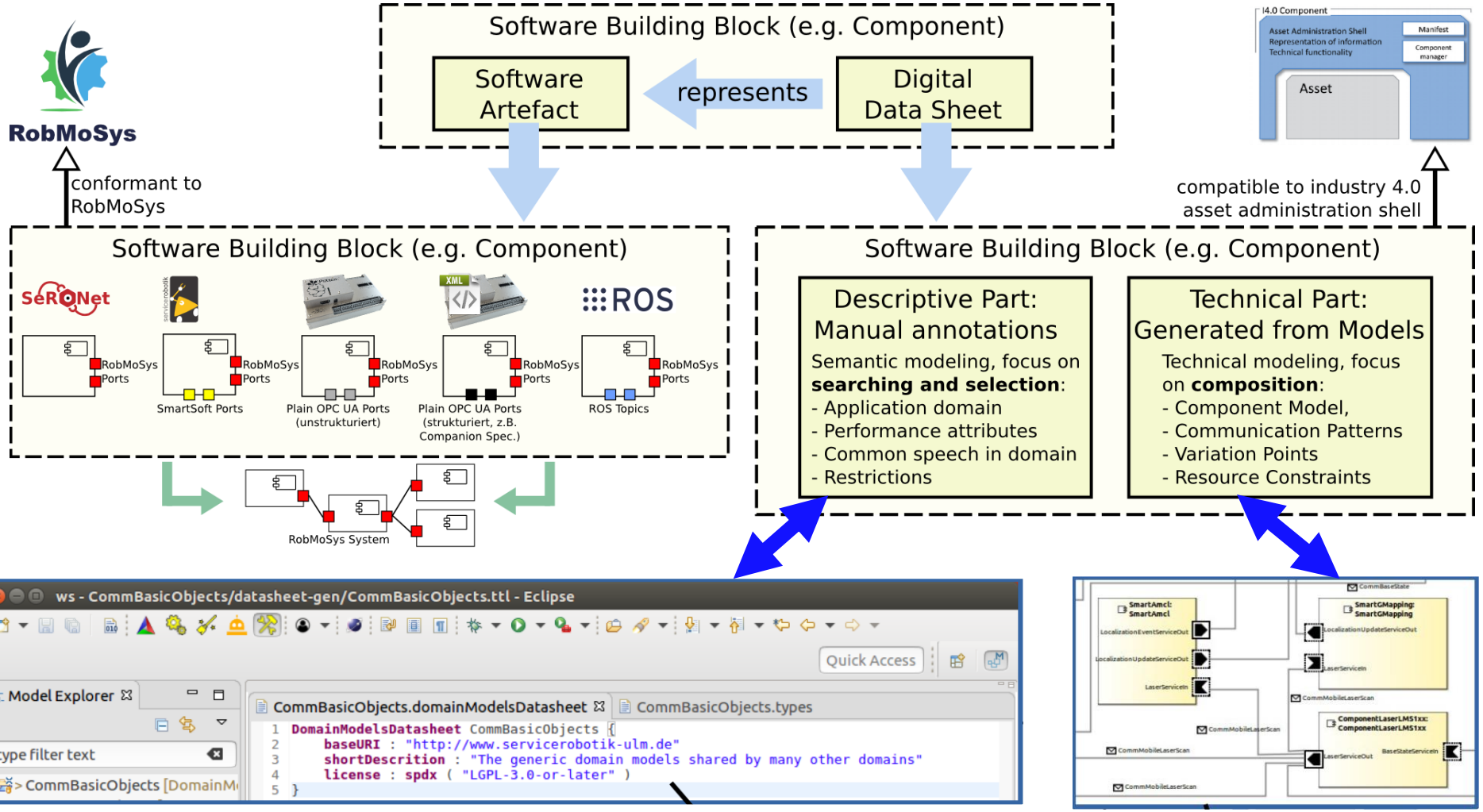
- Brokerage Platform
- Online marketplace

- Component selection
- Component composition
- Component configuration



- Runtime adaptation
- Context awareness
- Robustness and self-X

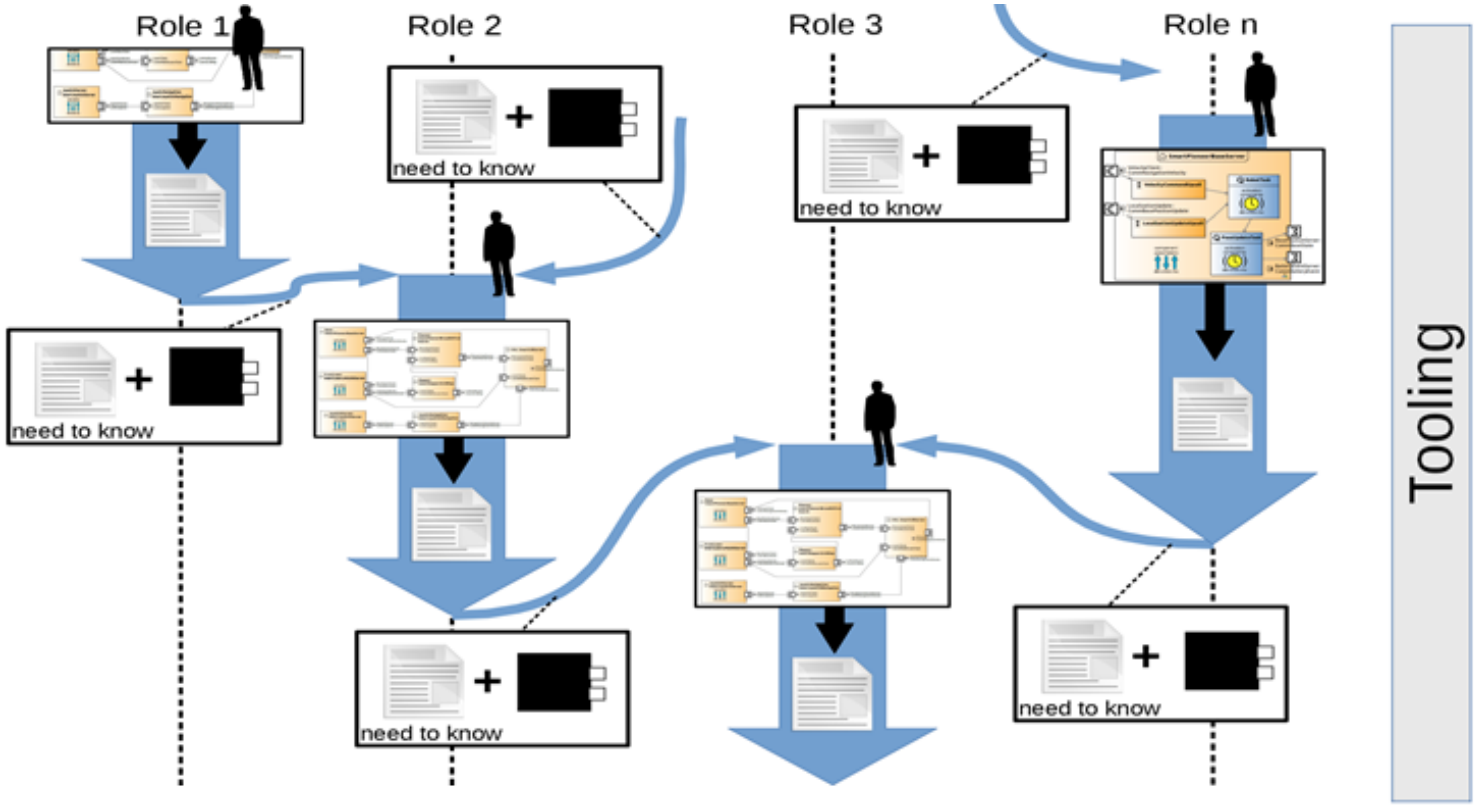
Digital Data Sheet as Submodel of Industry 4.0 Asset Administration Shell

