



09:00 – 09:20 Talk: Towards an open and industry-grade European robotics software ecosystem

Presenter: Christian Schlegel

09:20 – 10:00 Interactive Tool Demo: Open source Eclipse-based tooling for system composition: Piecing together software components to pilot applications *Presenter: Dennis Stampfer*

10:00 – 10:45 Interactive Tool Demo: Open Source Eclipse-Based Tooling for Component Builders: Middleware Agnostic Robotics Software Components *Presenter: Alex Lotz*

10:45 – 11:15 Coffee Break

11:15 – 11:55 Interactive Tool Demo: Safety-analysis by model-driven tooling *Selma Kchir*

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. *Presenter: Dennis Stampfer, Alex Lotz*

12:35 - 12:55 Talk: Overview, motivation, benefits of model-driven approaches in robotics: What-if analysis, compliance, predictability, etc. *Presenter: Herman Bruyninckx*

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

*Tutorial “Model-Driven Robot Software Engineering:
From ROS-specific coding to framework-agnostic modeling”*

Interactive Tool Demo: Piecing together software components to robotics pilot applications



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





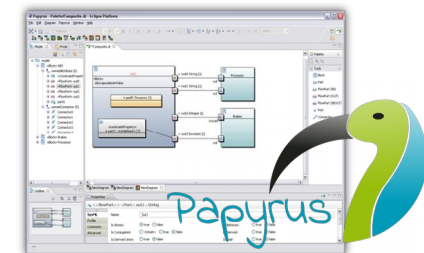
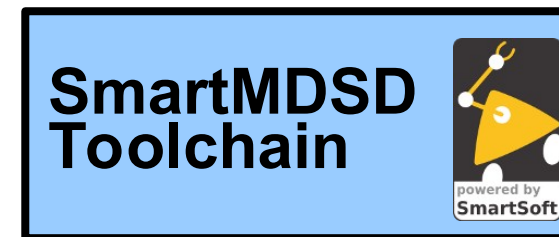
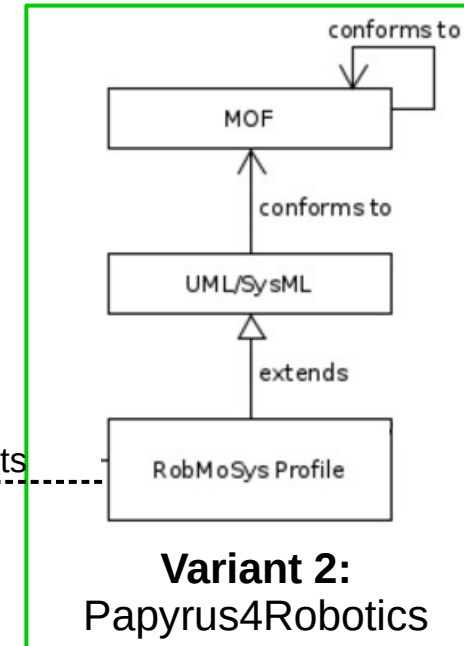
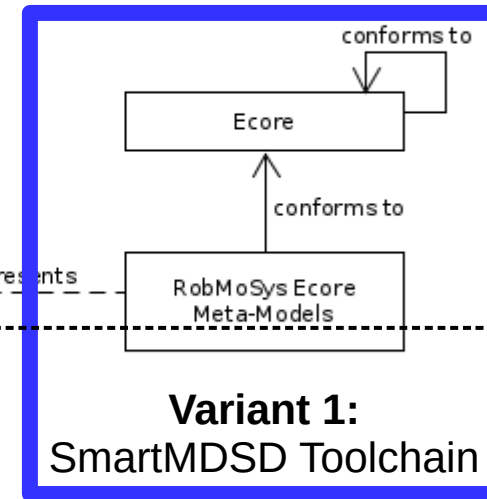
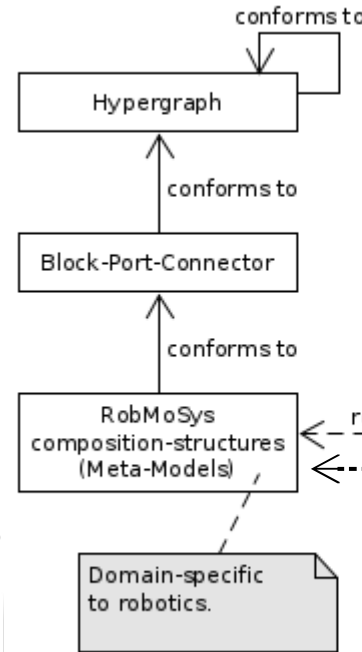
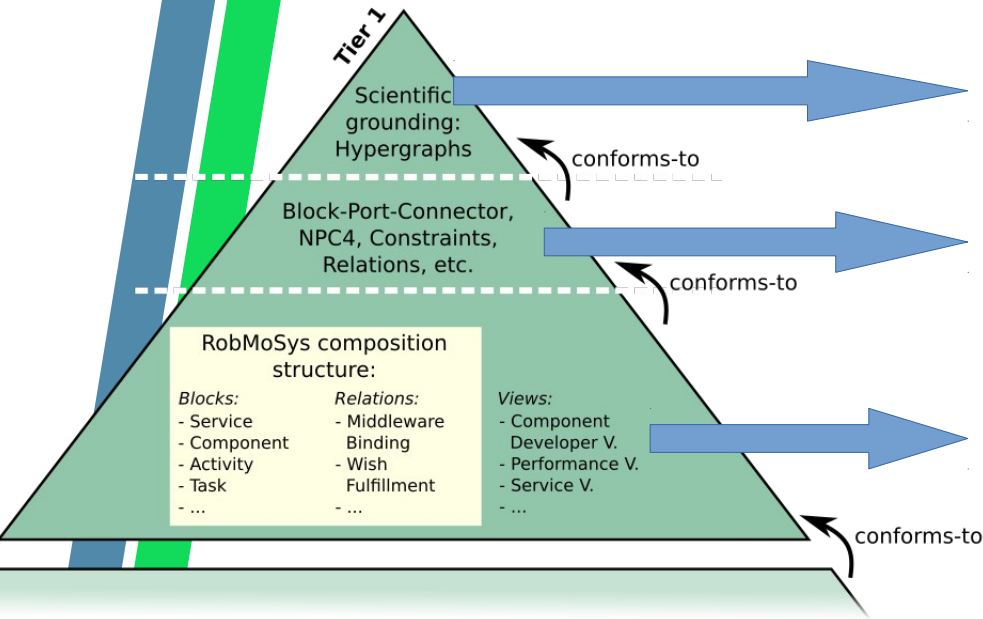
Outline

- Introduction
- Modeling systems
- Modeling and implementing components
- The Digital Data Sheet

RobMoSys Tooling: Realization alternatives

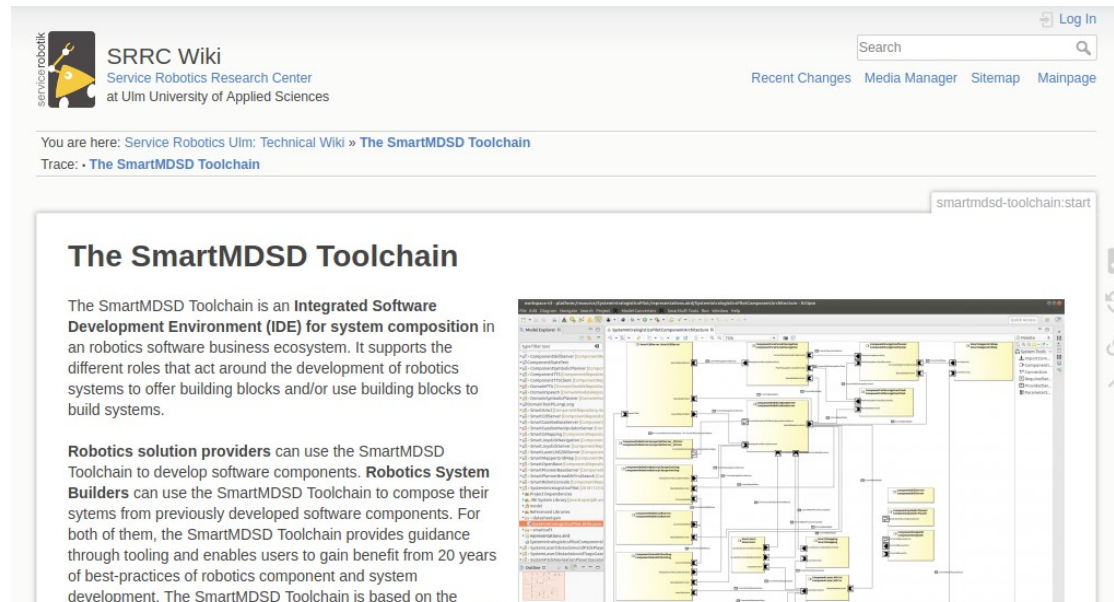


RobMoSys



Where to get the SmartMDSD Toolchain

- **SmartMDSD Toolchain easy entry:**
 - Available as standalone installation
 - and **pre-installed/ready-to-go virtual machine image!**
 - https://robmosys.eu/wiki/baseline:environment_tools:smartsoft:start
- **Tutorials**
 - <https://wiki.servicerobotik-ulm.de/tutorials:start>
 - <https://wiki.servicerobotik-ulm.de/how-tos:start>



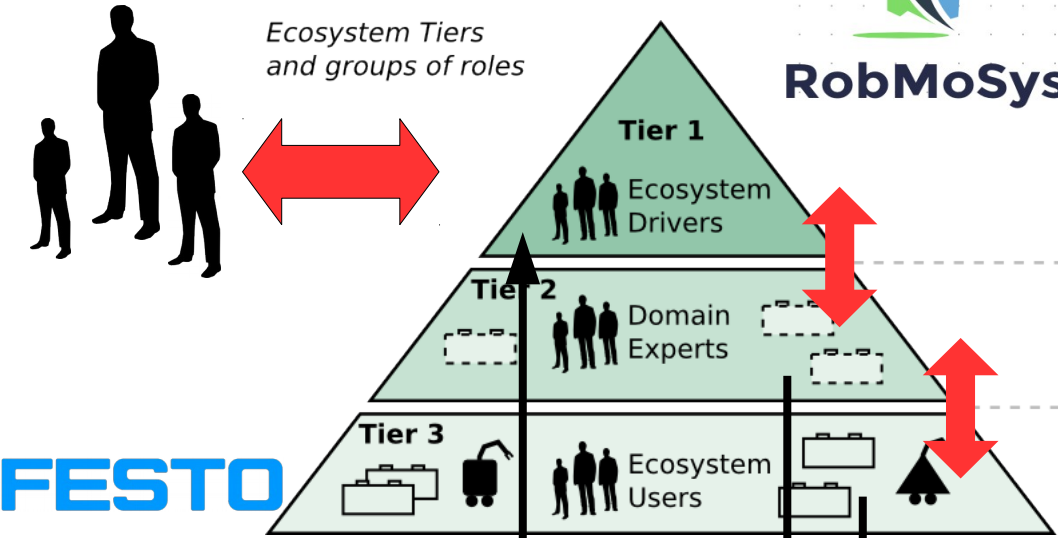
The screenshot shows the SRRC Wiki page for the SmartMDSD Toolchain. The page header includes the SRRC Wiki logo, the text "Service Robotics Research Center at Ulm University of Applied Sciences", a search bar, and navigation links for "Recent Changes", "Media Manager", "Sitemap", and "Mainpage". The breadcrumb trail indicates the user is in "Service Robotics Ulm: Technical Wiki » The SmartMDSD Toolchain". The main content area is titled "The SmartMDSD Toolchain" and contains a paragraph describing it as an "Integrated Software Development Environment (IDE) for system composition in a robotics software business ecosystem". It also mentions that "Robotics solution providers" and "Robotics System Builders" can use the toolchain. To the right of the text is a screenshot of the SmartMDSD Toolchain interface, which displays a complex block diagram of a system architecture with various components and their interconnections.

