Integrating ML and AI in Model-Based Systems Engineering for Trusted Autonomy

John S. Baras, Sandeep Damera, Praveen Kumar,
Clinton Enwerem, Daniel Hunter, Erfaun Noorani
Institute for Systems Research
University of Maryland College Park
USA

February 5-7,2024
18th MODPROD Workshop on Model-Based Cyber-Physical Product Development
Linköping University, Sweden

Advancing the Foundations of AI and ML for Trusted Autonomy

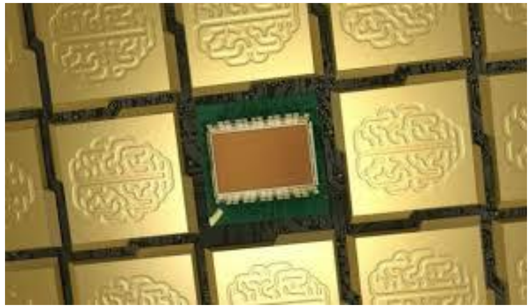
- Rigorous Mathematics for Deep Networks – Universal Architecture emerging (“One Learning Algorithm Hypothesis”)
- Non von-Neumann computing – do not separate CPU from Memory – Synaptic NN, in-memory processing -- HTM
- Universal ML -- Integrate Deep NN and Synaptic NN
- Knowledge Representation and Reasoning: Integrate Knowledge Graphs and Semantic Vector Spaces
- Progressive Learning, Knowledge Compacting
- Link Machine Learning with Knowledge Representation and Reasoning
- Inspirations from neuroscience: attention, memory, time scales

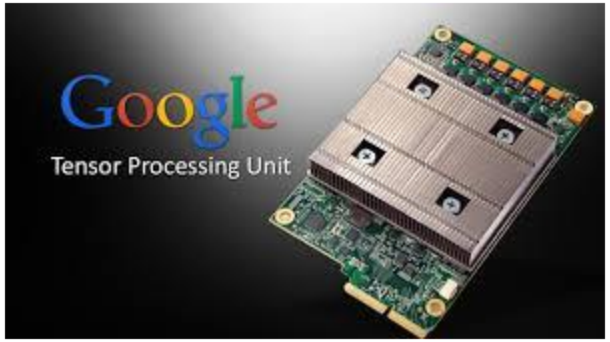
The Institute for Systems Research

Brain-Like Computers

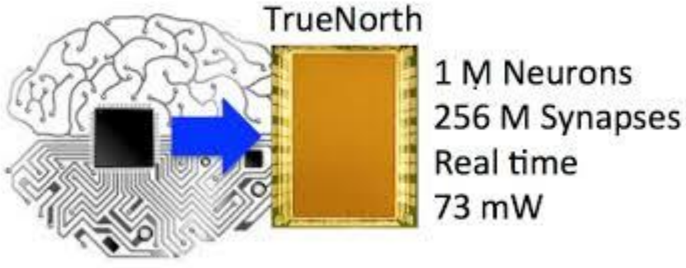
Race to design and manufacture “brain-like” computers is on

IBM





NEUROMOPRPHIC?



Feb 2018 INTEL establishes INTEL Neuromorphic Research Community (INRC) -- academic-industry-government group/consortium

INTEL LOIHI

1000x more energy efficient

Spike based info processing

Storing info on synapses

130K neurons, 130M synapses

We Pursue:

Hyperdimensional Computing

Symbolic Vector Architectures

Hierarchical Temporal Memory

Reservoir Computers

PROBLEM ADDRESSED AND SIGNIFICANCE

Systematic Methodology and Software Tool Suite for Trusted Autonomous Systems

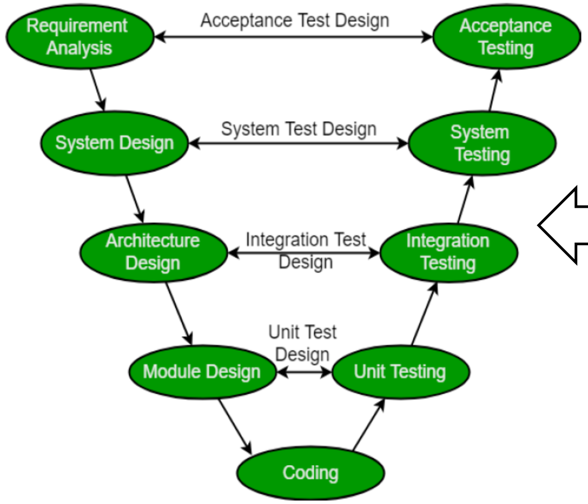
Critical need for many US Army and DoD missions, and also many commercial applications

HOW

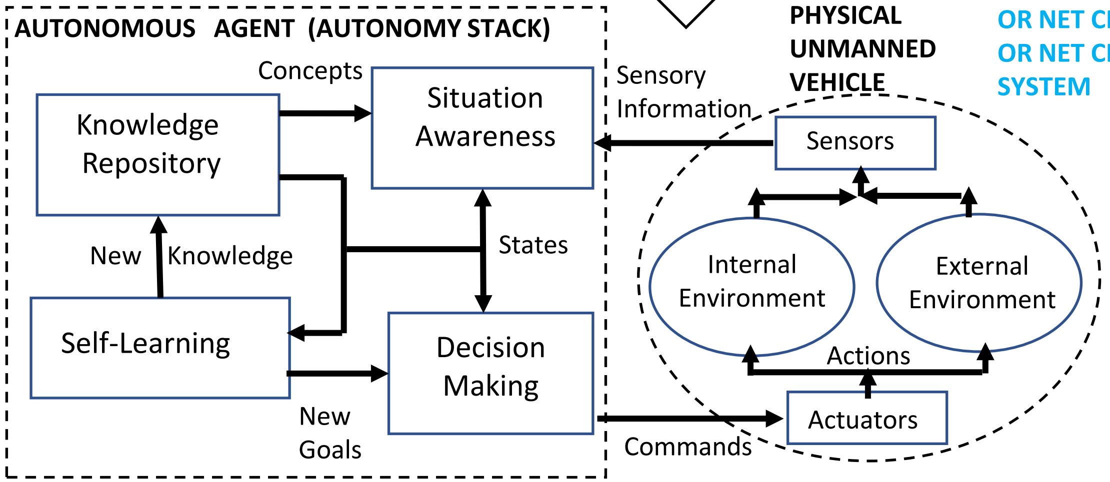
MBSE – Static, known Requirements, Verifiable

MBSE defeated by complexity and diversity of autonomous systems

Large Data Sets Driven ML - AI to the rescue: simulations, experiments, operations



Digital Twin or Testbed System



OR NET CPS
OR NET CHPS
SYSTEM

Design space exploration via tradeoffs to prioritize potential investments from portfolio of modules: sensors, actuators, cyber chips, materials, engines, architectures, algorithms, new technologies, etc.

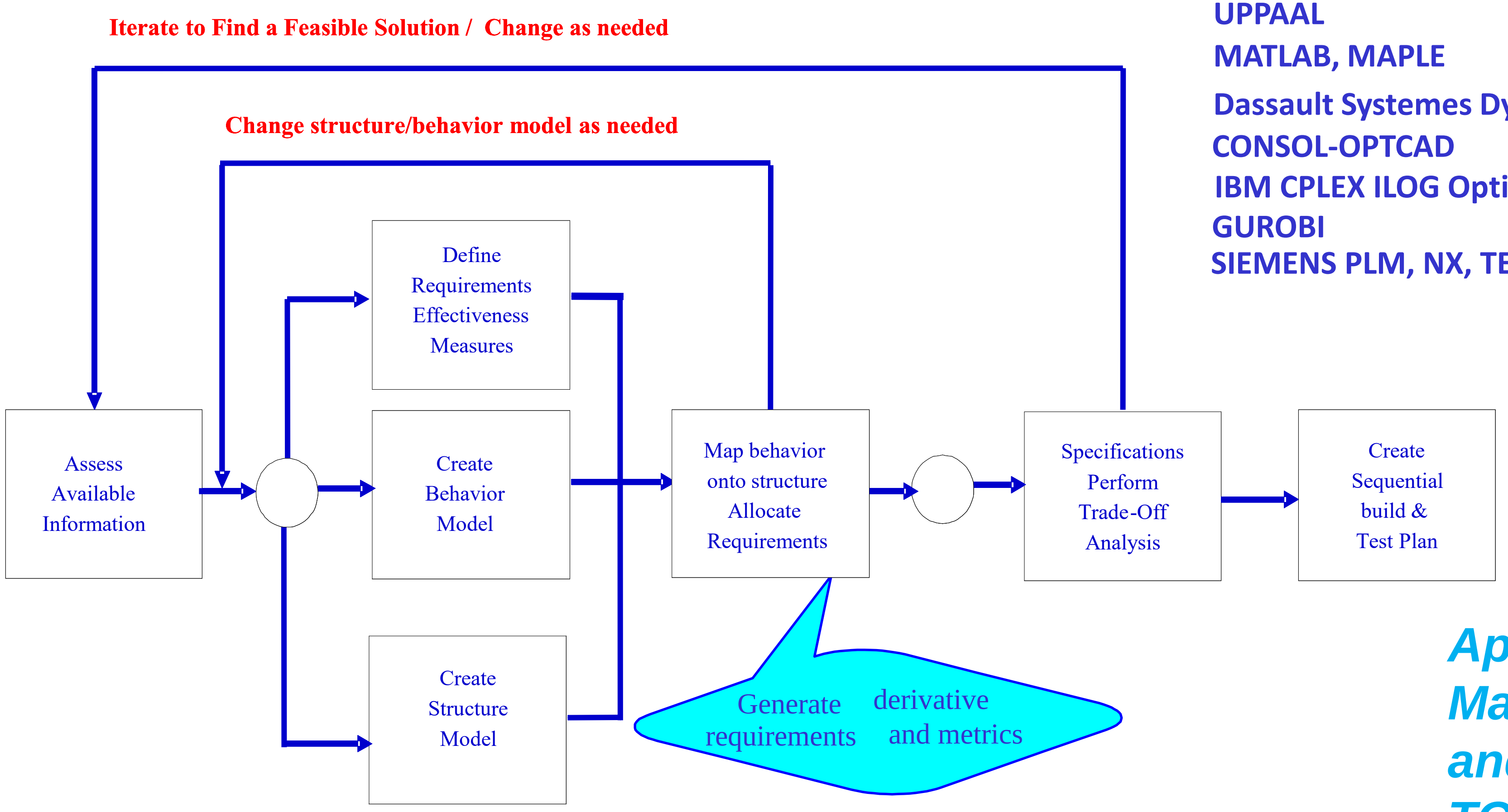
NOVELTY and VALUE

Integrating large data sets makes feasible the design of **high performance trustworthy autonomous systems** trough empirical (DD) **and** formal (MBSE) validation, with changing requirements and scenarios.

Not possible otherwise. Currently major open problem.

UMD Model- Based Systems Engineering Process

PRODUCT: Integrated System Synthesis Methods & Software Tool Suites



UML - SysML - GME - eMFLON
ANSYS Model Center
Rapsody
UPPAAL
MATLAB, MAPLE
Dassault Systemes Dymola, CATIA, PLM
CONSOL-OPTCAD
IBM CPLEX ILOG Optimization Studio
GUROBI
SIEMENS PLM, NX, TEAM CENTER

Apply this to: Design, Manufacturing, Operations and Management
TO THE WHOLE LIFE-CYCLE
$\Rrightarrow$ MBE

Copyright © John S. Baras 2023

UMD Rigorous Framework for Model-Based Systems Engineering ⊗ Data-Driven Methods (ML-AI)

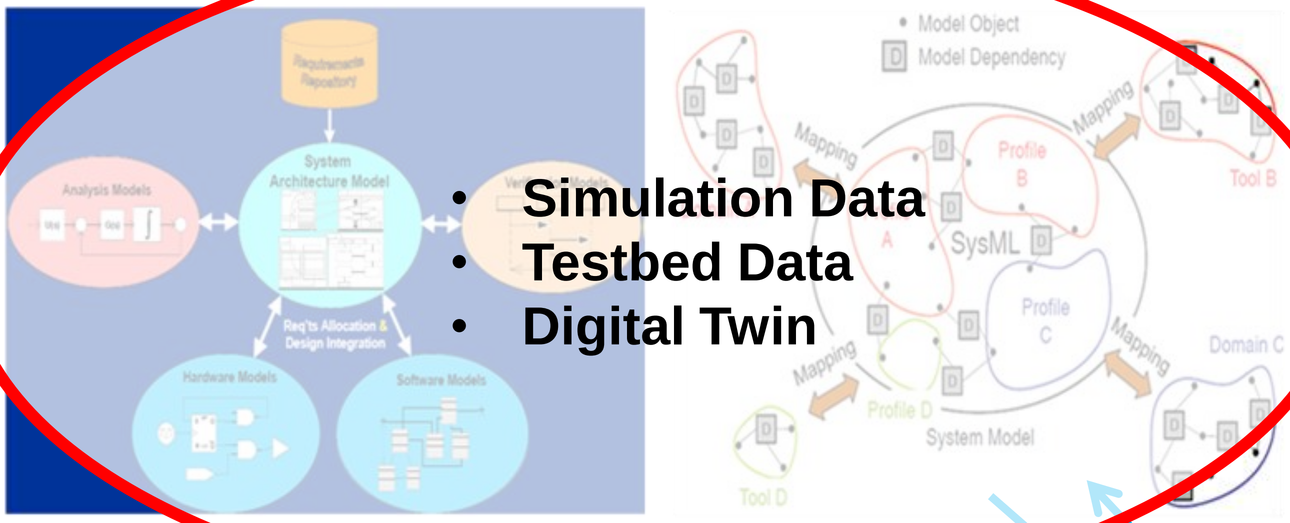
The Institute for Systems Research

PRODUCT – Proposed DATA DRIVEN ENHANCEMENTS

Scalable holistic methods, models, tools for enterprise level SE

Multi-domain Model Integration via System Architecture Model (SysML)