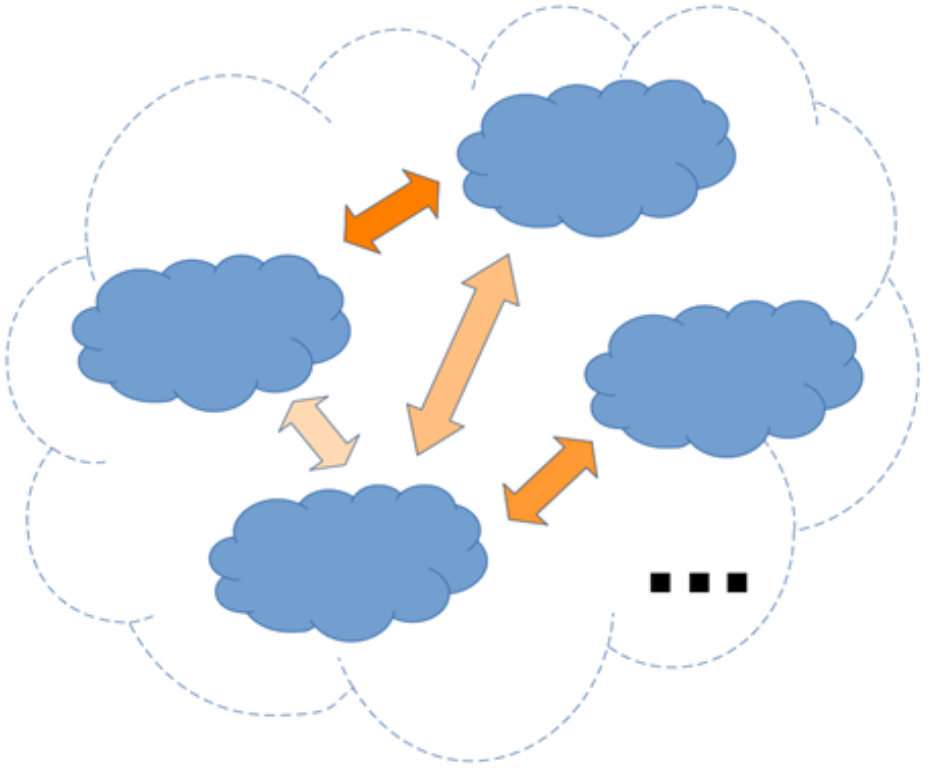


Data Sheets, Handover, Coverage and Conformance

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, …

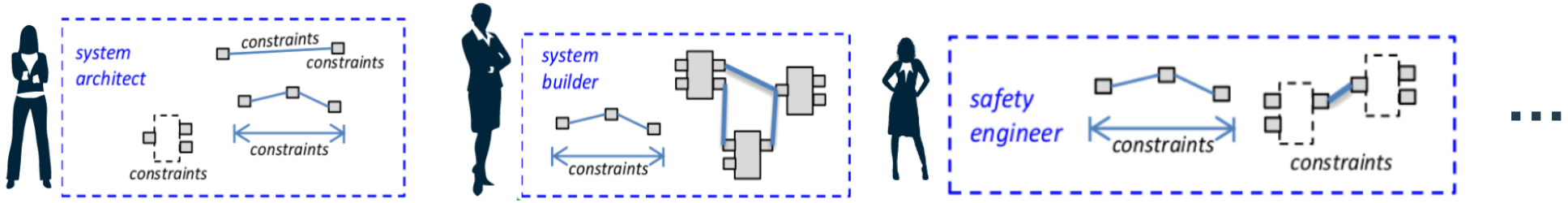




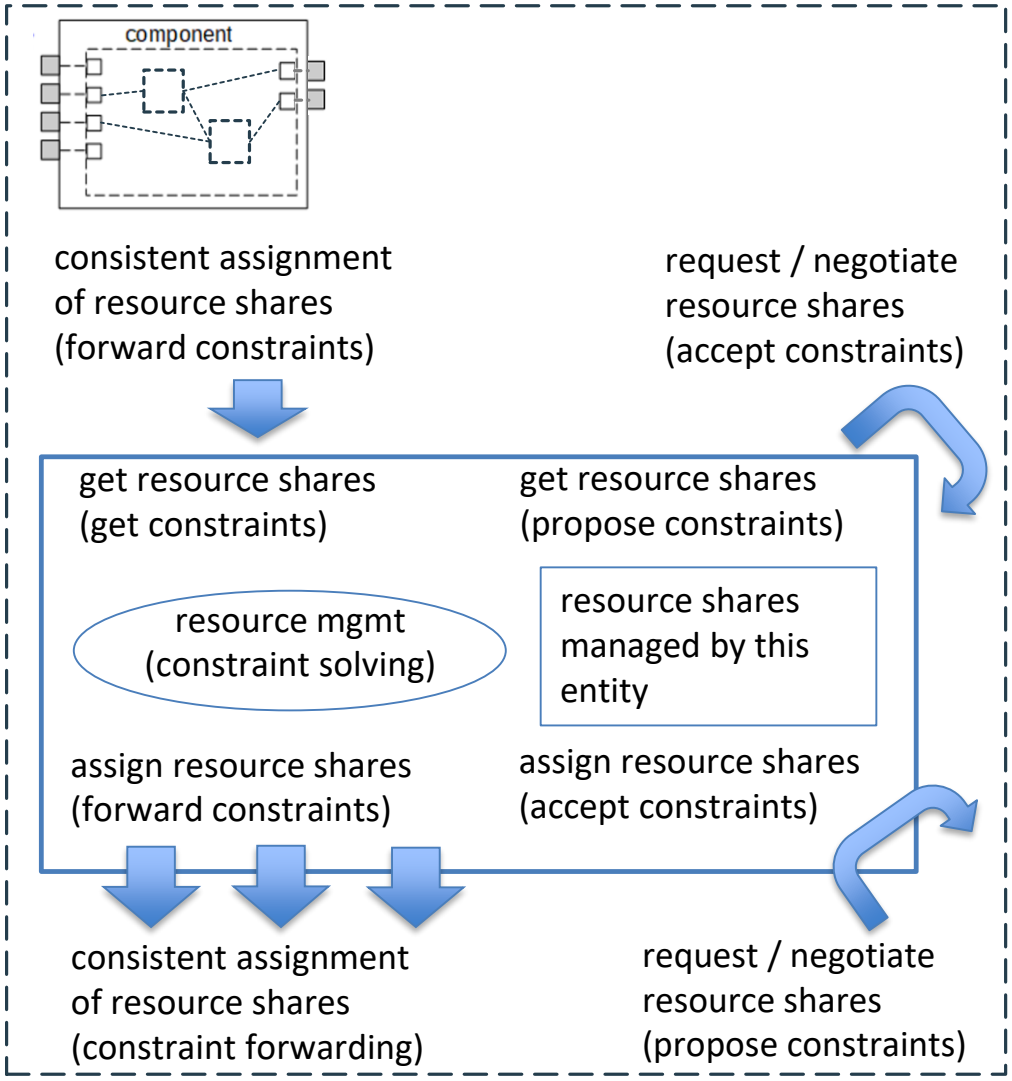
Wiki „Incubator“ => Wiki „Stable Body-of-Knowledge“

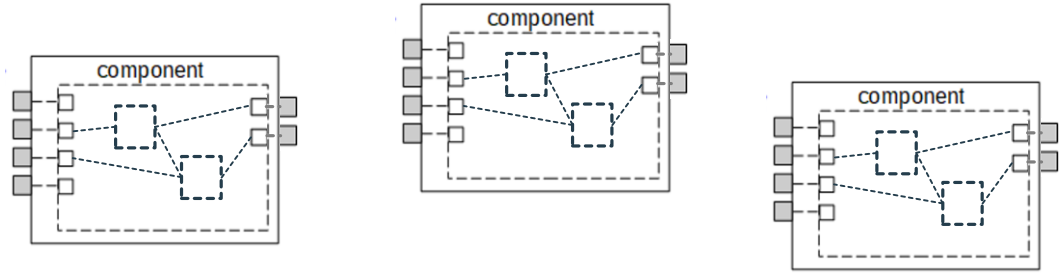
The Concept of Dependency Graphs

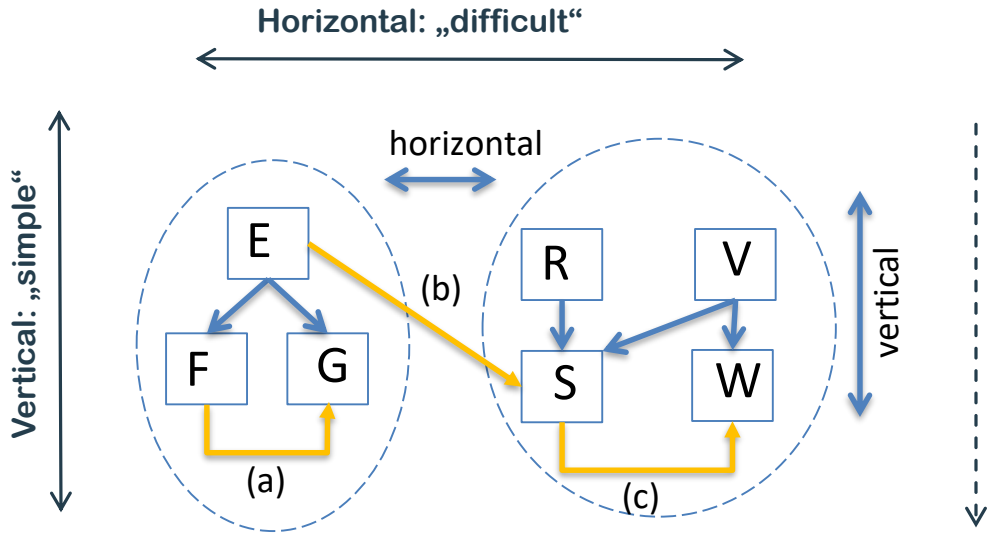
- *Horizontal / vertical composition and the challenge of managing resources*
- *Separation of control flow and data flow*
- *Composability:*
  - *Resource shares, reservation based mechanisms, constraints are composable*
  - *Priorities are not composable*



Horizontal / Vertical Composition: Separation of Control / Data Flow, Resource Management