SmartMDSD Toolchain

- The SmartMDSD Toolchain is an Integrated Development Environment (**IDE**) for robotics software to support system composition according to the structures of RobMoSys.
- Strong in building **real systems**
- Very **mature**, Used in Products delivered by **FESTO**



Industry 4.0 Intralogistics Pilot



Enabling access
to the RobMoSys
Approach

Composition
of previously
developed
building
blocks

eclipse
Sirius
Xtext

SmartMDSD
Toolchain

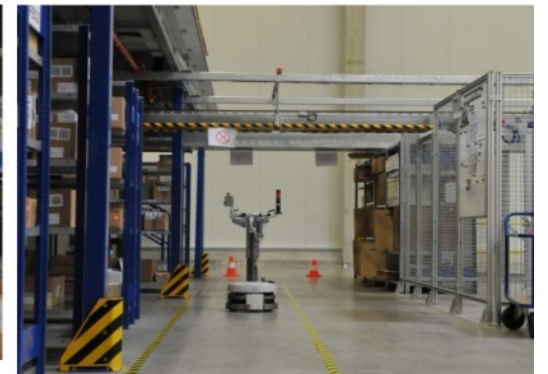
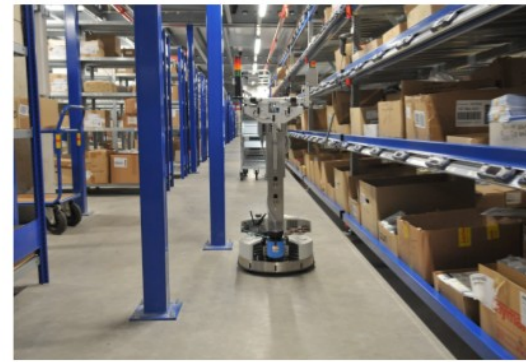
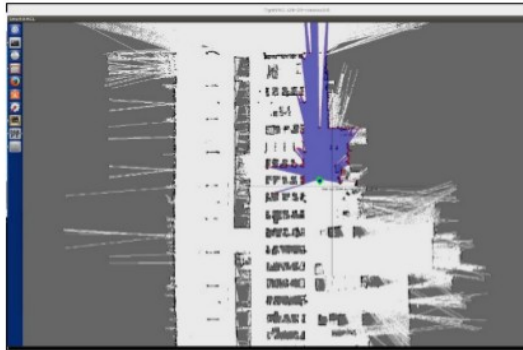


Applications built with the SmartMDSD Toolchain

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



RobMoSys





YouTube^{DE}

Search



RobMoSys

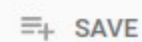


The operator places a box on top of the robot and presses a button to indicate it is ready to proceed to the delivery position.

The video player interface shows a room with a robot and a control panel. The control panel has two graphs: SAFETY and PERFORMANCE. The SAFETY graph shows a red bar at 1.0. The PERFORMANCE graph shows a blue bar at 0.0. The video player controls at the bottom show a play button, a progress bar at 1:44 / 4:30, and a volume icon.

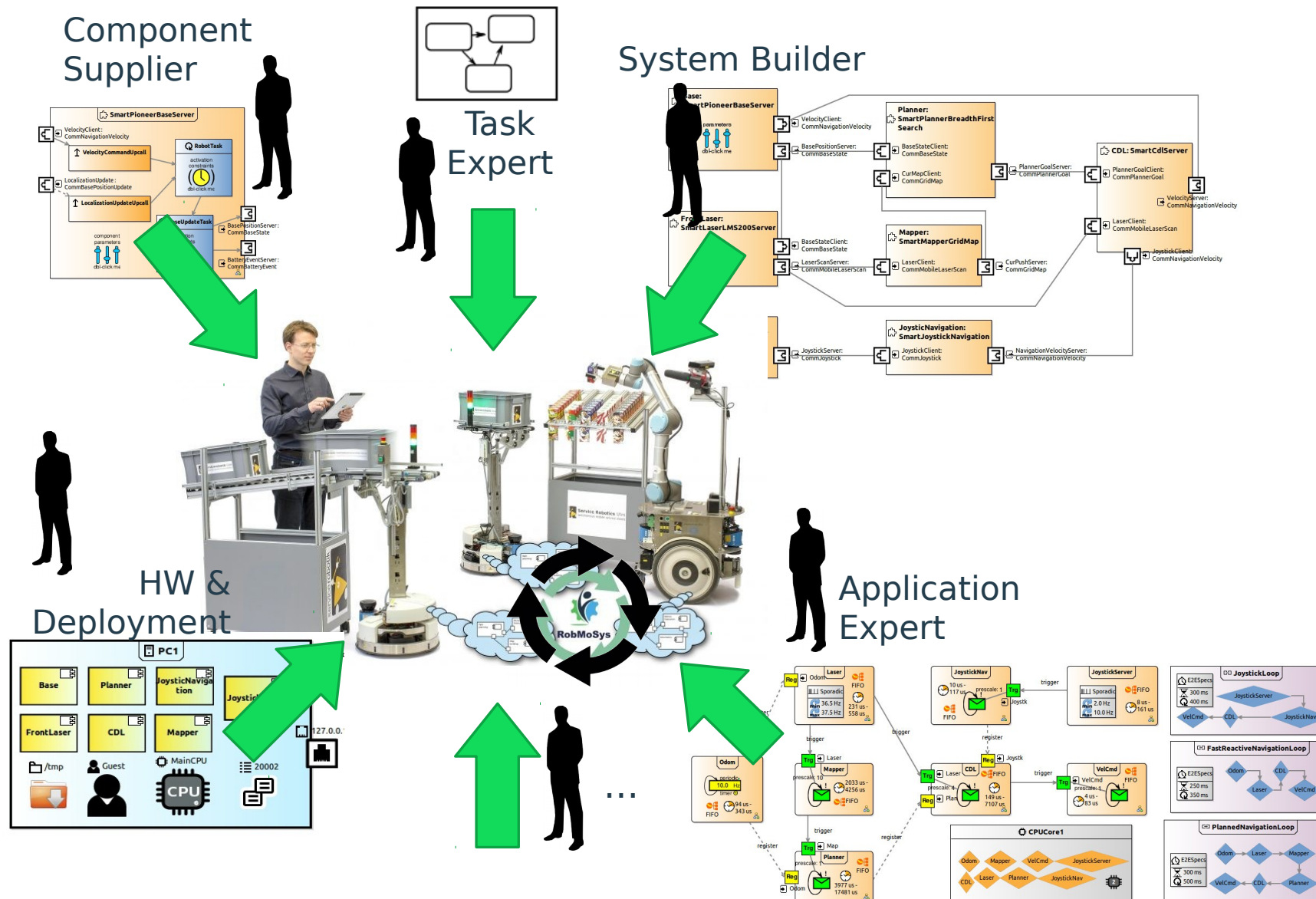
Dealing with Metrics on Non-Functional Properties in RobMoSys

43 views



Ro bot
Q uality
MΣ tries

Ecosystem, Separation of Roles, Composition



RobMoSys Modeling Directory

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	Status
CommBasicObjects	A collection of very basic service definitions and communication objects for use in almost every robotics system.	Universal	HSU	SmartMDSD Toolchain v3	Ready
CommNavigationObjects	A collection of domain models for navigation.				
CommRobotinoObjects	A collection of domain models for the Robotino robot.				
CommLocalizationObjects	A collection of domain models for localization.				
CommManipulationPlannerObjects	A collection of domain models for manipulation.				
CommManipulatorObjects	A collection of domain models for manipulation.				
CommObjectRecognitionObjects	A collection of domain models for object recognition.				

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

Tier 3 Component Models

Name	Description	Purpose	Vendor	Tooling	Status	Fig
SmartCdlServer	Implements the Curvature Distance Lookup (CDL) algorithm for fast local obstacle avoidance. It considers the dynamics and kinematics of the robot.	Navigation	HSU	SmartMDSD Toolchain v3	Ready	   
ComponentLaserObstacleAvoidance						
ComponentPlayerStageSimulation						
ComponentSymbolicPlanner						

Tier 3 Systems

Name	Description	Purpose	Vendor	Tooling
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
SystemP3dxNavigationRealWorld	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 P3dx Robot in real world.	Navigation	HSU	SmartMDSD Toolchain v3



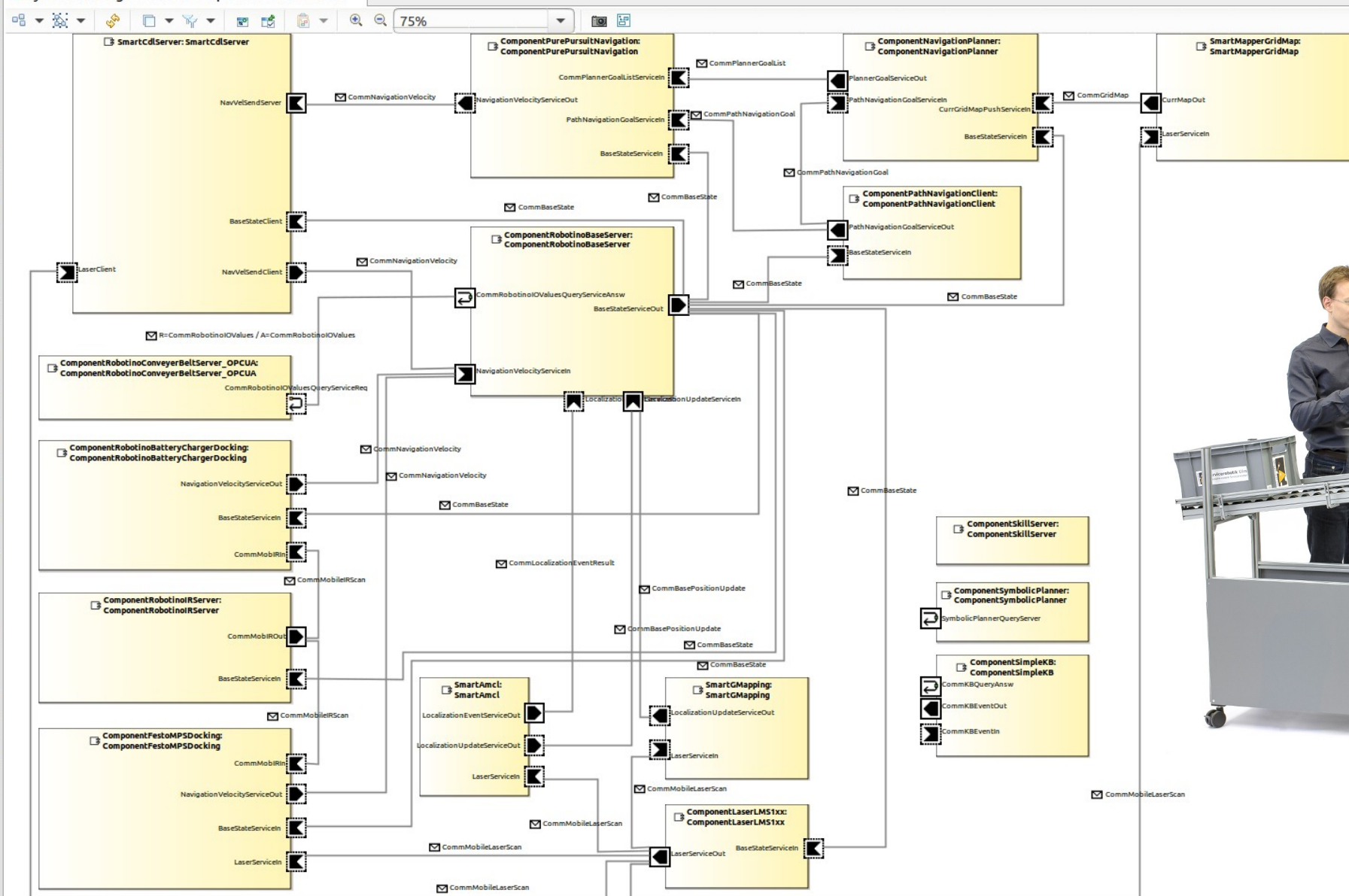
Model Explorer

- type filter text
- > ComponentSkillServer [ComponentRepository]
 - > ComponentStateTest
 - > ComponentSymbolicPlanner [ComponentRepository]
 - > ComponentTTTS [ComponentRepository]
 - > ComponentTTSClnt [ComponentRepository]
 - > DomainPTU [DomainModelsRepository]
 - > DomainSpeech [DomainModelsRepository]
 - > DomainSymbolicPlanner [DomainModelsRepository]
 - > DomainTestPiLongLong
 - > SmartAmcl [ComponentRepository]
 - > SmartCdServer [ComponentRepository]
 - > SmartGazeboBaseServer [ComponentRepository]
 - > SmartGazeboManipulatorServer [ComponentRepository]
 - > SmartGMapping [ComponentRepository]
 - > SmartJoystickNavigation [ComponentRepository]
 - > SmartJoystickServer [ComponentRepository]
 - > SmartLaserLMS200Server [ComponentRepository]
 - > SmartMapperGridMap [ComponentRepository]
 - > SmartOpenRave [ComponentRepository]
 - > SmartPioneerBaseServer [ComponentRepository]
 - > SmartPlannerBreadthFirstSearch [ComponentRepository]
 - > SmartRobotConsole [ComponentRepository]
 - > SystemIntralogisticsPilot [20181123-D]
 - > Project Dependencies
 - > JRE System Library [java-8-openjdk-arm64]
 - > model
 - > Referenced Libraries
 - > > datasheet-gen
 - > SystemIntralogisticsPilot.skills.json
 - > smartsoft
 - > representations.aird
 - > SystemIntralogisticsPilotComponentArchitecture
 - > SystemLaserObstacleAvoidP3dxPlayer
 - > SystemLaserObstacleAvoidTiagoGazebo
 - > SystemP3dxNavigationPlayerStageSim

Outline



SystemIntralogisticsPilotComponentArchitecture



Quick Access

Palette

System Tools

ImportCom...