System Modeling Transformations



- **Simulation Data**
- **Testbed Data**
- **Digital Twin**

" Master System Model"

Design space exploration

Tradeoffs

IBM Optimization Studio, CPLEX, CONSOL-OPTCAD

Machine Learning & Artificial Intelligence

ADD & INTEGRATE

- Multiple domain modeling tools
- Tradeoff Tools (MCO & CP)
- Validation / Verification Tools
- Databases and Libraries of annotated component models from all disciplines

components and models

Machine Learning & Artificial Intelligence

BENEFITS

- Broader Exploration of the design space
- Modularity, re-use
- Increased flexibility, adaptability, agility
- Engineering tools allowing conceptual design, leading to full product models and easy modifications
- Automated validation/verification

APPLICATIONS

- Avionics
- Automotive
- Robotics
- Smart Buildings
- Power Grid
- Health care
- Telecomm and WSN
- Smart PDAs
- Smart Manufacturing

AI/ML Value Addition in the IDDMBSE Framework

Requirements

- AI/ML tools for converting Natural Language requirements into formal (including temporal logic) specifications.
- Automated checking for Consistency, Completeness and Correctness of the requirements.
- Automated ranking of requirements based on significance and impact
- Integration of model-checking tools such as UPPAAL and PRISM for formalized specifications

Design Space Exploration

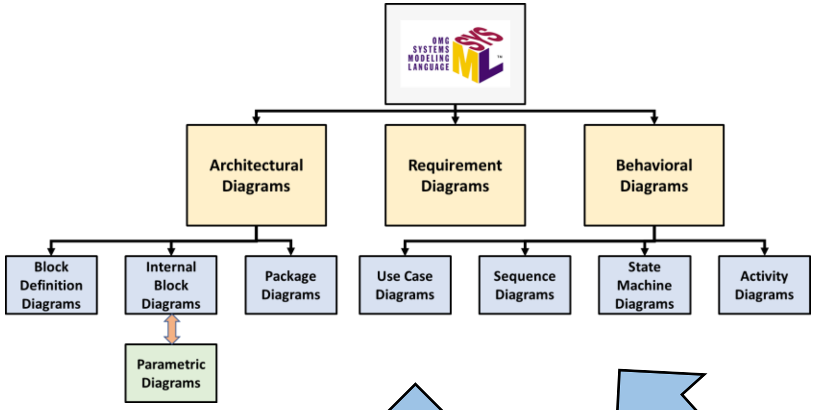
- The number of potential design configurations grows exponentially with the complexity of system design.
- Evaluating performance via purely data-driven methods (i.e. simulations) computationally and time costly.
- Ongoing work on providing theoretical tools for “informed” design space exploration (Functional optimization, Constraint-based reasoning, etc.) – to reduce the number of simulation runs and provide statistical guarantees.

Verification and Validation

- Verifying robustness and risk-sensitivity in design against system requirements.
- Domain Randomization for transferring IDDMBSE results from simulation to the real world—**THE SIM-TO REAL GAP**

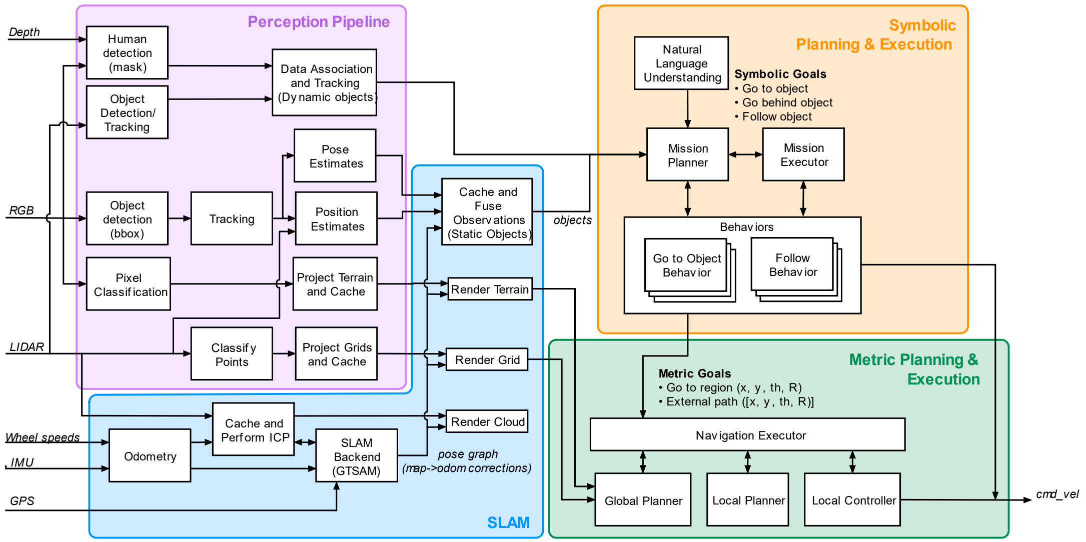
Our Approach

SysML Models and Diagrams



Mapping AS components to SysML models

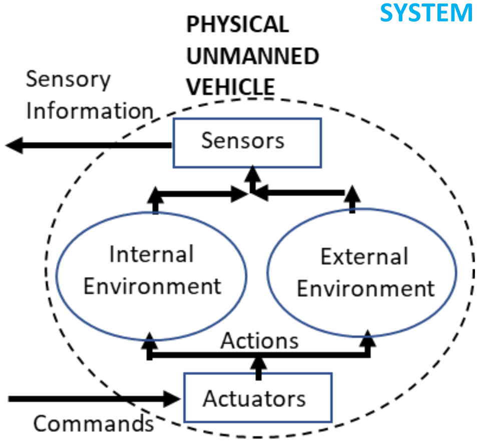
Autonomy Stack (AS)



LINK TO Formal Model Tools (UPPAAL, PRISM)
for Correct Task Execution, Timing analysis, Safety, Specification satisfaction, Robustness, Autonomy, Learning, Intelligence …

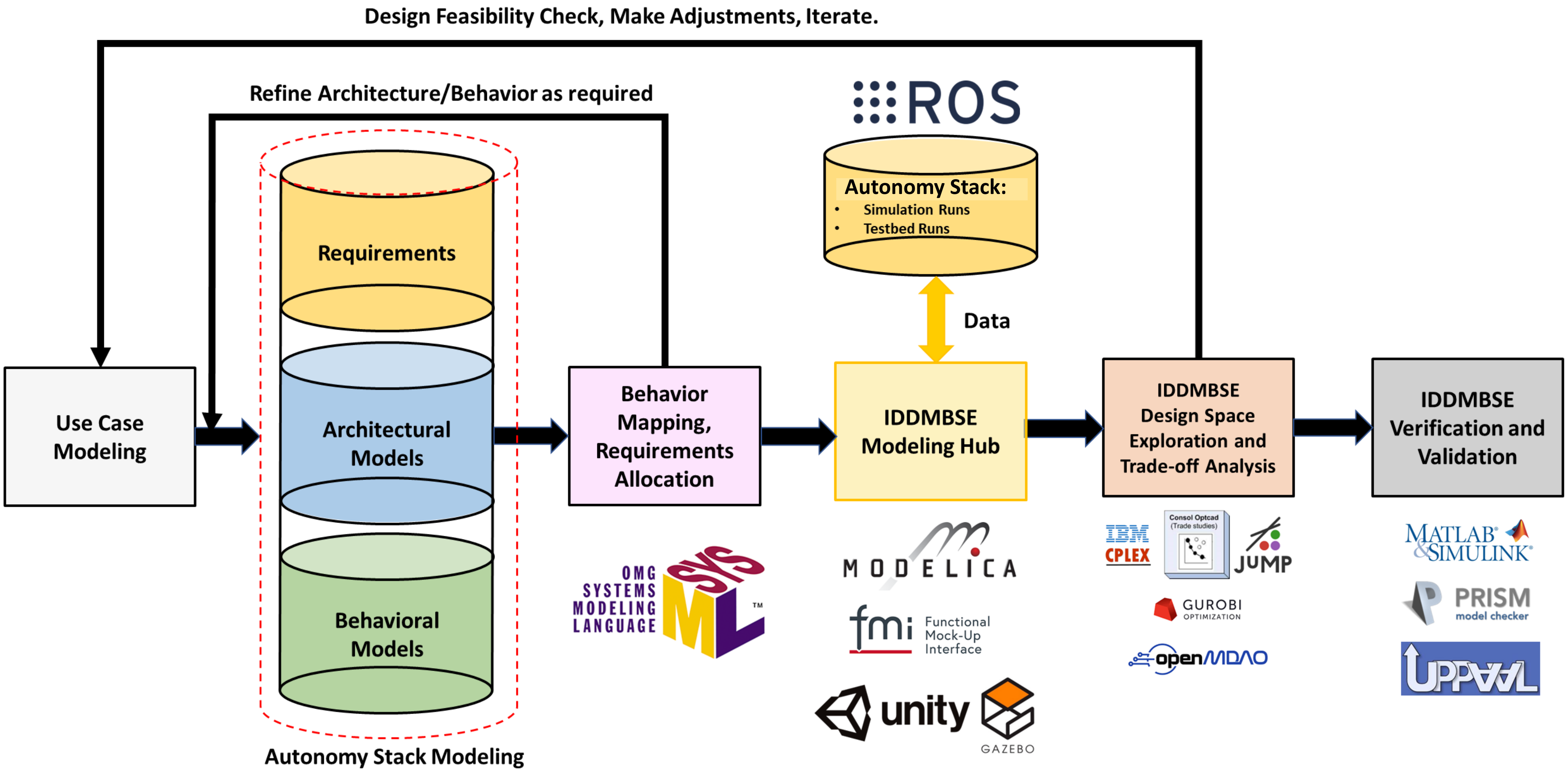
Design space exploration via tradeoffs to prioritize design decisions, investments, from portfolio of modules:
sensors, actuators, cyber chips, materials, engines,
algorithms, architectures, and new technologies.

LINK TO simulations, experiments, operations
for data generation, ML, AI

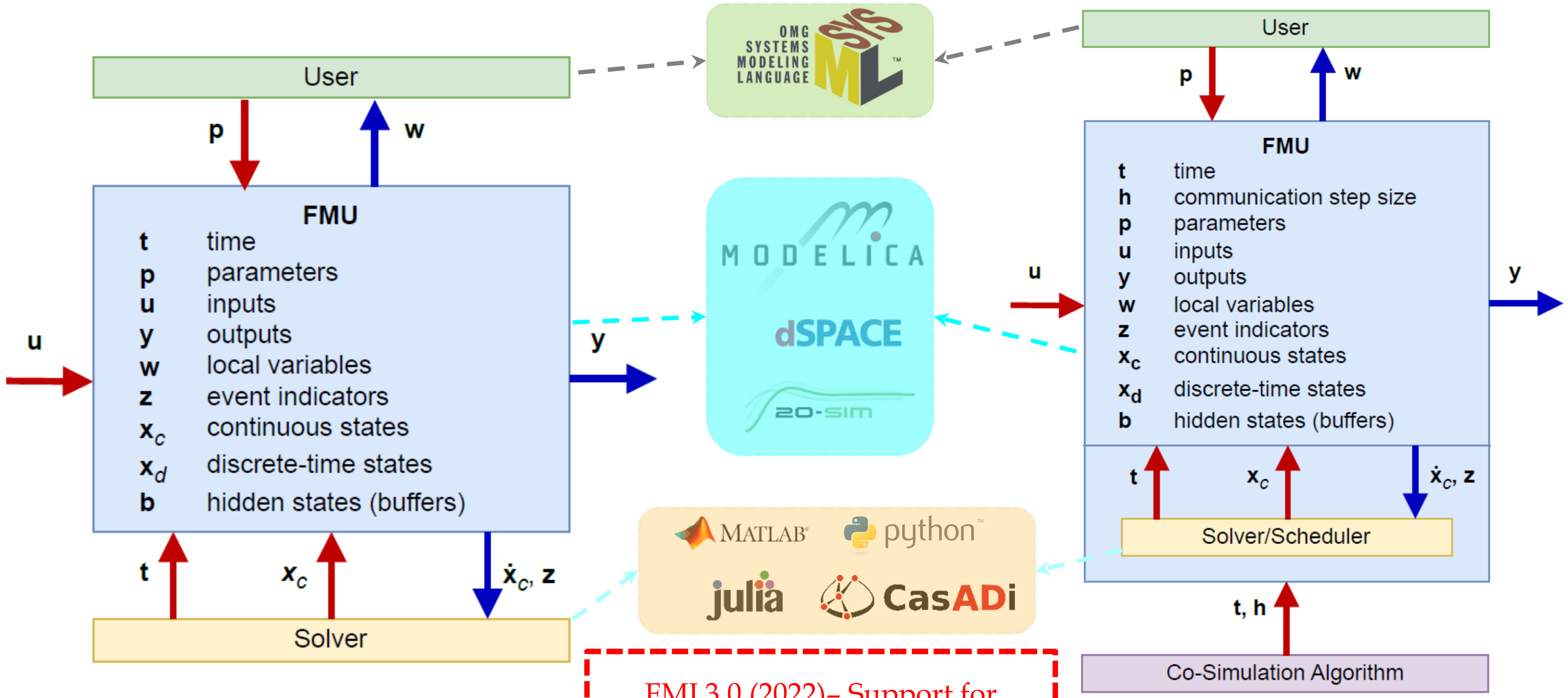


OR NET CPS
OR NET CHPS
SYSTEM

Our Approach: Specification of IDDMBSE and Tool Suite Architecture



The Solution: FMI and FMU for Model Exchange and Co-Simulation



FMI 3.0 (2022)– Support for Directional Derivatives! Unlocks the potential for hybrid data & model-based methods.

