RoSE 2022 Workshop

Dynamic Allocation of Service Robot Resources to an Order Picking Task Considering Functional and Non-Functional Properties

Timo Blender, Christian Schlegel

THU Technische Hochschule Ulm

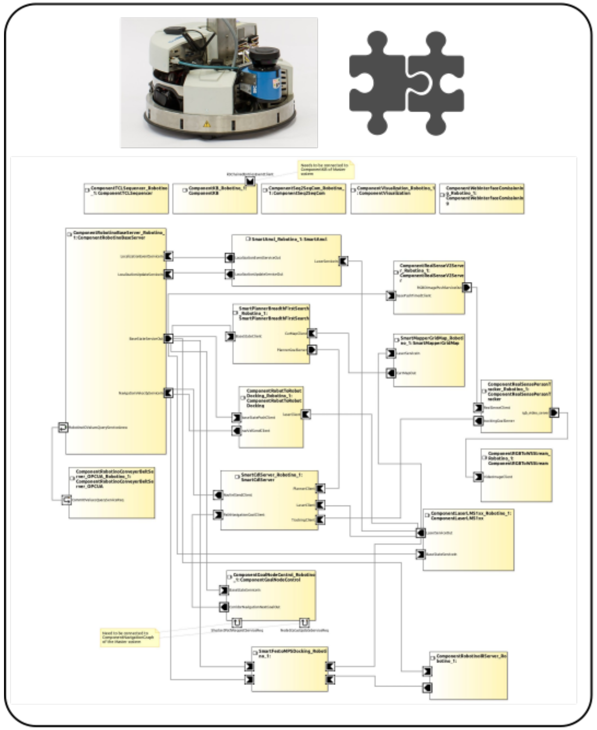
service robotik

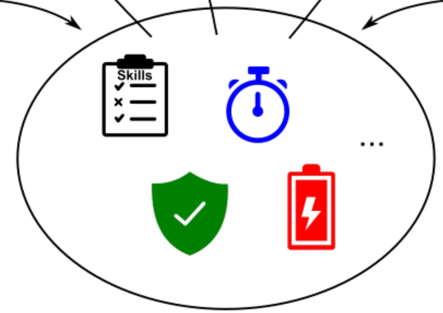
The Fundamental Problem: Flexible Machines and Varying Tasks

Capabilities *Non-Functional Properties*

Provided *Properties* Required

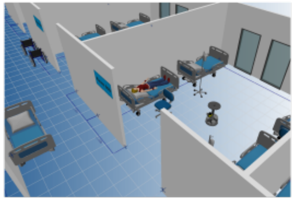
Possible Solutions





Possible Problems

Industry 4.0



Healthcare

Household

⋮

⋮

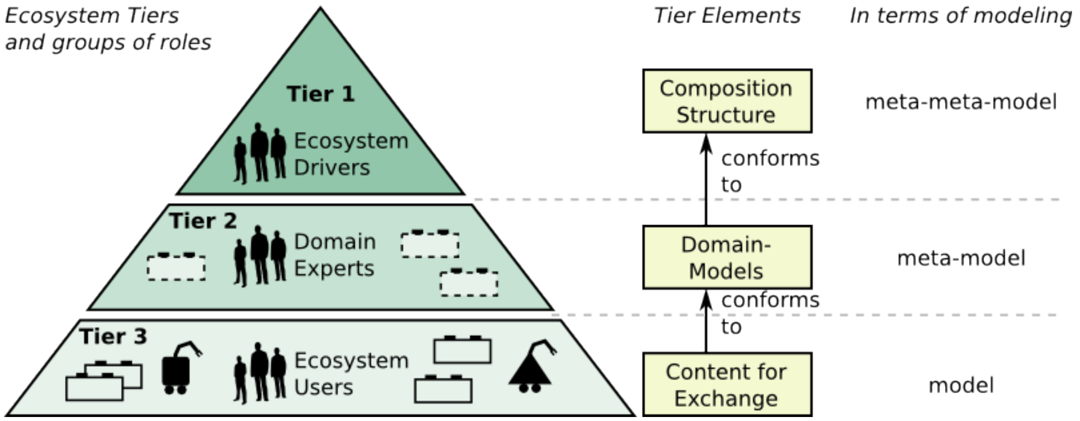
How can service robots decide based on its available software capabilities and variability whether and how to fulfill varying task requirements of end users in dynamic environments?