ComponentL...

Connection

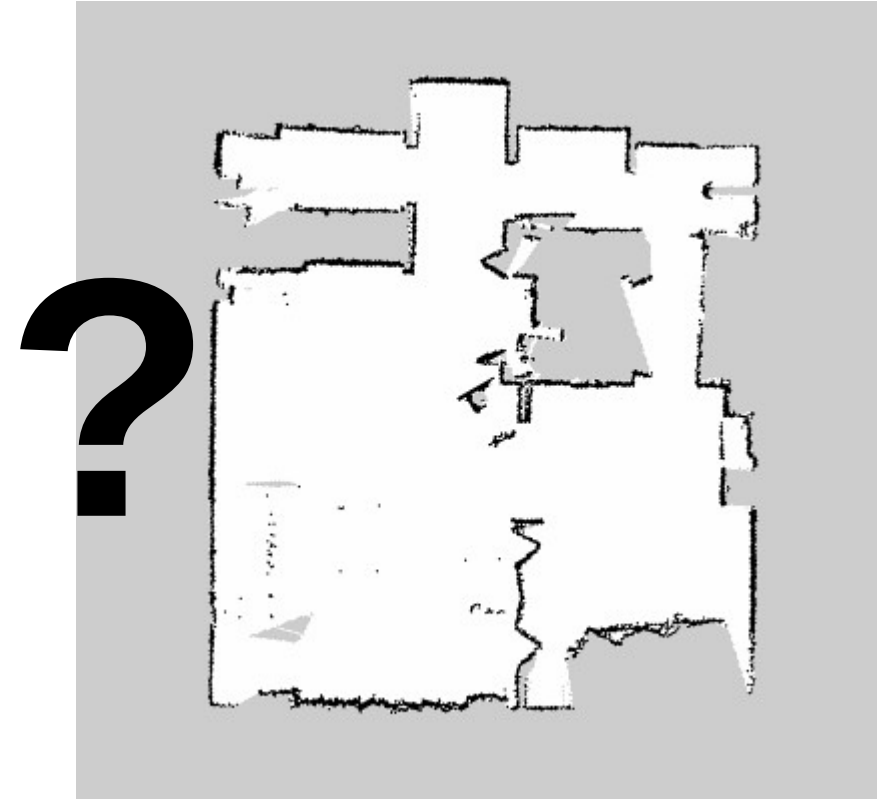
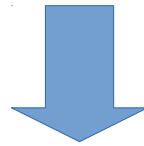
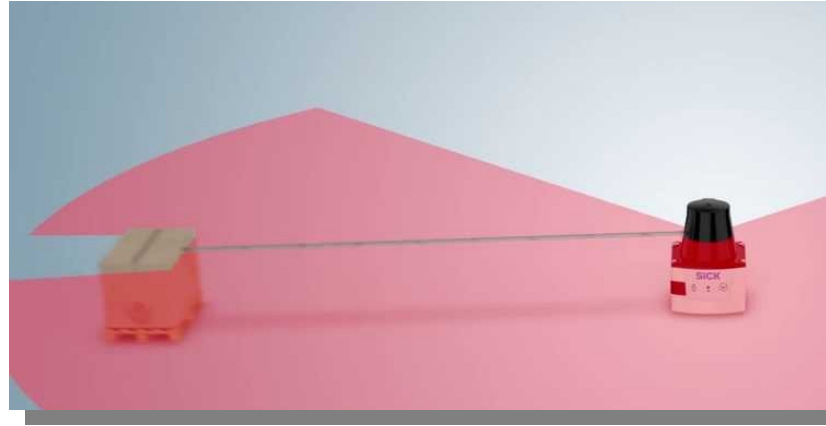
RequiredSer...

ProvidedSer...

ParametersS...

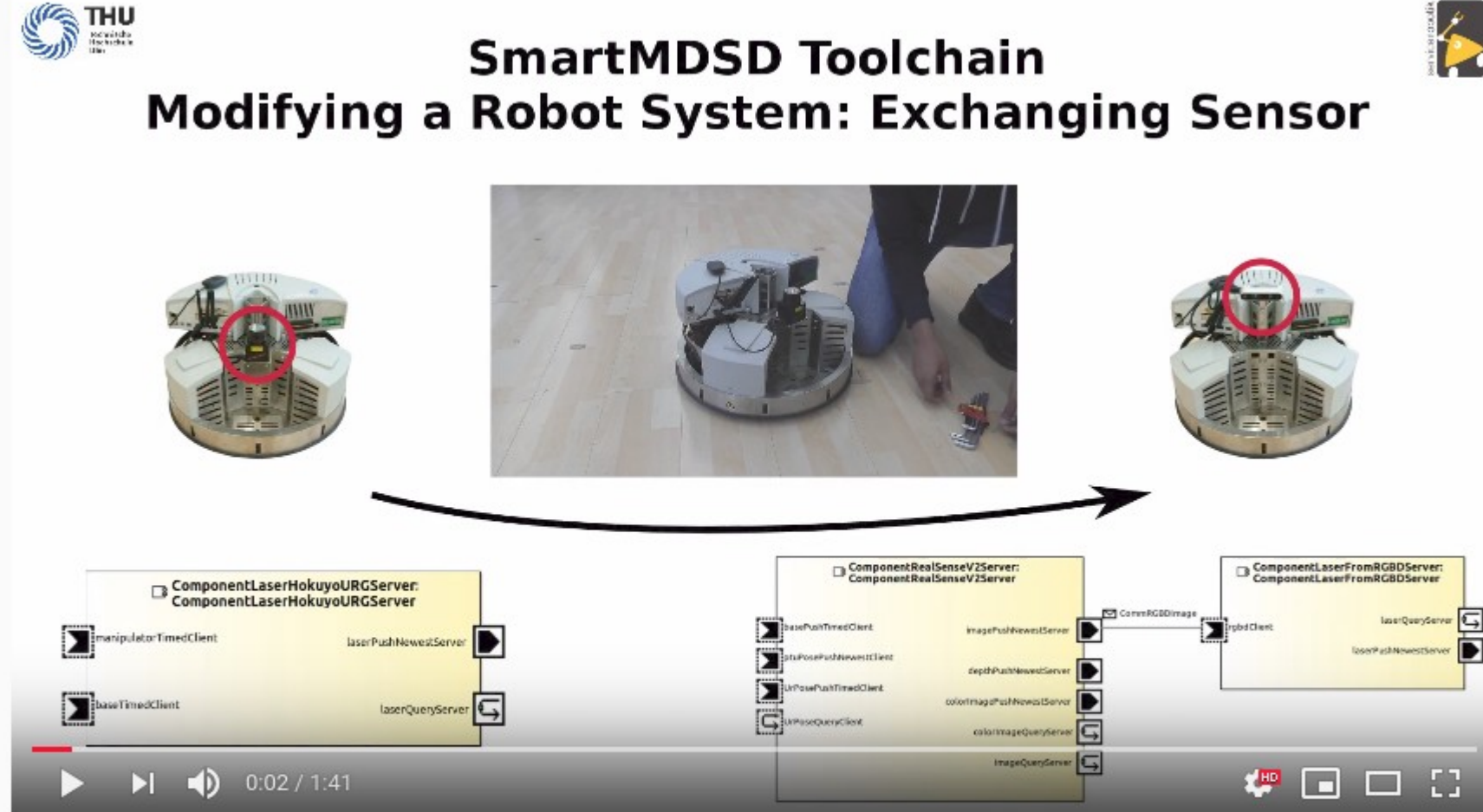


Replacement of Components



Replacement of Components

SmartMDSD Toolchain
Modifying a Robot System: Exchanging Sensor



The diagram illustrates the SmartMDSD toolchain architecture for modifying a robot system. It shows the replacement of a laser sensor component with a RealSense V2 sensor component. The architecture is divided into three main sections: the initial state (left), the modification process (middle), and the final state (right). The initial state shows a robot with a laser sensor (red circle) and a corresponding component in the toolchain. The modification process shows a person's hand interacting with the robot. The final state shows the robot with a new sensor (red circle) and the updated component in the toolchain. The toolchain diagram shows the following components and connections:

- Initial State (Left):**
 - ComponentLaserHokuyoURServer: ComponentLaserHokuyoURServer
 - manipulatorTimedClient
 - baseTimedClient
 - laserPushNewestServer
 - laserQueryServer
- Modification Process (Middle):**
 - ComponentRealSenseV2Server: ComponentRealSenseV2Server
 - basePushTimedClient
 - basePushNewestClient
 - basePushTimedClient
 - basePushQueryClient
 - imagePushNewestServer
 - depthPushNewestServer
 - colorImagePushNewestServer
 - colorImageQueryServer
 - imageQueryServer
- Final State (Right):**
 - ComponentLaserFromRGBDServer: ComponentLaserFromRGBDServer
 - laserQueryServer
 - laserPushNewestServer

The diagram also shows a 'ConvertRGBImage' block connecting the RealSense V2 server to the LaserFromRGBD server. The video player interface at the bottom shows a play button, a progress bar at 0:02 / 1:41, and a red 'HD' icon.

SmartMDSD Toolchain - Modifying a Robot System: Exchanging Sensor

34 views

1 0 SHARE SAVE ...

<https://www.youtube.com/watch?v=RHvvb6ITHG4>

Demo:

Piecing together software components

The **System Builder View**

Example: Simple Navigation Example

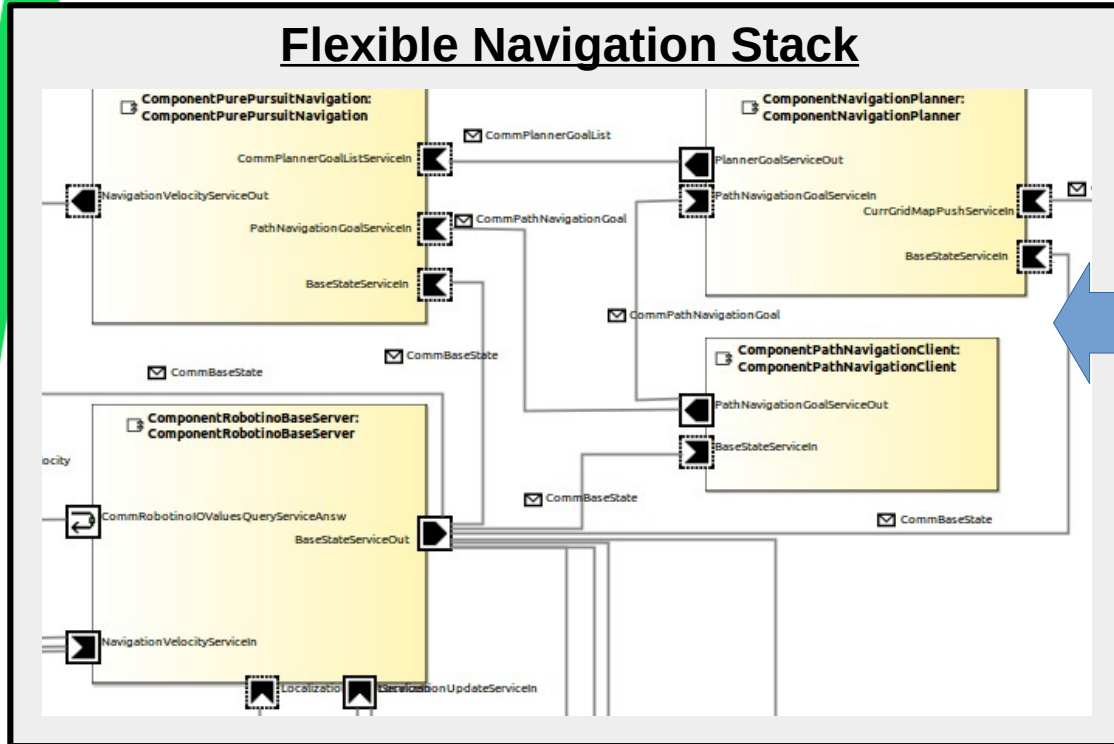
Tutorial and screencast to reproduce at home:

<https://wiki.servicerobotik-ulm.de/tutorials:develop-your-first-system:start>

<https://wiki.servicerobotik-ulm.de/tutorials:laser-obstacle-avoid-scenario:start>