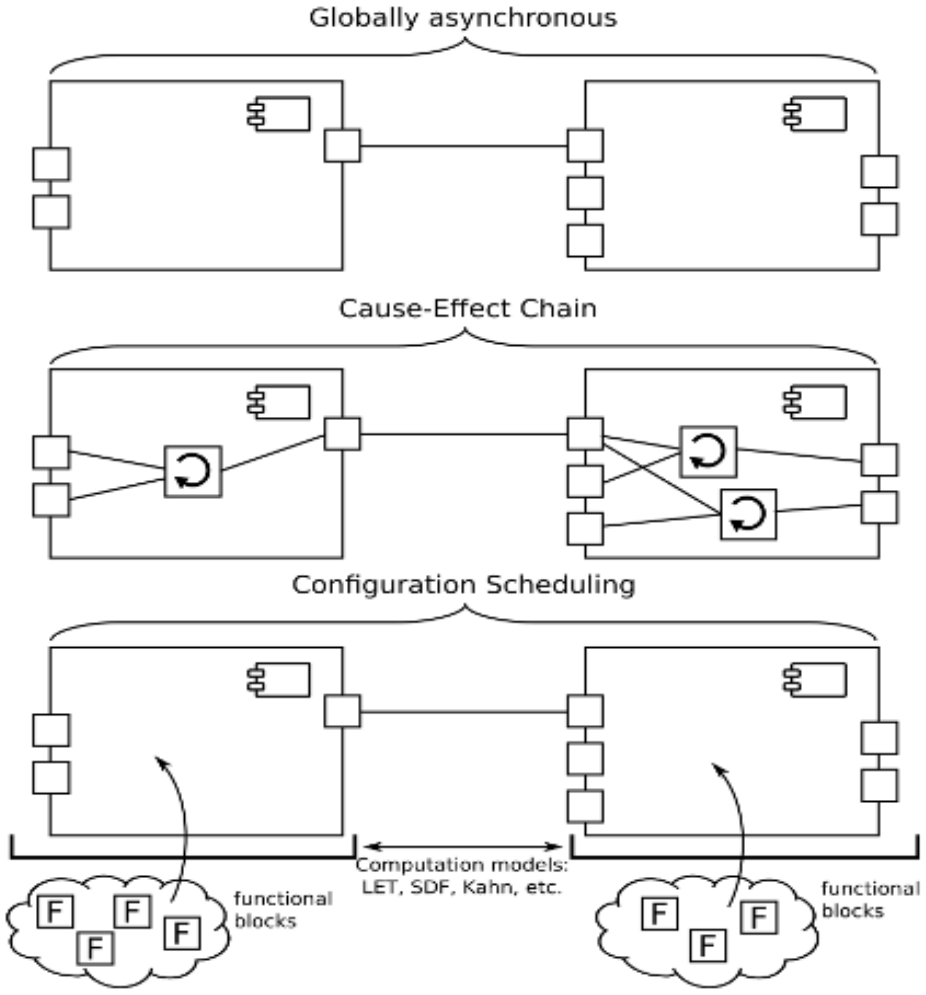


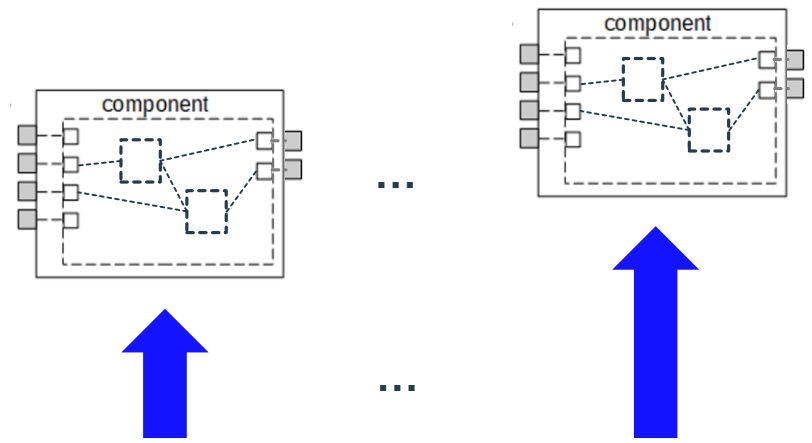
(a) Consistent because of either vertical coordination by E or horizontal coordination F asking G
(b) Consistent because of either vertical coordination above EFG and RSVW or just horizontal coordination E asking S
(c) Consistent because of either vertical coordination by RV for SW or S asking W

Horizontal / Vertical Composition: Dependency Graphs

Dependency graphs

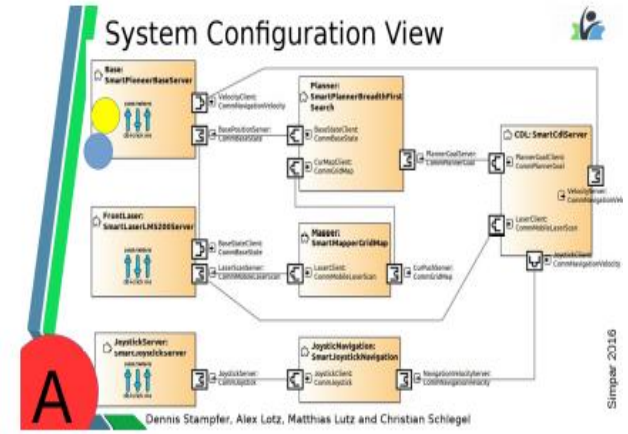
- to model needs for data consistency, data sync, data quality, data aging, cause-effect-chains, etc.
- to configure computation model

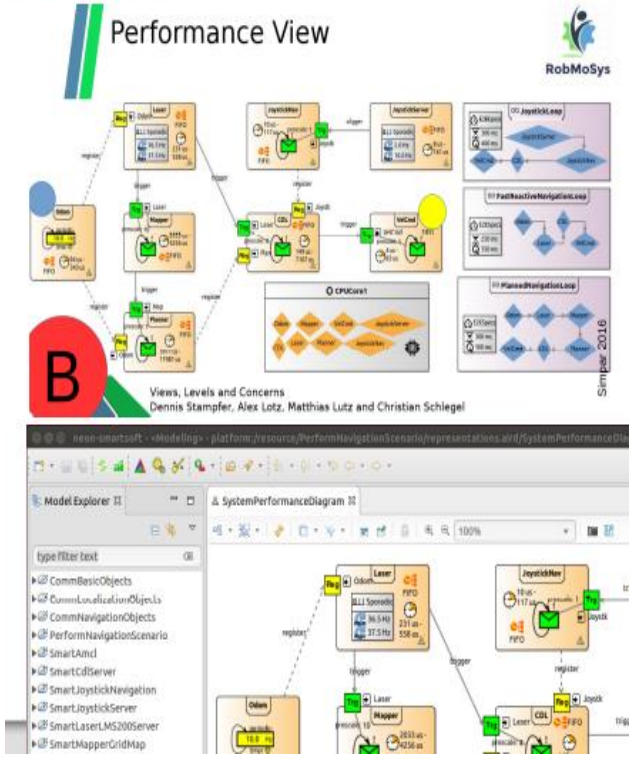




The software component model supports configuration of the following settings at system composition time (as represented in the data sheets) without recompilation:

- register / trigger semantic for communication
- port trigger / timed trigger for computation
- scheduling constraints
- sanity checks and run-time monitors
- …

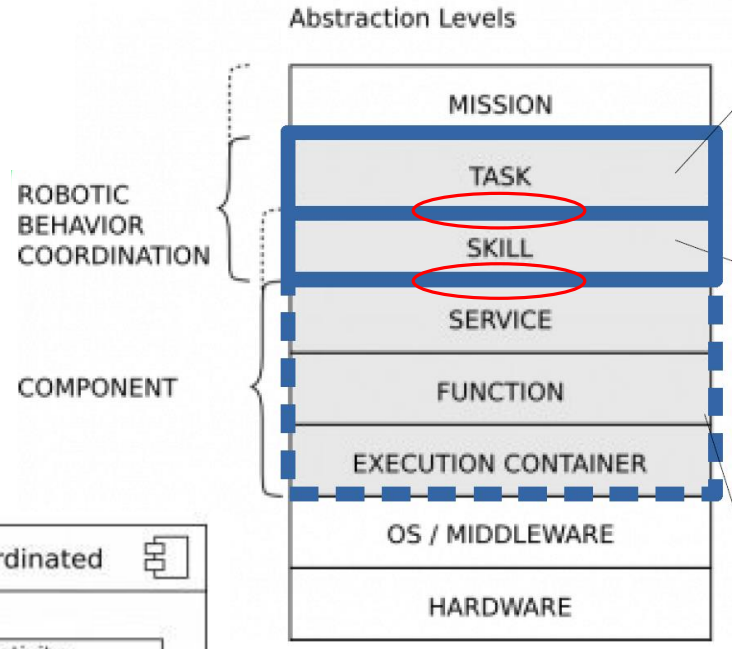




Behavior Coordination: Skill- / Task-Level Semantic Communication

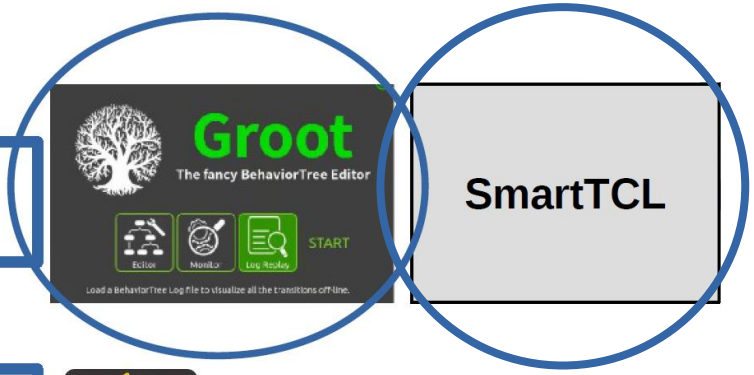
- not only „what to do“ (the task), but also „how to do it“ (quality of service, adequateness)
- Data sheets for skills: reuse of task models with robots coming with different capabilities

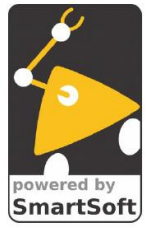
Behavior Coordination: Data Sheets for Skills

