Rigorous Modeling and Validation of Autonomous Driving Systems

International Symposium on
the Verification of Autonomous Mobile Systems

Palaiseau, March 9, 2023

Joseph Sifakis
Verimag Laboratory

ADS – Autonomous systems Main Characteristics

Autonomous systems are essential for reaching the Industrial IoT vision.

- ❑ They emerge from the needs to further automate existing organizations by progressive and incremental replacement of human operators by autonomous agents.
- ❑ They are very different from game-playing robots or intelligent personal assistants.
- ❑ They are often critical and should exhibit "broad intelligence" by handling knowledge in order to
  - Manage dynamically changing sets of possibly conflicting goals;
  - Cope with uncertainty of complex, unpredictable cyber physical environments;
  - Harmoniously collaborate with human agents e.g. "symbiotic" autonomy.

Autonomy is not just a Q&A game, it is a bold step from weak AI toward AGI.

Two different technical avenues both falling short of the ADS challenge:

- traditional model-based critical systems engineering, successfully applied to aircraft and production systems, proves to be inadequate.
- industrial end-to-end AI-enabled solutions currently available fail to provide the required strong trustworthiness guarantee.

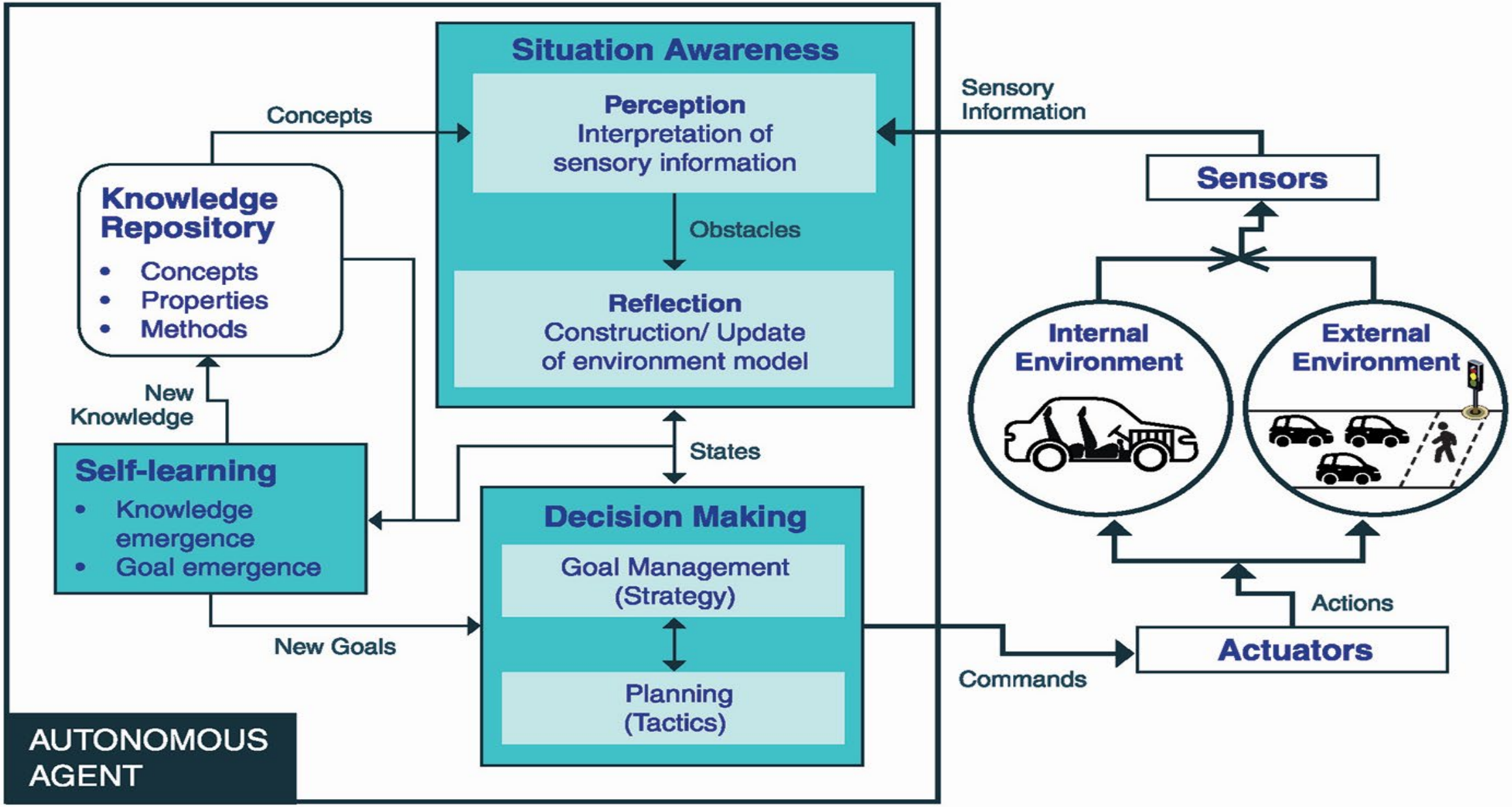
ADS – From Traditional to Autonomous Systems Engineering