FMI for Model Exchange

FMI for Co-Simulation

Summary of Most Recent Results

- In-depth investigation of needed software development and implementation for IDDMBSE toolsuite.
- Achieved First Instance of Mapping ROS-based Generic Autonomy Stack components to SysML components. First Instance of executable software implementation.
- Development of **PERFECT (PERFormance Evaluation Composable Toolsuite)**; planning patent submission. Demonstration on AGV robotic examples of execution of ROS-based Autonomy Stack modules from SysML commands.
- Initiated development of new tool for **TRadeoff Analysis and DEsign Space EXploration (TRADES-X)** on SysML side (formal) and improvements with data-driven methods (Autonomy Stack side). Demonstration on AGV robotic examples.
- Investigated robust path planning problem as focal/benchmark problem in framework.

Driving Use Case: Autonomous Robot Navigation Task

- Objective → Waypoint Navigation Task (Given a destination with respect to robot frame, plan a path and actuate the robot autonomously)
  - No prior map of the environment provided
  - Simultaneous Localization and Mapping (SLAM) via on-board sensors to explore the environment
  - Currently there is no perception module to reason about the environment
  - Global and Local planning modules to actuate the agent (husky robot) from point A to point B
- 4 test simulation environments
- 4 sensor modalities with multiple variations per modality
  - RGB camera
  - Depth camera
  - Laser range finders
  - LiDAR
- Multiple global and local planners

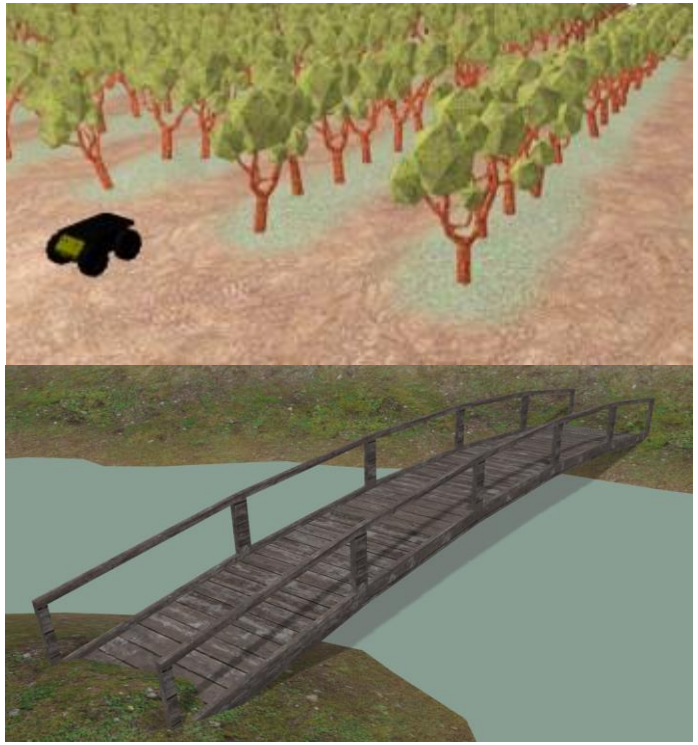
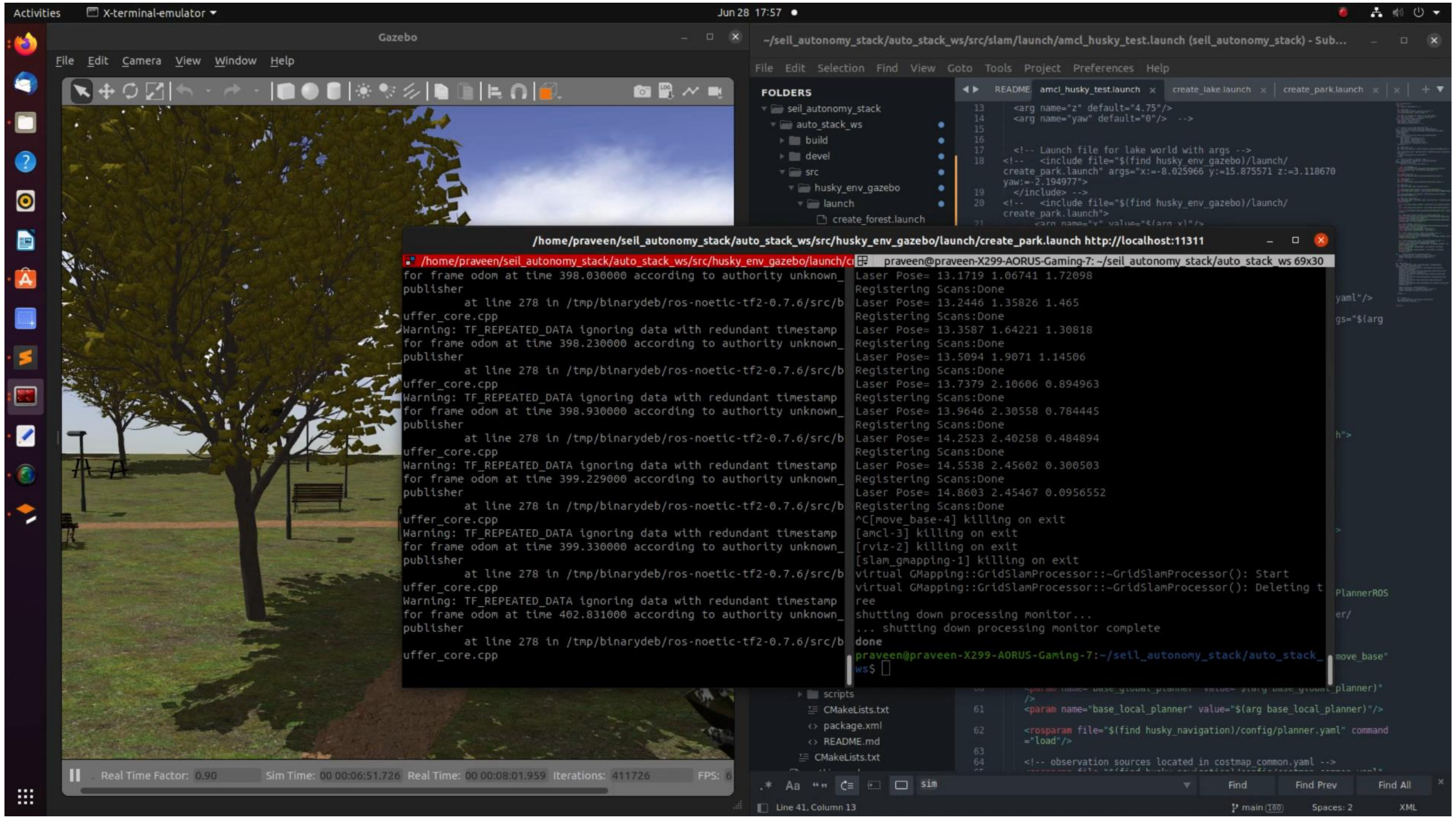


Fig: Two sample test simulation environments

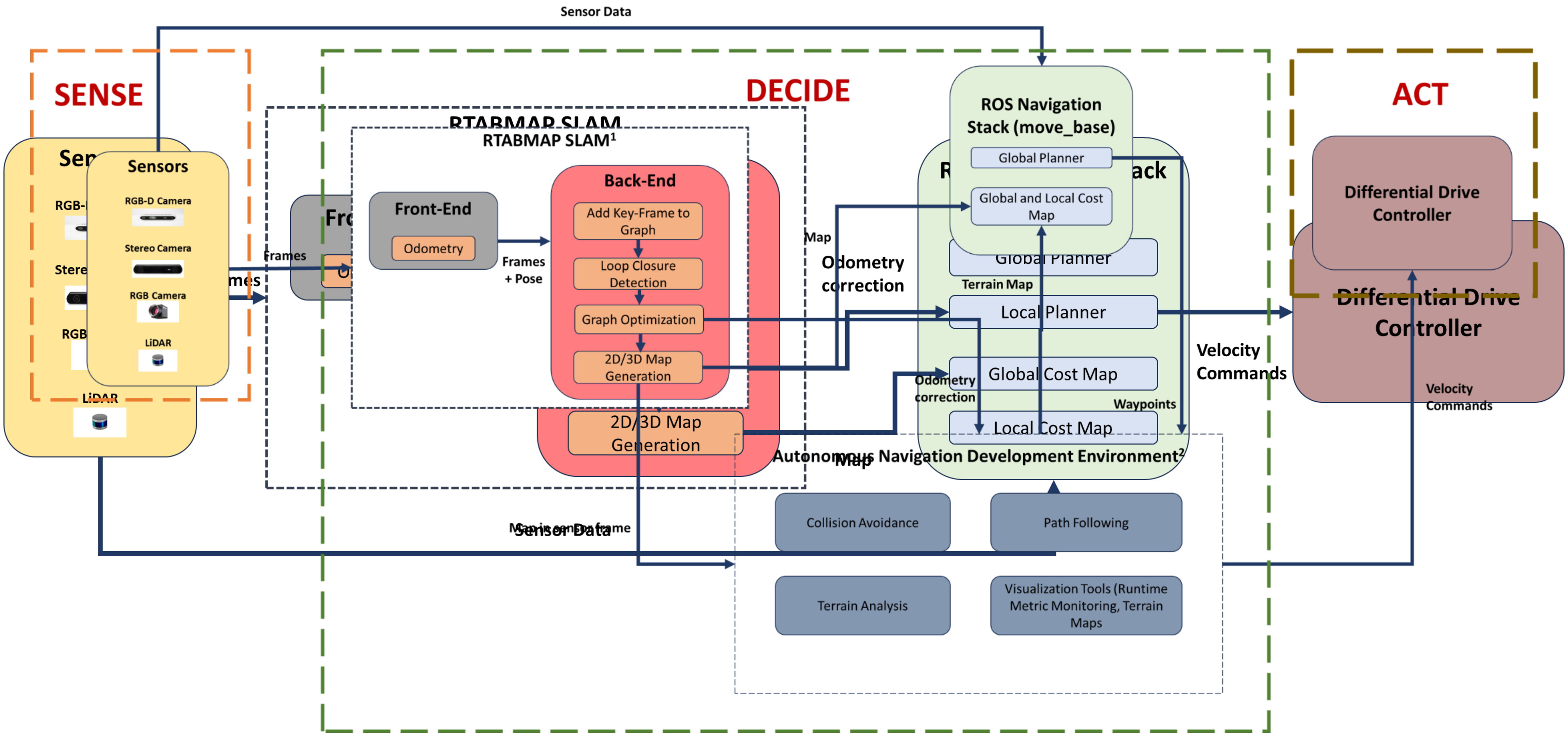
Simultaneous Localization and Mapping (SLAM) Pipeline



Demonstrating SLAM capability for Clearpath Husky Robot in Gazebo simulation environment

- LIDAR-based SLAM creates a 2D occupancy grid and cost map using LIDAR scan and odometry data from the Clearpath Husky robot.
- Default ROS global planner to plan the generate waypoints to the local planner.
  - Local planner - Dynamic Window Approach planner
  - Localization - Adaptive Monte Carlo Localization

