Composable Models and Software for Robotics Systems

Horizontal and Vertical Composition

Prof. Dr. Christian Schlegel
Forschungsgruppe Servicerobotik
Hochschule Ulm

http://www.hs-ulm.de/schlegel
schlegel@hs-ulm.de

http://www.servicerobotik-ulm.de/
https://www.youtube.com/user/RoboticsAtHsUlm/

Dr. Alex Lotz
M.Sc. Matthias Lutz
Dr. Dennis Stampfer
M.Sc. Timo Blender
Vineet Nagrath, PhD
M.Sc. Matthias Rollenhagen
M.Sc. Nayabrasul Shaik
Gautam Dange, PhD

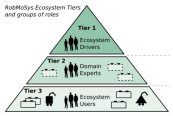
Service Robotics Ulm
autonomous mobile service robots

Hochschule Ulm

Enable success stories and finally produce success stories
Technical User Stories and Benefits

- Composable commodities for robot navigation with traceable and assured properties
- Description of building blocks via model-based data sheets
- Replacement of components and again matching all the attached constraints (requirements, system, building blocks)
- Composition of components and managing all the dependencies,e.g. mounting a camera on a manipulator
- Quality-of-service and management of resource shares
- Determinism when you change the processing platform,e.g. keep cause-effect-chains valid
- Free from hidden interference when you add further components to a system
- Management of non-functional properties and tool-supported trade-off
- Manage gap between design time assumptions and run time situation via e.g. sanity checks
- System analysis tools for what-if questions, trade-off analysis, etc.
- Task modeling for task-oriented robot programming
- Safety and shift from fail-safe to safe-operational (not just “the following things cannot happen” but “the system only behaves like that”)
- How to configure coordinate systems?
- How to inject schedules and manage budgets?
- How to partition synchronous islands(local sync, global async)?
- How to present composable computation models (budget, resource shares) to robotics experts?
- How do uncertainties propagate through chains of computation, motion, etc.?
- How to always reach a safe state (mode transitions)?
- …

Forschungsgruppe Servicerobotik



OPC UA

RobMoSys - Composable Models and Software for Robotics Systems

www.robmosys.eu

01.01.2017 – 31.12.2020, EU H2020-ICT-2016

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

SeRoNet – Eine Plattform zur arbeitsteiligen Entwicklung von Serviceroboter-Lösungen

www.seronet-projekt.de

01.03.2017 – 28.02.2021, BMWi – Technologieprogramm „PAiCE“

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

LogiRob - Multi-Robot-Transportsystem im mit Menschen geteilten Arbeitsraum

http://www.softwaresysteme.pt-dlr.de/media/content/Infoblatt_LogiRob.pdf

01.06.2016 – 31.05.2019, BMBF KMU innovativ

Gefördert vom: Bundesministerium für Bildung und Forschung
DLR Projektträger

ZAFH Intralogistik - Kollaborative Systeme zur Flexibilisierung der Intralogistik

http://zafh-intralogistik.de/

01.03.2017 – 28.02.2020, Land Baden-Württemberg und EU EFRE

Kooperatives Promotionskolleg „Cognitive Computing in Socio-Technical Systems“

Sprecher: Prof. Reichert, UUlm, Prof. Schlegel, HSU

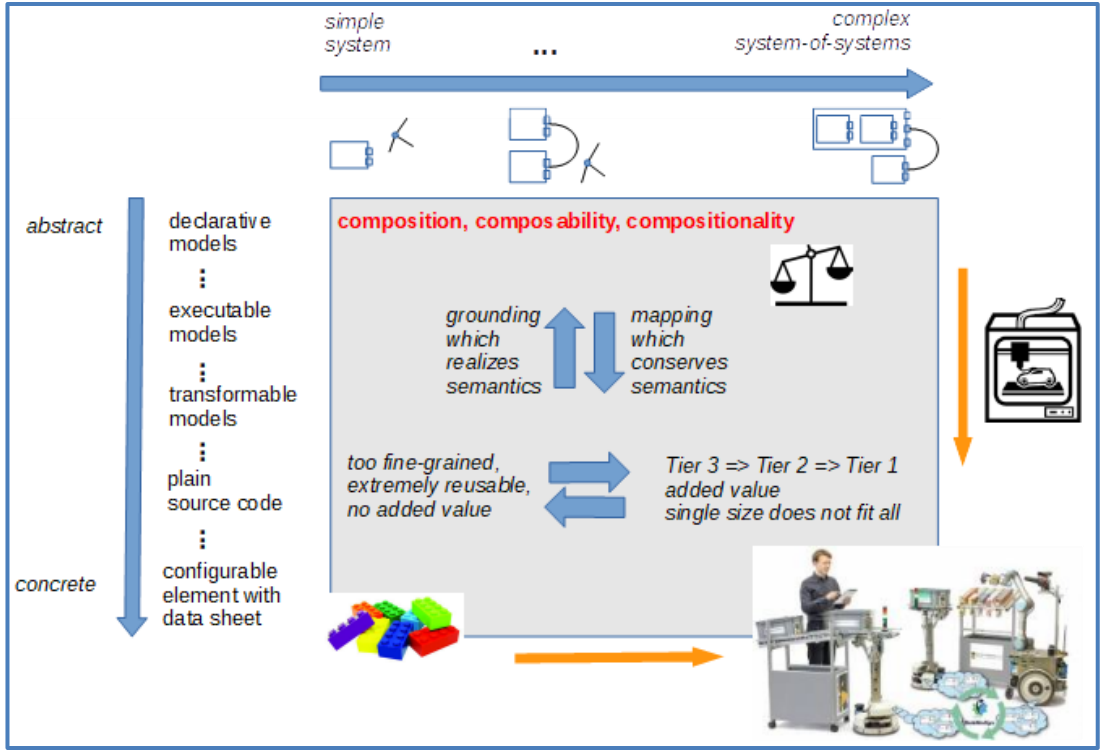
https://www.uni-ulm.de/in/koop-promotionskolleg-cognitive-computing-in-socio-technical-systems.html

Splitting Apart, Putting Together

Separation of Concerns
- computation
- communication
- coordination
- configuration

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

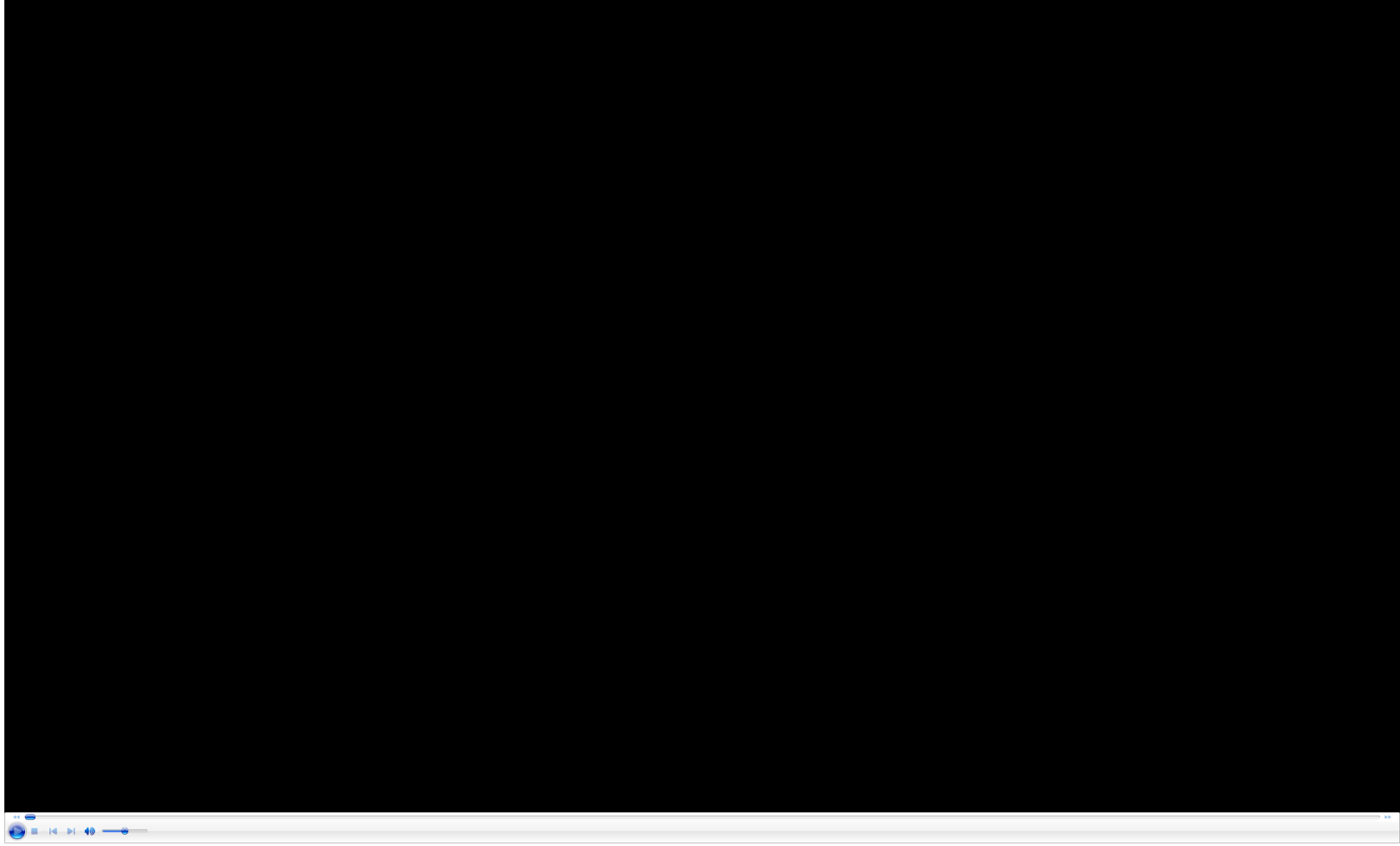
splitting apart… *…putting together*



- **composability** is the ability to combine and recombine as-is building blocks into different systems for different purposes. It requires that properties of sub-systems are invariant („remain satisfied“) under composition.
- **splittability** is the „inverse“ relationship of composability.
- **compositionality** requires that the behavior of a system is predictable from its sub-systems and that of the composition „glue“.
- **system composition (activity):** the activity of putting together a set of existing building blocks to match system needs with a focus on flexible (re-)combination.
- **system integration (activity)**: the activity that requires effort to combine components, requiring modifications or additional actions to make them work with others.