	Traditional Systems Engineering			Autonomous Systems Engineering		
Design Methodology	Model-based -- V-model			Hybrid (Model-based + Data-based) design		
Components	Reactive with predictable environment			Heterogeneous components e.g. cyber-physical, AI		
Architectures	Static, Centralized, usually hierarchical			Time and Space Dynamism, Decentralized, Self-organization		
Requirements	Elicitation before design			Progressive elicitation		
Validation	Formal Methods supported by tools e.g. MC			Application of Formal Methods limited to model-based components		
	Possibly by simulation and testing			Mainly by Simulation and testing (we need theory and methodology)		
Correctness	At design time			At runtime using monitoring techniques		
	Domain- specific standards – conclusive evidence			No standards!! – at best statistical evidence		
Modeling	Specific techniques. Matlab/Simulink, Automata-based techniques			New techniques: Cyber physical systems, knowledge based systems,		
	UML, SysML , AADL standards			New modelling methodologies supporting dynamism and reconfiguration		
SAFETY	Fault analysis	Fault Detection Isolation Recovery	Evaluation Reliability theory	Fault analysis	Fault Detection Isolation Recovery	Reliability theory
DAFETY& SECURITY				S&S risk analysis	Runtime Assurance techniques	Model-based and Experimental Evaluation
SECURITY	Attack analysis	Attack Detection Isolation	Evaluation – standards EAL	Attack analysis	Attack Detection Isolation	Evaluation – standards EAL