UMD-SEIL Autonomy Stack Architecture



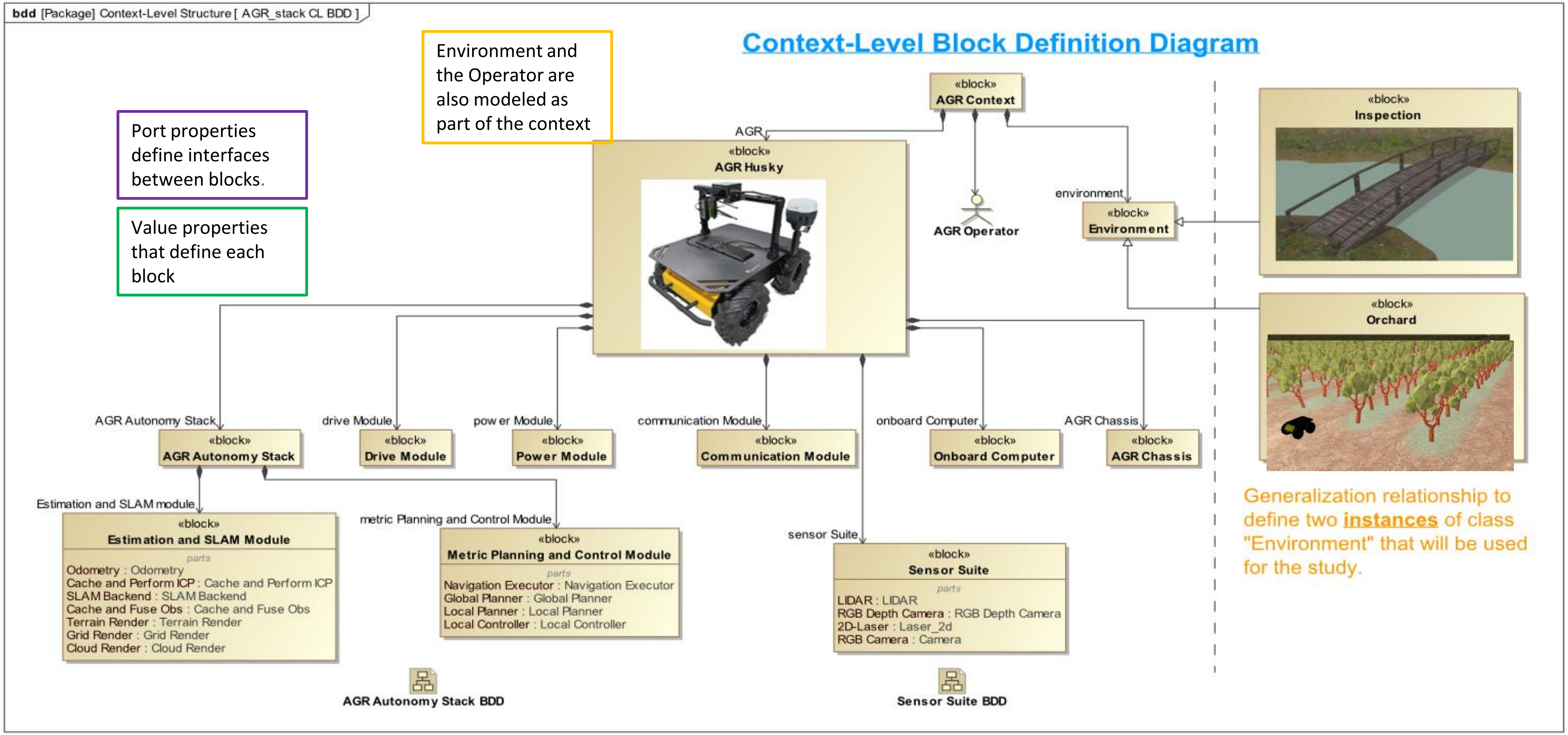
[1] RTABMAP SLAM https://introlab.3it.usherbrooke.ca/mediawiki-introlab/images/3/31/Labbe2015ULaval.pdf
[2] CMU Autonomous Exploration Development Environment https://www.cmu-exploration.com/

Progress on the UMD-SEIL Stack

- Autonomous Exploration Development Environment developed by CMU
- Contains a variety of simulation environments, autonomous navigation modules, and a set of visualization tools.
- Offers a flexible platform for run-time performance monitoring.
- Status: The tool currently works in a standalone manner
- Currently working on integration with the UMD SEIL Stack and the ***PERFECT*** toolsuite.

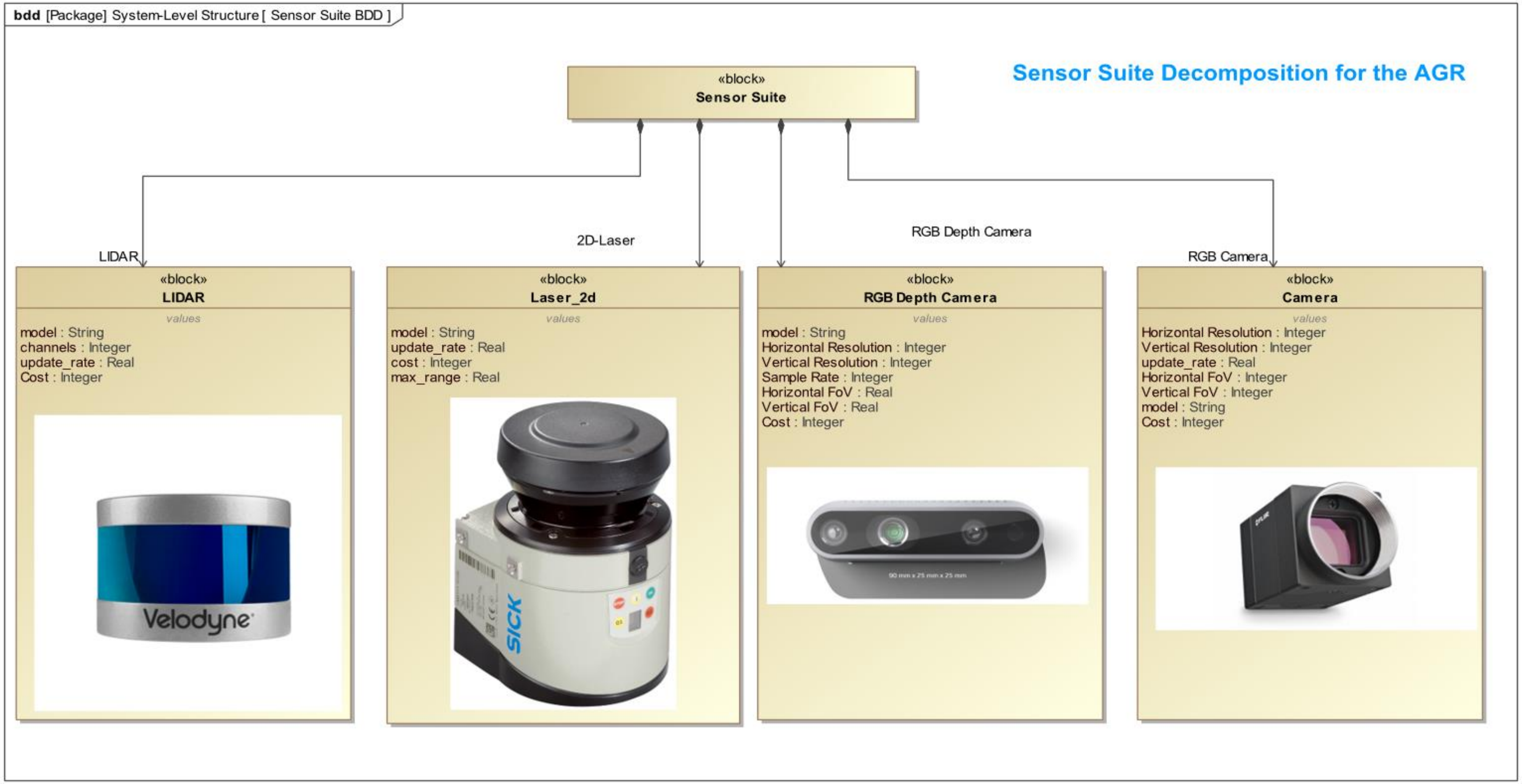
[1] **CMU Autonomous Exploration Development Environment** https://www.cmu-exploration.com/

SysML Structural Modeling



Context-Level Block Definition Diagram of the Autonomous Ground Robot (AGR). Defines the structural architecture of both the hardware (AGR) and software (AGR Stack). Directed Composition relationship used to show part components.

System-Level BDDs: Sensor Suite

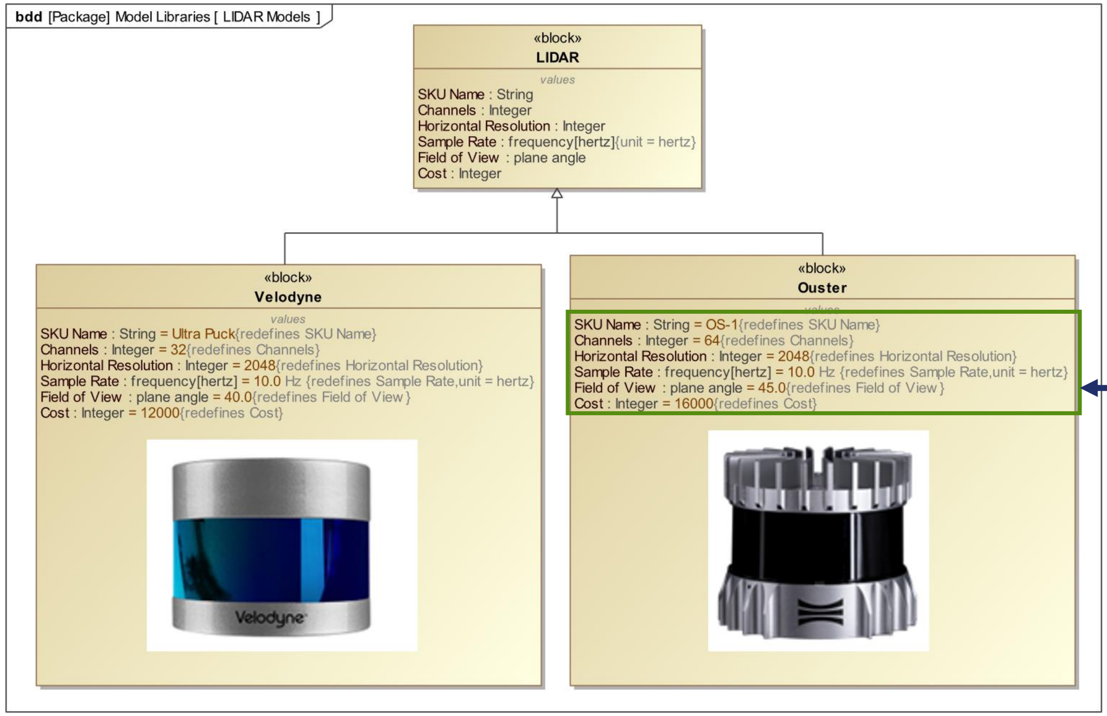


SysML Structural Architecture of the Sensor Suite Block using a Block Definition Diagram.

Value Properties of Sensor Class Blocks shown in the figure.

Mapping: SysML Structure Diagrams ⟷ ROS URDF Parameters

SysML Lidar Structure Specification



Lidar Structure Specification in ROS

ouster_parameters.xml

home > praveen > ouster_parameters.xml

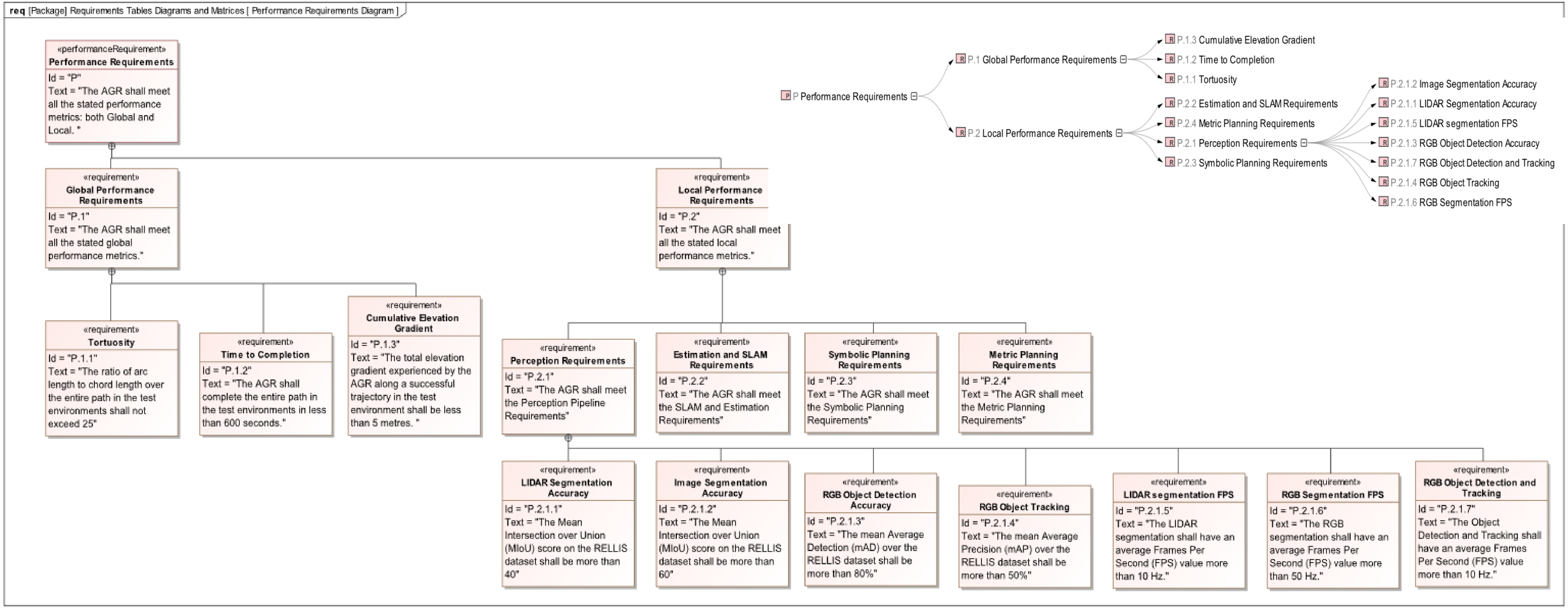
```xml
<?xml version="1.0"?>
<subsystem xmlns:xacro="http://ros.org/wiki/xacro">

  <!-- Handle all ouster types using argument list:
       verticalBeams
       horizontalBeams
       maxRange (meters)
       minVerticalAngle (degrees)
       maxVerticalAngle (degrees)
    -->
  <xacro:macro name="ouster" params="name args">
    <link name="${name}_link" />
    <unity reference="${name}">
      <spawn type="Lidar3D CPU" topic="${args.get('topic','lidar points')}" frame="${name} link"/>
      <configure command="verticalBeams:${args.get('verticalBeams',64)}
      horizontalBeams:${args.get('horizontalBeams',512)} maxRange:${args.get('maxRange',60)}
      minVerticalAngle:${args.get('minVerticalAngle',-16.6)} maxVerticalAngle:${args.get('maxVerticalAngle',16.6)}"/>
    </unity>
  </xacro:macro>

</subsystem>
```

Sensor model parameters defined in the ROS URDF file are mapped to the SysML BDD value parameters.