<https://wiki.servicerobotik-ulm.de/tutorials:flexible-navigation-task:start>

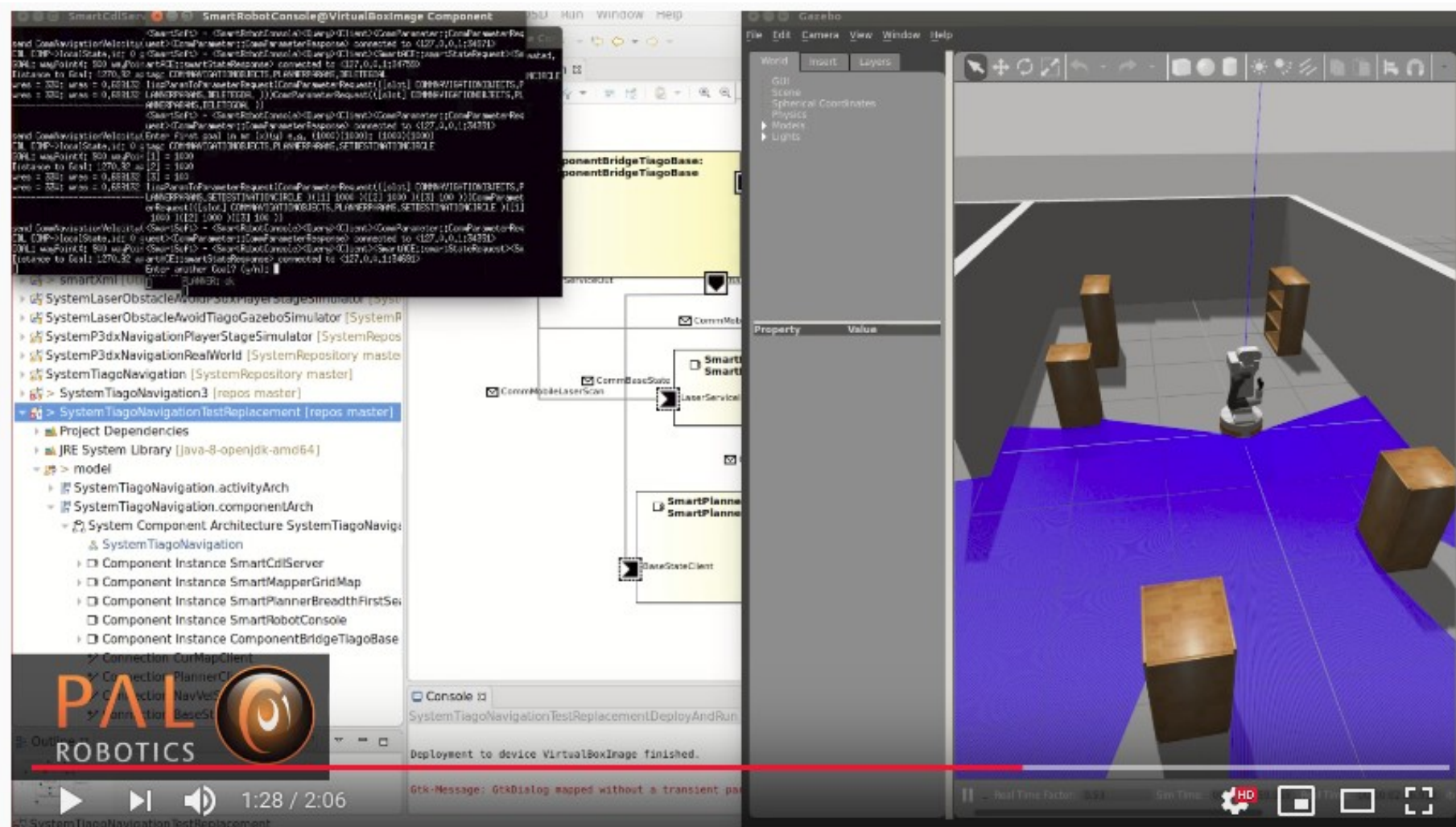
Replacement of Components: **Flexible Navigation Stack**



Replacement of Components



RobMoSys

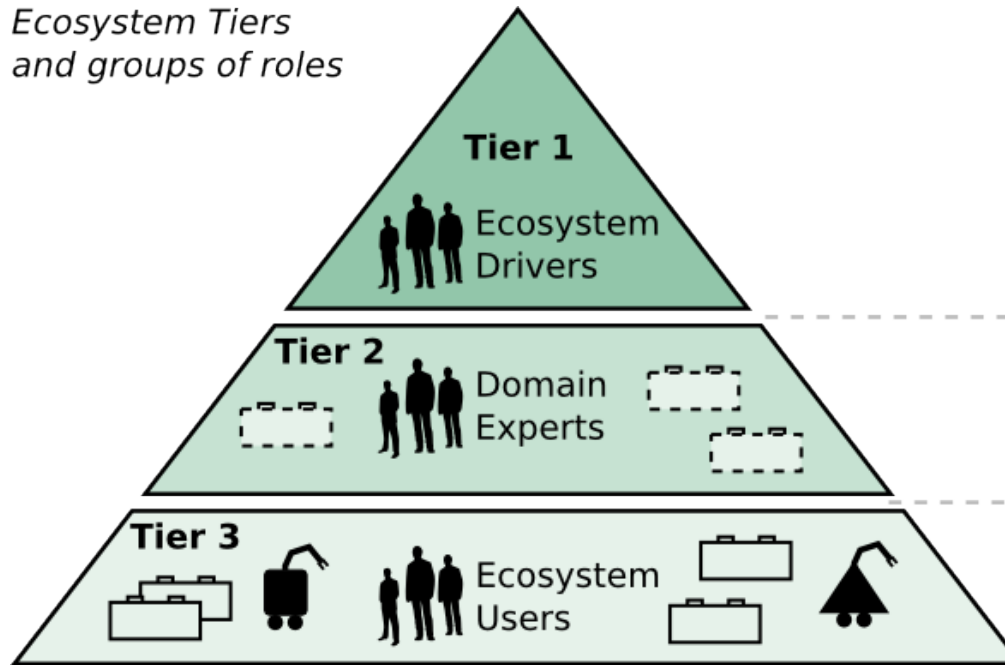


RobMoSys tooling in TIAGo robot

<https://www.youtube.com/watch?v=FCvK9dAZXPo>

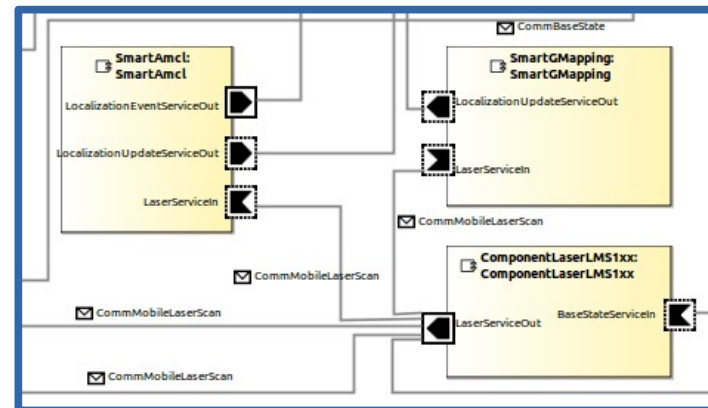
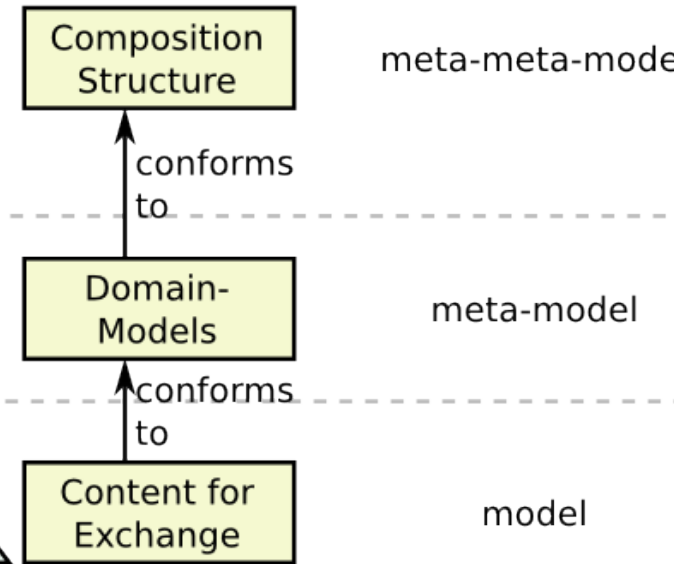
Ecosystem Organization

*Ecosystem Tiers
and groups of roles*



Tier Elements

In terms of modeling



Demo:

Modeling and implementing a Software Component: **The Component Supplier View**

Example: Simple Obstacle Avoidance Component

Tutorial and screencast to reproduce at home:

<https://wiki.servicerobotik-ulm.de/tutorials:develop-your-first-component:start>

<https://wiki.servicerobotik-ulm.de/tutorials:develop-your-first-domain-model:start>

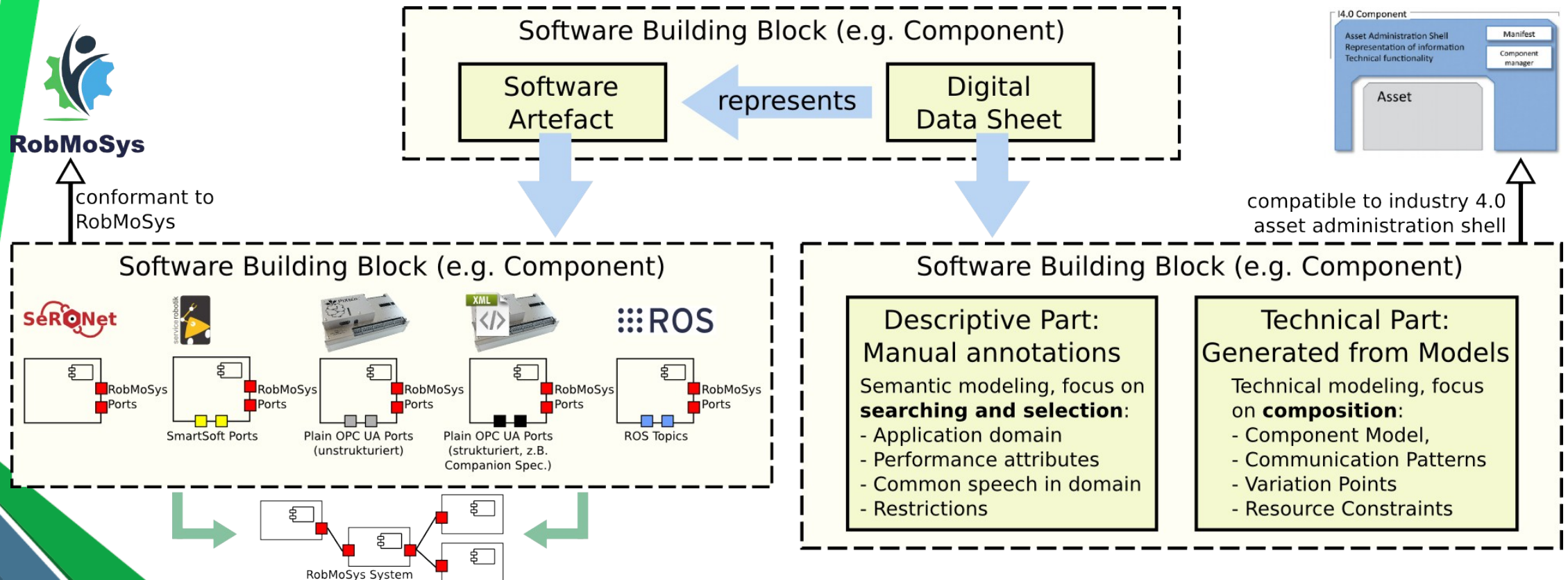


Outline

- Introduction
- Modeling Systems
- Modeling and implementing Components
- **The Digital Data Sheet**