The Context: Mastering the Software Engineering Challenge in Robotics

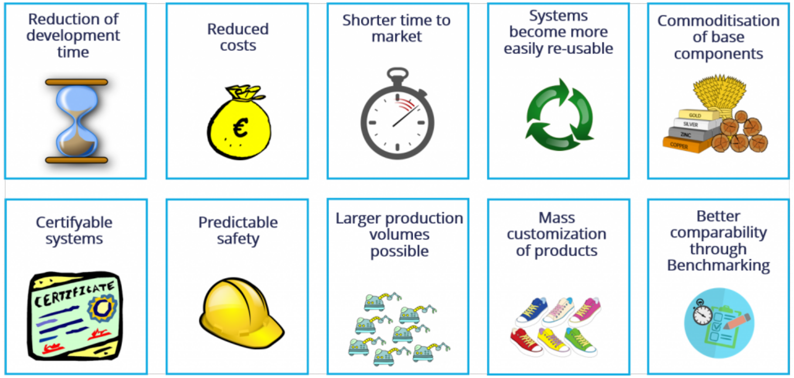
➢ Business ecosystem for robotics based on separation of roles and on composition

Step-change towards a European ecosystem for open and sustainable industry-grade software development for robotics

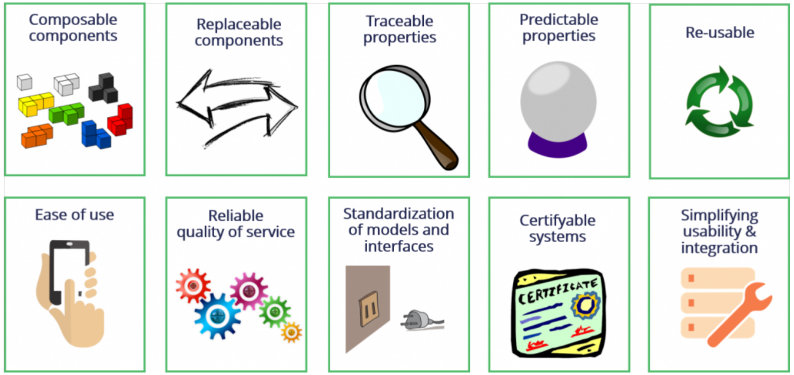
➢ **https://robmosys.eu/**



Reference: https://robmosys.eu/approach/

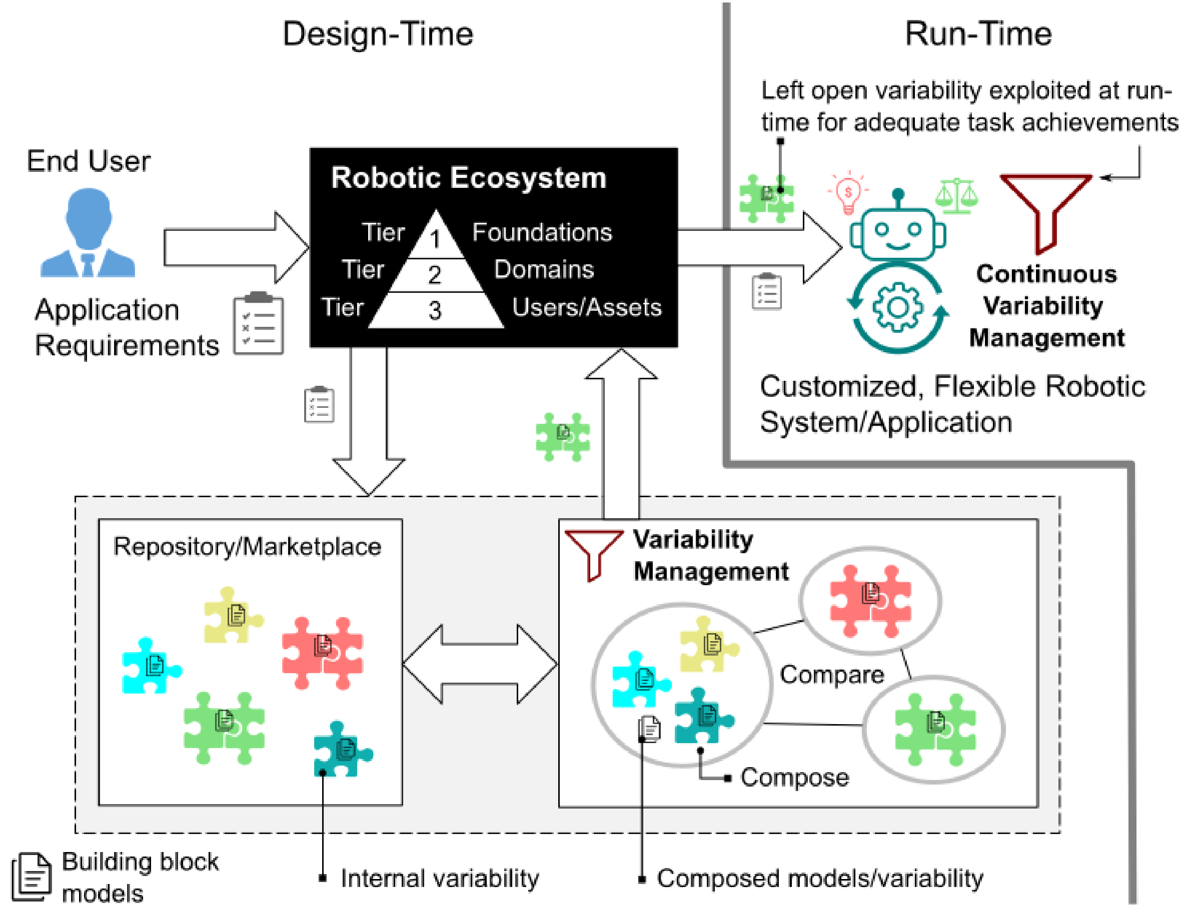


Reference: https://robmosys.eu/application/



Reference: https://robmosys.eu/application/

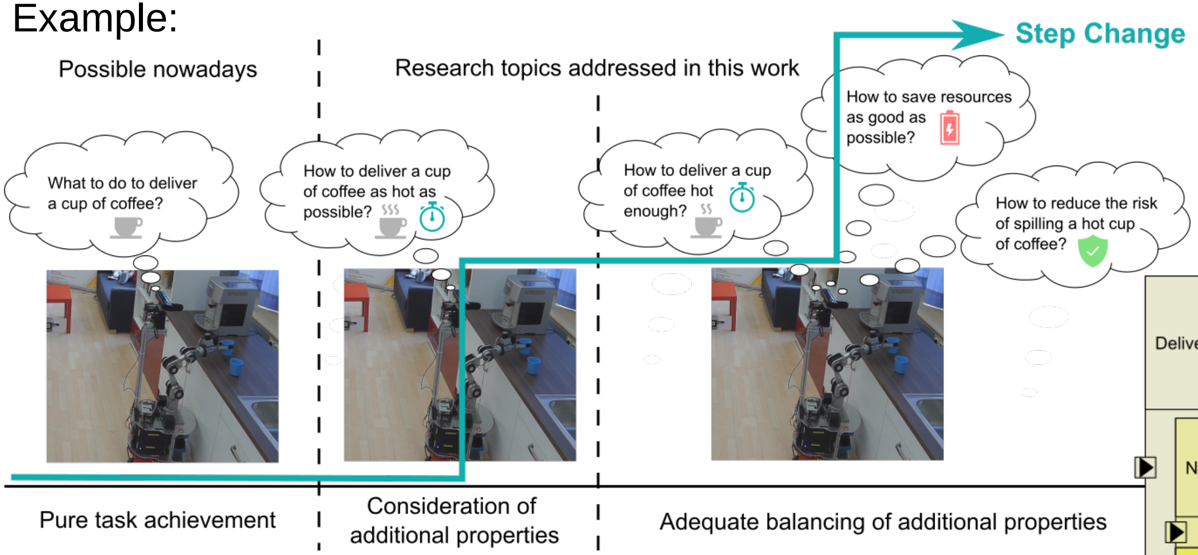
The Ideal: Synthesis in a Robotics Software Ecosystem (RSE)



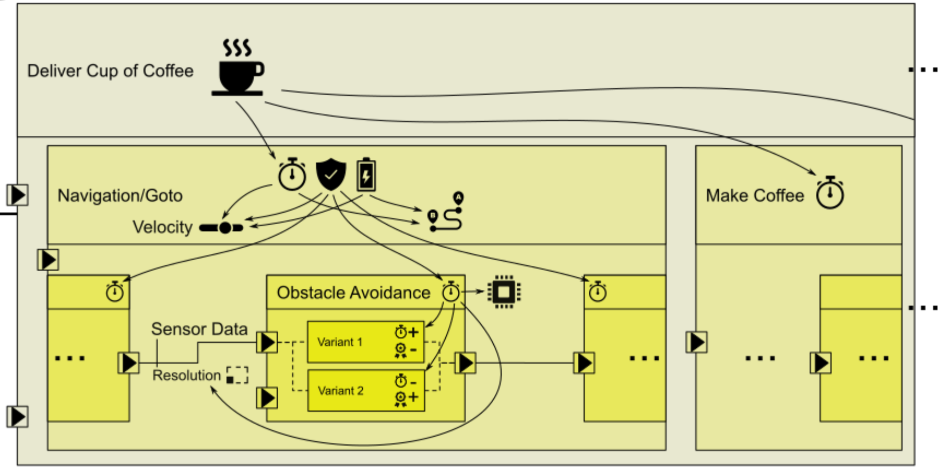