SysML Requirements Modeling of Autonomy Stack

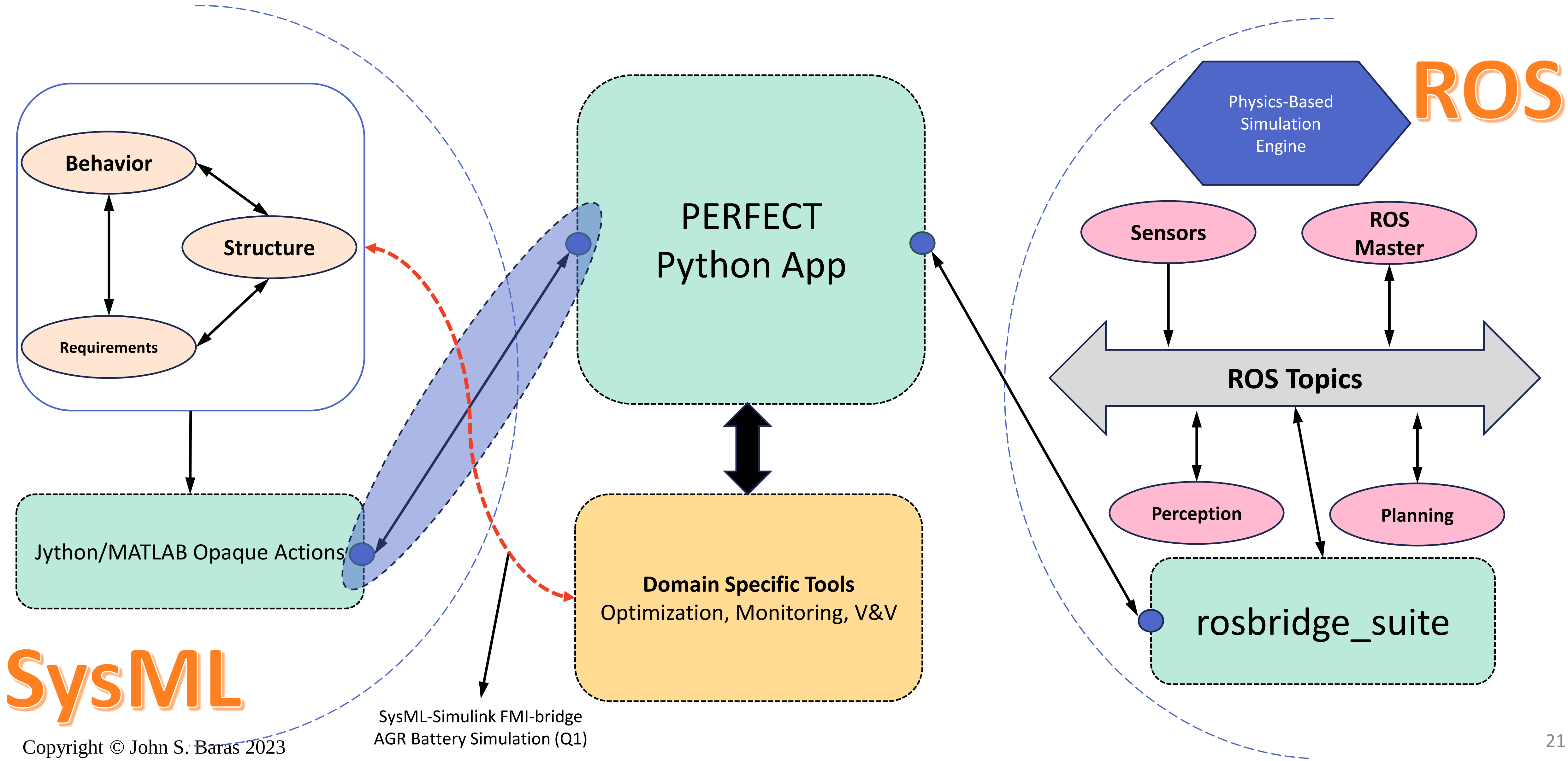


AGR Performance Requirements Decomposition using the Containment relationship. **Top right**: A requirements containment map to track effects. Text-based requirements generated from Metrics for sensor selection problem -- now quantified.

PERFECT: PERFormance Evaluation Composable Toolsuite

- ***PERFECT*** is a Python-based application that bridges the various tools needed for the IDDMBSE framework.
- It has the following salient features:
  - ***Distributed:*** The modeling (SysML), simulation (ROS-Gazebo), and Analysis (MATLAB) tools can operate independently on different workstations connected on the local network.
  - ***Real-Time:*** ***PERFECT*** enables real-time exchange of information between the tools with minimal network overhead.
  - ***Extendible:*** The modular structure of ***PERFECT***, coupled with its use on generic network microframework enables iterative design that can incorporate extended capabilities using a wider set of domain specific tools.

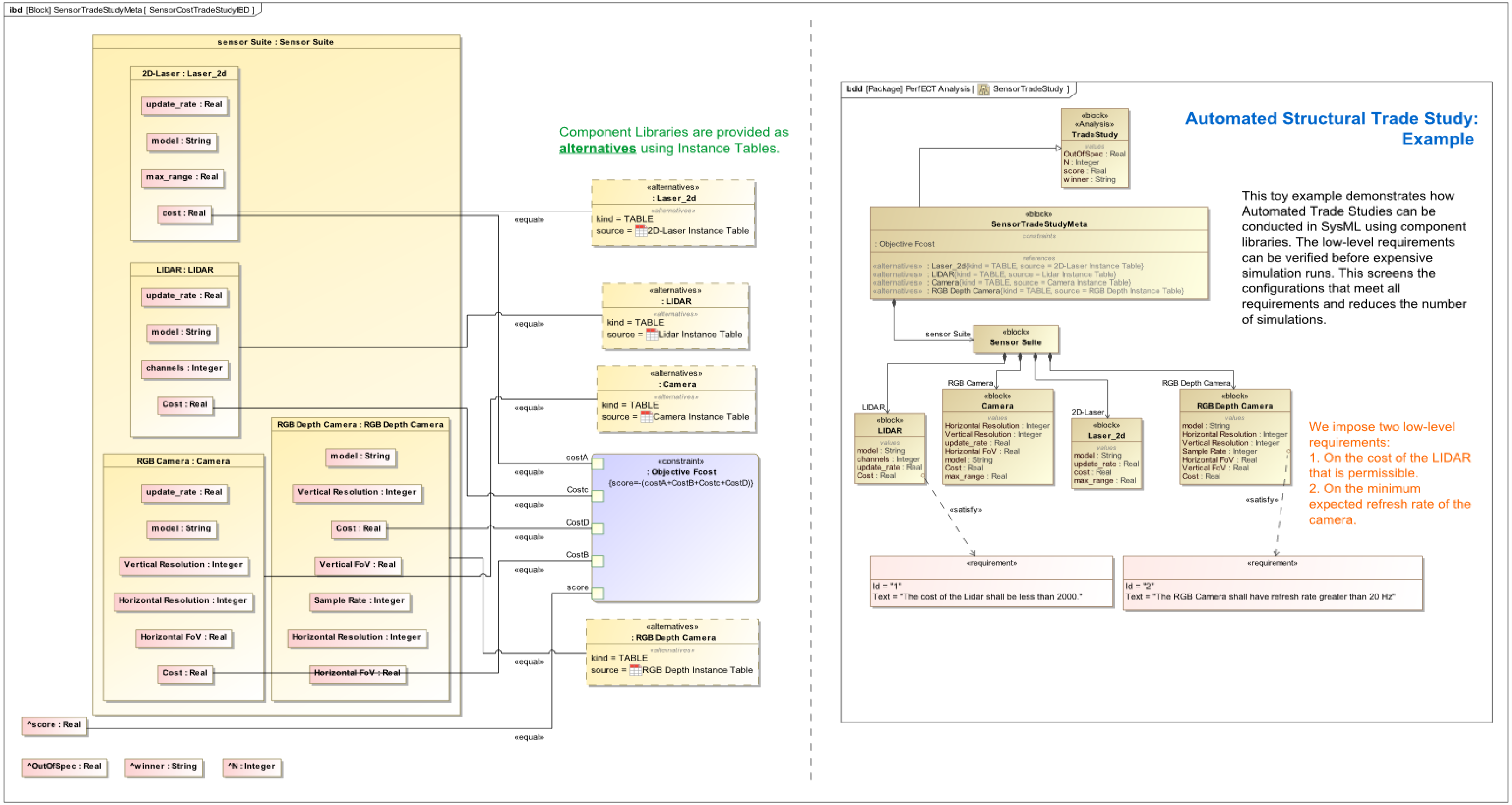
Performance Evaluation Composable Tool (PERFECT)



SysML-Simulink FMI-bridge
AGR Battery Simulation (Q1)

Linking PERFECT App with ROS-based Stack: Husky in Gazebo

SysML Driven Sensor Trade Study

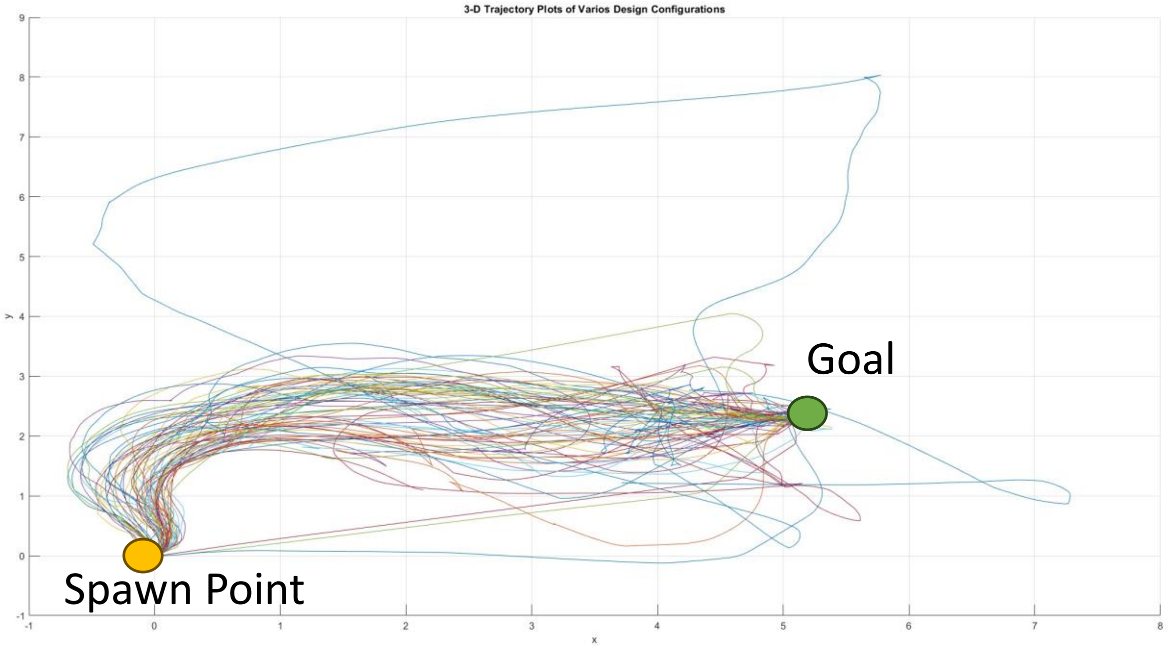


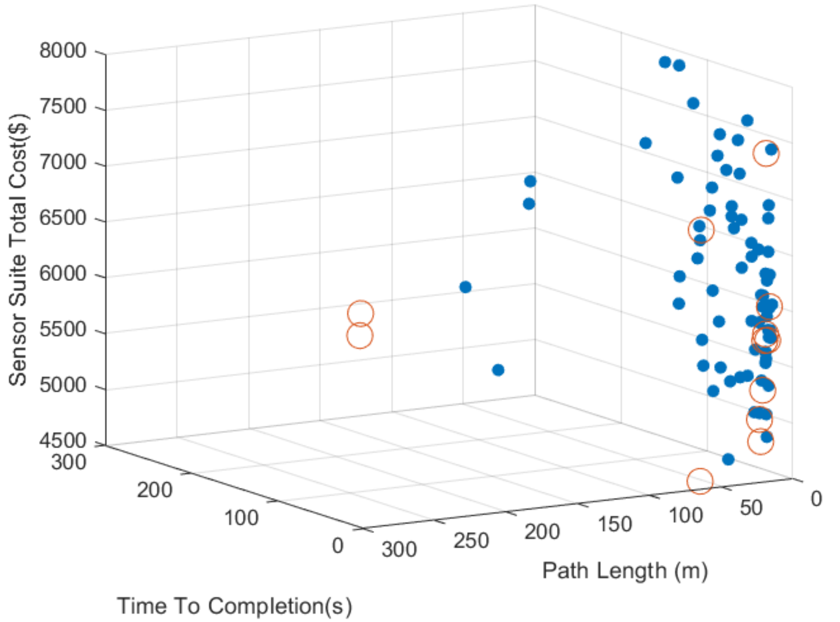
SysML-Driven Sensor Configuration Trade Study: Demo

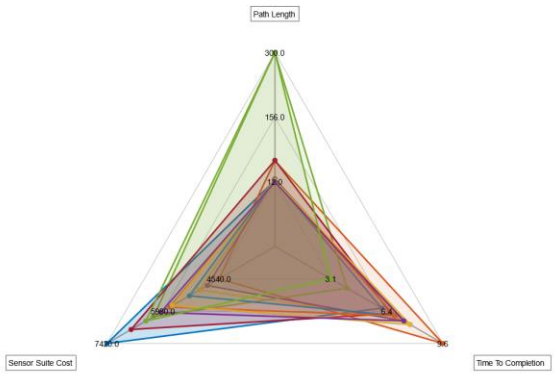
Data Driven Multi-Objective Trade-Off Analysis: Results

Trade Study for Sensor Suite Design:

- 96 possible configurations.
- 24 ruled out for requirement violations.
- Out of the 72 remaining, only 56 configurations succeeded in navigating to the goal.
- Pareto Analysis of the 56 design candidates against **Cost, Time to Completion** and **Path Length** objectives (minimize all), leads to a set of **12** Non-Dominating (pareto) solutions.