Separation of Roles!

RobMoSys



Video „What is the aim of RobMoSys?“ https://youtu.be/8GUSDTNoGRM

Ecosystem, Network Markets, Platform Approach

What is a platform?
- a platform offers your service without compensating you
- they go between you and your customer
- they take existing business relationships and squeeze in between you and your customer

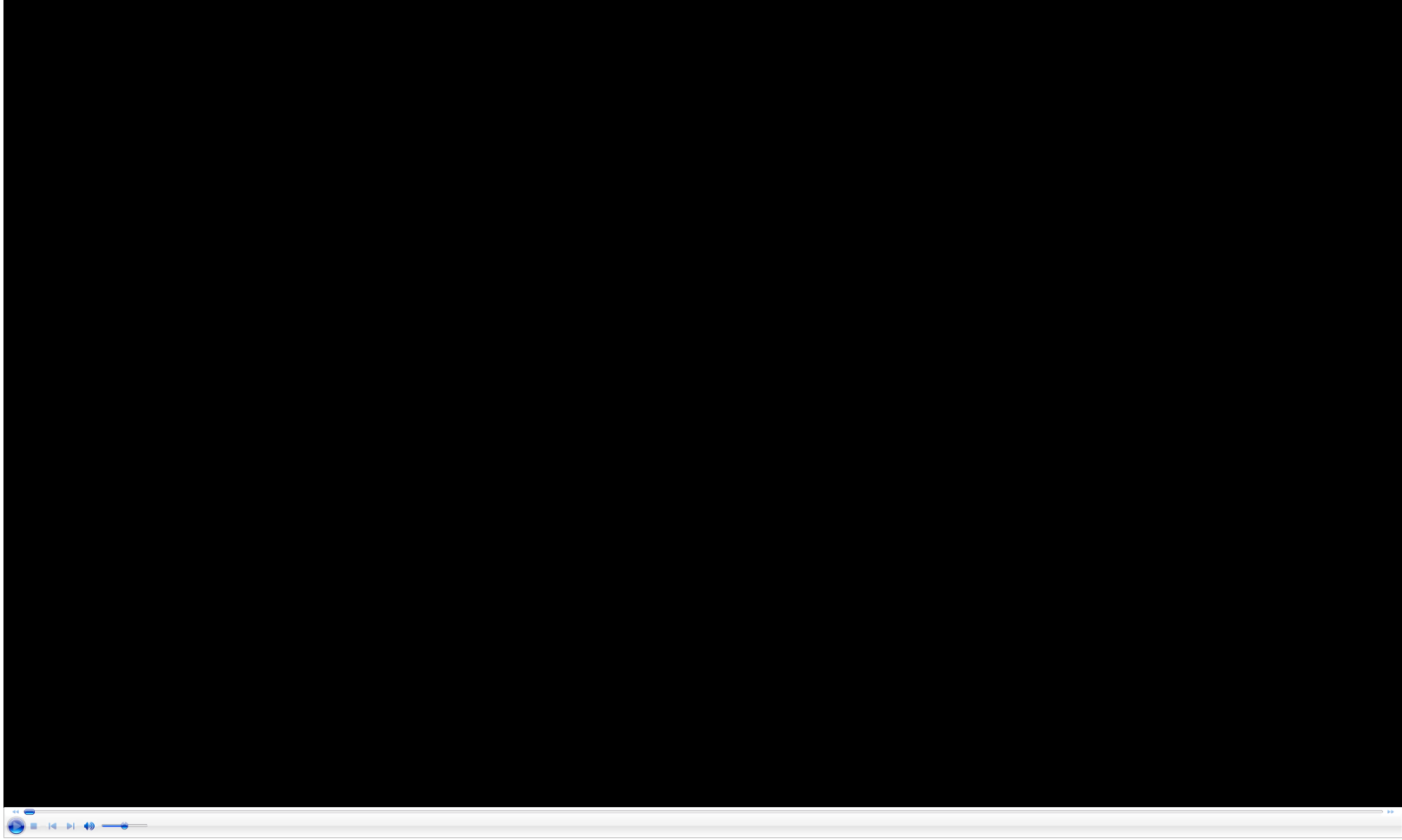
Example: Transportation (the same for music platforms, booking platforms, …)
- so far:
  - you (customer) paid taxi driver directly (service provider)
- now:
  - platform does not deliver transportation
  - you now pay platform and platform pays the taxi driver
  - you have a one-stop shop for transportation without extra costs
  - the taxi driver gets less per transport but hopes for better occupancy

In which markets does a platform approach work and why?
- it works in network markets because they have a different value proposition compared to „old markets“

Example “old market” and value proposition
- there is a glass of water
- one person drinks it with benefit
  and the others do not have a benefit, when the water is away
- the water being away is for the **benefit of others** either **neutral** (don’t care)
  or is a **disadvantage** (you suffer from not having it)

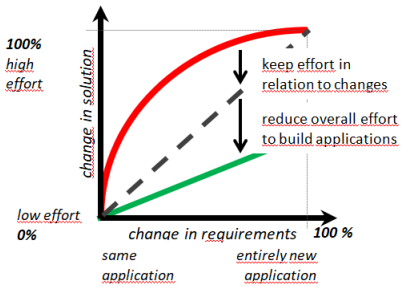
Example “network markets” (here, it is completely different with respect to value proposition!)
- A and B each have a telephone
  - A can call B (one participant), B can call A (one participant)
- now C has a telephone as well
  - A can now call B, C (two participants: +100%),
    B can call A, C (two participants: +100%),
    C can call A,B (from 0% to 200%)
- now D has a telephone as well …
  - A, B, C, D **all** have additional value
- **that market is n! (n faculty) value proposition, which is a steep rise of the value proposition where all benefit with each new participant**

Source: World Economic Forum/Accenture analysis

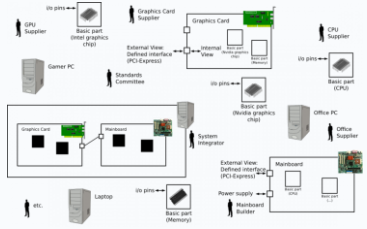


Projekt-Video „SeRoNet“ https://www.seronet-projekt.de/files/inhalte/Videos_fuer_Artikel/SeRoNet_v5.1_1080.mp4

Ecosystem, Network Markets, Platform Approach

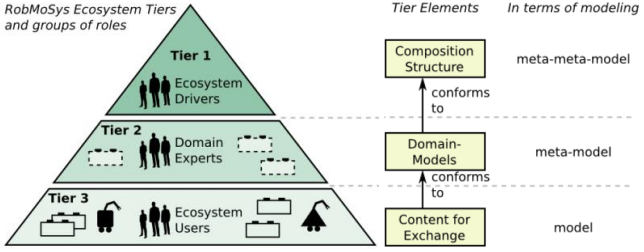


- All can be ***blocks*** with hierarchy (containment, collection).
- Blocks define structure where ***ports*** link inner parts of a block with the outer view on the block.
- Ports are linked via ***connectors***.
- Blocks come with ***data sheets***



- building blocks with data sheets (outer view on block)
- different stakeholders in different roles
- composition instead of integration

- Composition is about the **management** of the **interfaces** between different **roles** (participants in an ecosystem) in an efficient and systematic way.
- Composition is about guiding the roles via **superordinate composition-structures**.
- Composition is about explicating and managing **properties**.
- Composition is about the right **levels of abstraction and views** for roles.



Freedom of Choice

- Not a universal positive
- High price to pay since there is **no guidance** with respect to ensuring composability and system level conformance

Freedom from Choice

- Not a universal negative
- Structures that **ensure composability** restrict freedom of choice to achieve system level conformance

- Structures enable collaboration: **organization by structure** rather than by "management"
- Structure and t**ooling** go hand-in-hand
- Tooling enables **access** to structures to benefit from them

Which patterns and structures form the *Sweet Spot* between *Freedom of Choice* and *Freedom from Choice*?