- Automatic composition and configuration of available (software) building blocks from a marketplace to a customized, flexible robotic application based on a formal requirement specification of an user
  - Enables deployment of requirement-related robotic applications
  - Significant effort reduction of the associated development and configuration process

The Challenge: Managing Variability

Example:

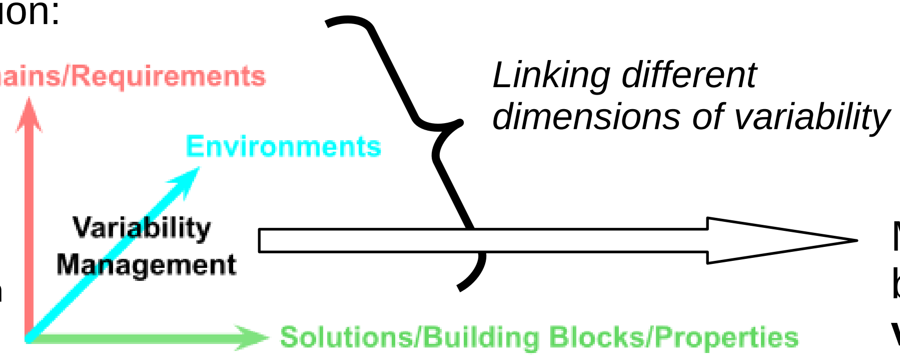


Service robot systems composed out of **different** building blocks with **different** configuration variants and resulting **different** properties, influenced additionally by **dynamic** environments should configure themselves according to **variable** user requirements