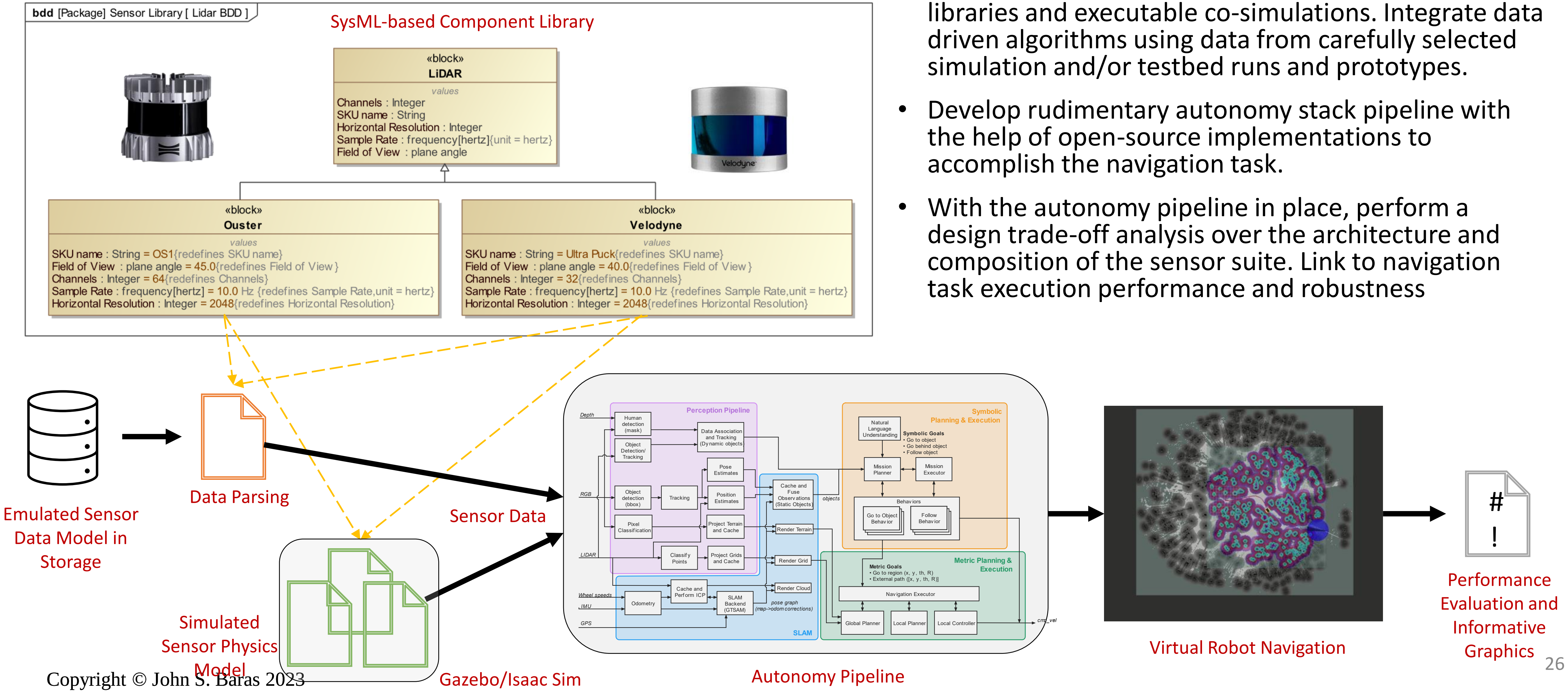


Left: Trajectory plots of design configurations for a given goal task.

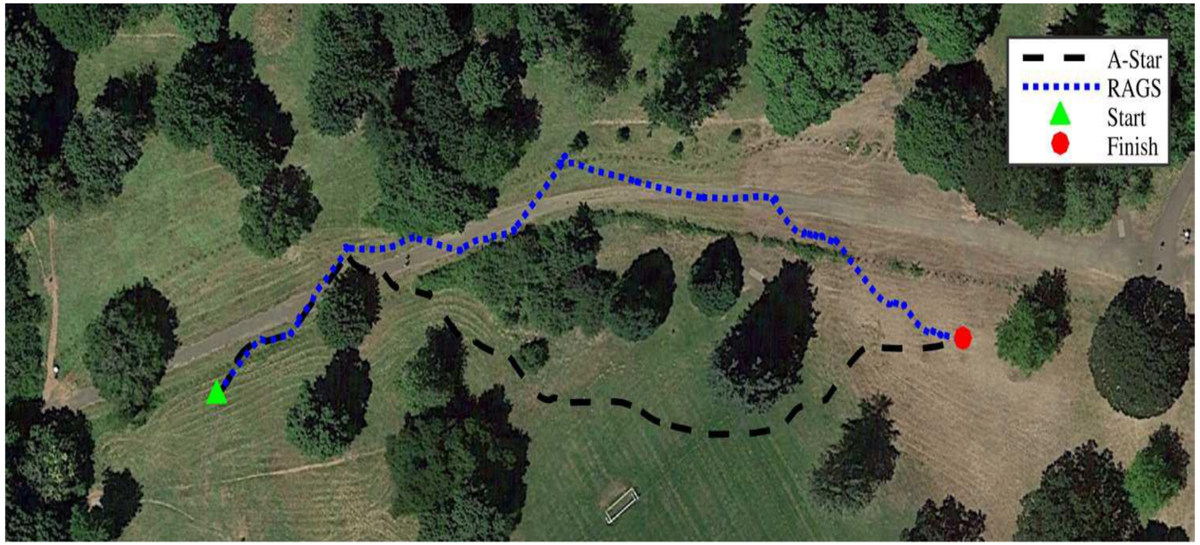
Right Top: The Pareto Frontier of the design candidates Right Bottom: A Spider Plot of the Pareto Design Candidates

SysML-Driven Design Trade-off Analysis

- Use SysML as the IDDMBSE hub to create component libraries and executable co-simulations. Integrate data driven algorithms using data from carefully selected simulation and/or testbed runs and prototypes.
- Develop rudimentary autonomy stack pipeline with the help of open-source implementations to accomplish the navigation task.
- With the autonomy pipeline in place, perform a design trade-off analysis over the architecture and composition of the sensor suite. Link to navigation task execution performance and robustness



Robust Path Planning and Path Following



Uncertainty: Models and Data-Driven Robustness via Risk-Sensitive Optimization and ML / RL

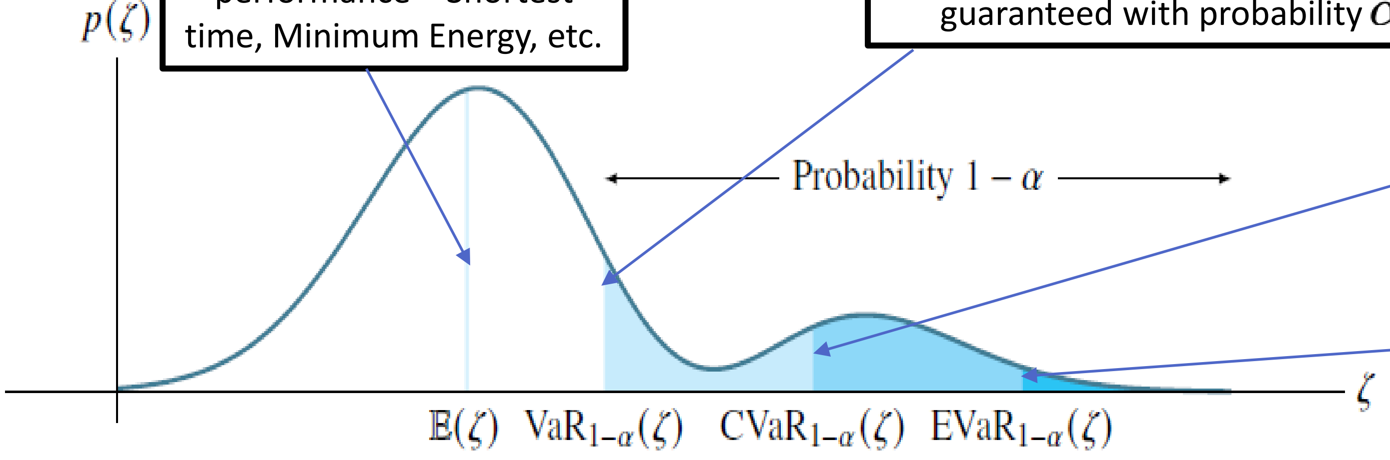
RISK MEASURES

Expected optimal performance – Shortest time, Minimum Energy, etc.

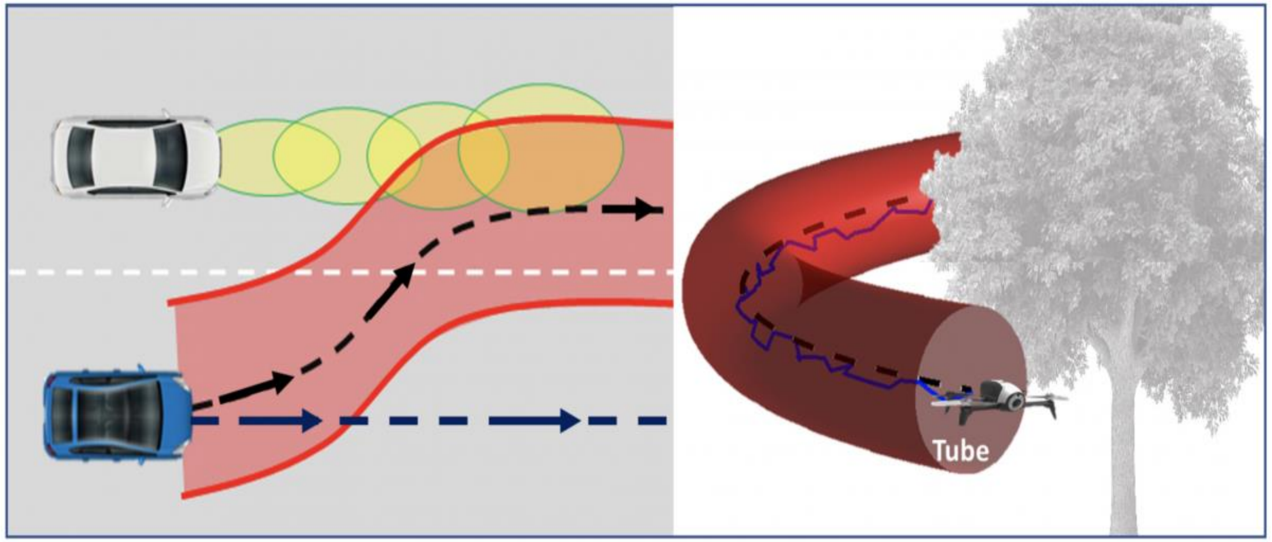
Best performance that can be guaranteed with probability α

Guarantees that the actual performance will **fall short of** the expected performance only $1-\alpha$ of the time

Allows more flexibility with α



α = Confidence level. 90 – 95%, typically.