Guidance for separation of concerns by superordinate objectives like the need for separation of roles and the need for composition

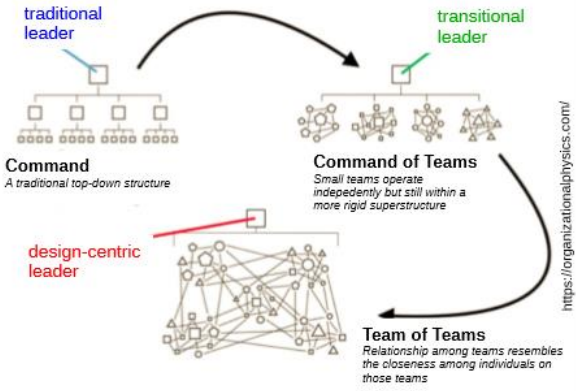
Support as much freedom as possible while still ensuring composability despite separation of roles

computation
communication
coordination
configuration

achieve separation of roles
support composition

Hierarchy and Heterarchy

Agility, Flexibility, Adaptability, …



Fachtagung Web 2.0: Technische Gestaltung, Folgerungen für die Prozesse und Interaktion durch den Menschen

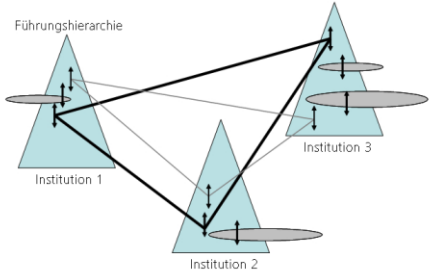
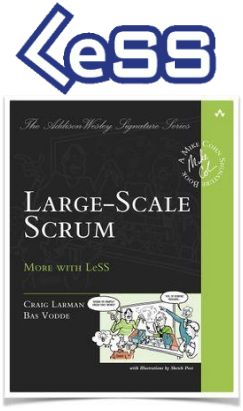
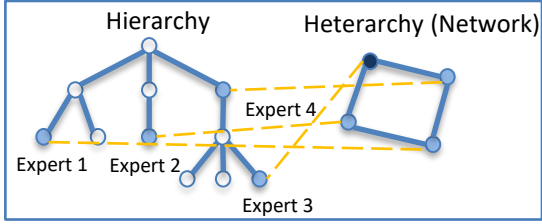


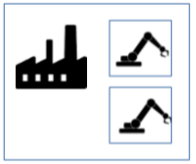
Abbildung 1: Kommunikation der Netzteilnehmer in Hierarchie und Heterarchie

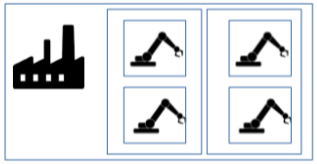


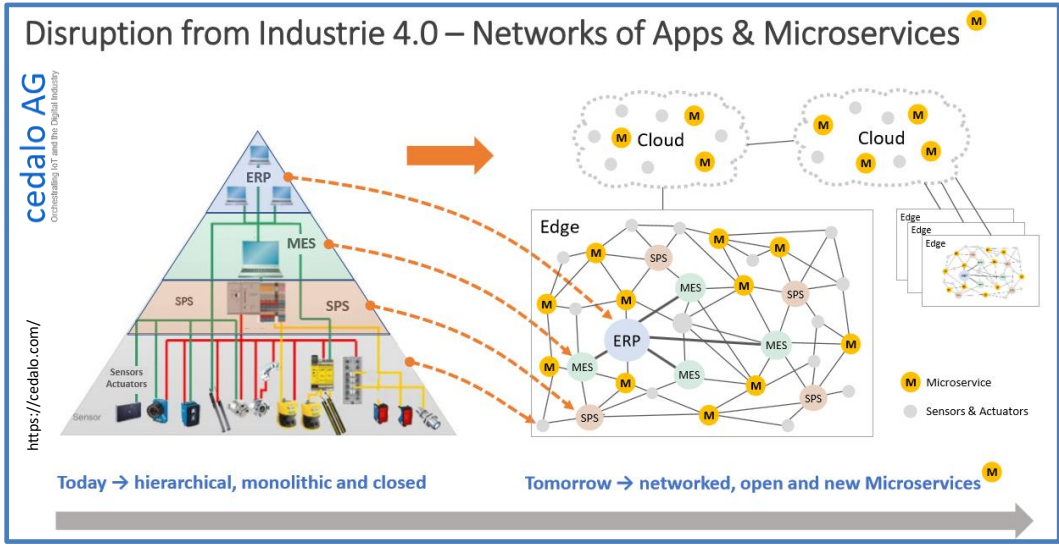
*Achieve **adequate** results with respect to quality and resource usage:*

- *resource budgets, resource shares*
- *local sync, global async*
- *nesting versus flattening*
- *extra-functional properties*

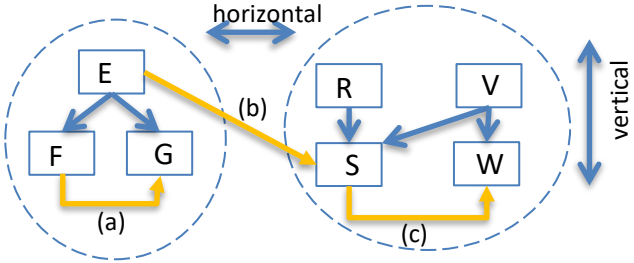








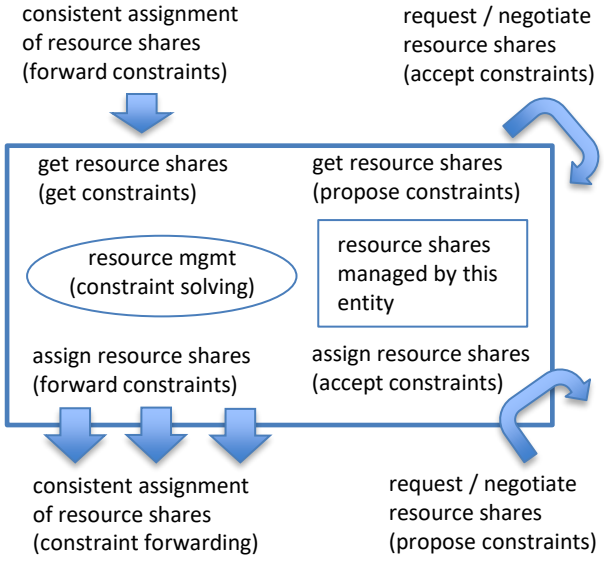
Horizontal and Vertical Interaction in Robotics



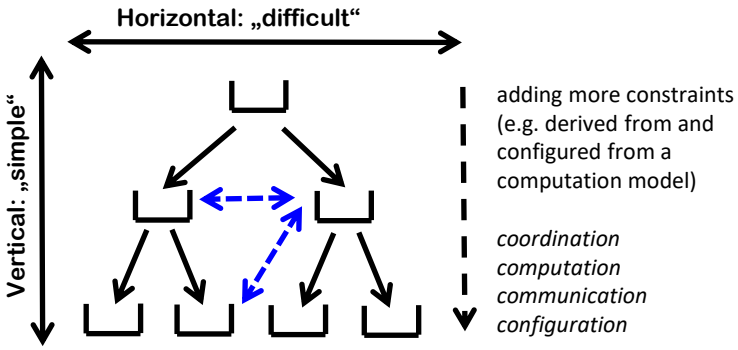
(a) Consistent because of either vertical coordination by E or horizontal coordination F asking G
(b) Consistent because of either vertical coordination above EFG and RSVW or just horizontal coordination E asking S
(c) Consistent because of either vertical coordination by RV for SW or S asking W
(d)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

Dynamically changing control hierarchy but with always a consistent and clear responsibility for resource shares
- best effort, contract based with guarantees, orthogonal assignment, …
- design time, deployment time, run time, …

Distributed and linked Models:
- Model of factory
- Model of production cell
- Model of robot resources and skills in the robot knowledge base
- Model of object in object recognition skill
- …



Horizontal and Vertical Interaction in I4.0



adding more constraints (e.g. derived from and configured from a computation model)

coordination
computation
communication
configuration

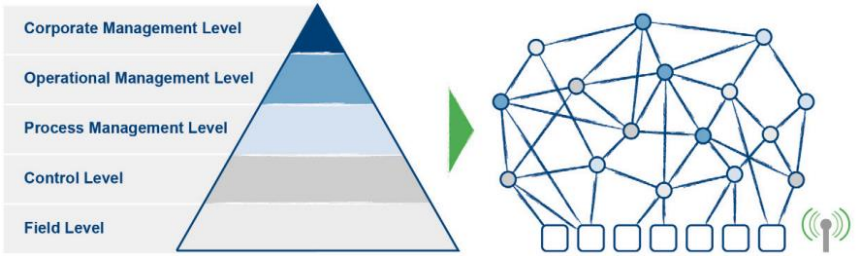
Horizontal and vertical interaction:

- technical aspect
- process aspect
- organizational aspect
- decision making aspect
- …