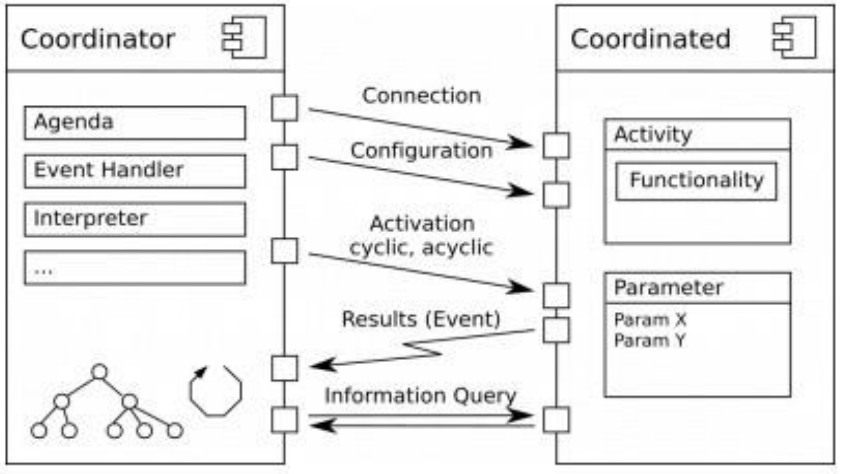


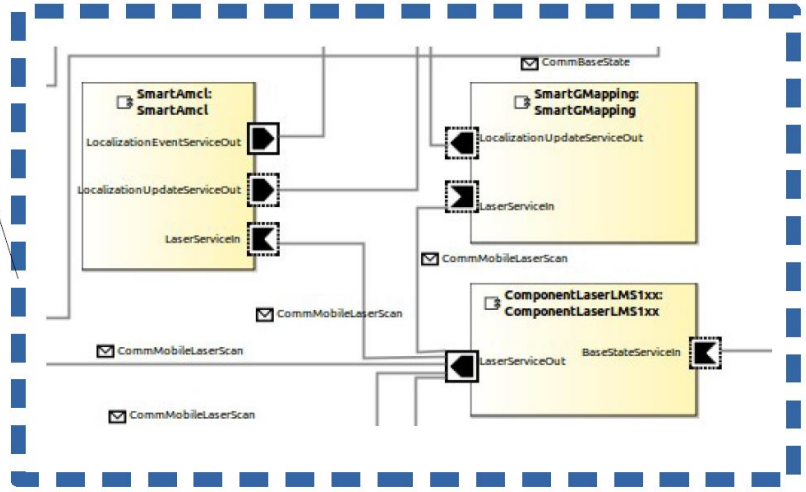
Tasks
Delivery

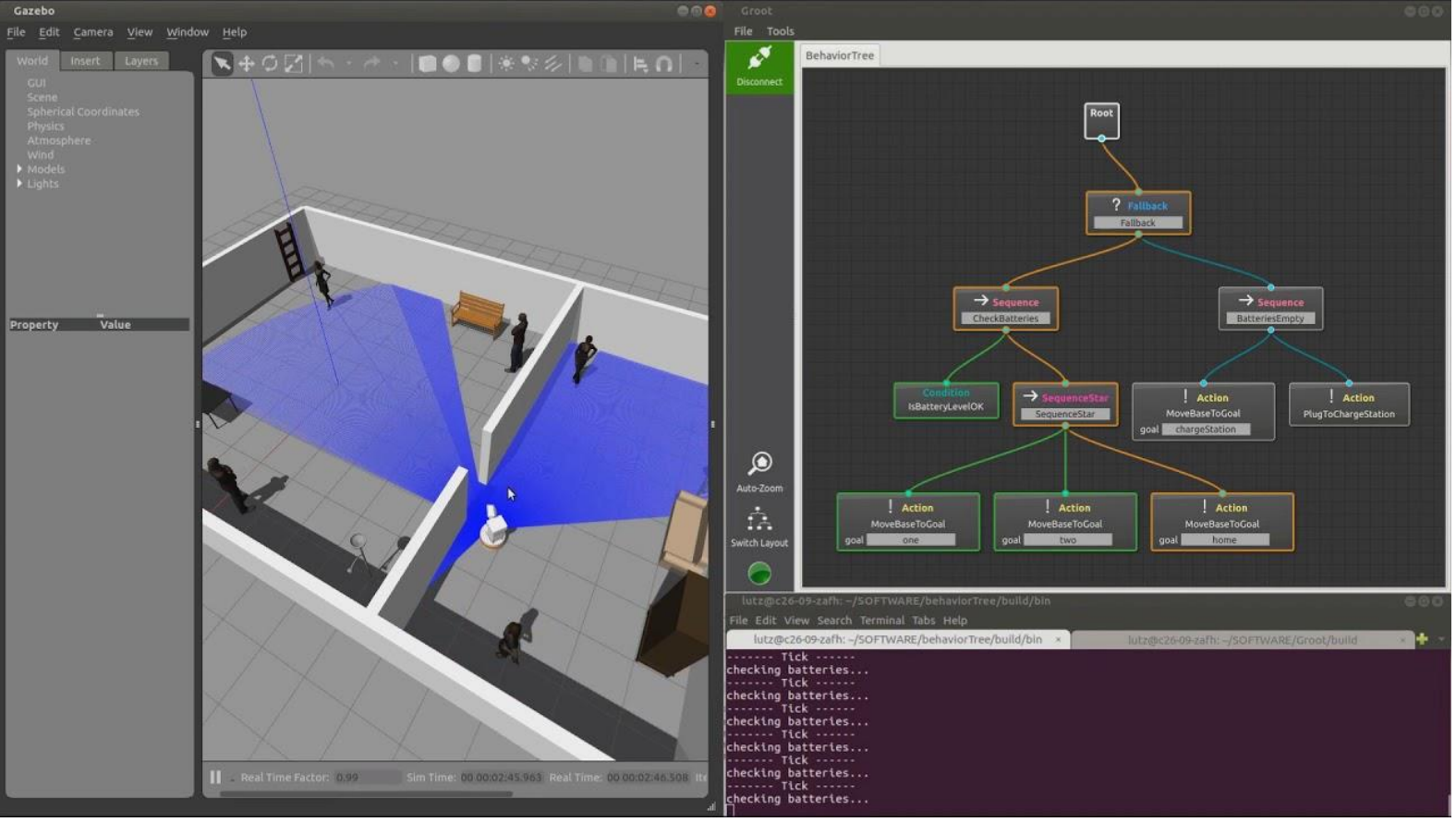
Skills:
pick up box
go to
drop box



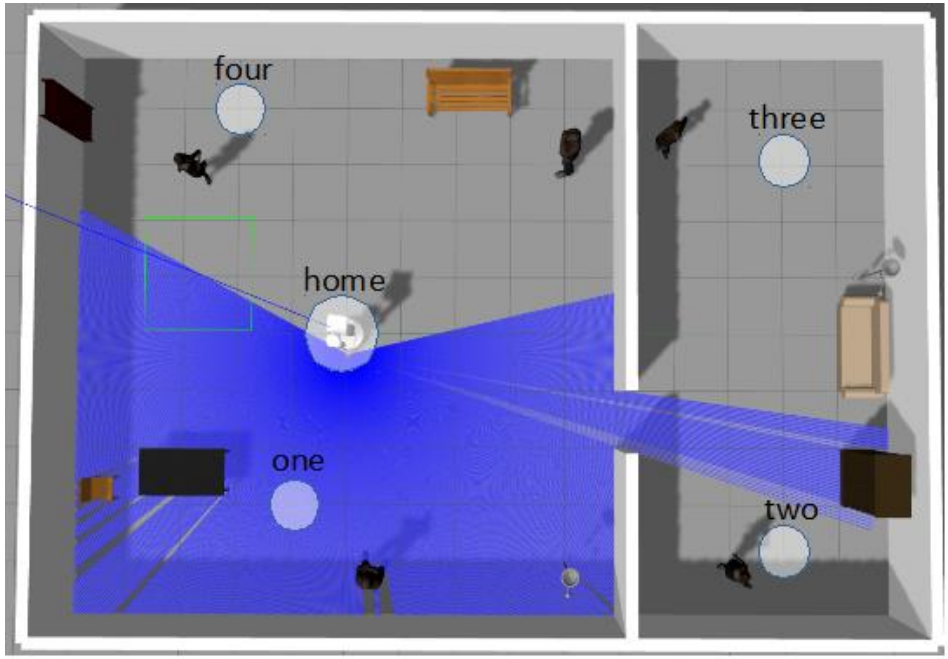


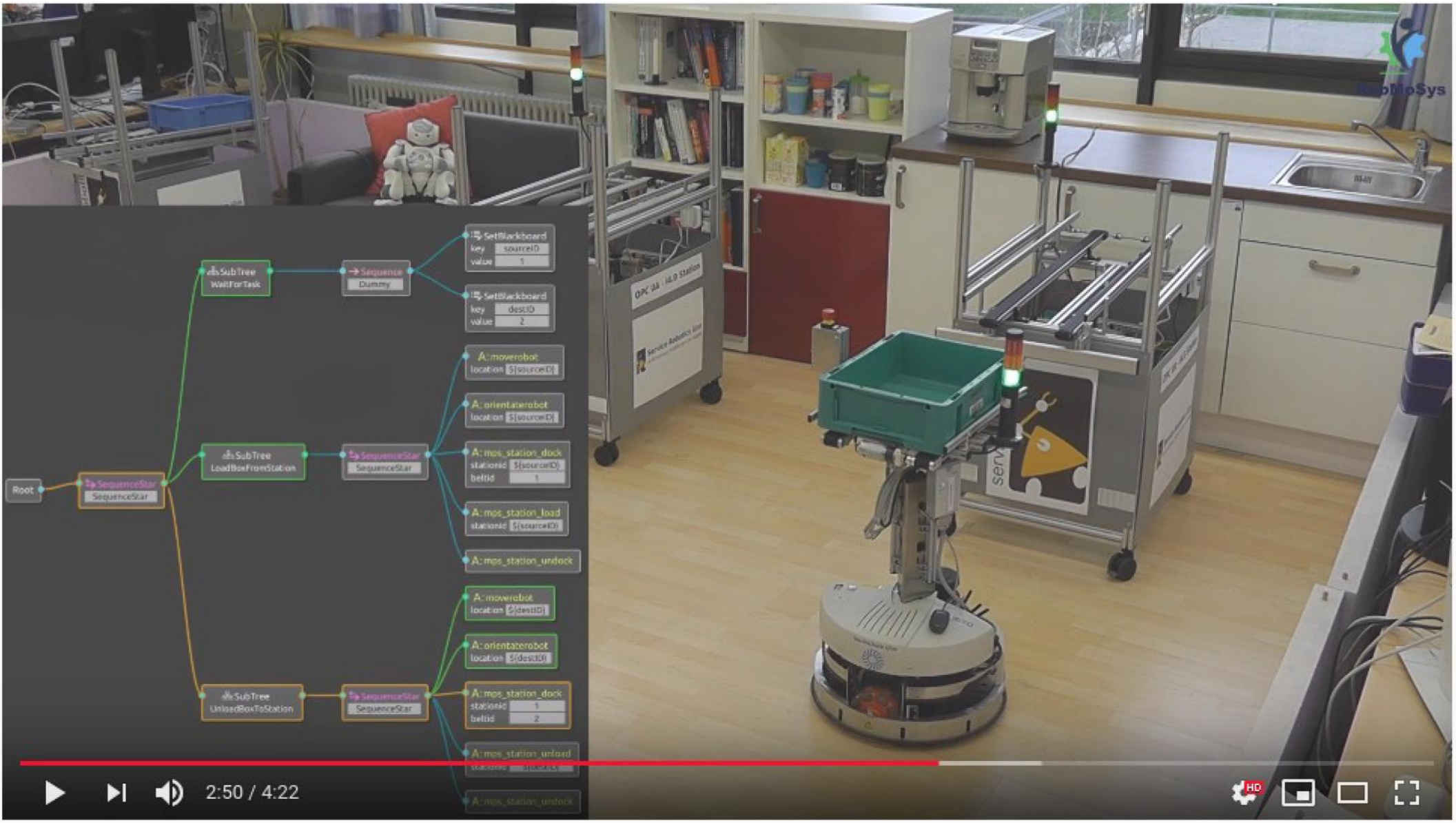




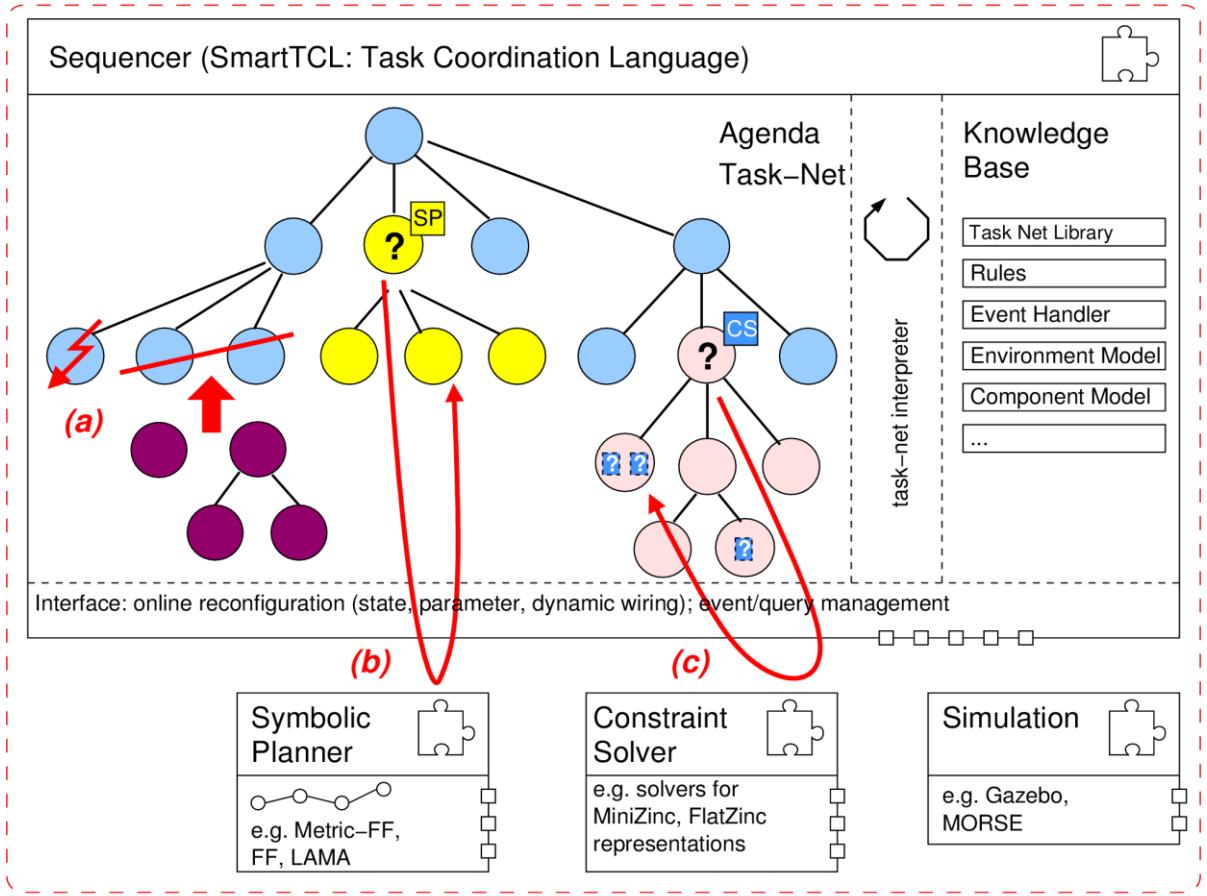


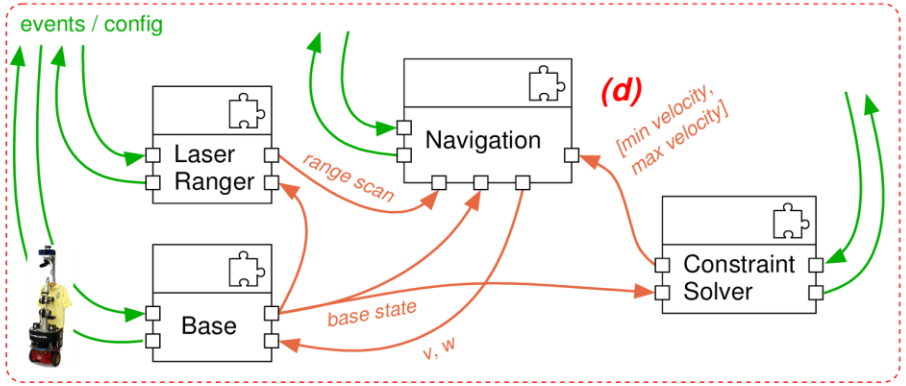
https://robmosys.eu/wiki/community:behavior-tree-demo:start





https://www.youtube.com/watch?v=54_skOuHsds





(a) SmartTCL handles a contingency by exchanging a sub-tree

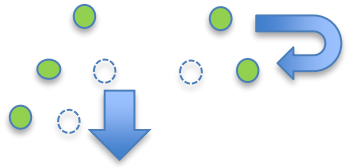
(b) SmartTCL uses a symbolic planner to refine a sub-tree

(c) SmartTCL triggers a constraint solver which executes the VML models

(d) VML binds left open variation point "max velocity" as a continuous service

Integration of "Variability in Task **Sequencing**" and "Variability in Task **Execution Quality**"

SmartTCL