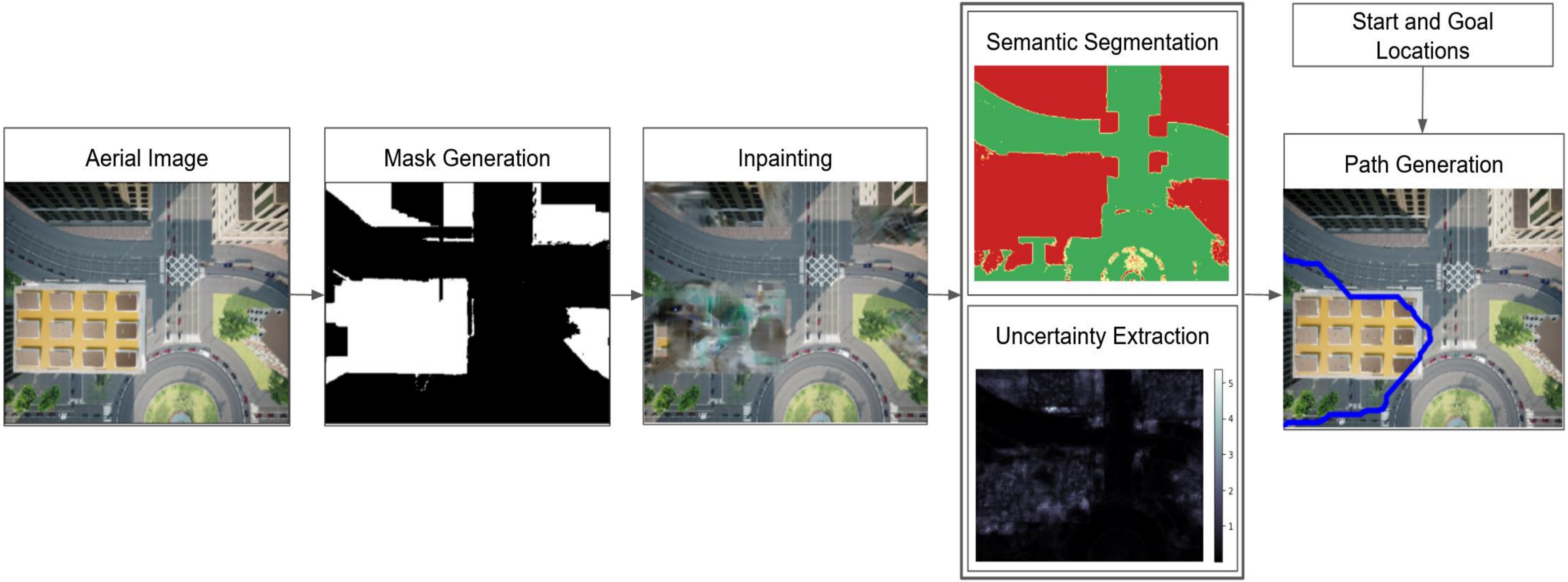


VaR : Value-at-Risk

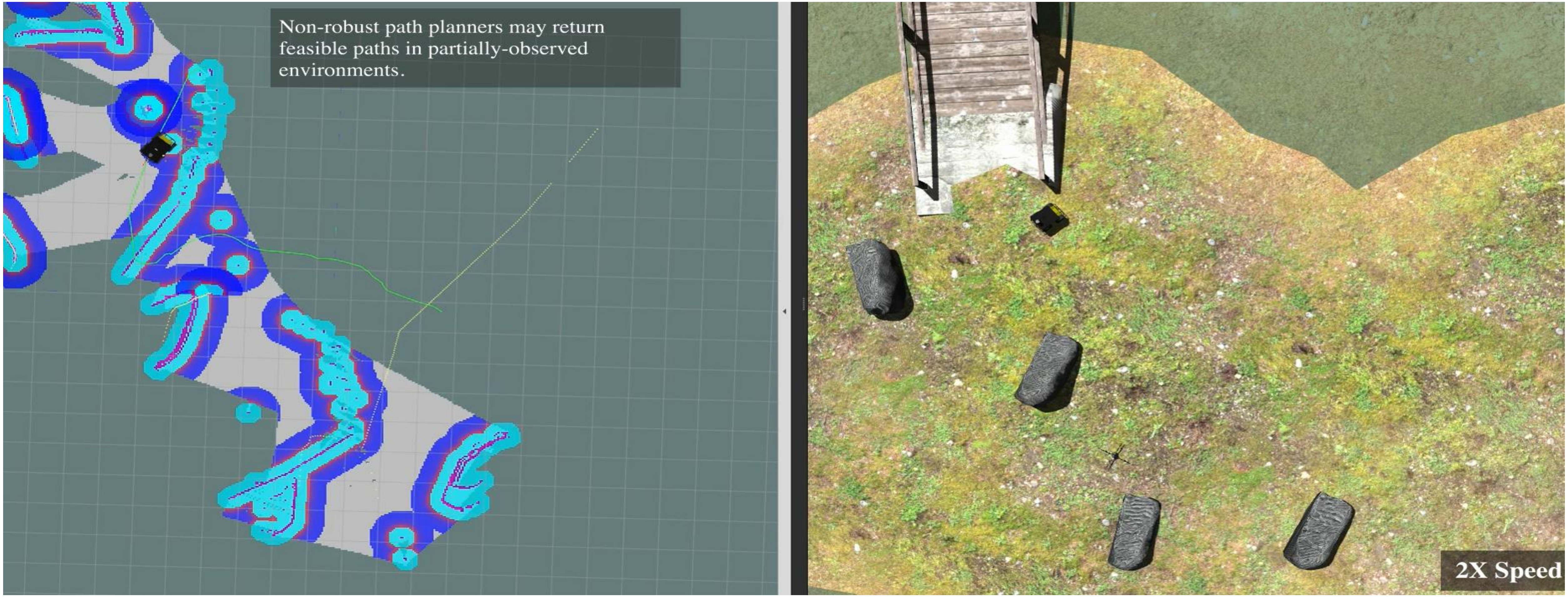
CVaR: **Conditional** Value-at-Risk

EVaR: **Expected** Value-at-Risk

Help from Aerial Images – even noisy ones
Multiple scales from sensor frequency tuning



Motivating the Need for Robust Path Planning



- Here, the UGV is planning its path using non-robust planners: **Dijkstra's algorithm** (Global) and the **Dynamic Window Approach** (local).

The Institute for Systems Research

Robust Path Planning via Risk Sensitivity in Partially-Observed Environments (AGV, AGV-UAV collaboration, AGV-UAV teams)

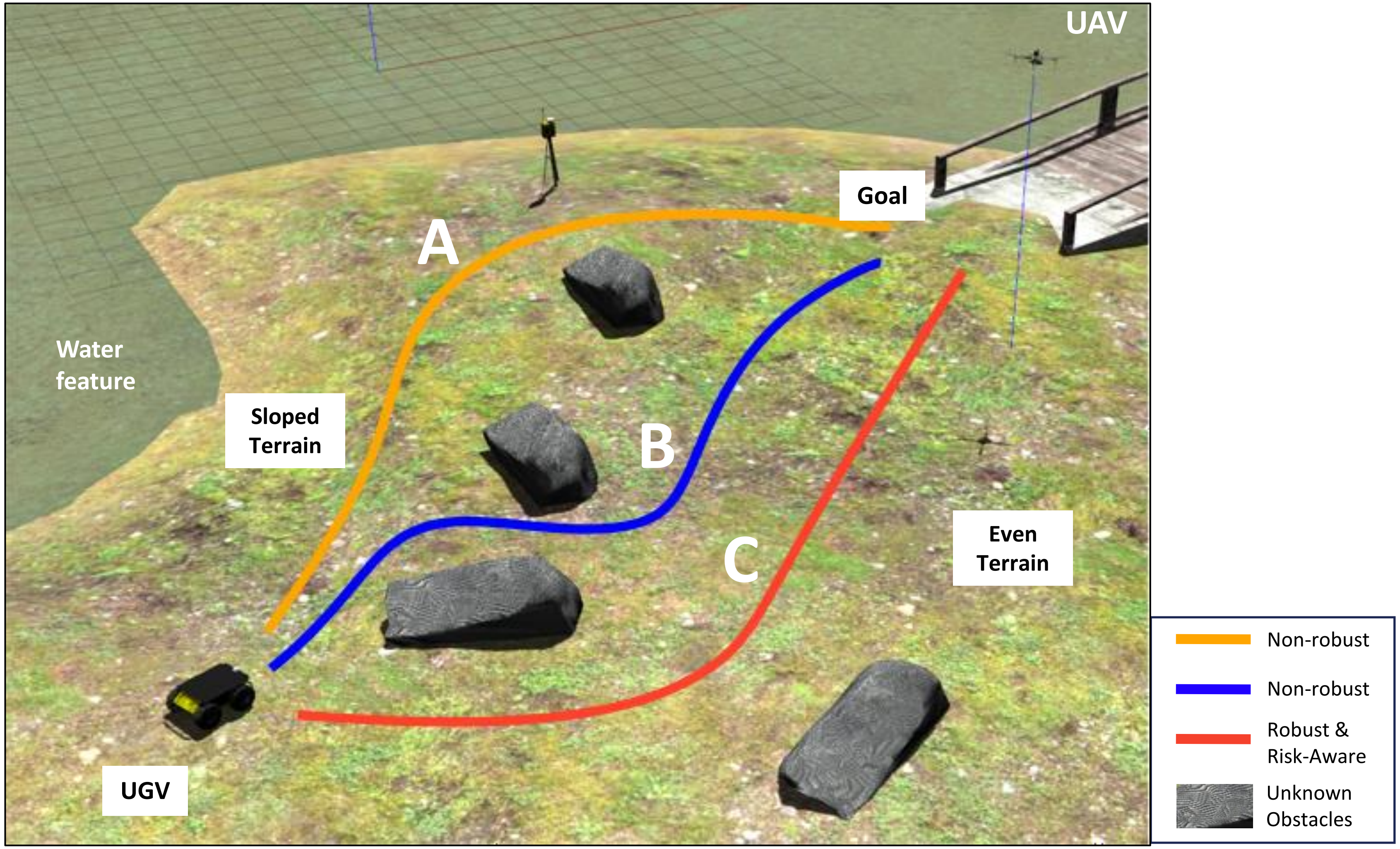
Technical Challenges

- **Dynamic** environments and **high-dimensional** state-action spaces make online path planning challenging.

- Sampling-based techniques require **rollouts** of prohibitively **many trajectories (or one single and very long trajectory)** to guarantee (an often **slow**) convergence to an optimal plan.

- Generated paths may be traversable but **non-robust**, e.g.,
  - **Path A**: robot *falls into water feature* en route to goal due to steep terrain slope.
  - **Path B**: robot *crashes into obstacles*.

- Uncertainty quantification may be far too conservative or imprecise for real-world perturbations.

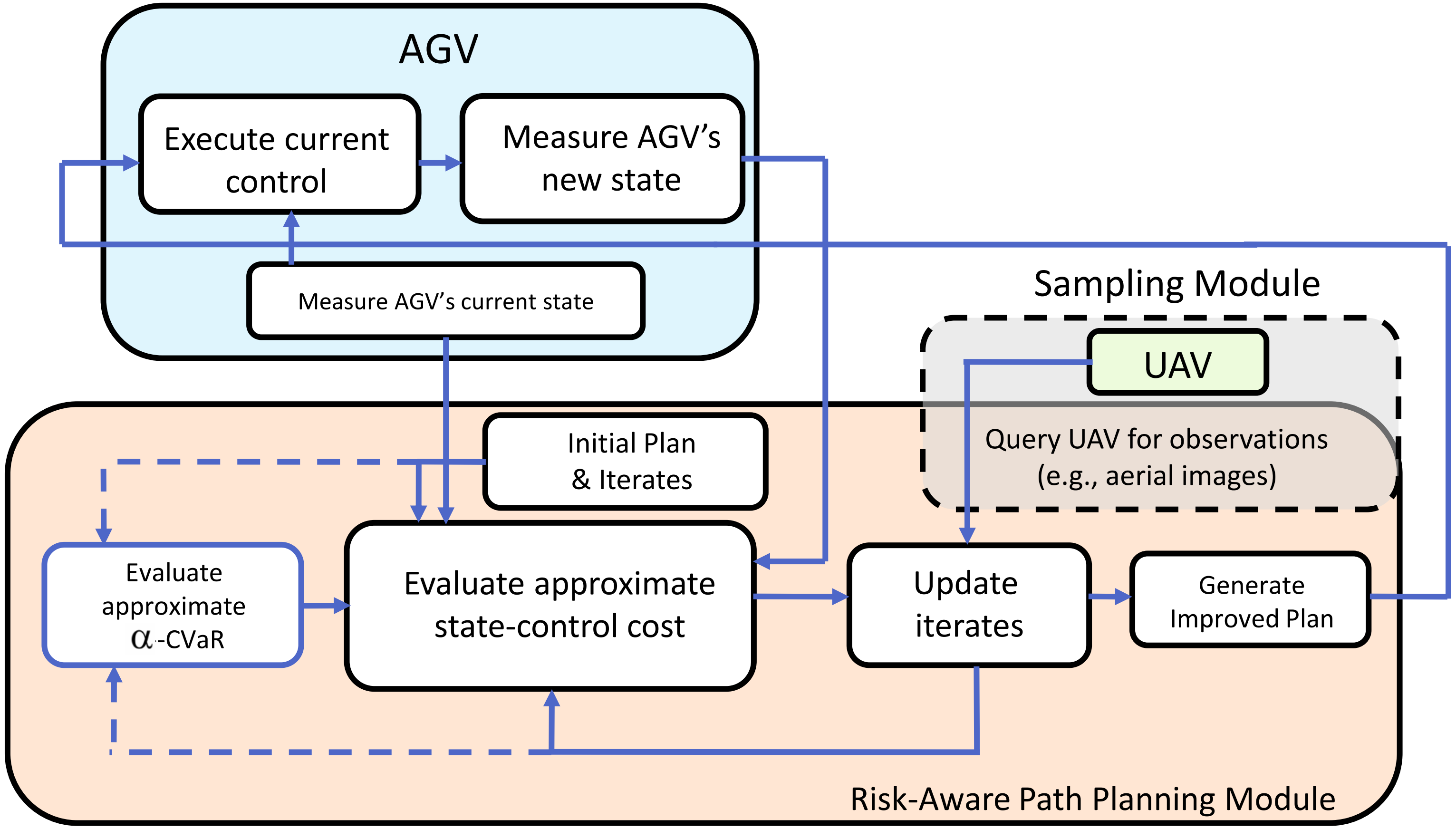


Motivating Example

Robust Path Planning via Risk Sensitivity in Partially-Observed Environments (AGV, AGV-UAV collaboration, AGV-UAV teams)

Approach:

- Adopt **function approximation** of state-control cost using **noisy real-time samples (local and from UAV)**
- Update cost approximation using estimated future cost via **stochastic gradient descent**
- **Efficient sampling** of risky regions
  - Sample from **risky regions** as the planning algorithm progresses using **importance sampling**
  - Importance sampling --- Use **regression** and **parametric cost approximation** to learn **minimum-risk** sampling distribution
- Efficient sampling **with** and **without tunable** risk levels (α).