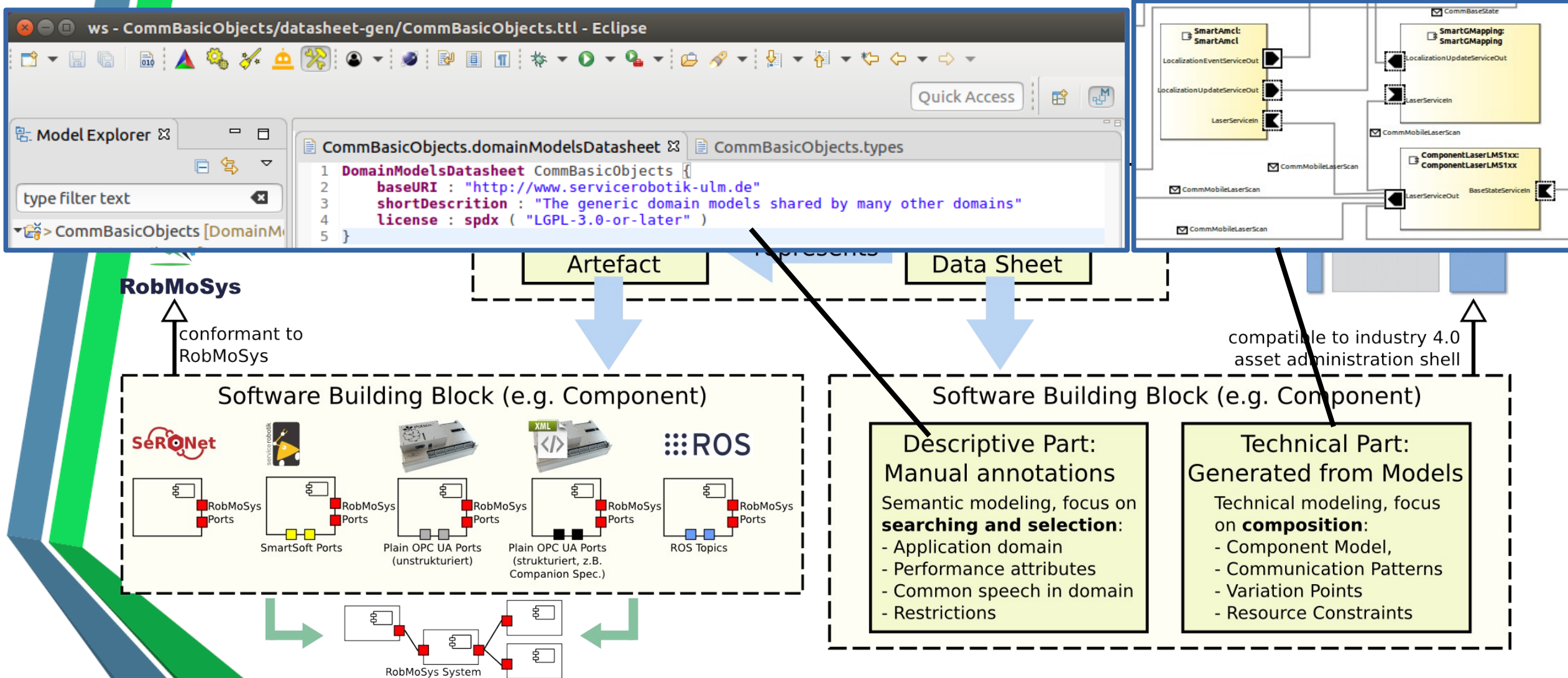
Digital Data Sheet



Digital Data Sheet



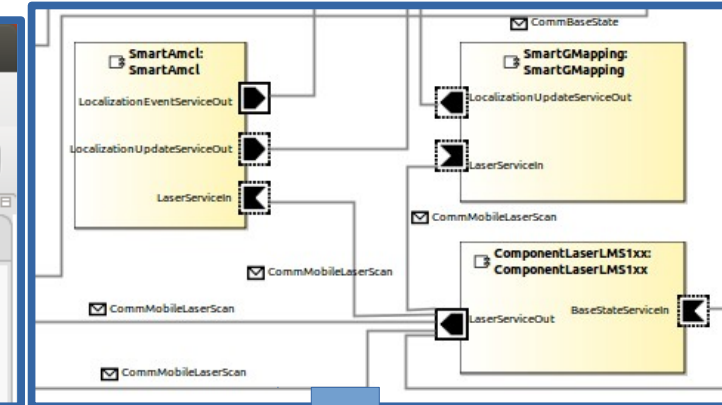
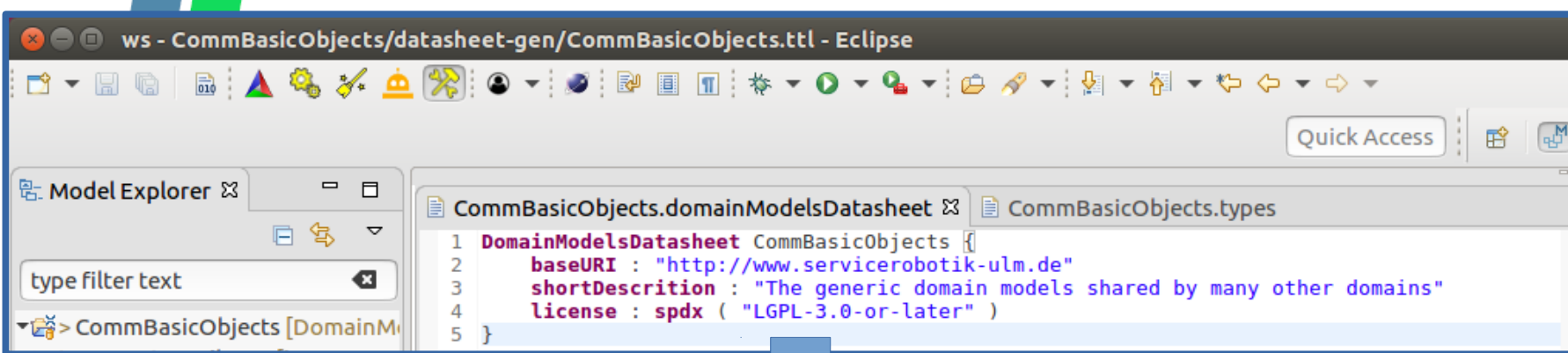
RobMoSys



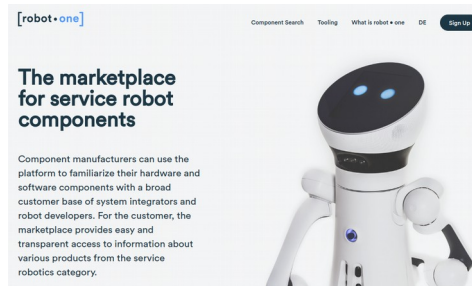
Digital Data Sheet



RobMoSys



RDF, Ontologies, ...



Documentation
for humans

Component Parameters SmartCdlServerParams

InternalParameter PathNav

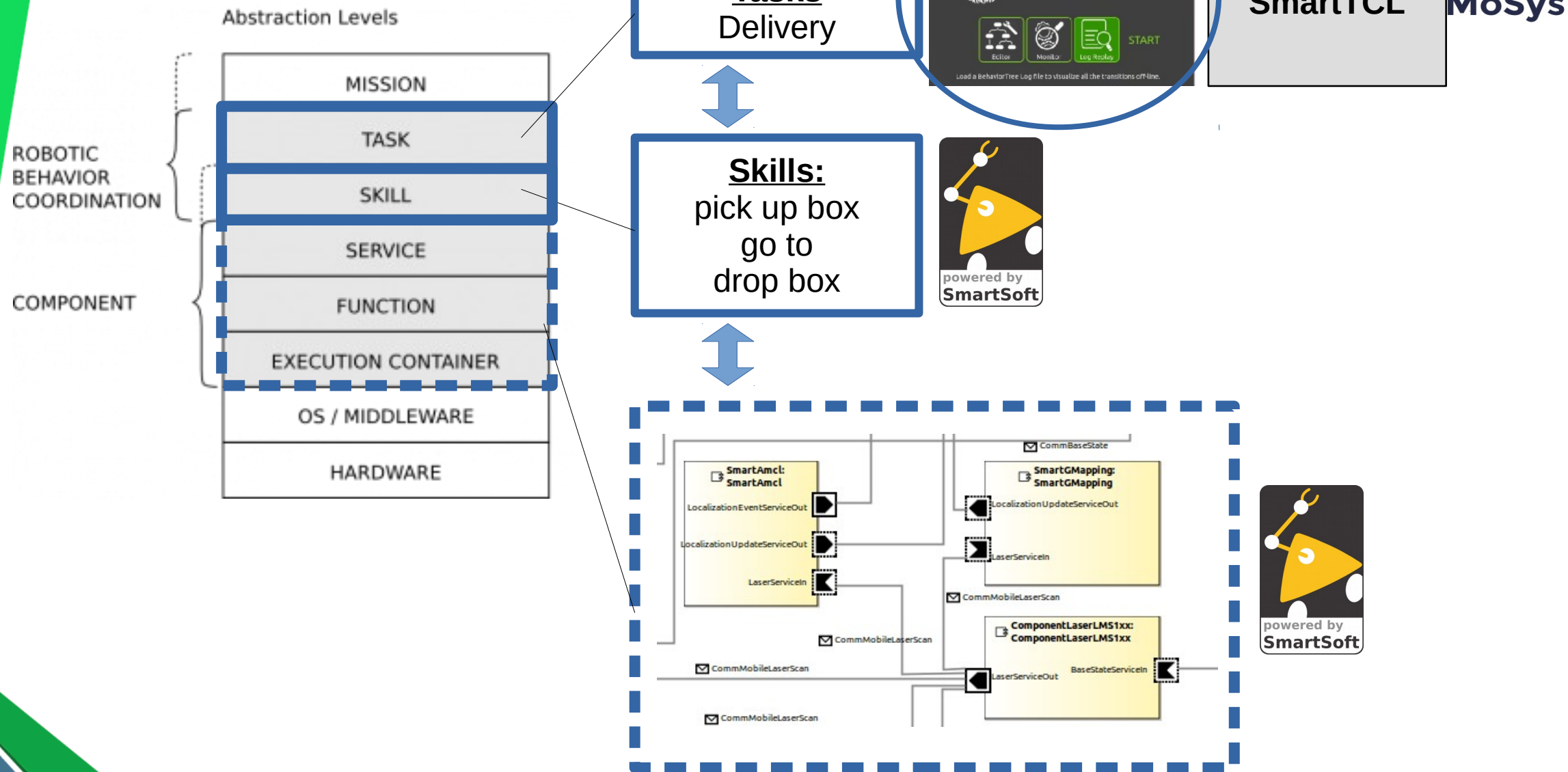
Attribute Name	Attribute Type	Description
pathNavPredictedGoalPose_control1_dist	Double	
pathNavPredictedGoalPose_control1_speed	Double	
pathNavPredictedGoalPose_control2_dist	Double	
pathNavPredictedGoalPose_control2_speed	Double	
pathNavPredictedGoalPose_control3_dist	Double	
pathNavPredictedGoalPose_control3_speed	Double	
pathNavPredictedGoalPose_minDist	Double	
pathNavRecover_max_dist	Double	
robotBlocked_event_timeout	UInt16	timeout for robot being block in seconds

Tooling
Interoperability



...