OVERVIEW

- Autonomous Systems
- Autopilot Design
- Global ADS Validation
- Discussion

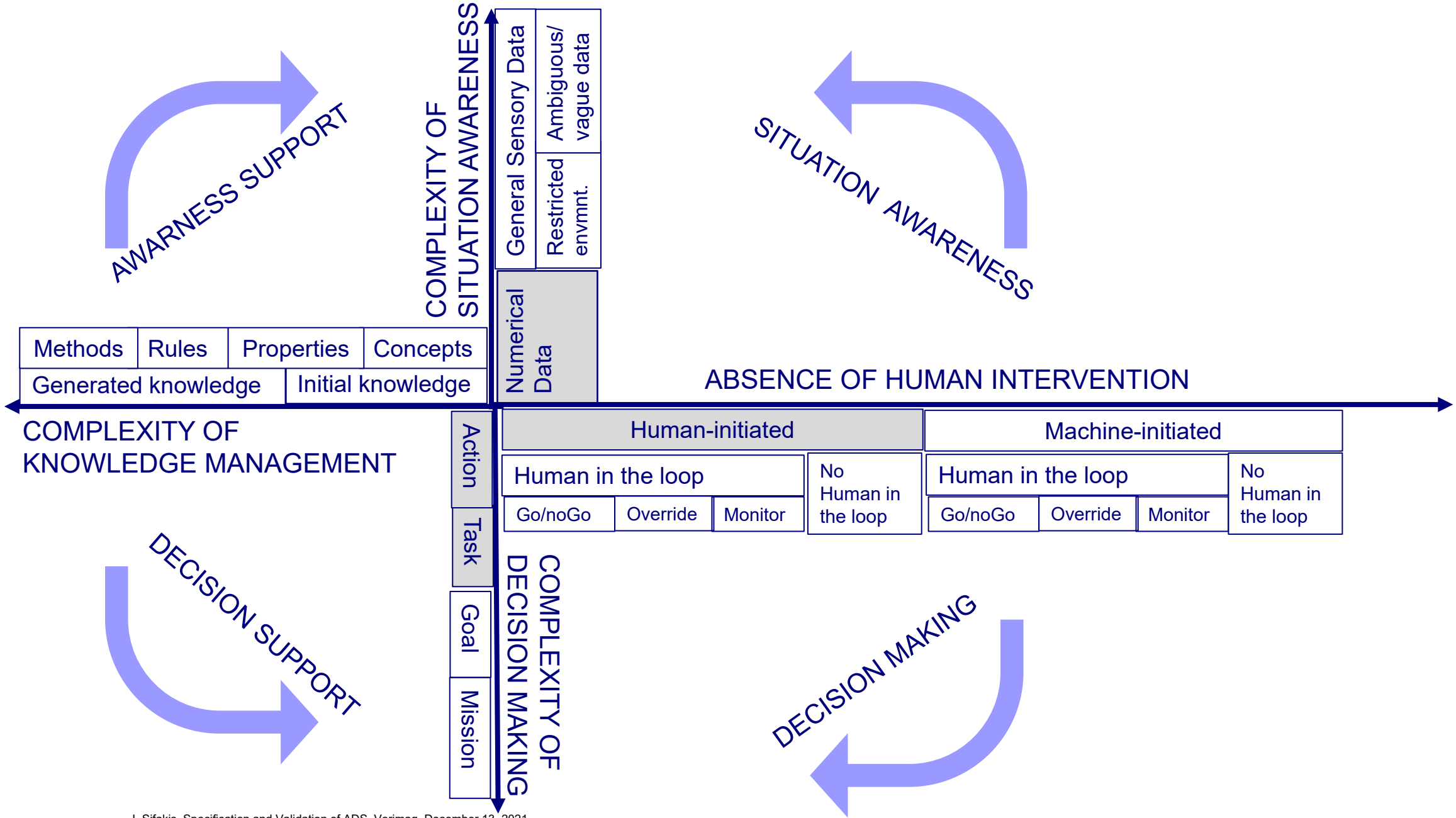
Autonomous Systems – Agent Architecture



Autonomous Systems – From Automation to Autonomy

	SAE AUTONOMY LEVELS	
Level 0	No automation	AUTOMATION (ADAS)
Level 1	Driver assistance required	
	The driver still needs to maintain full situational awareness and control of the vehicle e.g. cruise control.	
Level 2	Partial automation options available	
	Autopilot manages both speed and steering under certain conditions, e.g. highway driving.	
Level 3	Supervised Autonomy	AUTONOMY (ADS)
	The car, rather than the driver, takes over actively monitoring the environment when the system is engaged. However, human drivers must be prepared to respond to a "request to intervene"	
Level 4	Geofence autonomy	
	Self driving is supported only in limited areas or under special circumstances, like traffic jams	
Level 5	Full autonomy	
	No human intervention is required e.g. a robotic taxi	

Autonomous Systems – From Automation to Autonomy (2)