- *component operating mode*
- *assignment of resource shares*
- *constraints forwarding*

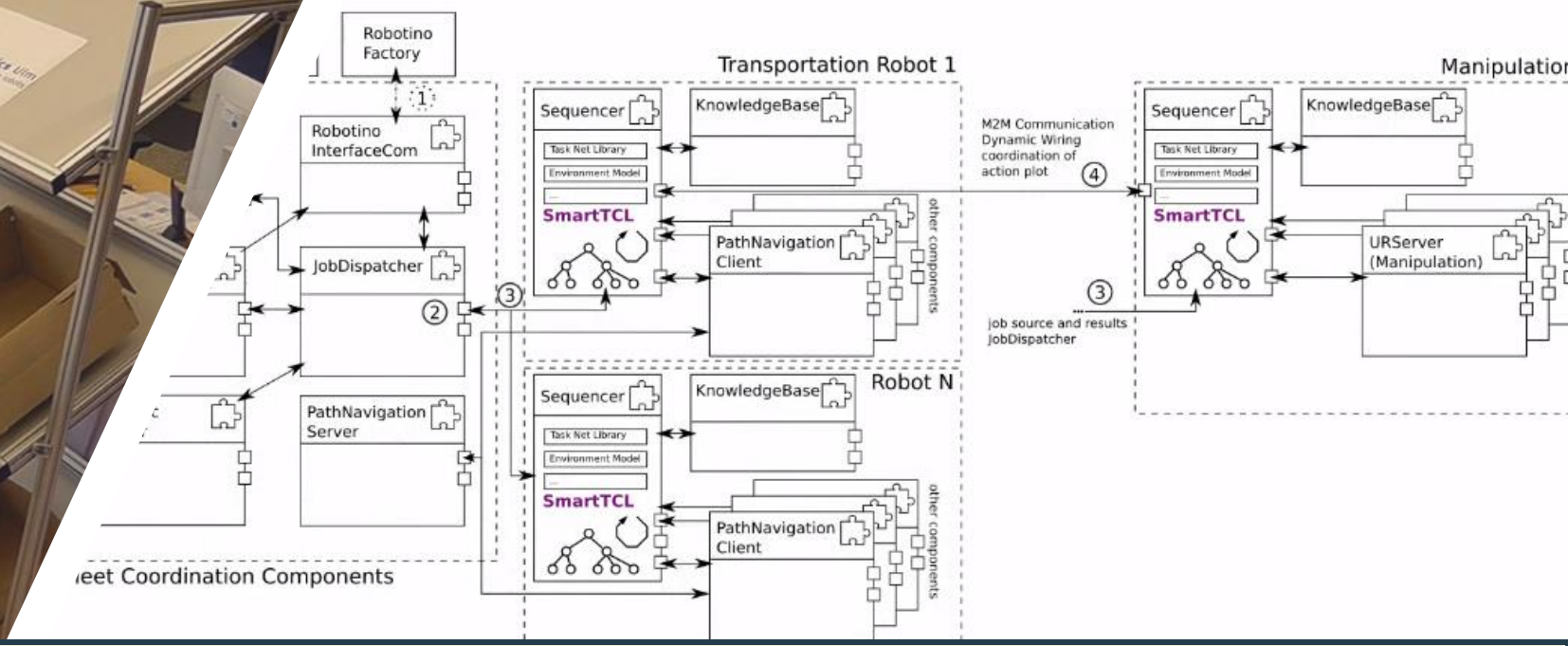
- *resource gripper fully occupied*
- *constraint „hold always upside"*

dynamic behavior tree
- parallel (one-of, all), sequential, expansion, …

horizontal
- e.g. resource share reservation in knowledge base

vertical
- e.g. expand task node under constraints and forward constraints

https://youtu.be/Zz66I4NVtNU



SeRoNet / RobMoSys conformant S/W building blocks

Behavior Coordination
Sequencer (SmartTCL)
SmartSimpleKB
SmartSeq2SeqCom

Navigation
SmartCdlServer
SmartNavigationPlanner
SmartPlannerBreadthFirstSea
SmartMapperGridMap
SmartPathNavigationClient
SmartPurePursuitNavigati

Hardware Abstraction
SmartLaserLMS1xxServer
SmartRobotinoBaseServer
SmartRobotinoImageServer
SmartRobotinoIRServer
SmartRobotinoLaserServer
SmartRobotinoConveyerBeltServer