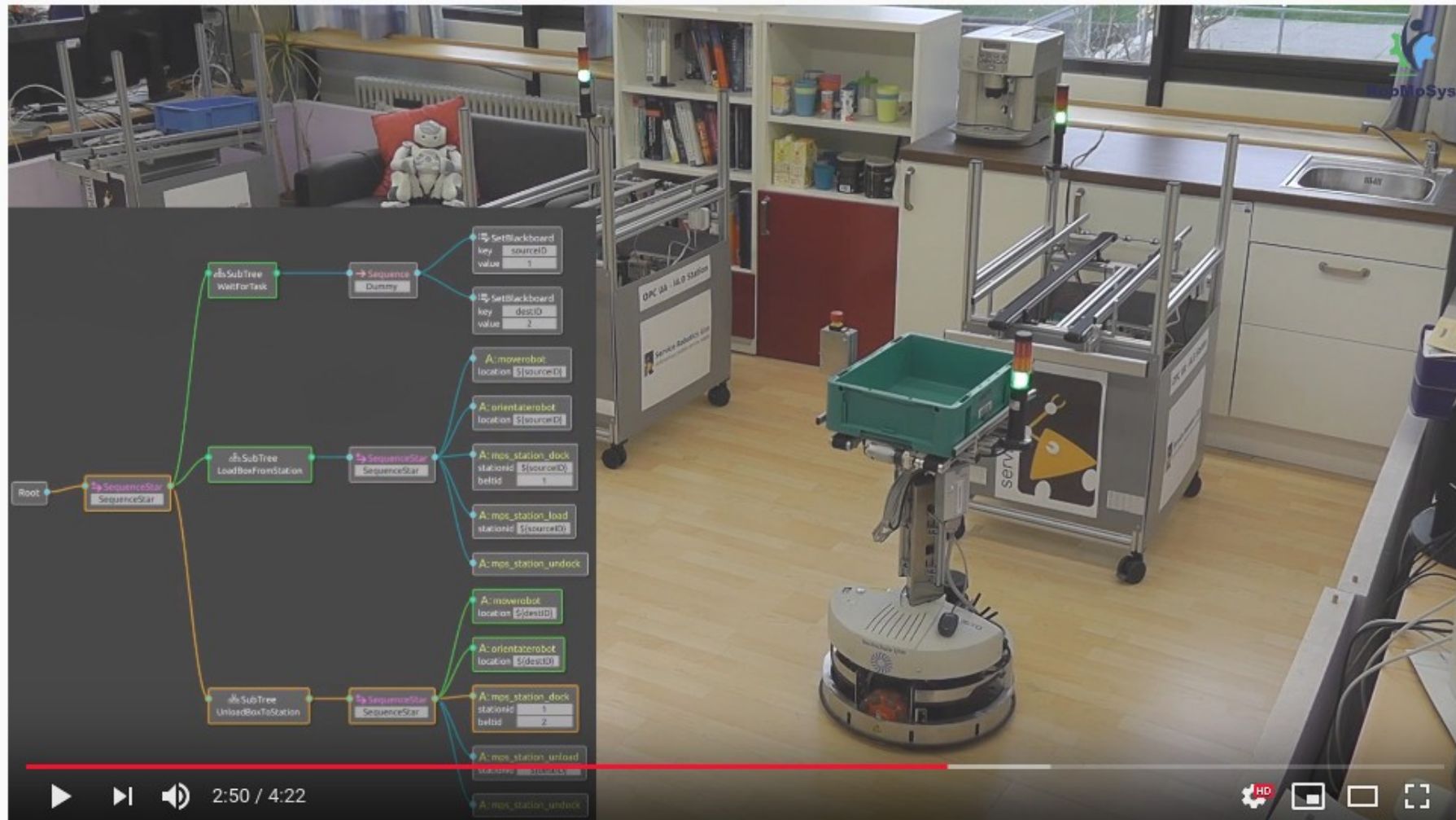
Behaviour Coordination



Behaviour Coordination



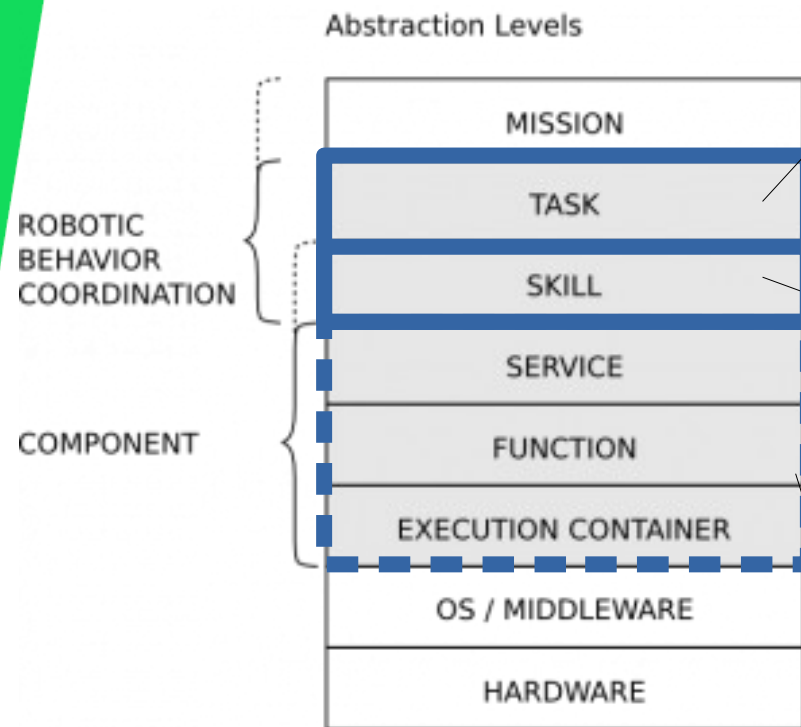
RobMoSys



Robotic Behavior in RobMoSys using Behavior Trees and the SmartMDSD Toolchain

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

Behaviour Coordination



Tasks
Delivery

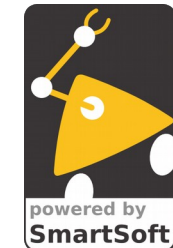
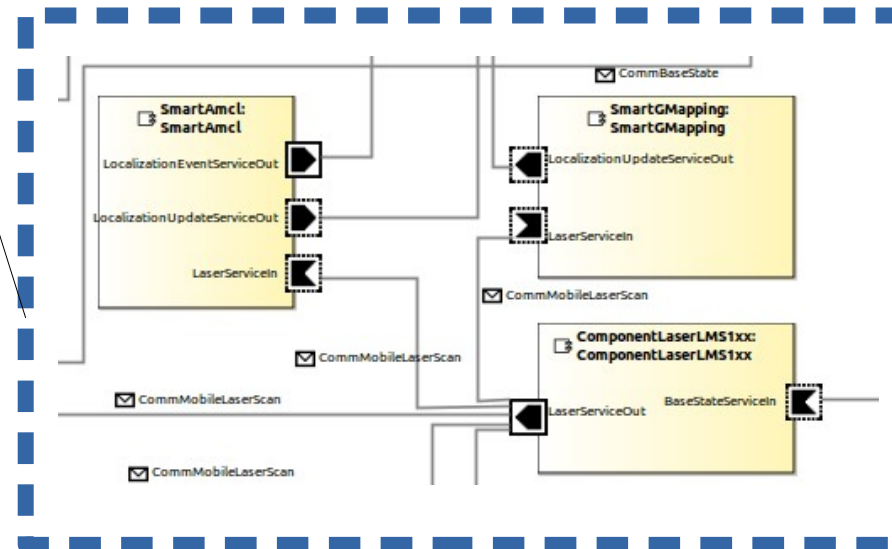
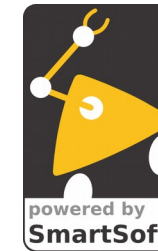


Skills:
pick up box
go to
drop box



SmartTCL

MoSys



Behaviour Coordination



RobMoSys



Intralogistics Scenario: Flexible Context Model Adaptation for Robotic Order Picking

<https://www.youtube.com/watch?v=5116bGhXBr8>

IROS 2019 – RobMoSys Tutorial – November 8th 2019 – **Alex Lotz** lotz@hs-ulm.de



Take away messages

- RobMoSys serves the robotics community by collecting and consolidating best practices and approaches related to **model-driven, composition-oriented robotics software engineering**
- The RobMoSys philosophy is to use **the right approach for the right problem by the right expert**, and to support the handover of information and artefacts between different experts
- RobMoSys provides **matured and ready-to-use modeling tools** with code generation and model checks that offer immediate benefit for the design and implementation of real robotic software systems



09:00 – 09:20 Talk: Towards an open and industry-grade European robotics software ecosystem

Presenter: Christian Schlegel

09:20 – 10:00 Interactive Tool Demo: Open source Eclipse-based tooling for system composition: Piecing together software components to pilot applications *Presenter: Dennis Stampfer*

10:00 – 10:45 Interactive Tool Demo: Open Source Eclipse-Based Tooling for Component Builders: Middleware Agnostic Robotics Software Components *Presenter: Alex Lotz*

10:45 – 11:15 Coffee Break

11:15 – 11:55 Interactive Tool Demo: Safety-analysis by model-driven tooling *Selma Kchir*

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. *Presenter: Dennis Stampfer, Alex Lotz*

12:35 - 12:55 Talk: Overview, motivation, benefits of model-driven approaches in robotics: What-if analysis, compliance, predictability, etc. *Presenter: Herman Bruyninckx*

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

*Tutorial “Model-Driven Robot Software Engineering:
From ROS-specific coding to framework-agnostic modeling”*

Interactive Tool Demo: Stepwise Migration to Model-Driven Development



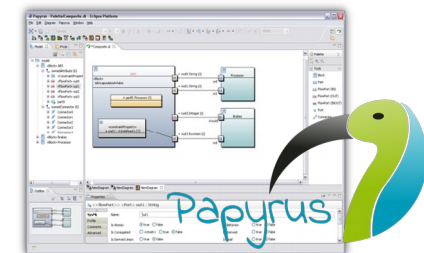
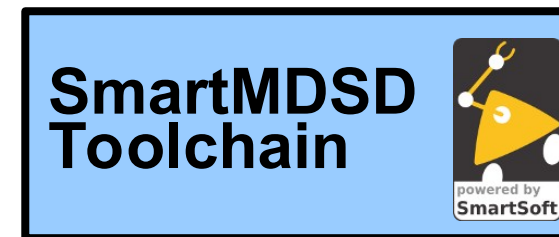
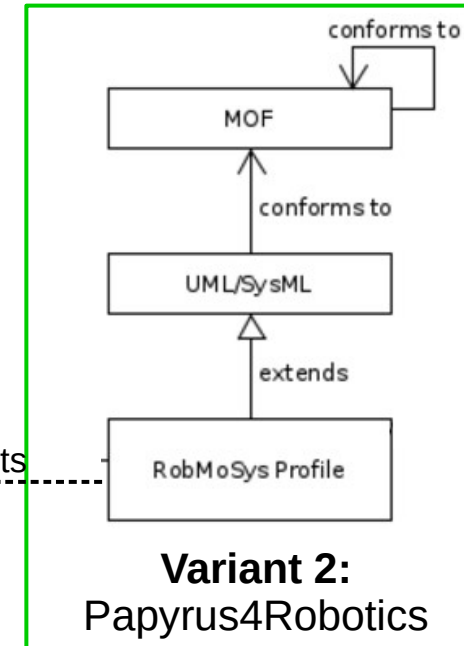
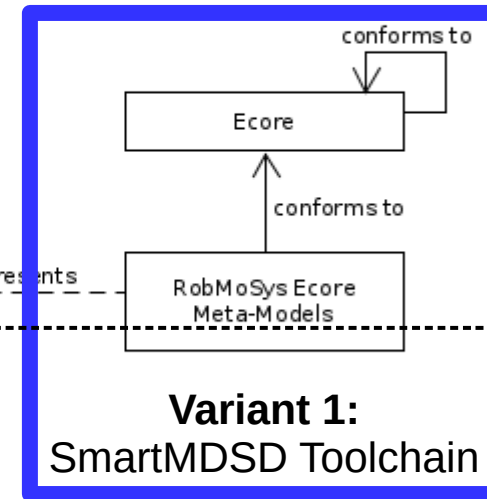
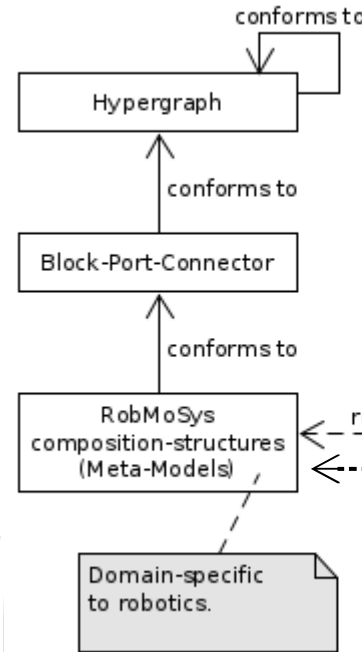
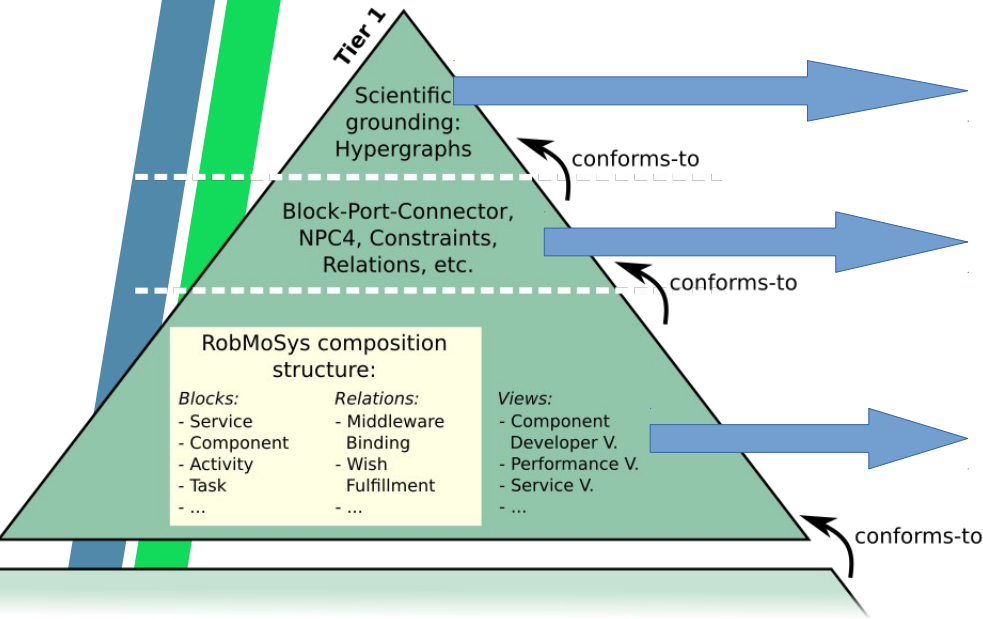
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 732410.