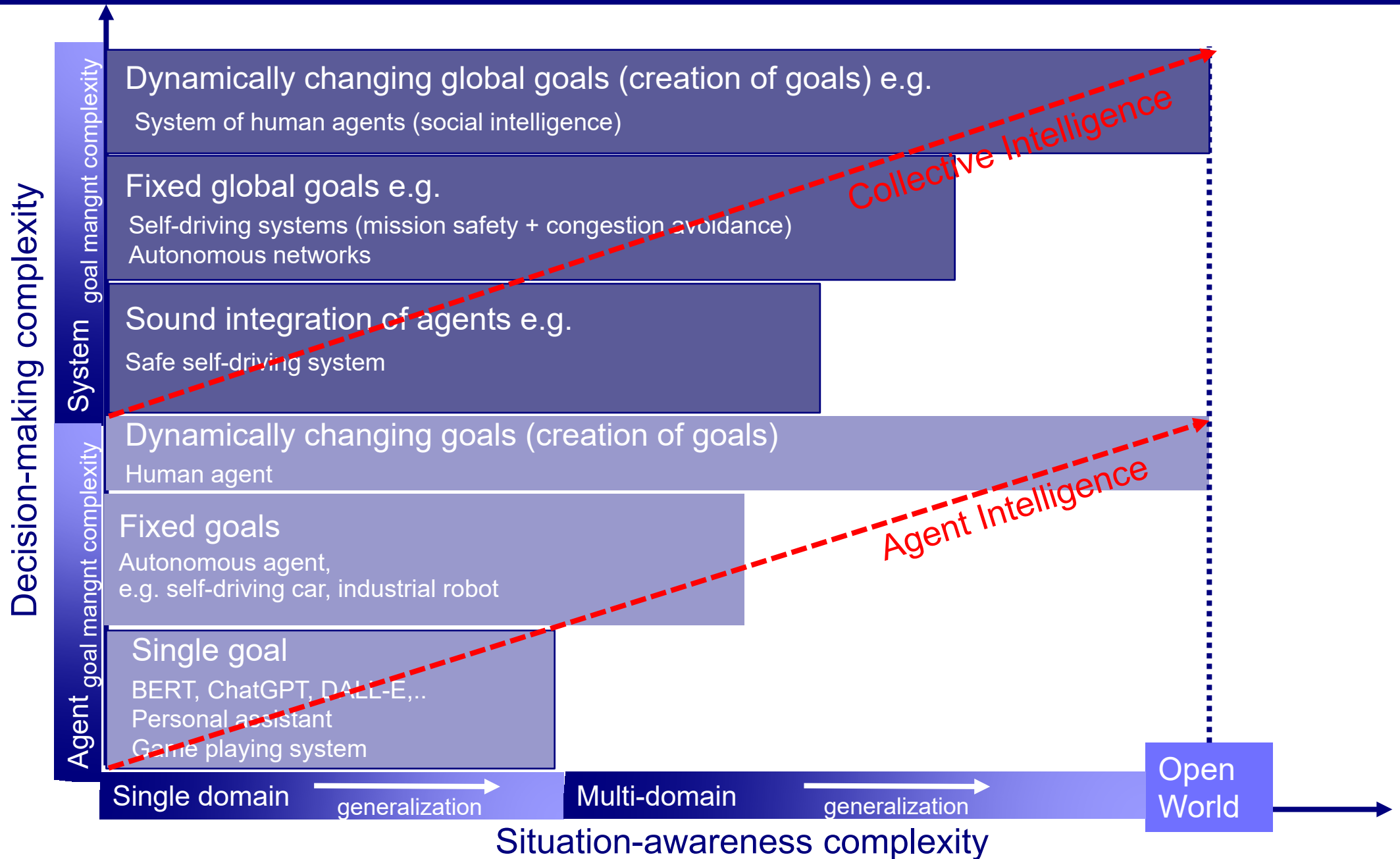


Autonomous Systems – Complexity Issues

- Autonomous agents rely on computational intelligence to overcome complexity limitations
  - Complexity of perception due to the difficulty to interpret stimuli (cope with ambiguity, vagueness) and to timely generate corresponding inputs for the agent environment model.
  - Complexity of uncertainty due to situations involving imperfect or unknown information implying lack of predictability about the environment such as dynamic change caused by physical or human processes, rare events, critical events such as failures and attacks.
  - Complexity of decision reflected in the complexity of the agent's decision process (goal management and planning) and impacted by factors such as diversity of goals and size of the space of solutions for planning.