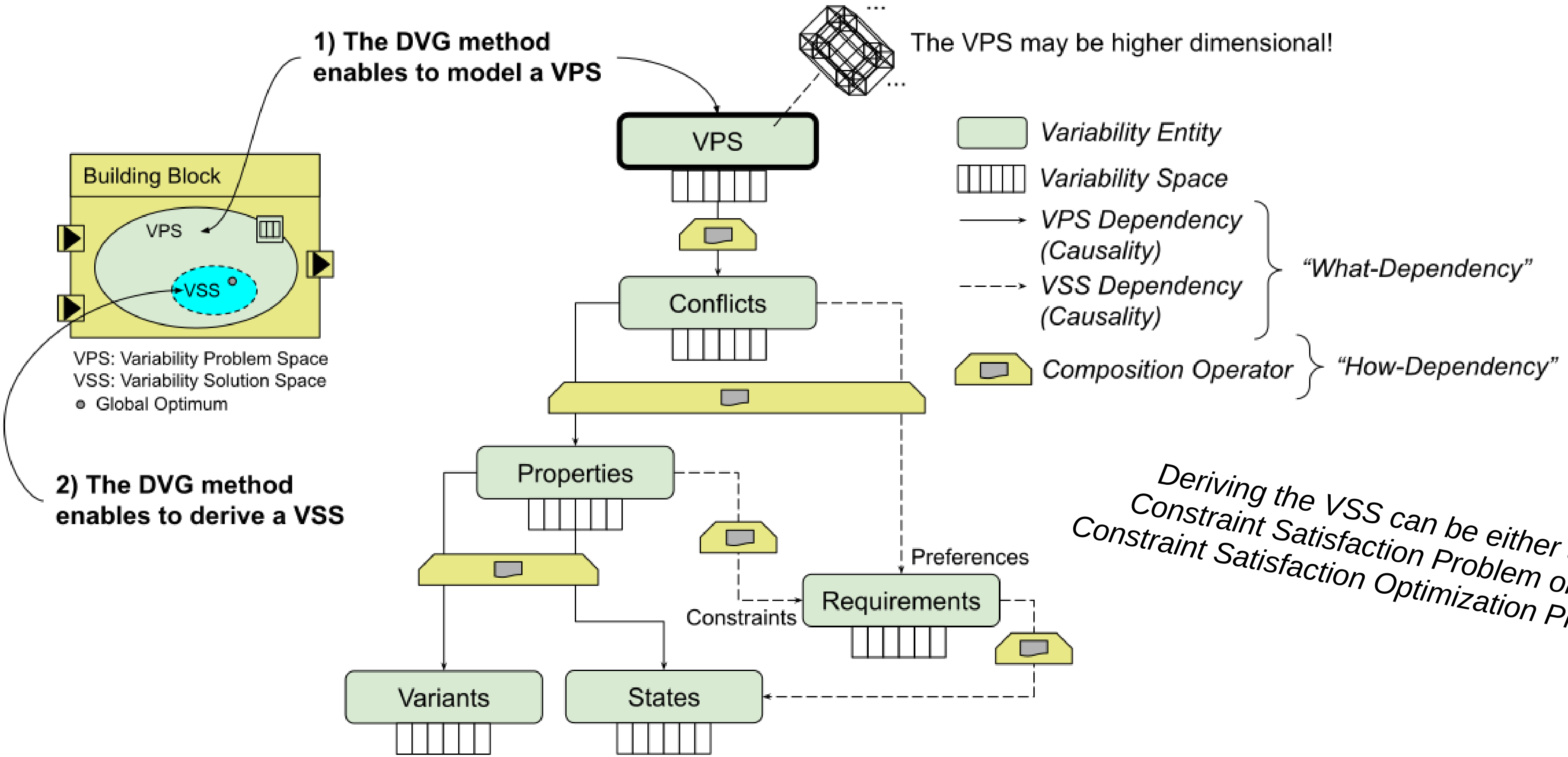
Generalization:

Several variability entities may exist in each dimension



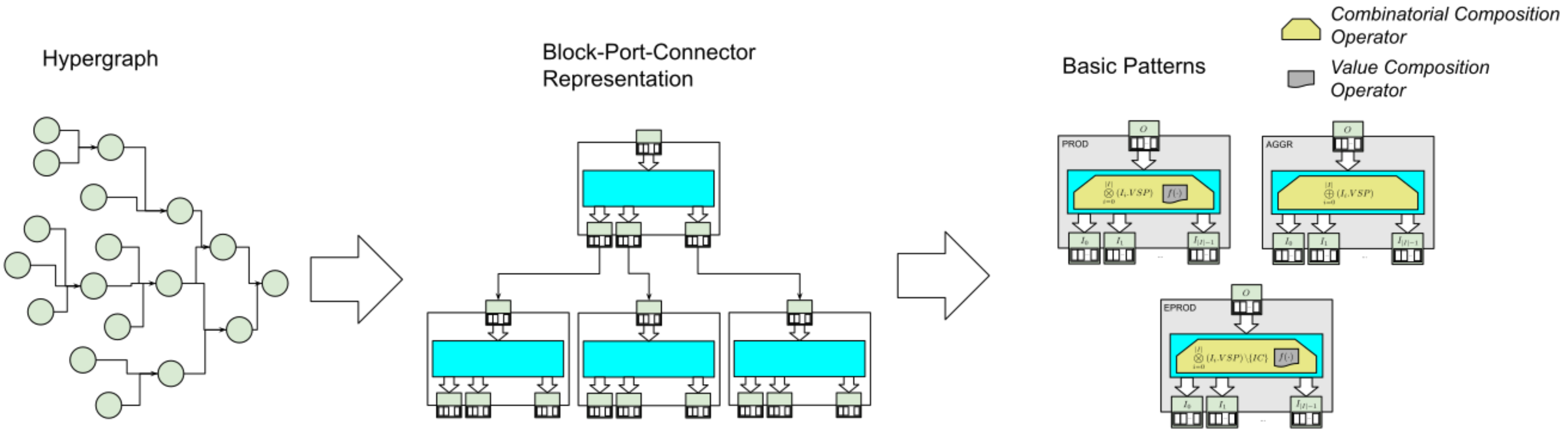
Modeling and resolving dependency relationships between **variability entities** and their **variability spaces** of different **building blocks**

The Approach: Dependency Variability Graphs (DVGs)



Deriving the VSS can be either a
Constraint Satisfaction Problem or a
Constraint Satisfaction Optimization Problem

DVGs: Basics

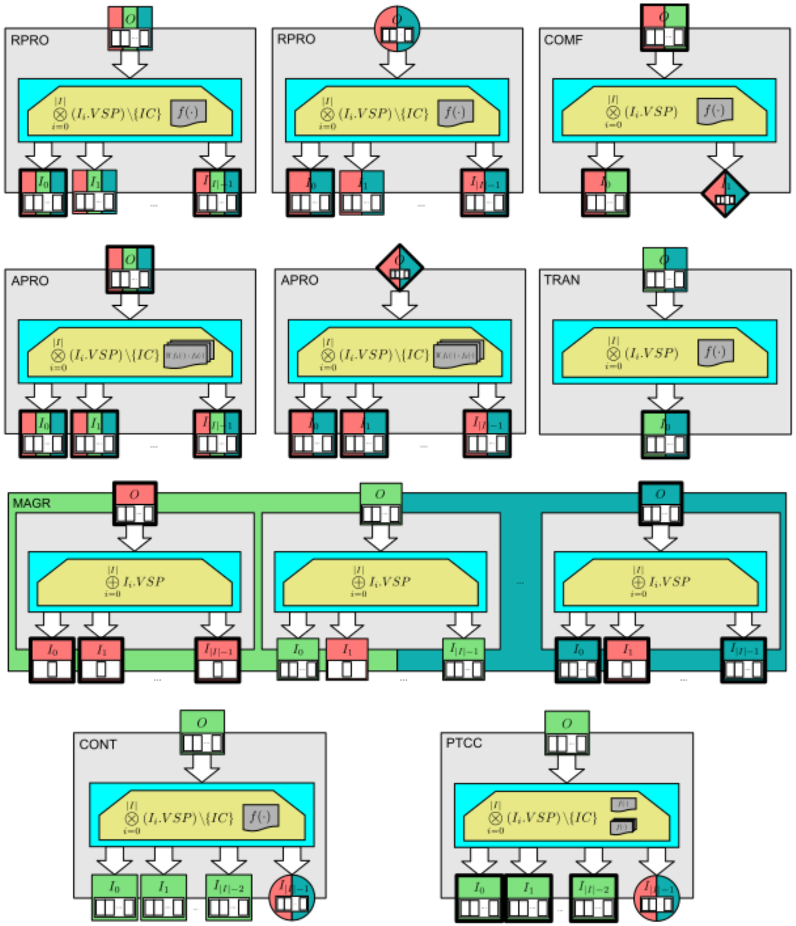


- DVGs are a novel combination of different methods and structures:
  - Dependency graphs
  - Value trees
  - Decision trees
  - Constraint graphs
  - Function composition