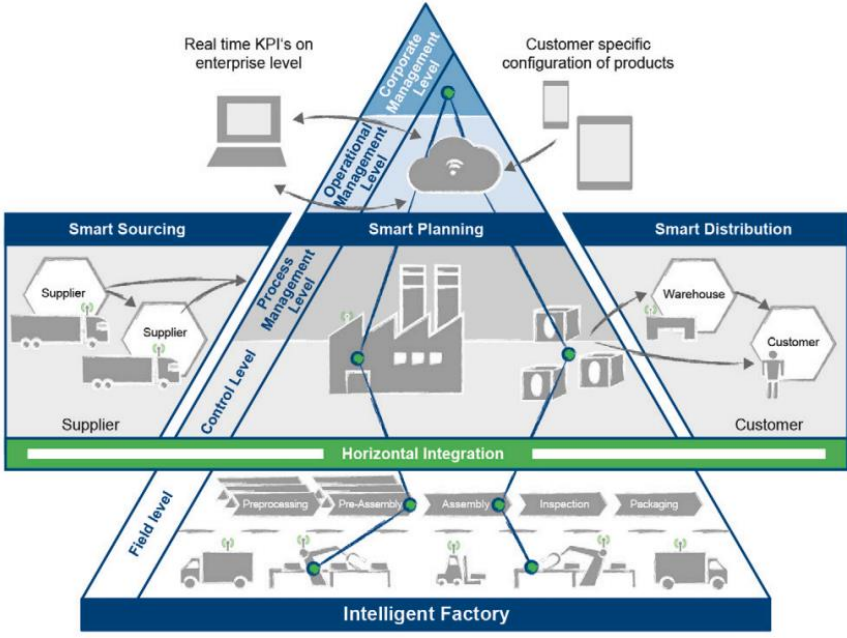
Challenge:

- resource management
- proper granularity and size of „entities“

- *constraints networks (dependency graphs)*
- *structural granularity along blocks, ports, connectors driven by composition and by separation of roles*



Verein Deutscher Ingenieure e.V.: Thesen und Handlungsfelder Cyber-Physical Systems: Chancen und Nutzen aus Sicht der Automation, April 2013



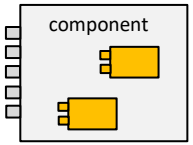
https://www.unity.at/fileadmin/Insights/OPPORTUNITY/OPPORTUNITY_Seize_OPPORTUNITY_Industrie_4.0.pdf

Composition in Robotics

function development

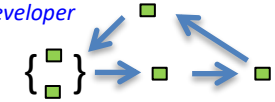
service definitions by domain experts

component supplier

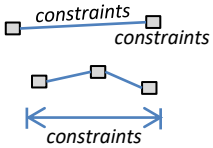


communication
coordination
configuration
computation

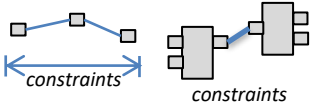
behavior developer



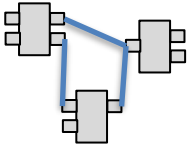
system architect

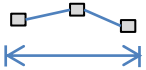


safety engineer



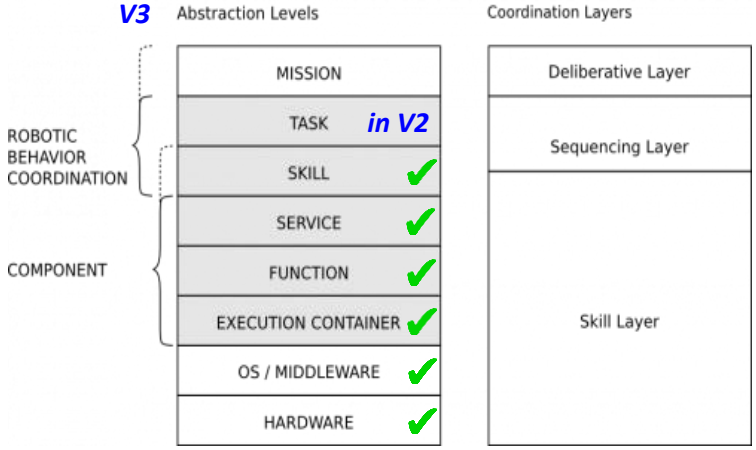
system builder



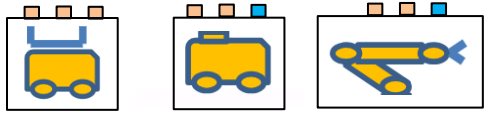


dependency graphs,
data cconsistenvy constraints:

- *cause-effect-chains*
- *data aging constraints*
- *data accurarcy constraints*
- *scheduling constraints*
- *etc.*



composition operators within the different abstractions levels and for their mutual linkage (structural, behavioral, resource shares, …)



Tier 2:
service definition

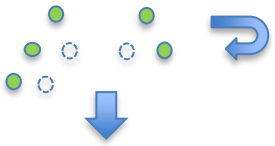
Tier 3:
component supplier

Tier 3:
system architect
system builder

exploit variation points
trade-off and manage consistent bindings
achieve adeqaute solutions

design time deployment time run time …

Horizontal and Vertical Interaction in Robotics



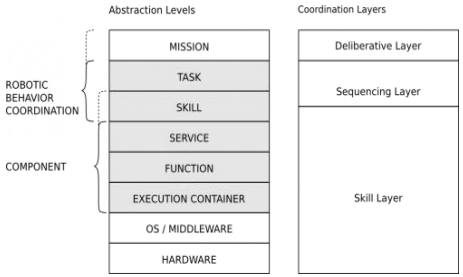
- *component operating mode*
- *assignment of resource shares*
- *constraints forwarding*

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

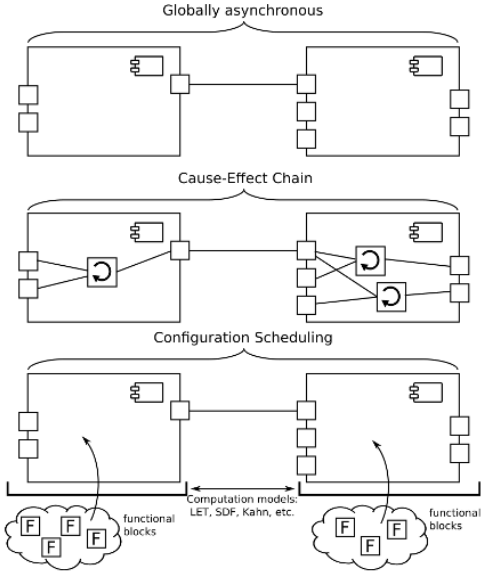
dynamic behavior tree
- parallel (one-of, all), sequential, expansion, …

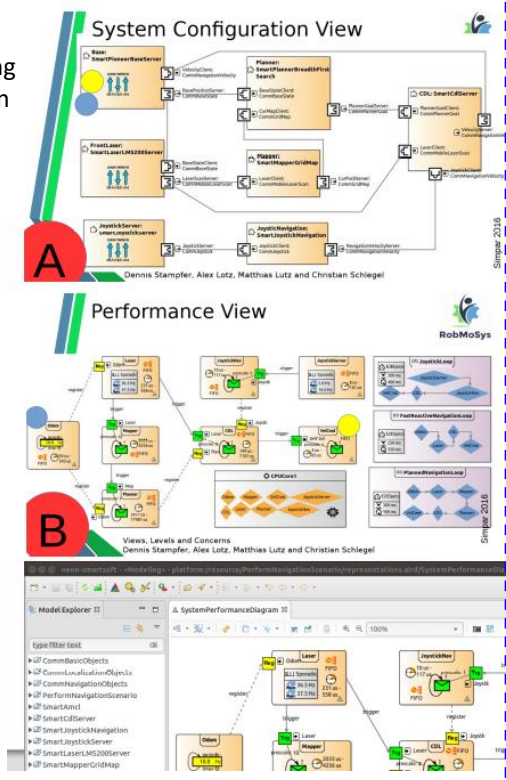
horizontal
- e.g. resource share reservation in knowledge base