- However, building autonomous systems involves difficult systems engineering problems that are not related to the fact that agents are intelligent -- problems that could explain the setbacks of autonomous car industry,
  - System agents should be
    - integrated in complex cyber physical environments systems e.g. electromechanical systems
    - be able to harmoniously collaborate with human operators – It's not just an HMI problem!
  - System agents should be adequately coordinated to achieve
    - Symbiosis: the coordination of agents does not impede the achievement of their individual goals
    - Synergy: agents collaborate to achieve global system goals by demonstrating collective intelligence.

Autonomous Systems – From Agent Intelligence to Collective Intelligence



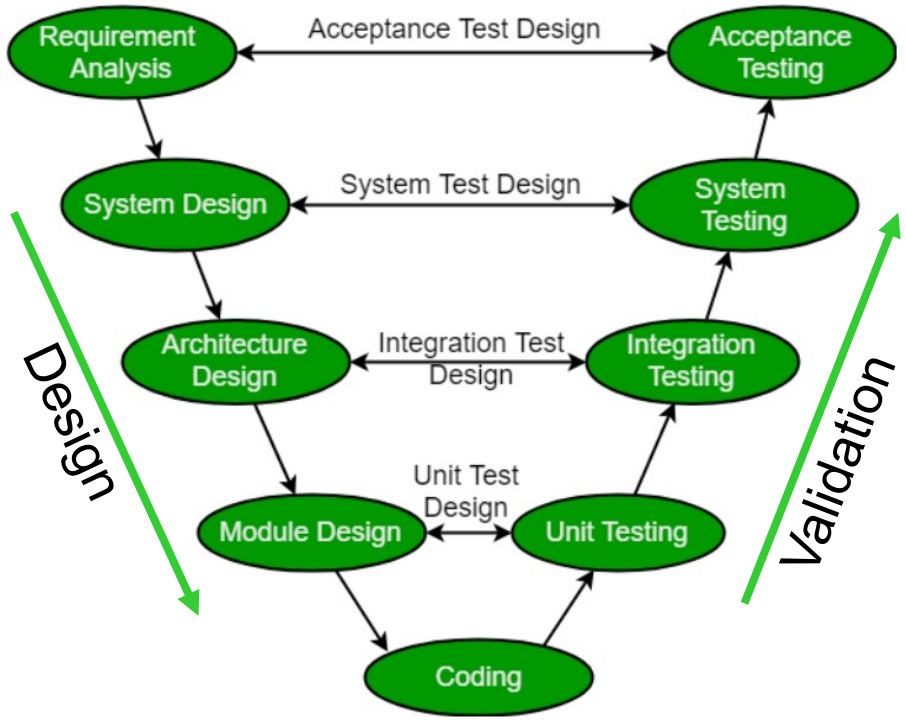
OVERVIEW

- Autonomous Systems
- Autopilot Design
- Global ADS Validation