Localization + SLAM
SmartAmcl
SmartGMapping

Relative Movement

Robot Fleet Co

ion Robot - Larry

vior Coordination
quencer (SmartTCL)
martSimpleKB
SmartSeq2SeqCom

Object Recognition
SmartBoxDetection
SmartRackDetection

Mobile Manipulation
SmartOpenRave

Localization + SLAM

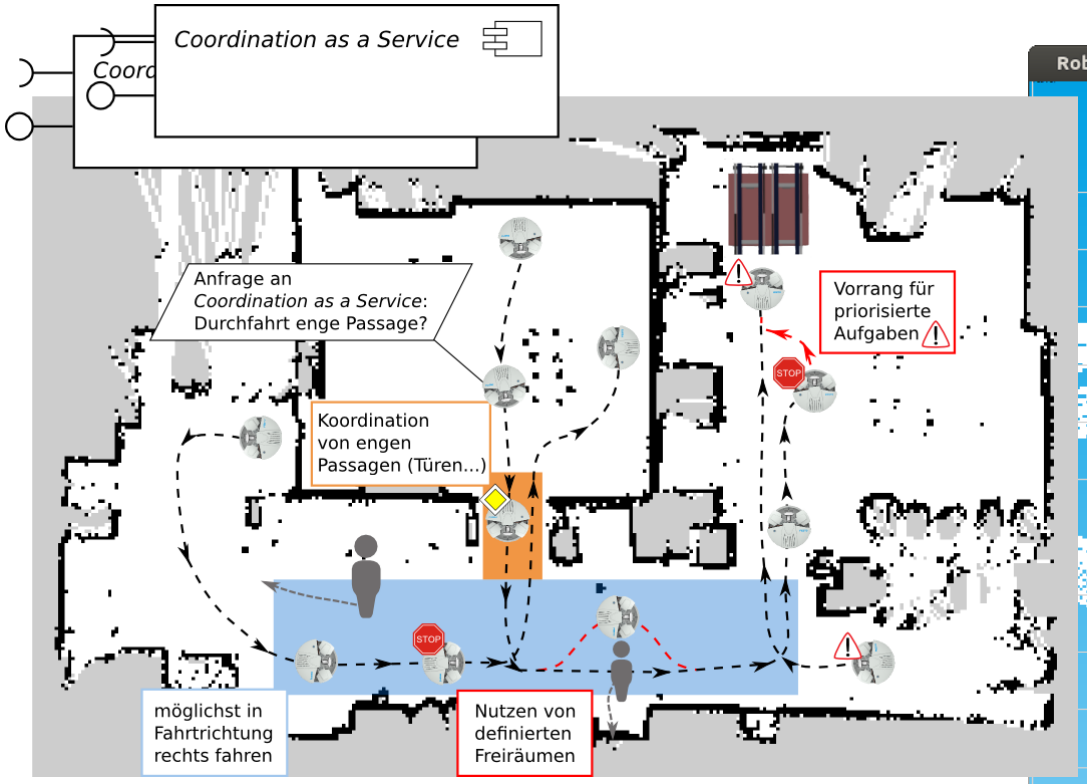
Navigation
SmartMapperGridMap
SmartNavigationPlanner
SmartCdlServer

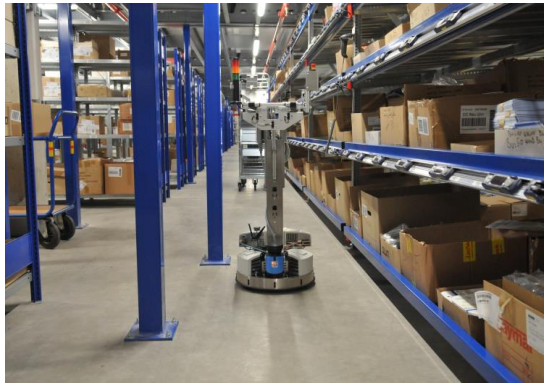
Hardware Abstraction
SmartPTUServer
SmartKinectV2Server
SmartRMPBaseServer
SmartLaserLMS1xxServer
SmartURServer
SmartVacuumGripper

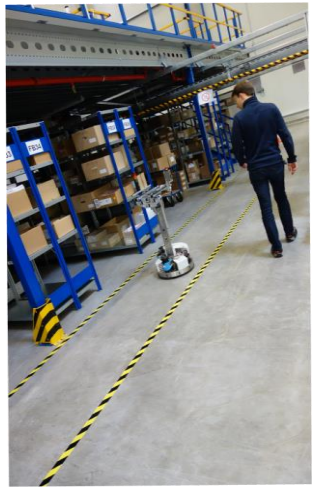
Robot Fleet Communication

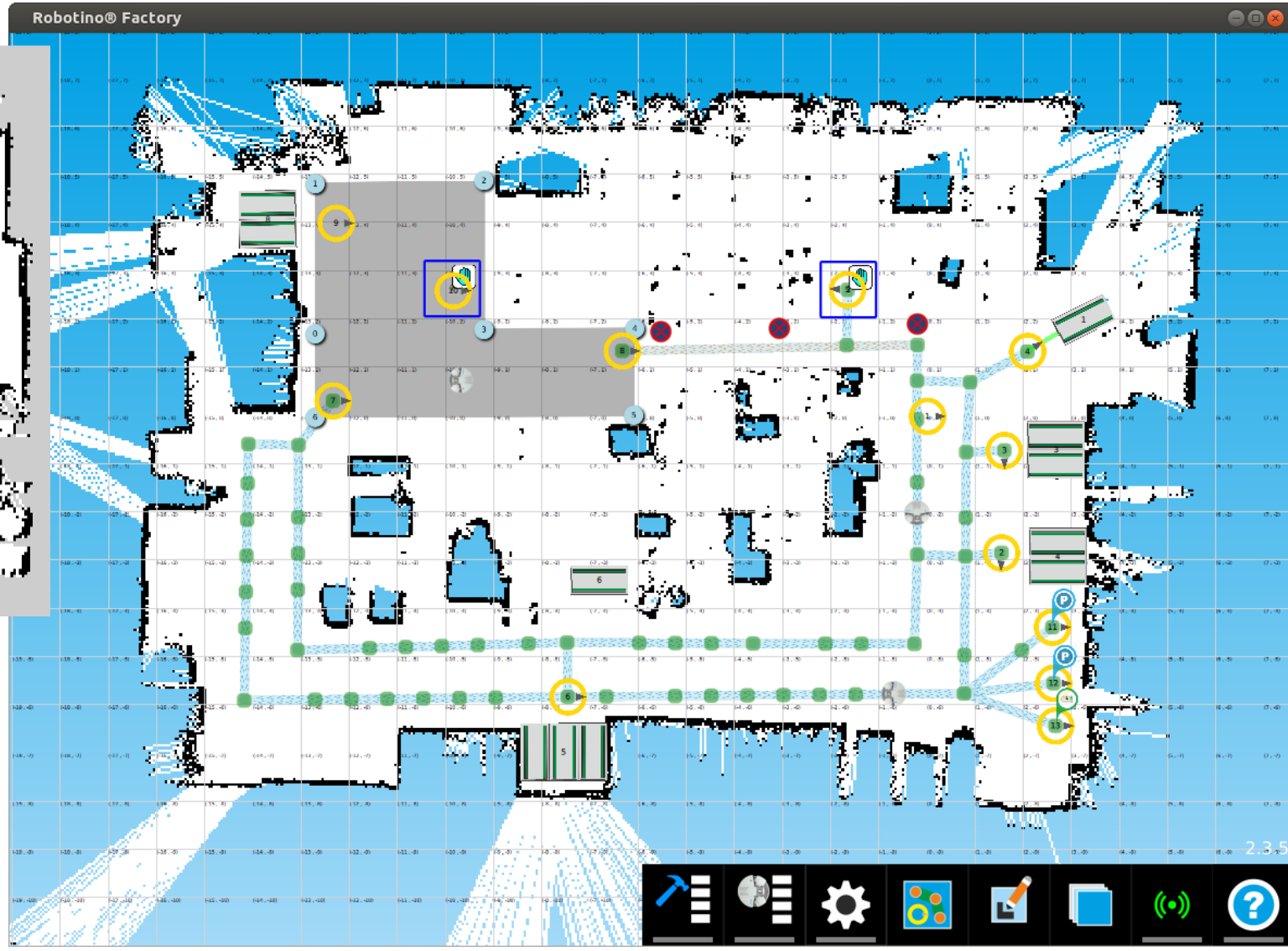
SmartTCL

https://youtu.be/RHvvb6lTHG4


Coordination as a Service
Anfrage an
Coordination as a Service:
Durchfahrt enge Passage?
Koordination
von engen
Passagen (Türen...)
Vorrang für
priorisierte
Aufgaben
möglichst in
Fahrtrichtung
rechts fahren
Nutzen von
definierten
Freiräumen