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

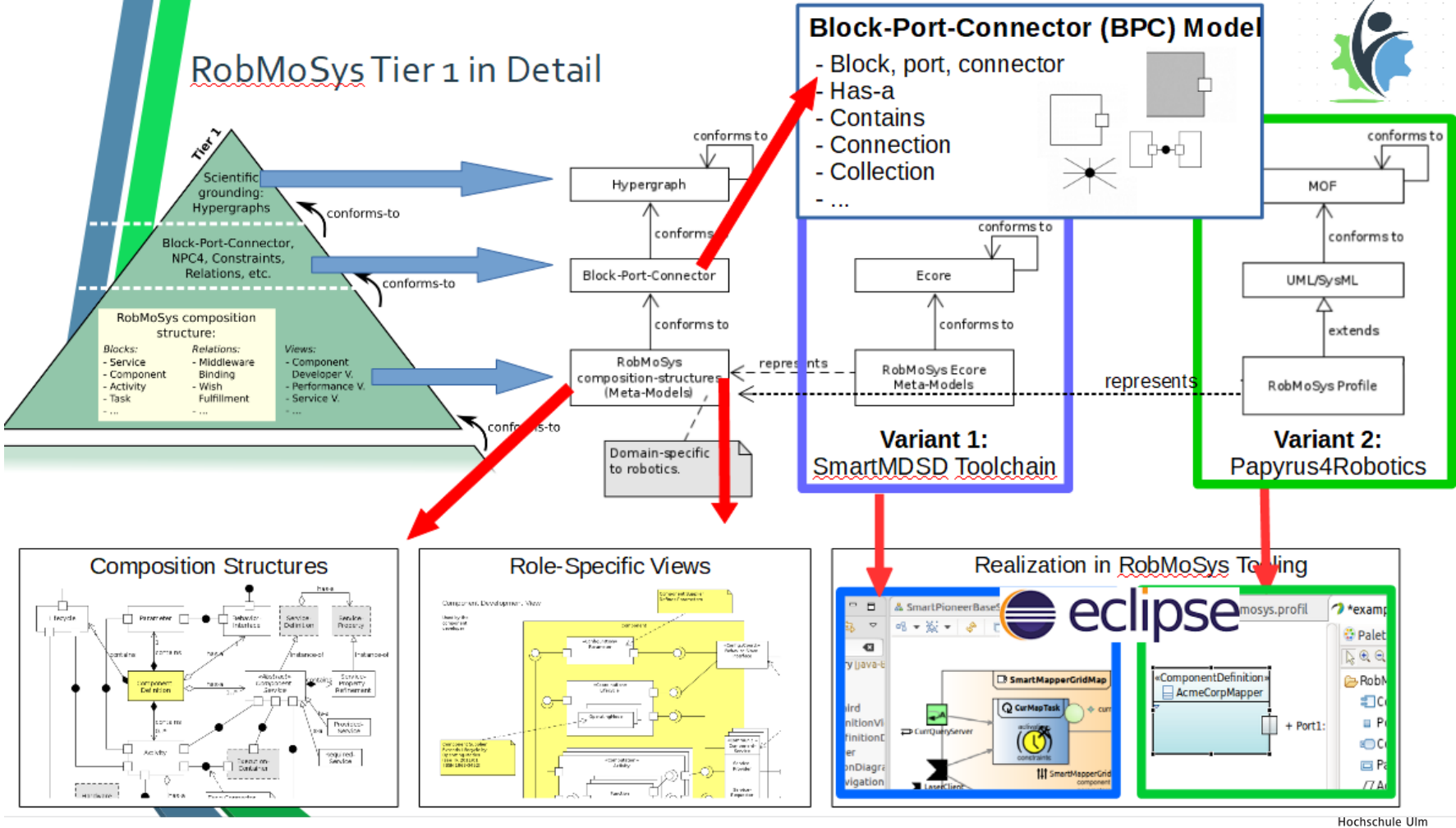


computation model configuration
- data consistency / data sync / data quality / data aging
  - inject register / trigger semantic for communication
  - inject port trigger / timed trigger for computation
  - inject scheduling constraints





RobMoSys Tier 1 in Detail

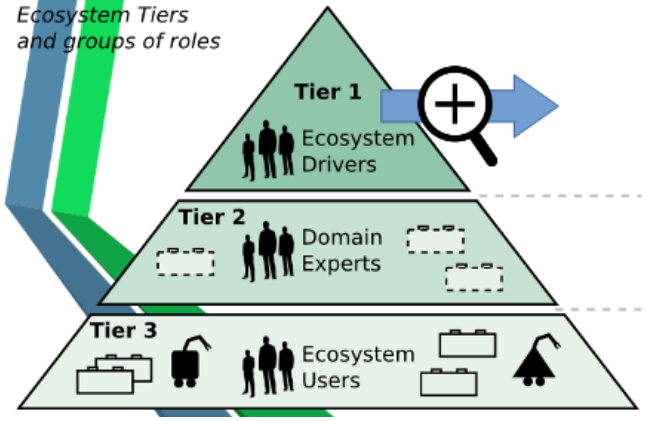


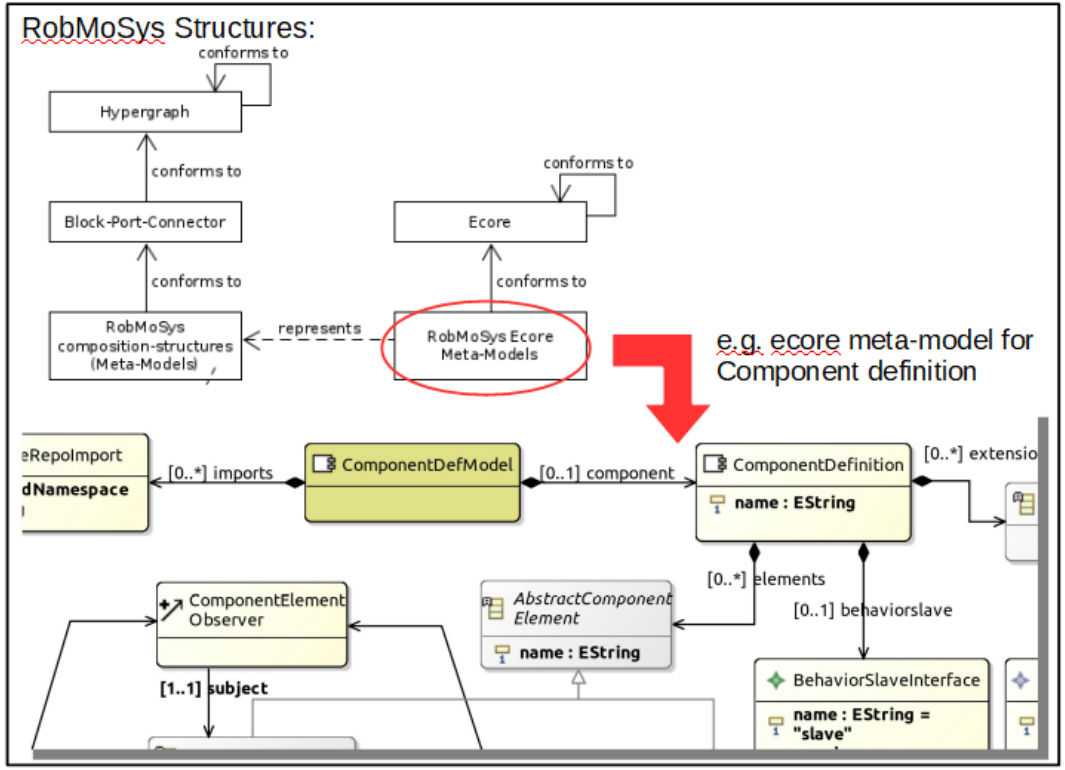
SmartMDSD Toolchain Walkthrough
Support for RobMoSys Tier 1

RobMoSys

The SmartMDSD Toolchain implements the RobMoSys composition structures using Ecore. RobMoSys **structures become accessible** to Tier 2 and Tier 3 users!

Example: Component Definition Meta-Model





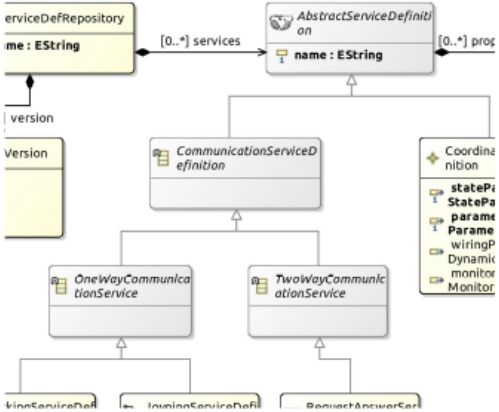
SmartMDSD Toolchain Walkthrough
Support for RobMoSys Tier 2

RobMoSys

The SmartMDSD Toolchain supports in modeling domain structures (**domain models)** according to the RobMoSys composition structures.

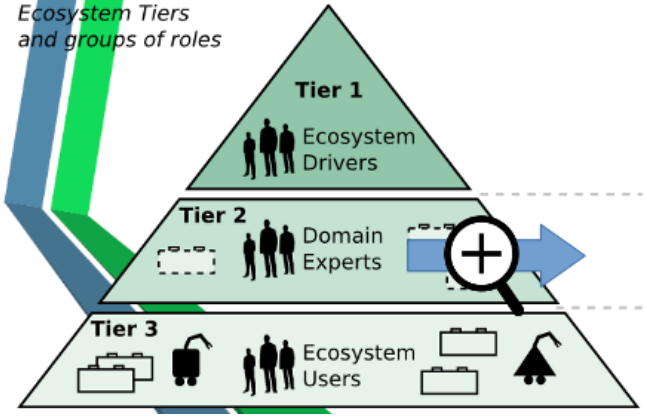
Example: **Service Definitions**

RobMoSys Structures:

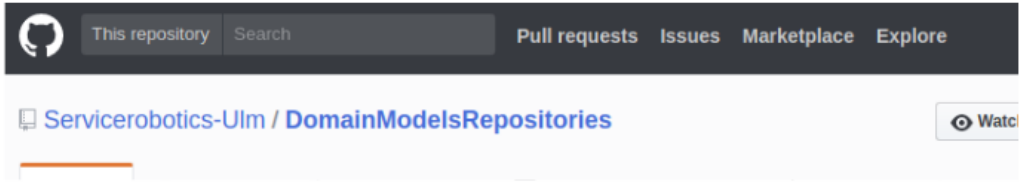


Role-Specific View:

Ecosystem Tiers and groups of roles



Available content: Domain Models
See https://github.com/Servicerobotics-Ulm/DomainModelsRepositories



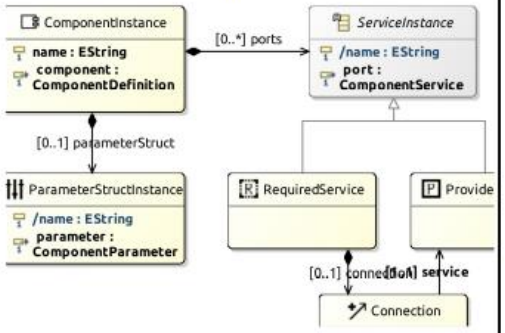
SmartMDSD Toolchain Walkthrough
Support for RobMoSys Tier 3

RobMoSys

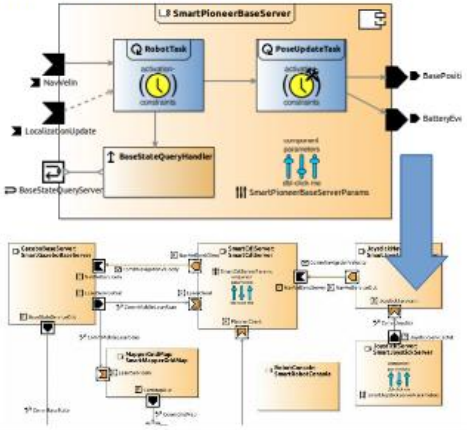
The SmartMDSD Toolchain supports in **developing components** and in **composing** previously developed components to **systems**.

Example: TiaGO Navigation

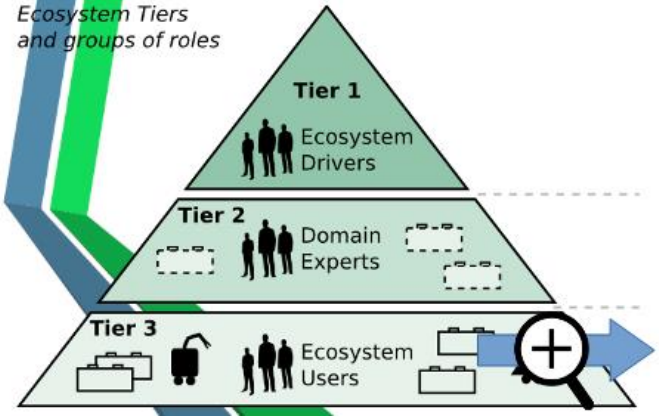
RobMoSys Structures:
e.g. Component Definition
and System Component Architecture



Role-Specific View:



Ecosystem Tiers and groups of roles

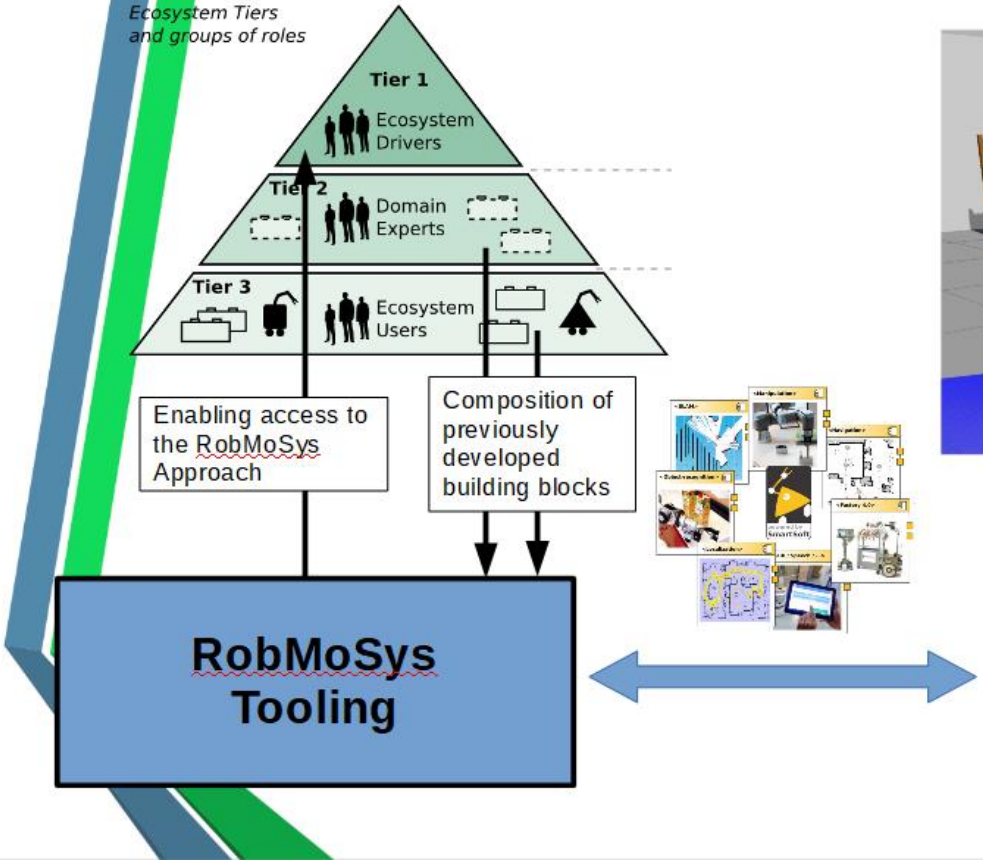


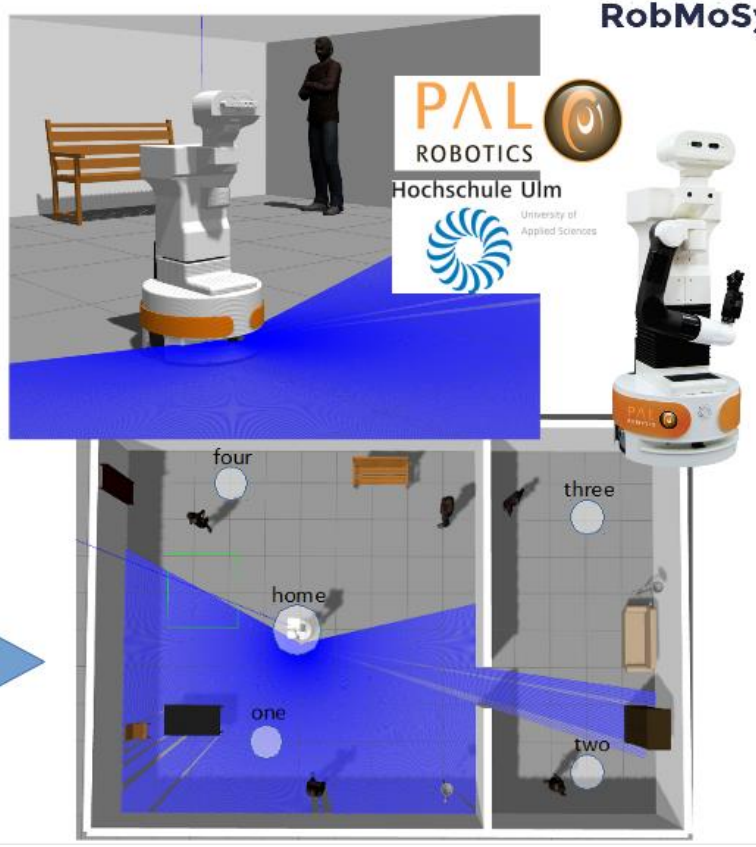
Available content: previously developed/modeled building blocks:
See https://robmosys.eu/wiki/baseline:components:smartsoft



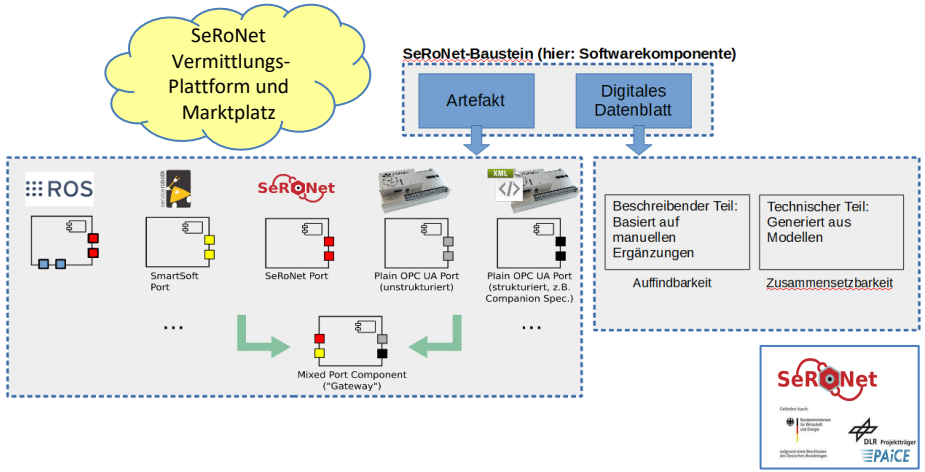
Available Software Baseline: The Gazebo/TIAGo/SmartSoft Scenario
https://robmosys.eu/wiki/baseline:environment_tools:smartsoft:start

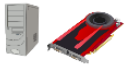
RobMoSys





Composition in Robotics





Examples of the PC Analogy

e.g. Semiconductor standards, architecture, USB, PCIe, mode ethernet, etc.