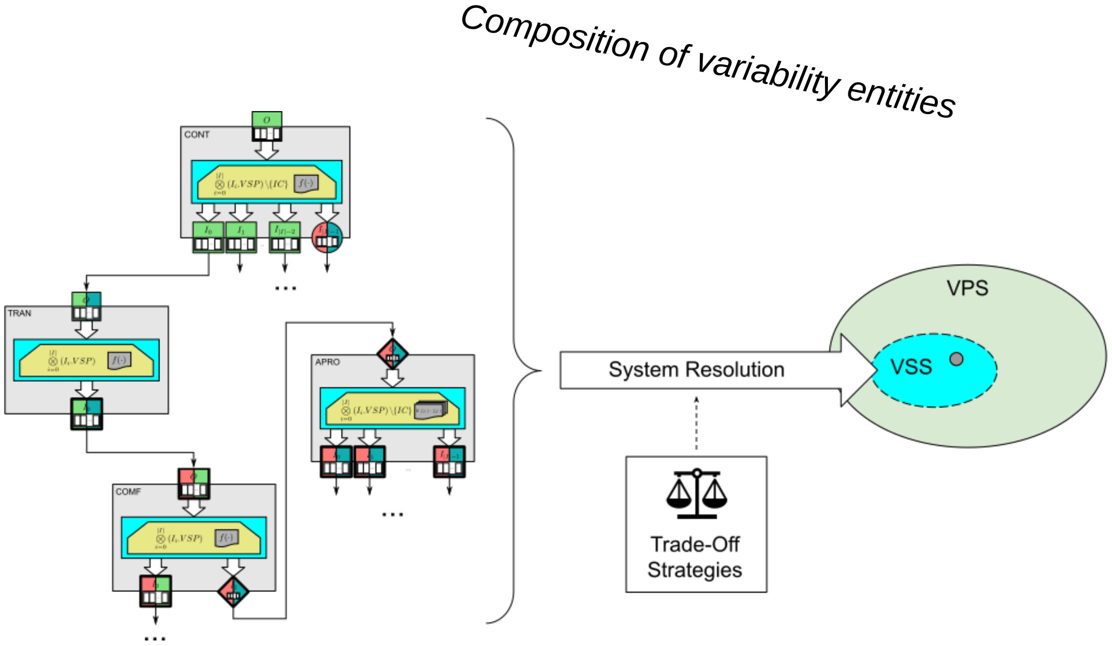
This novel combination results in a generalized, more powerful expressiveness required to represent appropriate problem spaces in the context of a RSE

- DVGs are tailored to the boundary conditions of a RSE (composition, separation of roles)