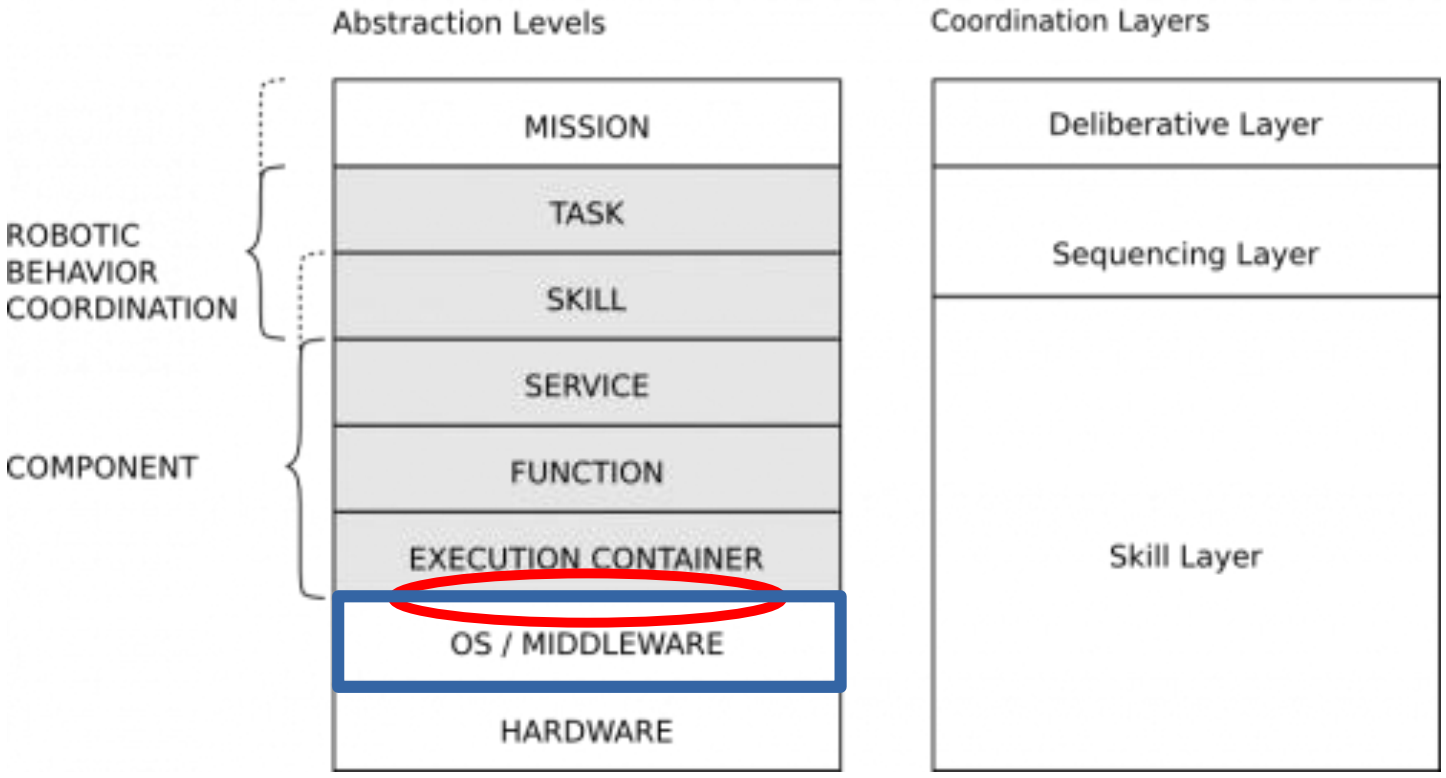

Robotino® Factory

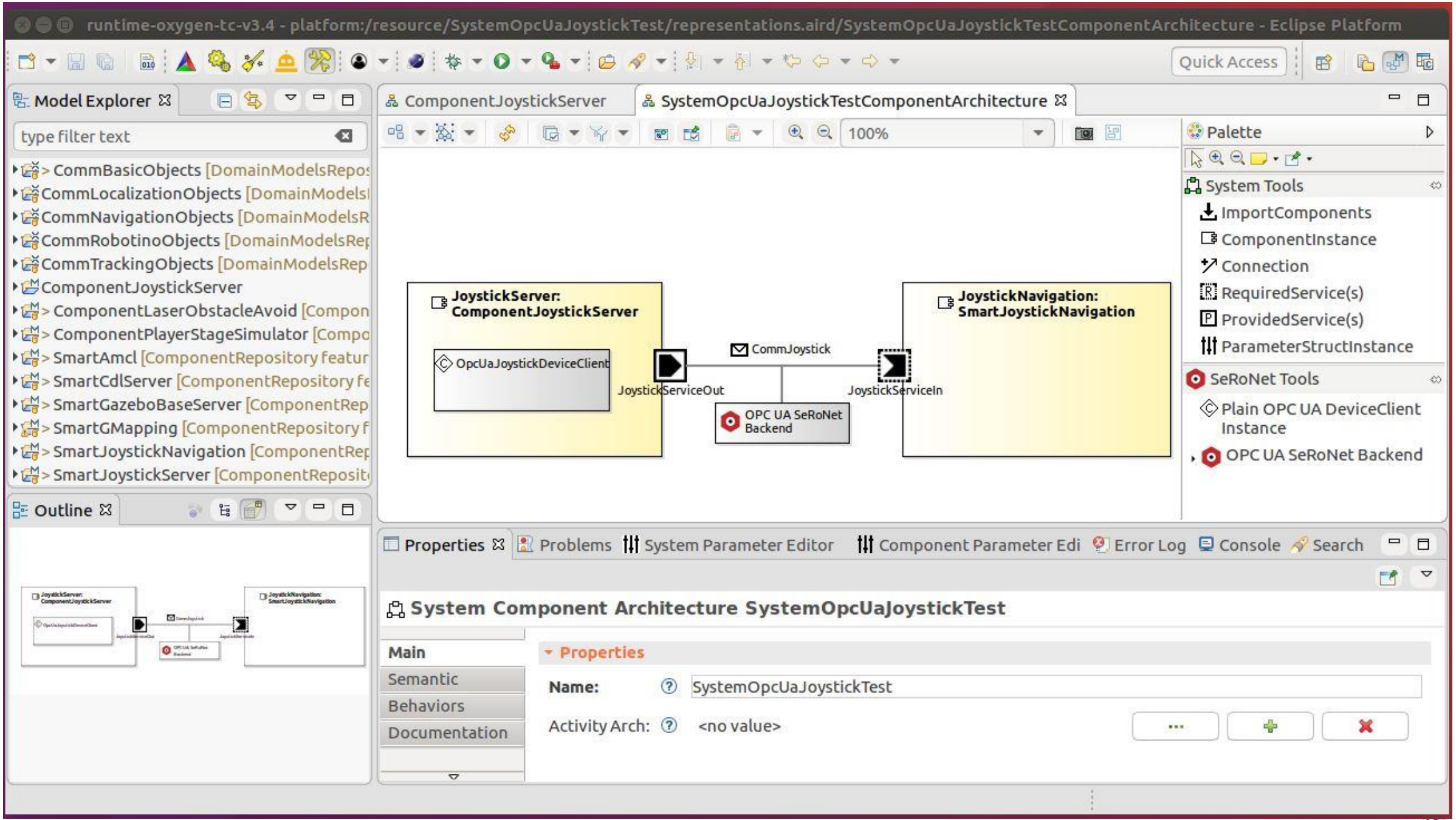
The concept of early binding of semantics but late binding of technology

Middleware Late Binding / Middleware Agnostic Modeling

Middleware Late Binding / Middleware Agnostic Modeling

- *Mixture of different middlewares within a single system*
- *Middleware can be decided per connection*
- *Late binding of middleware at deployment without recompilation of components*




runtime-oxygen-tc-v3.4 - platform:/resource/SystemOpcUaJoystickTest/representations.aird/SystemOpcUaJoystickTestComponentArchitecture - Eclipse Platform
Quick Access
Model Explorer
type filter text
CommBasicObjects [DomainModelsRepos
CommLocalizationObjects [DomainModels
CommNavigationObjects [DomainModelsR
CommRobotinoObjects [DomainModelsRep
CommTrackingObjects [DomainModelsRep
ComponentJoystickServer
ComponentLaserObstacleAvoid [Compon
ComponentPlayerStageSimulator [Compo
SmartAmcl [ComponentRepository featur
SmartCdlServer [ComponentRepository fe
SmartGazeboBaseServer [ComponentRep
SmartGMapping [ComponentRepository f
SmartJoystickNavigation [ComponentRep
SmartJoystickServer [ComponentReposit
Outline
ComponentJoystickServer
SystemOpcUaJoystickTestComponentArchitecture
100%
JoystickServer: ComponentJoystickServer
OpcUaJoystickDeviceClient
JoystickServiceOut
CommJoystick
OPC UA SeRoNet Backend
JoystickServiceIn
JoystickNavigation: SmartJoystickNavigation
Palette
System Tools
ImportComponents
ComponentInstance
Connection
RequiredService(s)
ProvidedService(s)
ParameterStructInstance
SeRoNet Tools
Plain OPC UA DeviceClient Instance
OPC UA SeRoNet Backend
Properties
Problems
System Parameter Editor
Component Parameter Edi
Error Log
Console
Search
System Component Architecture SystemOpcUaJoystickTest
Main
Semantic
Behaviors
Documentation
Properties
Name:
SystemOpcUaJoystickTest
Activity Arch:
<no value>

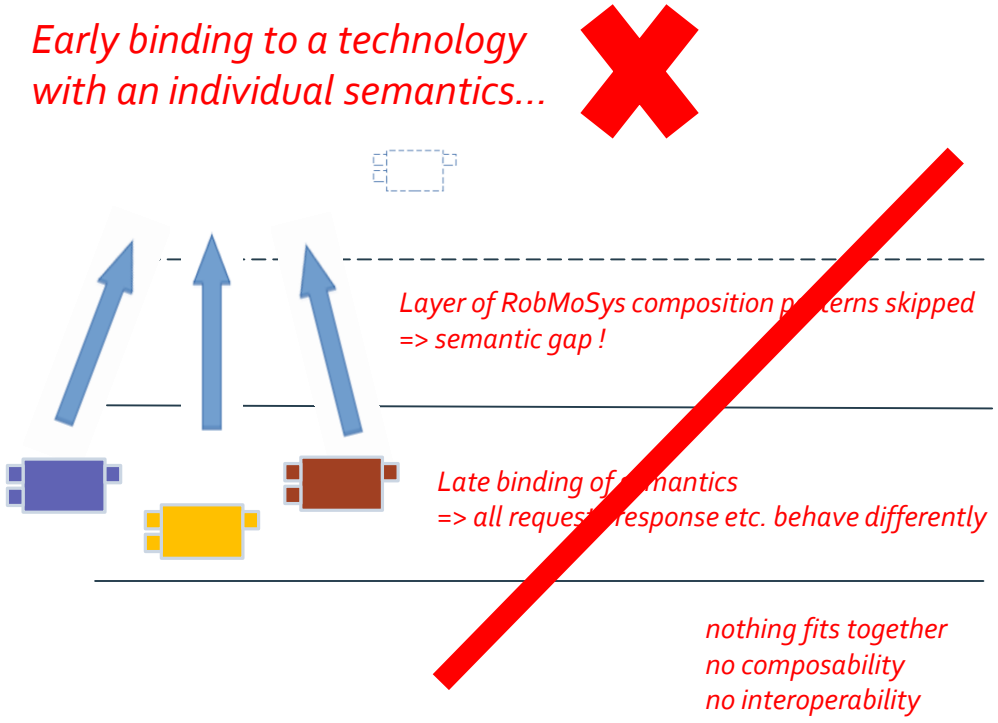
Middleware Late Binding / Middleware Agnostic Modeling