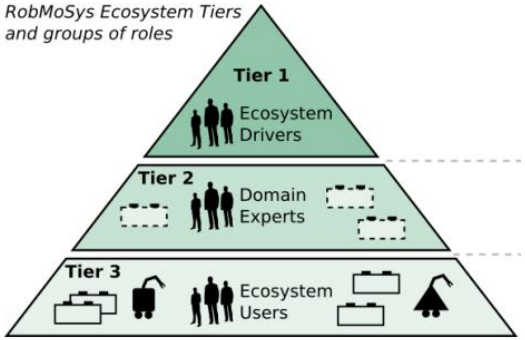
e.g. laptop PC, desktop PC, inc ATX, ITX, Mini-ITX, VGA, HDM CPU socket, GPU socket, USB storage, etc.

e.g. graphics card, CPU, TPM, Memory, power supply, USB SSD Hard disc, USB stick, etc.

RobMoSys - Composable Models and Software for Robotics Systems

RobMoSys

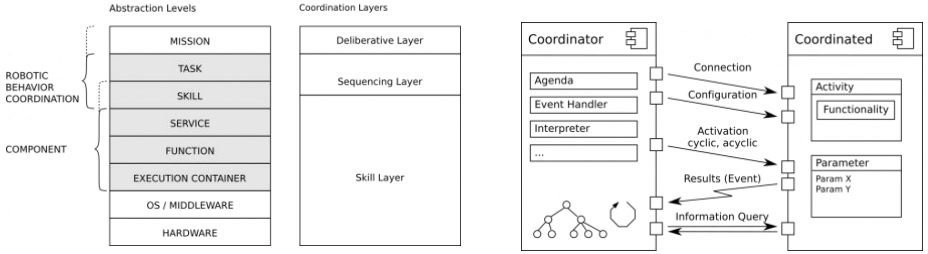
RobMoSys Ecosystem Tiers and groups of roles

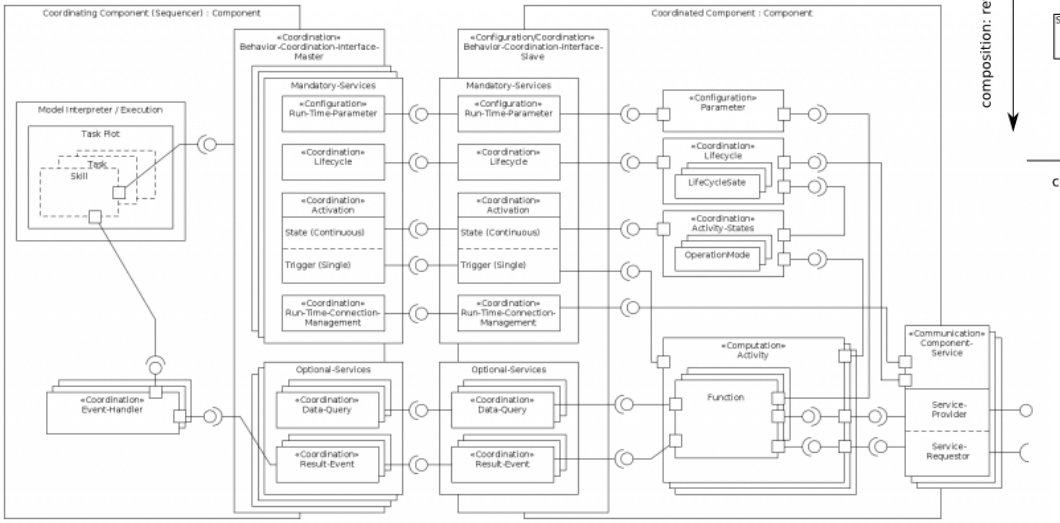


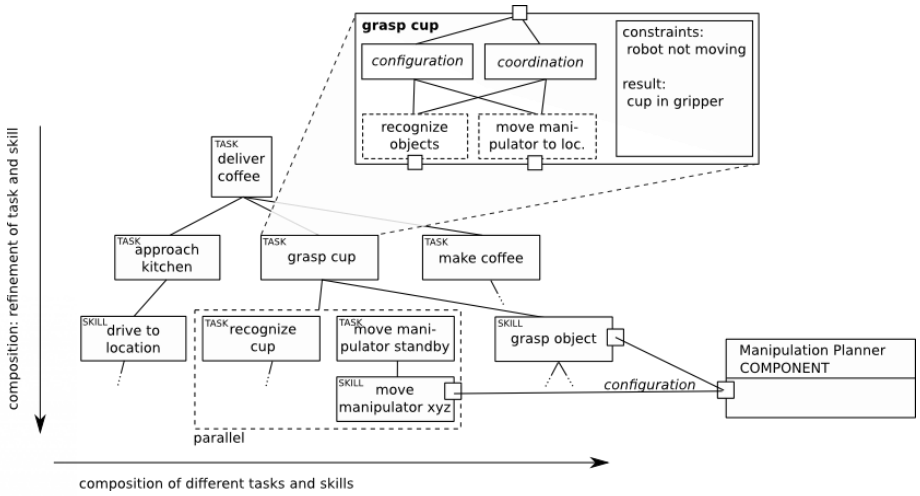
Tier Elements	*In terms of modeling*	RobMoSys	*OPC UA World* OPC UA
Composition Structure	meta-meta-model	e.g. service-oriented software component model, robotics task models, communication patterns	OPC UA *Standard* — e.g. information models, variables, methods
conforms to			
Domain-Models	meta-model	e.g. vision, flexible navigation stack, motion-perception-world model stack, manipulation.	OPC UA *Companion Specifications* — e.g. vision, robotics, devices, kitchen equipment
conforms to			
Content for Exchange	model	e.g. vision, flexible navigation stack, motion-perception-world model stack, manipulation.	OPC UA *Building Blocks and Systems* (Clients/Servers) — e.g. SPS, coffee machine, sensors, robots

Examples of System Composition

Examples of System Composition: Task Coordination







Management of resource shares, execution context constraints, configuration constraints in horizontal and vertical task interactions

Examples of System Composition: Robot Fleet

Kommissionierung Servicerobotik Fachbodenregal

- Distributionslogistik, stückgenau, Pharmaartikel in Schachteln
- kombiniert Ware-zu-Kommissioniergasse und Person/Roboter-zu-Ware in Kommissioniergasse
- Fokus: Pickaufgaben

LogiRob

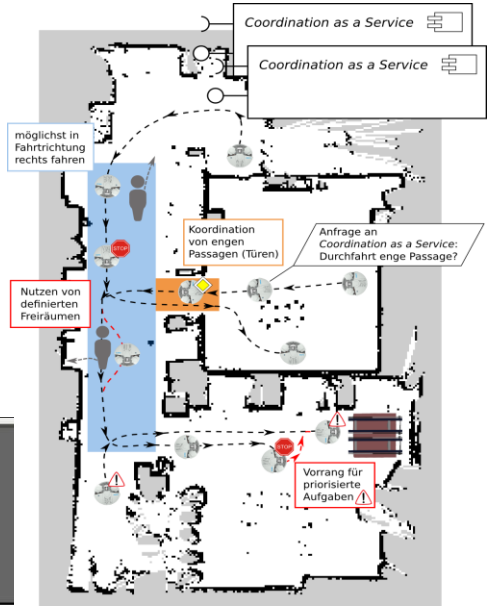
ZAFH Intralogistik

Hochschule Ulm

Robotics Equipment Corporation

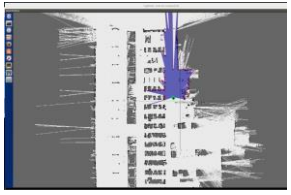
FESTO

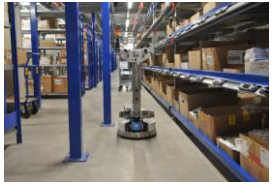
TRANSPHARM
competence.connection.courage



Kommissionierung Person-zu-Ware

- Distributionslogistik, sehr große Vielfalt einschließlich loser Artikel
- Roboter/Person-zu-Ware, Zone-Picking
- Fokus: gemischte Roboterflotte, mit Menschen geteilter Arbeitsraum, kollaborative Pick- und Transportaufgaben