DVGs: Modeling Patterns and Systems

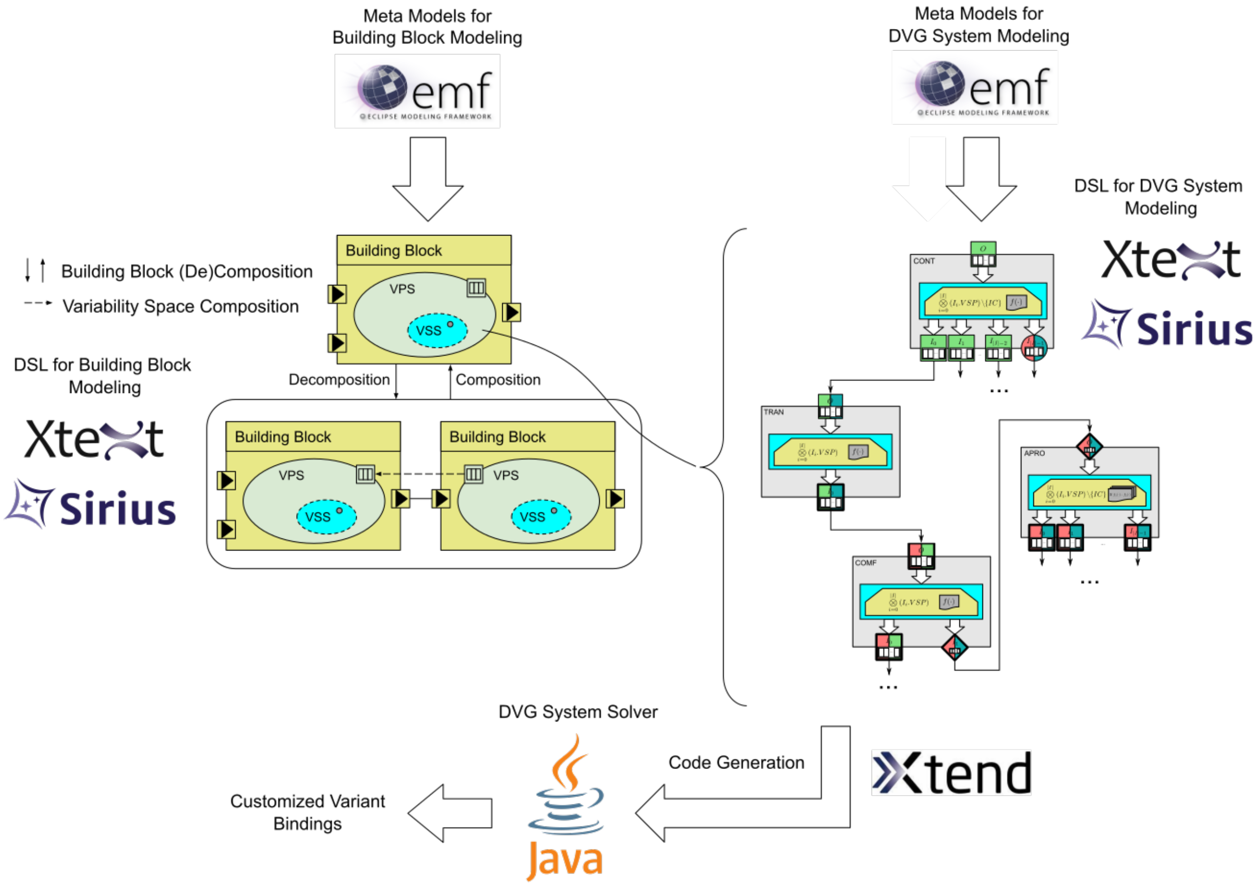


Based on the basics different modeling patterns were derived that serve as templates to compose DVG systems



Implementation: Variability Management in a RSE by MDSD

- Realization of Variability Management in a RSE by
  - building block modeling
  - DVG system modeling
  - Generating solver code for DVG system resolution



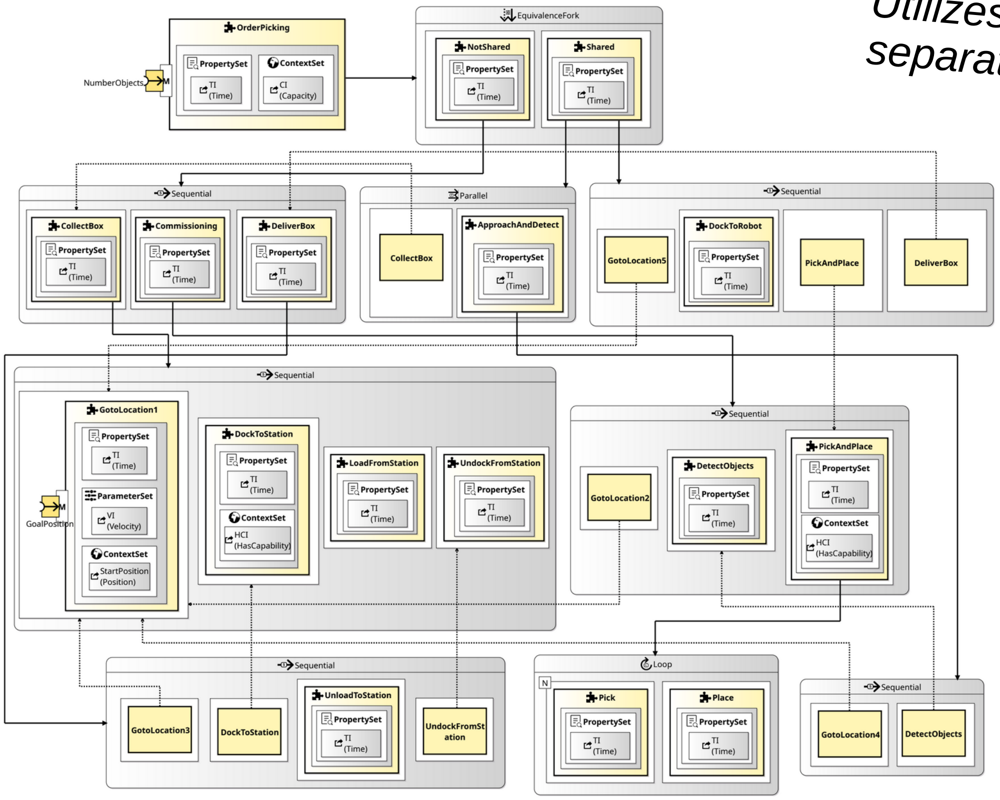
The Use Case of the Paper: Overview

- Order Picking task with variable requirements
  - Minimizing time or
  - Staying below a maximum time limit
- Different robot resources available with different capabilities and properties
  - Can grasp objects or not, maximum velocity, current position, …
- Different environment conditions
  - Layout of the warehouse