Robust Collaborative Path Planning via Risk Sensitivity

Robust Path Planning via Risk Sensitivity in Partially-Observed Environments (AGV, AGV-AAV collaboration, AGV-AAV teams)

Ongoing Work: Robust Path Planning via Risk-Sensitivity – Problem Formulation

Given: AGV's initial pose (x_0), a desired goal location (x^{goal}), an initial costmap, and a finite-length rollout of

A possibly inaccurate state (x_t), control ($u_t = [u_{1,t}, u_{2,t}]^T \in \mathcal{U}_t(x_t)$), and costs ($q, h$):

$$\boxed{(x_t, u_{1,t}, u_{2,t}, \psi_t, q, x_{t+1}, \ldots, h)}$$

Find: A time ($\mathrm{T}_\Sigma < \infty$) and a policy (π):

$$\boxed{\pi : [0, \mathrm{T}_\Sigma] \to \{u_t, 0 \le t \le \mathrm{T}_\Sigma \mid u_t \in \boxed{\mathcal{U}_t}(x_t)\}}$$

Space of admissible control inputs

Obstacle-free configuration space of AGV

So that: $\square[0, \mathrm{T}_\Sigma]\phi$, where ϕ represents the logical formula: $x : [0, \mathrm{T}_\Sigma] \to \boxed{\mathcal{X}} \subset \boxed{\mathcal{C}_{free}} \models \bar{B}_\epsilon(x^{\text{goal}})$

AGV state space

Robust Path Planning via Risk Sensitivity in Partially-Observed Environments (AGV, AGV-AAV collaboration, AGV-AAV teams)

Ongoing Work: Robust Path Planning via Risk-Sensitivity – Problem Formulation (Optimization)

The AGV path planning problem can be re-expressed as the following optimization problem:

Optimization Problem

$$\min_{\pi} \boxed{\mathop{\mathbb{E}}_{\psi_t}[J_\pi \mid \psi_t]}$$

Expected Cumulative Cost (with risk measure)

$$t = 0, 1, \ldots, \mathrm{T}_\Sigma - 1$$

$$\text{s.t.} \quad (x_t, u_{1,t}, u_{2,t}, \psi_t, q, x_{t+1}, \ldots, h)$$

$$\boxed{\psi_t \sim \Pi(\cdot | x_t, u_{1,t}, u_{2,t}, z_t),}$$

Random Process representing AGV Sensor Noise

Penalty Functional

$$J_\pi = \sum_{t=0}^{\mathrm{T}_\Sigma - 1} \underbrace{\boxed{q(x_t, u_{1,t}, u_{2,t})}}_{\text{Stage Cost}} + \underbrace{\boxed{h(x_{\mathrm{T}_\Sigma})}}_{\text{Terminal cost}}$$

Robust Path Planning via Risk Sensitivity in Partially-Observed Environments (AGV, AGV-AAV collaboration, AGV-AAV teams)

Ongoing Work: Robust Path Planning via Risk-Sensitivity – Problem Formulation (Optimization)

The AGV path planning problem can be re-expressed as the following optimization problem:

Penalty Functional

Stage Cost

Terminal cost

$$\boxed{J_\pi = \sum_{t=0}^{T_\Sigma - 1} \boxed{q(x_t, u_{1,t}, u_{2,t})} + \boxed{h(x_{T_\Sigma})}}$$

$$\boxed{||u_{1,t}||^2 + ||u_{2,t}||^2 + ||x_t - x^{\text{goal}}||^2 - c_{\text{coll}}}$$

$$\boxed{||x_{T_\Sigma} - x^{\text{goal}}||^2 \cdot \mathbb{1}\left\{x_{T_\Sigma} \in \bar{B}_\epsilon\left(x^{\text{goal}}\right)\right\}}$$

Robust Path Planning via Risk Sensitivity in Partially-Observed Environments (AGV, AGV-AAV collaboration, AGV-AAV teams)

Ongoing Work: Robust Path Planning via Risk Sensitivity – Optimization Problem Formulation

$$
\begin{aligned}
\underset{x(\cdot),u(\cdot),t_f}{\text{minimize}} \quad & \boxed{\mathbb{E}[J] + \lambda\,\mathrm{CVaR}^{\alpha}[J]} \\
\text{subject to} \quad & x(t_i) = x_{\text{init}}\,, \quad x(t_f) = x_{\text{goal}} \\
& x(t+\Delta t) = f(x(t), u(t)), && t \in [t_i, t_f] \\
& x(t) \in \mathcal{X}_{\text{free}}(t) \cap \mathcal{X}_{\text{valid}}\,, && t \in [t_i, t_f] \\
& u(t) \in \mathcal{U}, && t \in [t_i, t_f]\,.
\end{aligned}
$$

CVaR adds robustness to the optimization by considering worst-case scenarios

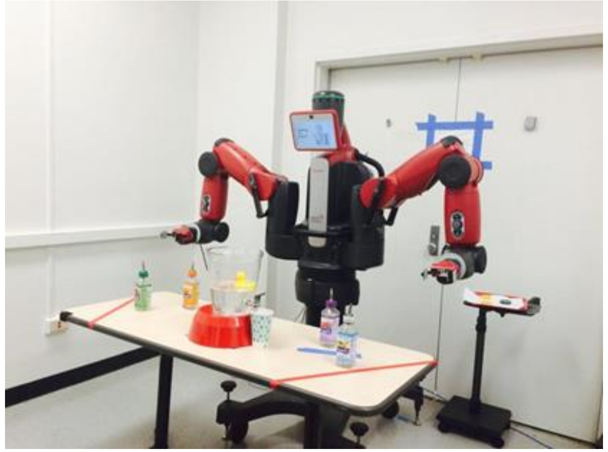
$$
\boxed{J = k\,(t_f - t_i) + \sum_{t=t_i}^{t_f} \left[u(t)^T R u(t) \right]}
$$

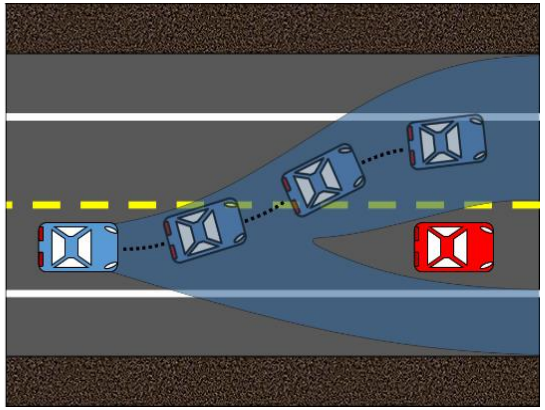
Performance measure = time taken to reach goal, path length, energy, etc.

Additions to our IDDMBSE Framework: Temporal Logic, Robots, Human-Robot Teams

Finite time logical constraints arise due to:

- Task description
- Decision making process
- Inherent inter–system interactions
- Other (a)causal dependencies



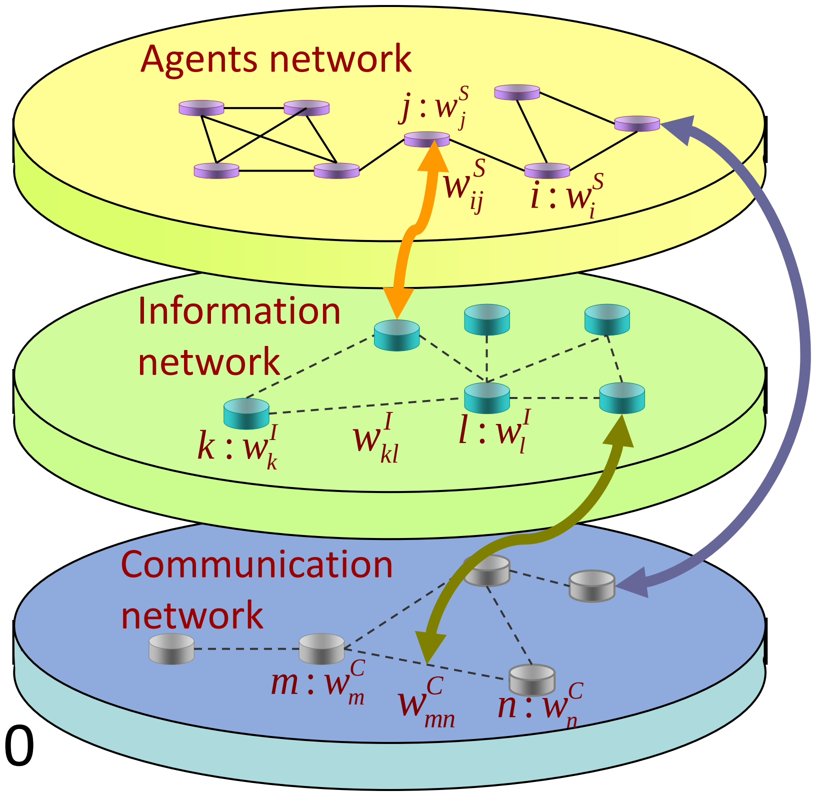


Constraints:

- Safety
- Human involvement
- Physical limitation

Multi-Agent Autonomous Systems: Multiple Coevolving Multigraphs

- Multiple Interacting Graphs
  - *Nodes*: agents, individuals, groups
  - Directed graphs
  - *Links*: ties, relationships
  - Weights on links : value (strength, significance) of tie
  - Weights on nodes : importance of node (agent)
- **Real-life problems: Dynamic, time varying graphs, relations, weights, policies**



- We introduced these models -- 2010
- Used them recently to model Net-CPS, Net-CHPS
- Investigated effects of topology: proved Small World Graphs speed up consensus (probabilistic argument)

Future Research Directions

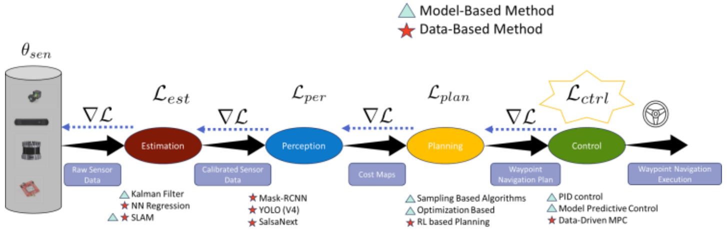
Integrate SysML with ROS

- Use the Functional Mock-up Interface (FMI). A standard for dynamical model-exchange and co-simulation.
- `fmi_adapter`, ROS package by Bosch. Supports co-simulation of FMUs from different tools as ROS Nodes.
- Use **Web Server for Cameo Simulation Toolkit** plug-in to build a SysML-ROS bridge for real-time message passing.
- Develop the framework into a functional software tool.

Develop New IDDMBSE Tools

- Physics-based models of components in **Dymola** using its extensive model libraries.
- Robotic and Autonomy Platform Simulators such as **Nvidia Isaac Sim** for high-fidelity Digital Twins.
- **Data Distribution Service (DDS)** based ROS2 implementation of a functional stack to exploit new capabilities of ROS2.
- **SysML v2** based implementation of the IDDMBSE framework.
- Extend **SCOUPE** for Trusted Autonomy. An earlier software by Prof. Baras enabling Computer-Assisted Generation of Activity Diagrams from Textual Scenarios. That is **from text to correct SysML**. Facilitating learning and use of SysML.

IDDMBSE Theory and Applications



- *Differentiable autonomy* pipeline with hybrid modules for modularity-preserving e2e learning.
- RL for *balancing risk and opportunity in the design* of autonomous systems by jointly learning the optimal design and optimal policy.
- *Data-Driven augmentation of physical* models of wheel-terrain interaction using sensor data to improve path planning/following performance.
- *Adaptive terramechanical design* of autonomous ground robots by optimizing wheel geometry and contact sensing pad for various terrain conditions.
- *IDDMBSE framework for multiple collaborating autonomous systems*. Start with a ground vehicle and an air vehicle. Investigate safety, robustness.

Thank you!

baras@umd.edu

301-405-6606

https://johnbaras.com/

Questions?