Recap: RobMoSys Tooling: Realization alternatives



RobMoSys



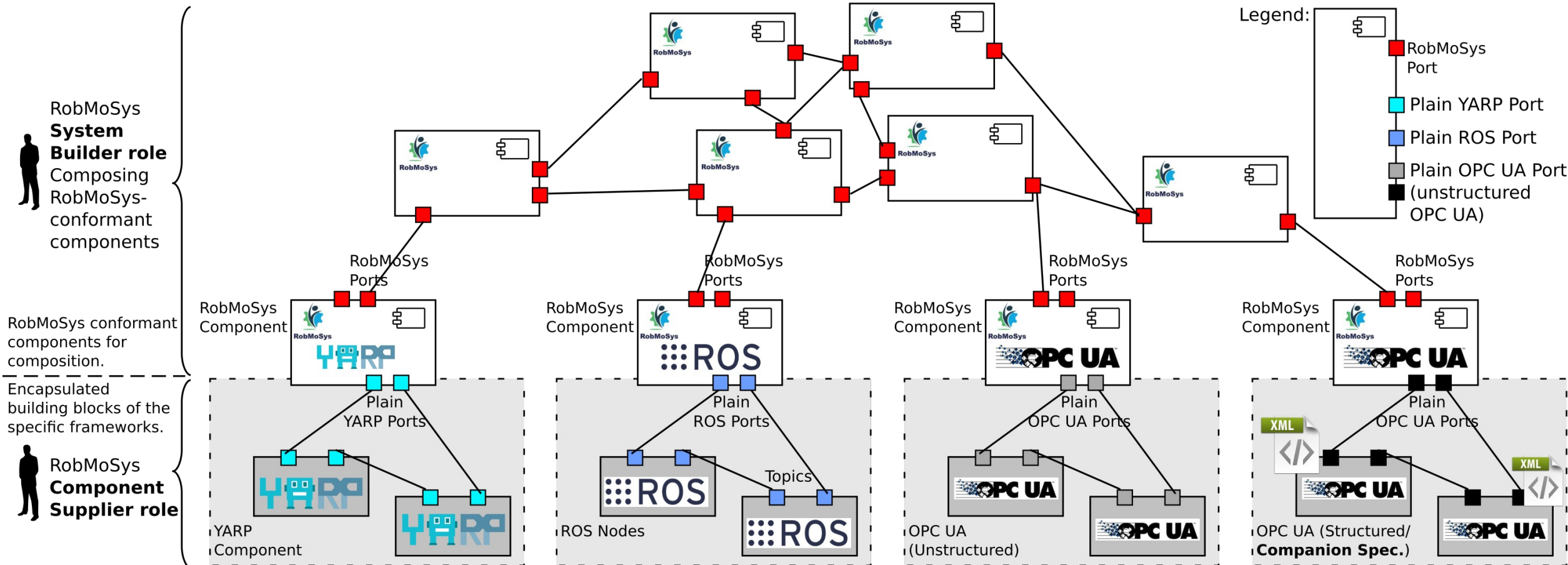


Outline

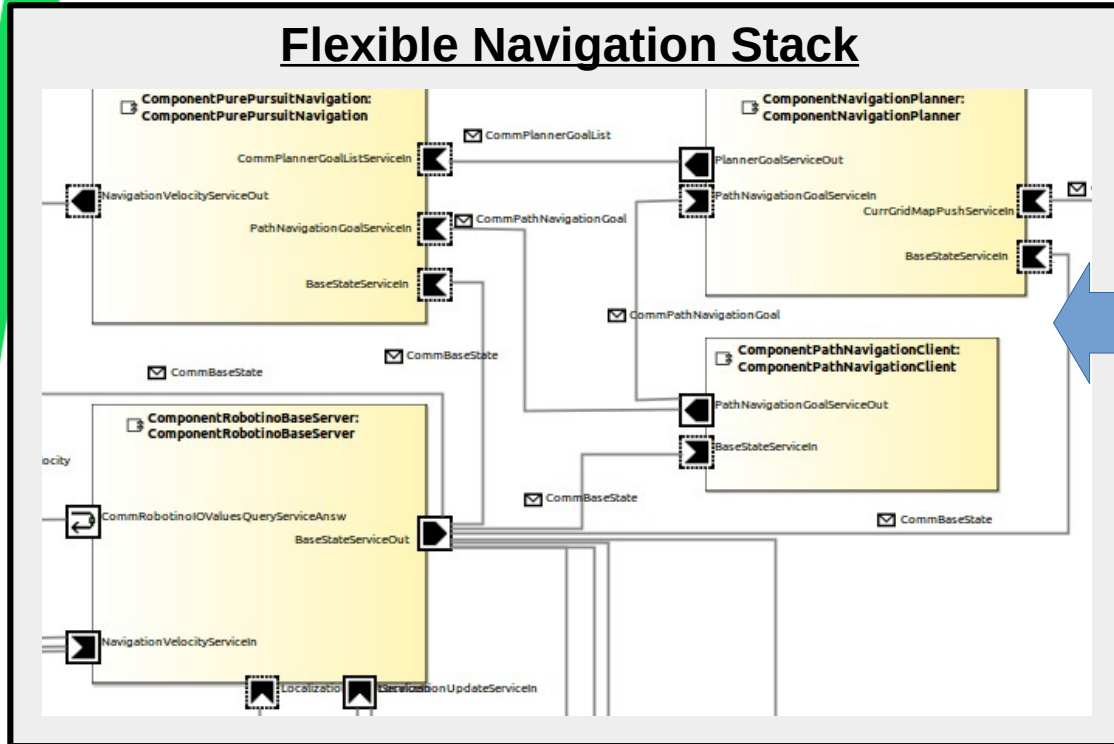
- **Part I: Mixed Port Components**
- Part II: Middleware Agnostic Modeling



Mixed-Port Components



Replacement of Components: **Flexible Navigation Stack**

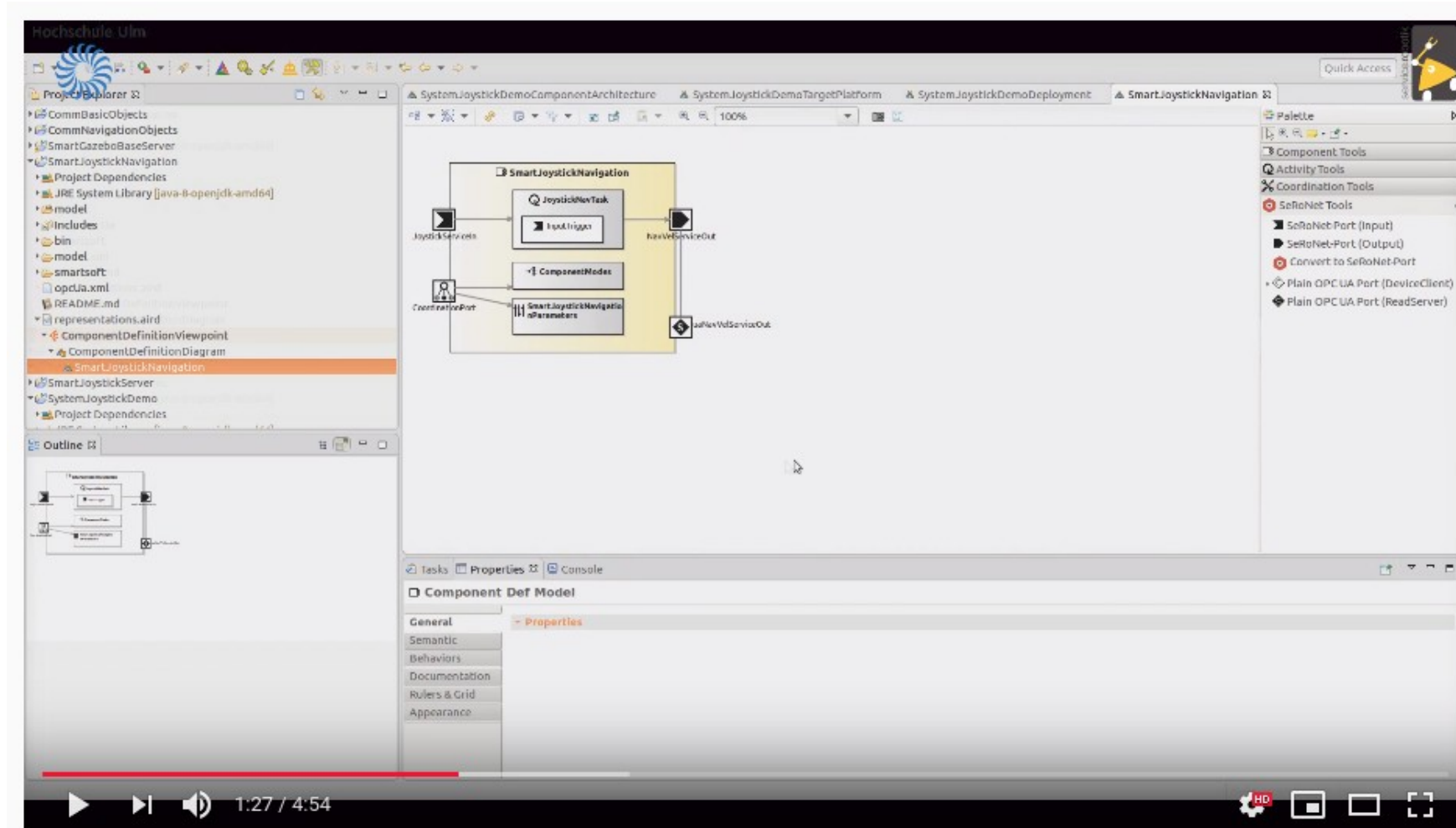


A screenshot of a ROS2 simulation environment, specifically Gazebo. In the center, a small white robot with a blue line extending from its top is positioned in a grey maze-like environment. A terminal window titled 'SmartRobotConsole@Local Component' is open in the foreground, displaying a series of command-line interactions. The terminal shows the execution of a script that connects to a SmartState service, sets a goal, and manages navigation objects. The background features a grid floor and white walls forming the maze. On the right side, there is a logo for 'CA' (Control Architecture) featuring a cloud and various colored squares. At the bottom, a video player interface is visible, showing a progress bar and a timestamp of 7:16 / 7:35.

Using the YARP Framework and the R1 robot with RobMoSys (CARVE ITP)

<https://www.youtube.com/watch?v=p466dW-IU4w&t=51s>

Mixed Port with OPC UA

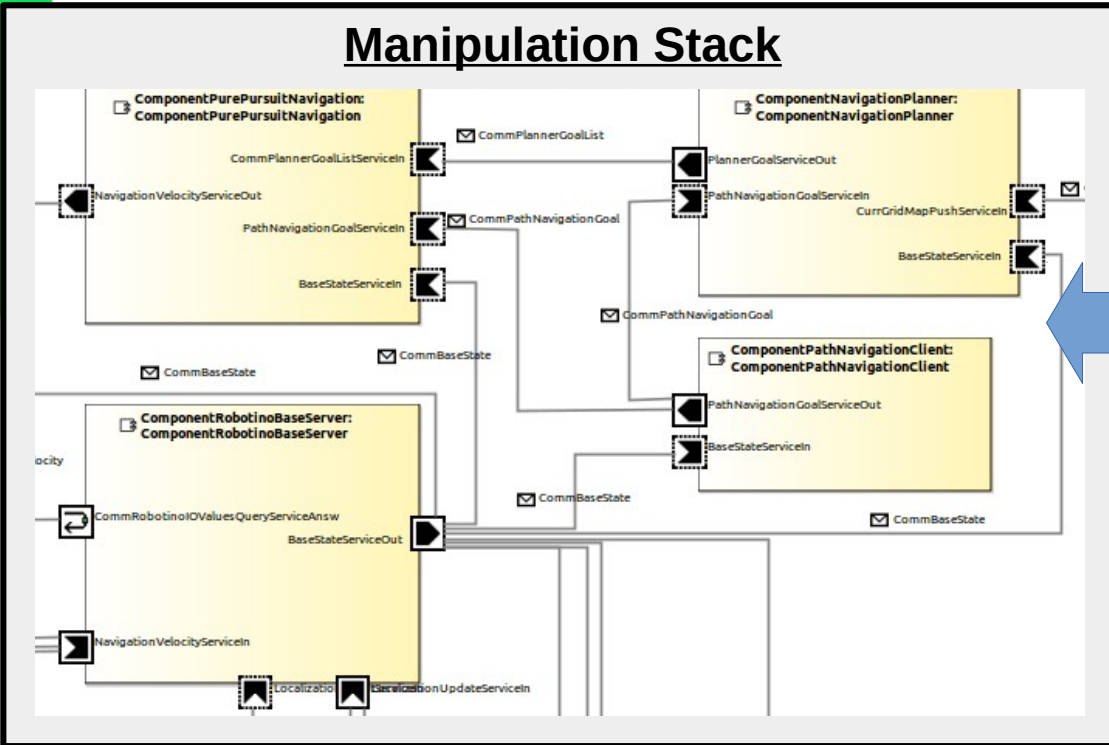


Developing an OPC UA Server: Using the Plain OPC UA Port (ReadServer)

62 views • Published on Jan 22, 2019

0 1 SHARE SAVE ...