Early binding of semantics and late binding of technology

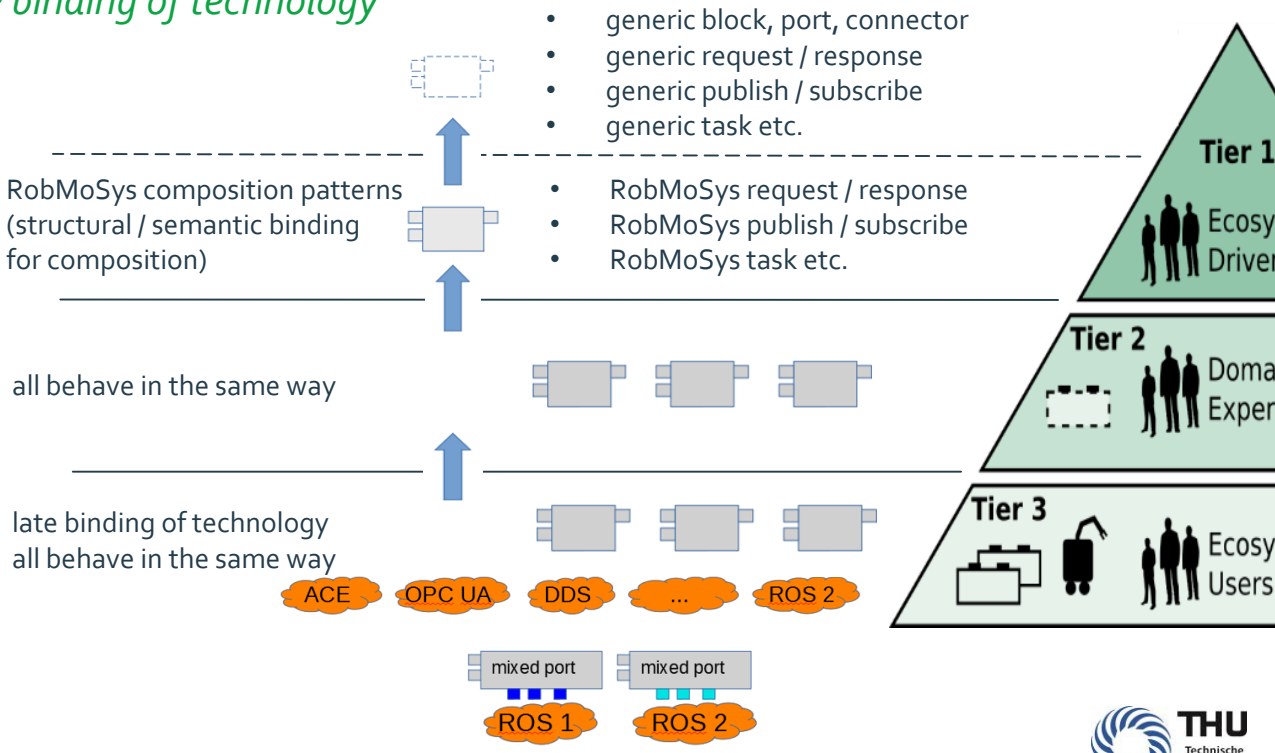
skipping RobMoSys composition patterns and making mapping shortcuts
⇒ ignoring semantics produces semantic gap
⇒ This destroys composition

instead, align with the RobMoSys structures
⇒ early (not late) binding of semantics, late (not early) binding of technology

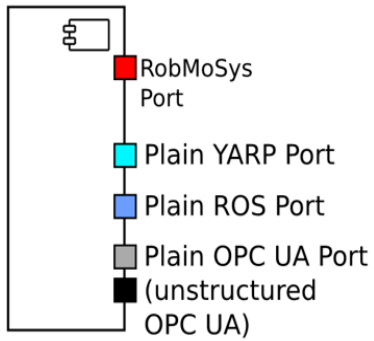
Early binding to a technology with an individual semantics...



Early binding of semantics with late binding of technology



Mixed Port Component as Migration Path



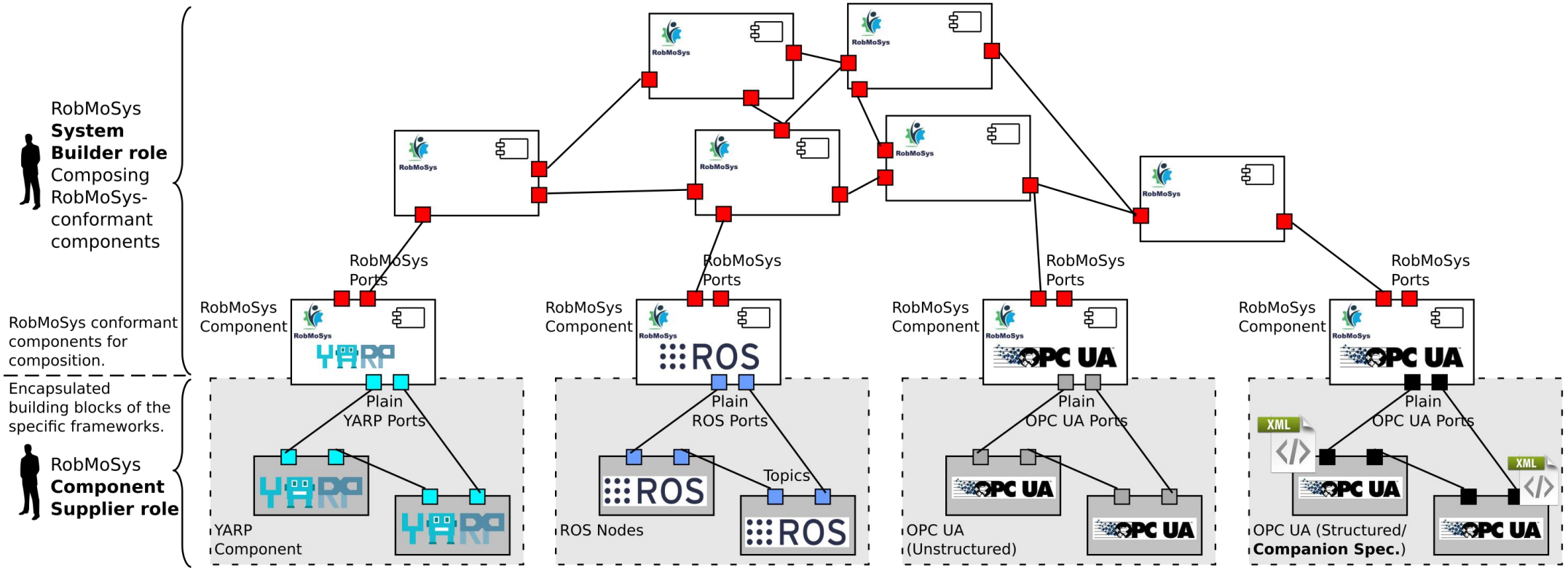
Supported Middlewares
- ACE (SmartSoft)
- DDS (SmartSoft)
- OPC UA (SeRoNet)

by RobMoSys ITP Carve

by PAiCE SeRoNet

for structured (Companion Specs)
and unstructured OPC UA by PAiCE SeRoNet

https://wiki.servicerobotik-ulm.de/tutorials:start



I want to use it …

I want to contribute …

I want to …

- 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 ulation/Gazebo.	Navigation	HSU	SmartMDSD Toolchain v3	Ready	
	ot skeleton that covers navigation aspect of ntralogistics Industry	Navigation	HSU	SmartMDSD Toolchain v3	Ready	-

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

RobMoSys Wiki
http://www.robmosys.eu

Search

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.

Table of Co
- RobMoSys
- Tier 2 Do
- Tier 3 Co
- Tier 3 Sy
- Explanat

Tier 2 Domain Models

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

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

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

Service Robotics Ulm
autonomous mobile service robots

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

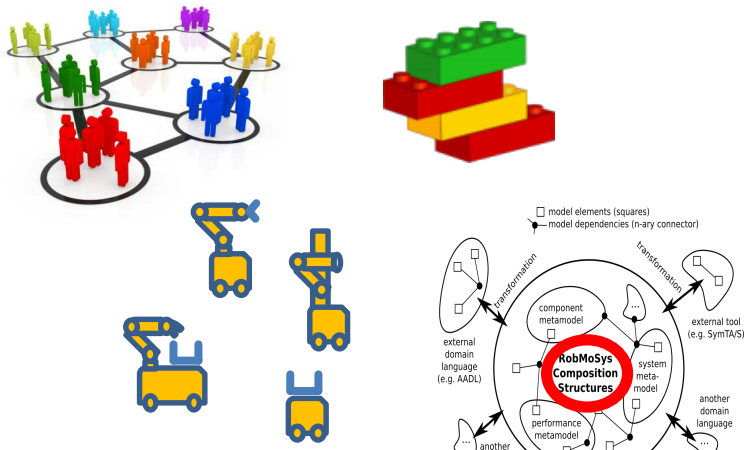
https://robmosys.eu/wiki/jumppage

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

Towards an EU Digital Industrial Platform for Robotics

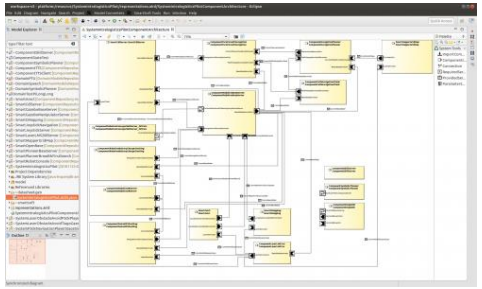
Models always have a purpose: overall purpose is consistency

- 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, responsibiilities, 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



Current Steps towards adequateness everywhere via data sheets:

- extensions of horizontal / vertical rresource and quality management
  - resource share reservations and contracting
  - from the task net level down to the OS / middleware, from design-time to run-time
- success stories for management of system level properties
  - safety, resource ⇔ quality, free of interference, ...