- Discussion

Autopilot Design – Critical Systems Engineering Limitations

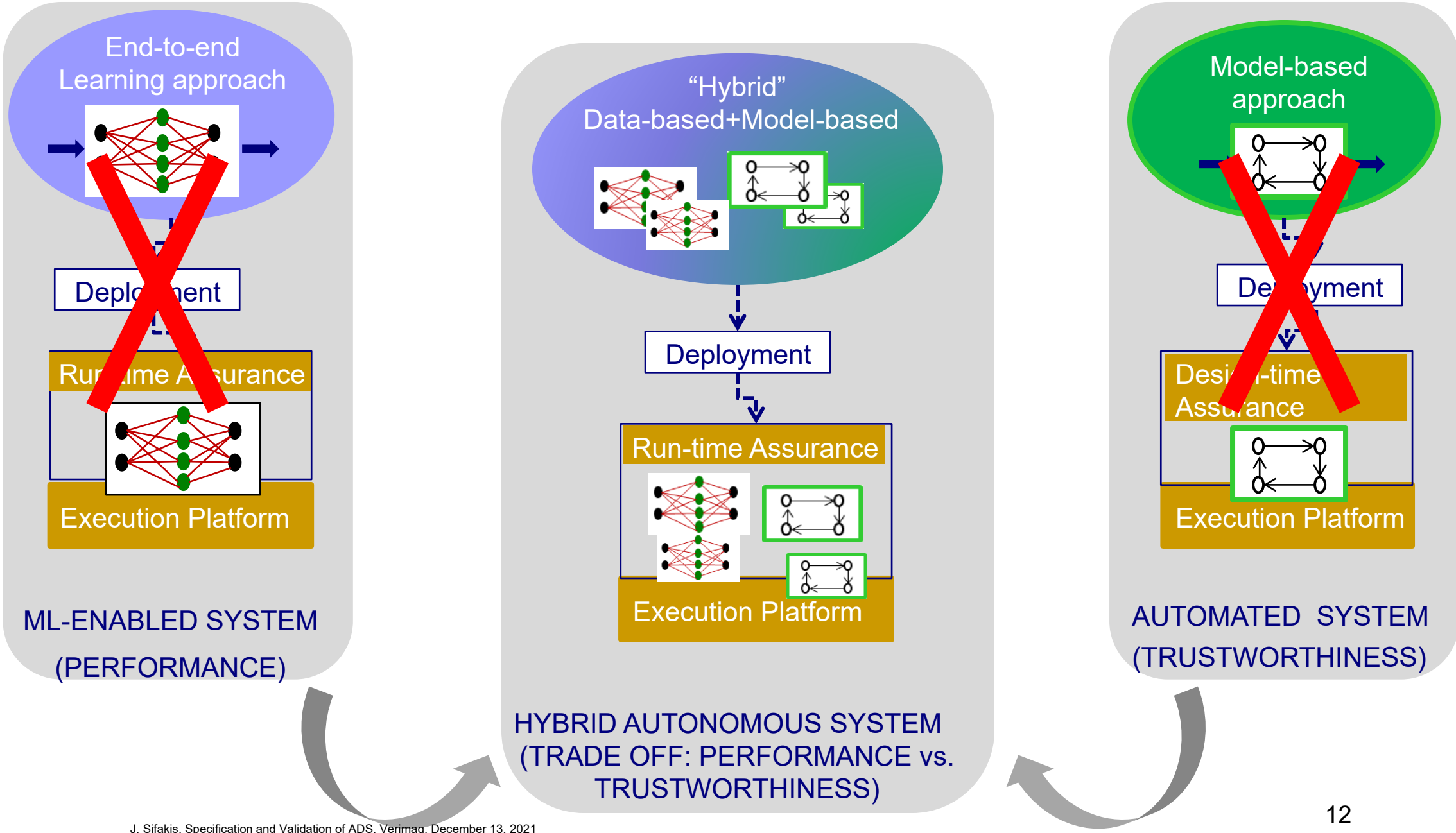
Critical systems design flows follow model-based prescriptive frameworks recommended by standards e.g. ISO26262

- Assume that system development is top-down and validation is bottom-up.
- Assume that all requirements are initially known, can be clearly formulated and understood.
- Consider that global system requirements can be broken down into requirements satisfied by system components.
- Focus on providing model-based conclusive evidence that the system is safe e.g.
  10^{-9} failures per hour of flight



- The model-based paradigm is defeated by the overwhelming complexity and diversity of autonomous systems
- This explains the adoption by industry of end-to-end machine-learning-enabled techniques which however preclude conclusive safety guarantees