Replacement of Components: Manipulation



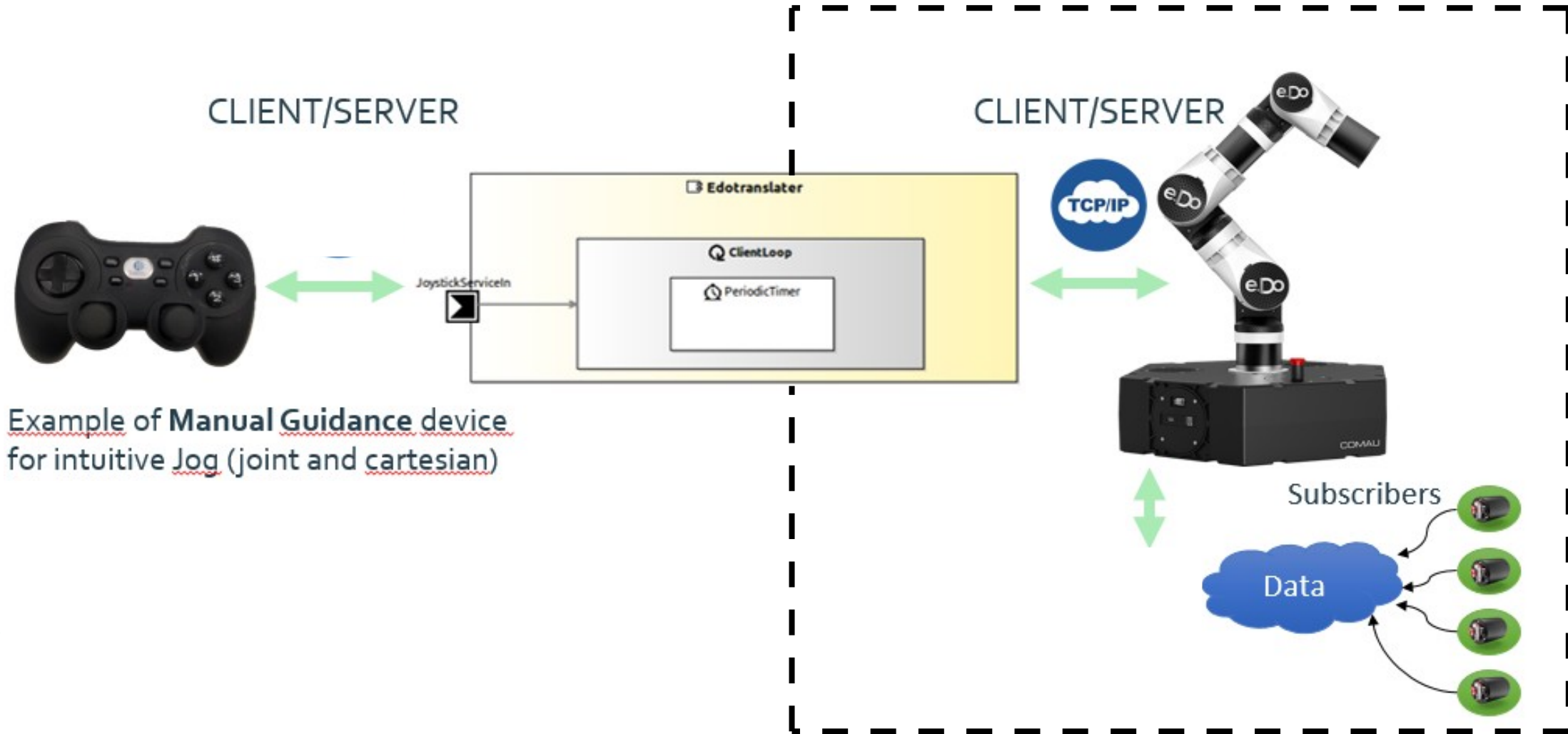
PAL
ROBOTICS



Replacement of Components: Manipulation with e.DO



RobMoSys



Demo:

Mixed-Port Components – **Accessing ROS nodes**

Example: ROS Joystick

Tutorial and screencast to reproduce at home:
<https://wiki.servicerobotik-ulm.de/tutorials:ros:mixed-port-component-ros>



RobMoSys

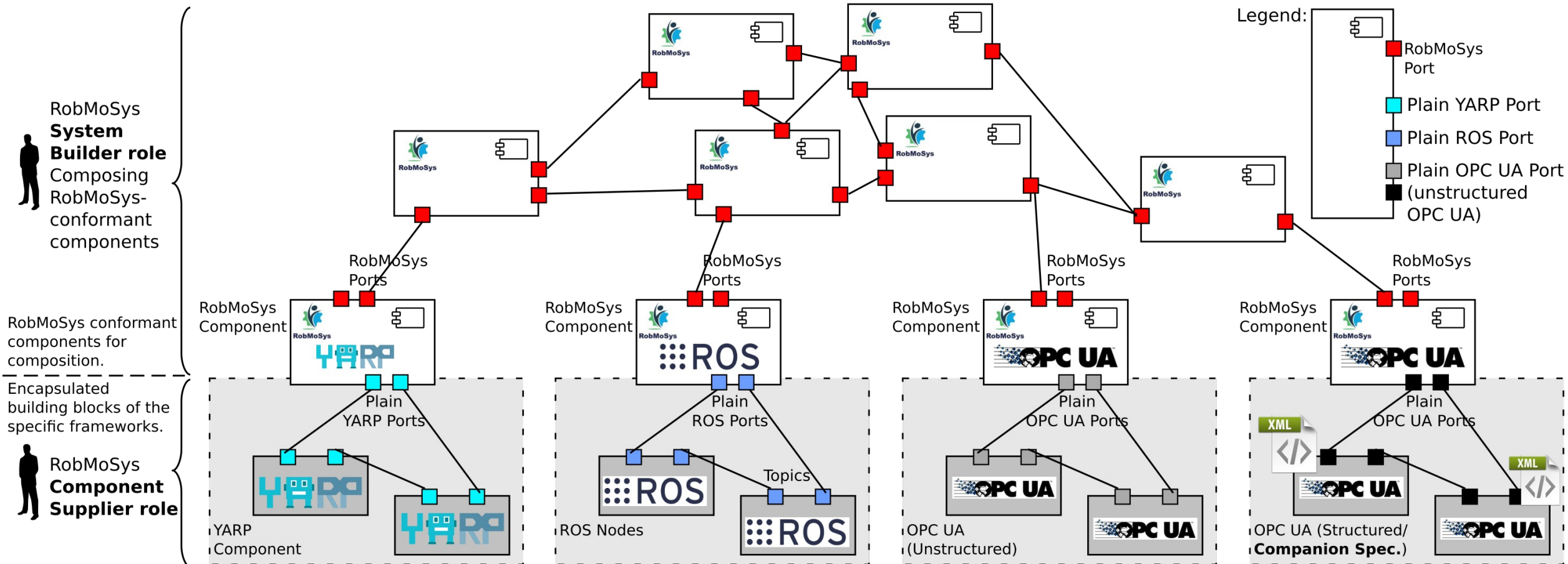
Outline

- Part I: Mixed Port Components
- **Part II: Middleware Agnostic Modeling**

Mixed-Port Components vs. Middleware Abstraction



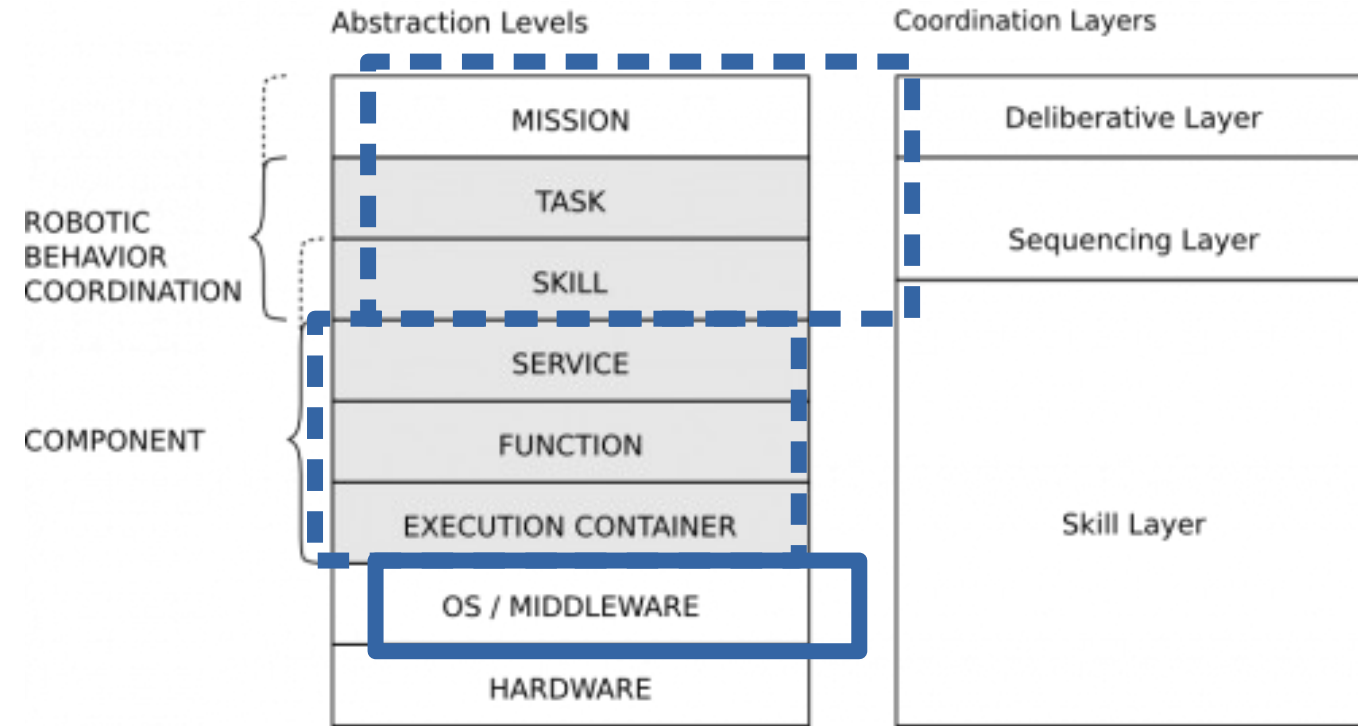
RobMoSys



RobMoSys User-Story: Middleware Late Binding / Middleware agnostic modeling



RobMoSys



Model Explorer

type filter text

- > CommBasicObjects [DomainModelsRepo]
- > 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 [ComponentRepositi]

Outline

ComponentJoystickServer

SystemOpcUaJoystickTestComponentArchitecture

100%

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>

```
graph LR
    subgraph JoystickServer [JoystickServer: ComponentJoystickServer]
        OpcUaJoystickDeviceClient
    end
    subgraph JoystickNavigation [JoystickNavigation: SmartJoystickNavigation]
    end
    subgraph Backend [OPC UA SeRoNet Backend]
    end
    JoystickServer -- JoystickServiceOut --> Backend
    JoystickNavigation -- JoystickServiceIn --> Backend
    Backend -- CommJoystick --> JoystickServer
```

Demo:

Middleware agnostic modeling

Example: Navigation Stack



Take away messages

- Expressive and consistent communication semantics together with a **middleware abstraction layer** and a **middleware-independent API** are important aspects that exist in RobMoSys from day one
 - Middleware technologies evolve over time, so it must be possible **to easily adapt to new middleware solutions (or even support multiple middlewares)** with ideally no modifications of the component code
- **RobMoSys does not exclude approaches as ROS**, instead, it actively supports linking to and gradually migrating from such approaches using the **Mixed Port Component** idea
 - **RobMoSys goes beyond what is currently possible with approaches like ROS**, by implementing the consolidated body of knowledge within model-driven tools with respect to composition-oriented robotics software system development