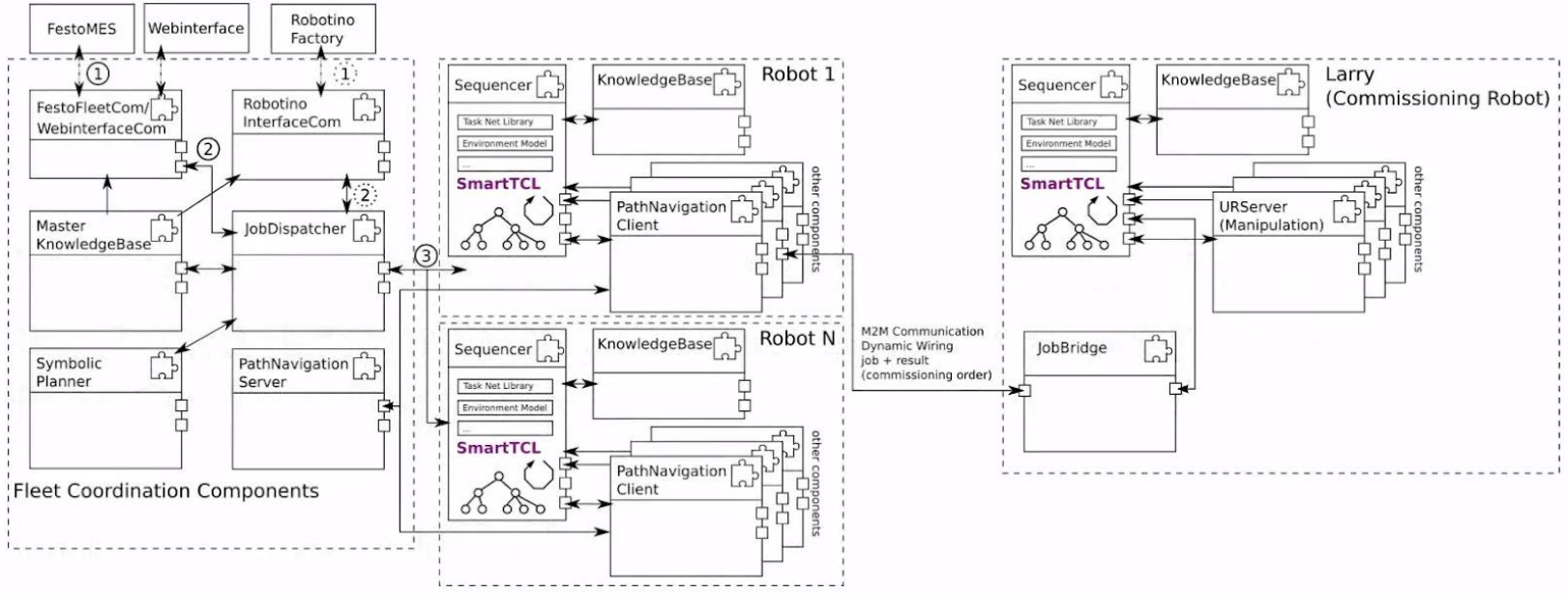


Examples of System Composition: Robot Fleet

Management-Components (zentral)

Transportroboter-Komponenten

Pickroboter-Komponenten

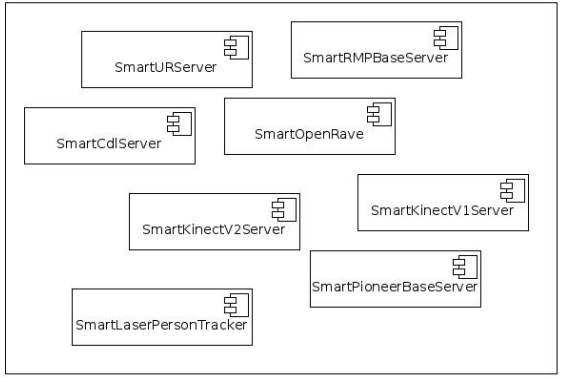


Examples of System Composition: Robot Fleet

Composition:

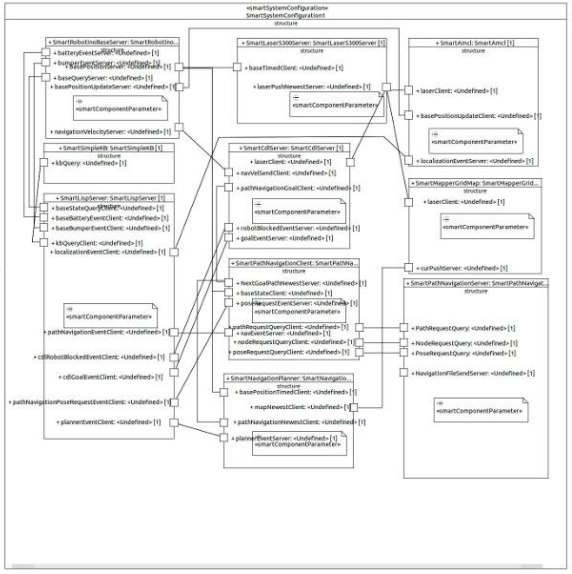
- Components to Robot System
- Skills to Robot Tasks and Fleet Tasks
- Robots to Fleet

Building-Block Repository

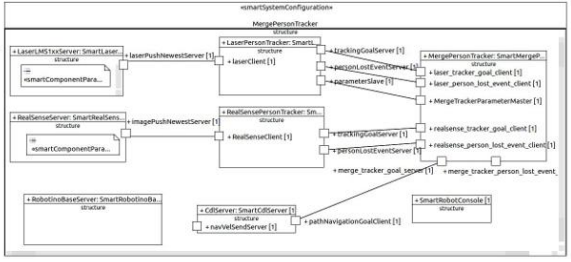


System Composition

System A



System B



Video Order Picking by Robot (https://youtu.be/cggCY-cvdJ8)

Video Person Following for Order Picking (https://youtu.be/r4mgPgyYISQ)

Examples of System Composition: Produktionslogistik

Pilot 3: Montage in der Produktion

klassische manuelle Montage mit unterschiedlich komplexen Teilprozessen im industriellen Umfeld sukzessive automatisieren und wandlungsfähiger gestalten.

Daimler TSS
Enabling Excellence

Hochschule Ulm

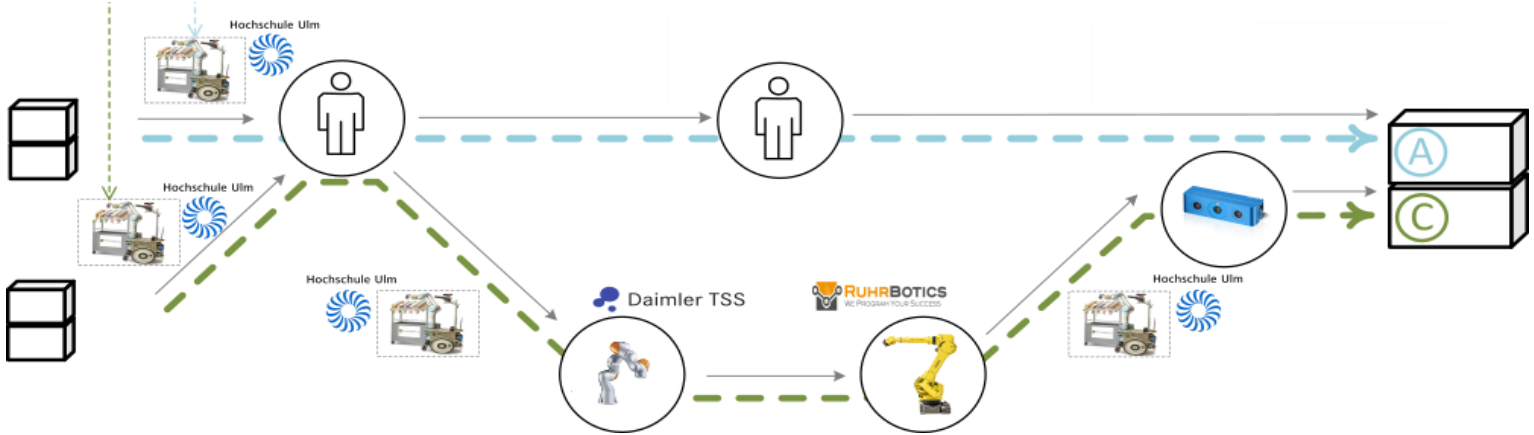
RUHRBOTICS

SeRoNet

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

DLR Projektträger

PAiCE



Horizontal and Vertical Composition

Determinism in compositions with respect to adequacy (costs, quality, etc. via resource share allocations and dependencies via constraints):

- robotics capabilities with explicated quality / resource relationships, i.e. know about adequateness: what do you get with what resources
- predictability and knowing about properties of system compositions
- explicitly address non-functional properties within a composition oriented approach
- keep assured properties like data processing order and thus data consistency, data quality, data accurarcy, data aging , response times etc. when adding, removing, replacing components or when moving them from one platform to another
- ensure consistent system mode transitions

Blocks, Ports, Connectors as structural elements

- bundle the four concerns (communication, coordination, computation, configuration) such that composability and separation of roles can be achieved
- be able to inject configurations into compositions of as-is building blocks via variation points
- digital data sheets for blocks with ports (task plots, capabilities, components, devices, etc.) which explicate variation points
- use constraint solving to exploit the offered variability for conformance to system level properties (either at design time, at deployment time, at run time, …)

Generate trust into systems

- by validation & verification, by simulation, by testing, by maturity levels along TRLs, …
- „not all possible combinations of activities are checked for safety but check that you can always reach a safe state“