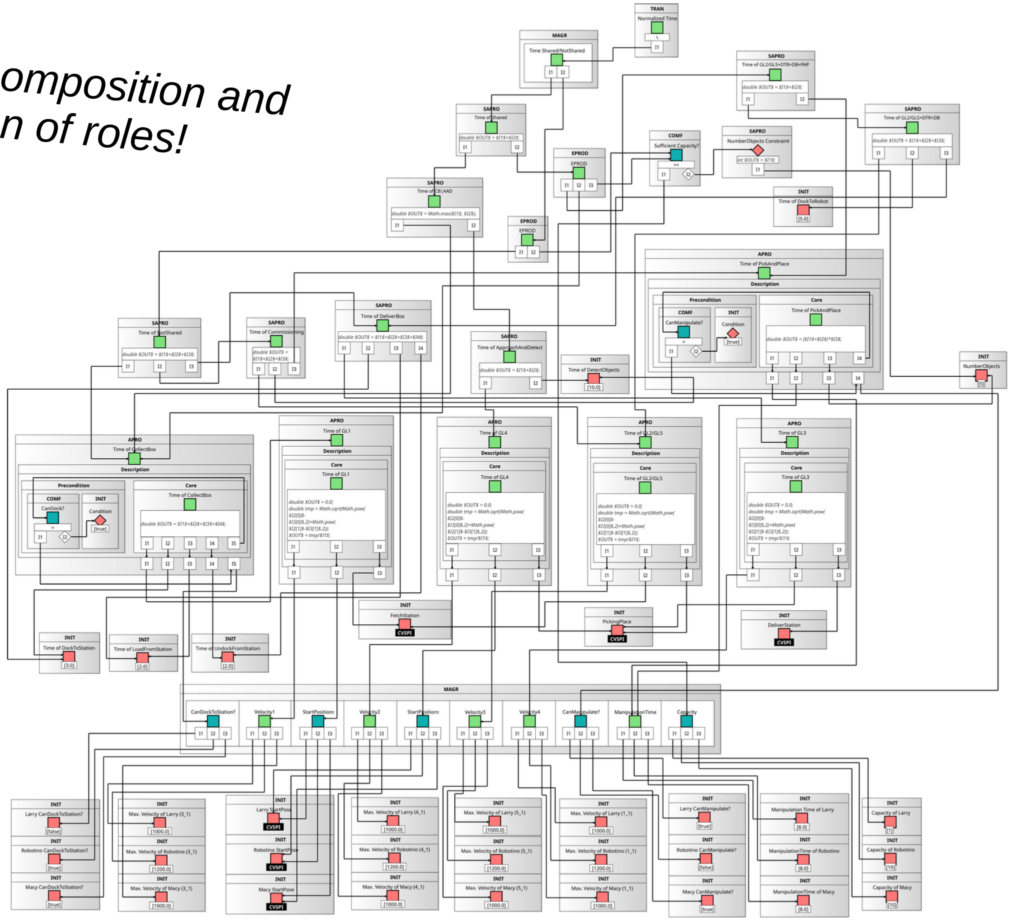
We apply our general method for variability management to model and solve this problem!

Which combination of available robots in which individual configuration is suitable for the current environment to fulfill the specified task requirements?

The Use Case of the Paper: Building Block and DVG System Model



Utilizes composition and separation of roles!



Experiment of the Use Case: Minimizing Time (3 Robot Candidates)

Table 1: Properties of our service robot candidates.

Robot	DockToStation	PickAndPlace	Max Vel.
Robotino	true	false	1200.0
Larry	false	true	1000.0
Macy	true	true	1000.0

Table 2: The possible task allocations in this example based on the available service robots (table 1).

Allocation	R^{S_T}	R^{S_M}	R^N
1	-	-	Macy
2	Robotino	Larry	-
3	Robotino	Macy	-
4	Macy	Larry	-

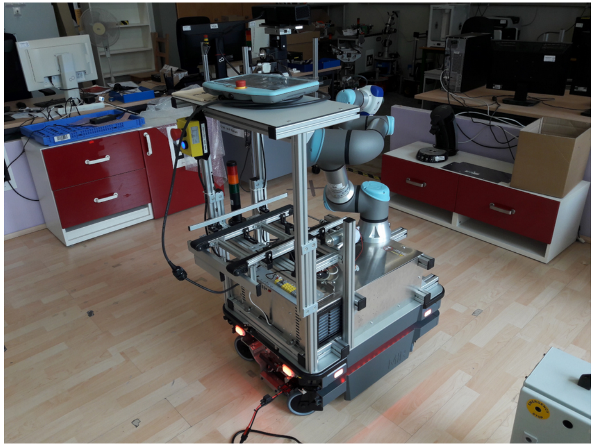


Table 3: Constant values in the example

Names	Values
NumberObjects | *FetchStation*	5 | (6.0, 7.0)
PickingPlace | *DeliverStation*	(5.0, -10.0) | (-3.0, 6.0)
$t_{DockToStation}$ | $t_{LoadFromStation}$	3.0 | 2.0
$t_{UnloadToStation}$ | $t_{UndockFromStation}$	2.0 | 2.0
$t_{DetectObjects}$ | $t_{DockToRobot}$	10.0 | 5.0
t_{Pick} | t_{Place}	8.0 | 8.0

Table 4: Varying start positions for the service robots in the different experiments

Experiment	Robotino	Larry	Macy
1	(-11.0, -14.0)	(3.0, 12.0)	(1.0, 1.0)
2	(1.0, 1.0)	(-2.0, -8.0)	(-11.0, -14.0)
3	(-11.0, -14.0)	(-2.0, -8.0)	(1.0, 1.0)

DVG solver determines the corresponding allocation for the current situation fulfilling the specified requirements

ZAFH
Intralogistik

EUROPÄISCHE UNION
Europäischer Fonds für regionale Entwicklung

EFRE
Investition in Ihre Zukunft.

Baden-Württemberg