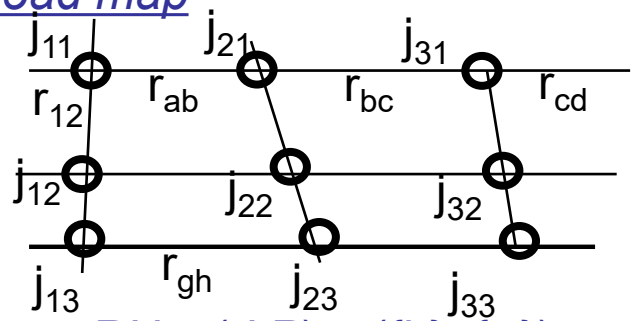
Autopilot Design –Taking the Best from Each



Autopilot Design – Why We Need Maps?

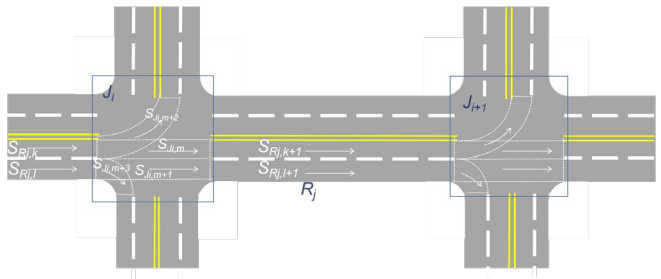
Maps

Road map



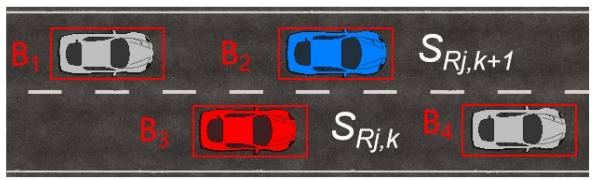
$RN = (J,R) = (\{j_k\}, \{r_s\})$

Lane map



$SN = (S,E)$

Synthesized local map



Hierarchical Autopilot

Mission Planning – Layer 4

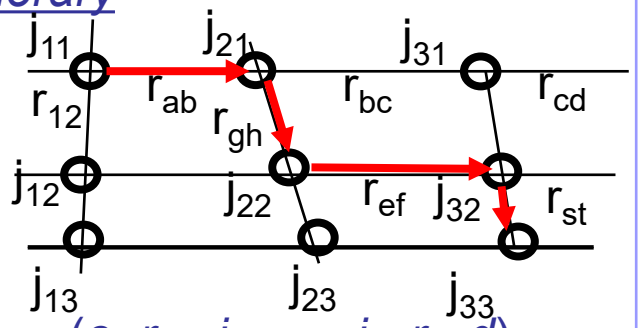
Path Planning - Layer 3

Maneuver Planning & protocols- Layer 2

Trajectory Planning - Layer 1

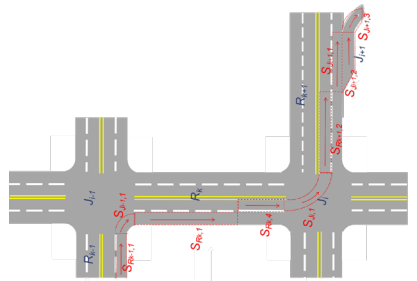
Goals

Itinerary



$(o, r_{ab}, j_{21}, \ldots j_k, r_k, d)$

Path

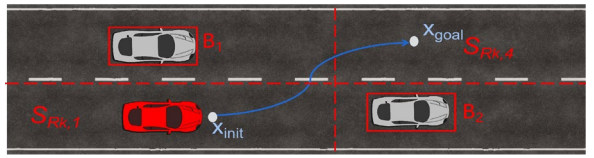


$(s_{Rk-1,1}, s_{Ji-1,1}, s_{Rk,1}, s_{Rk,4}, s_{Ji,1}, s_{Rk+1,2})$

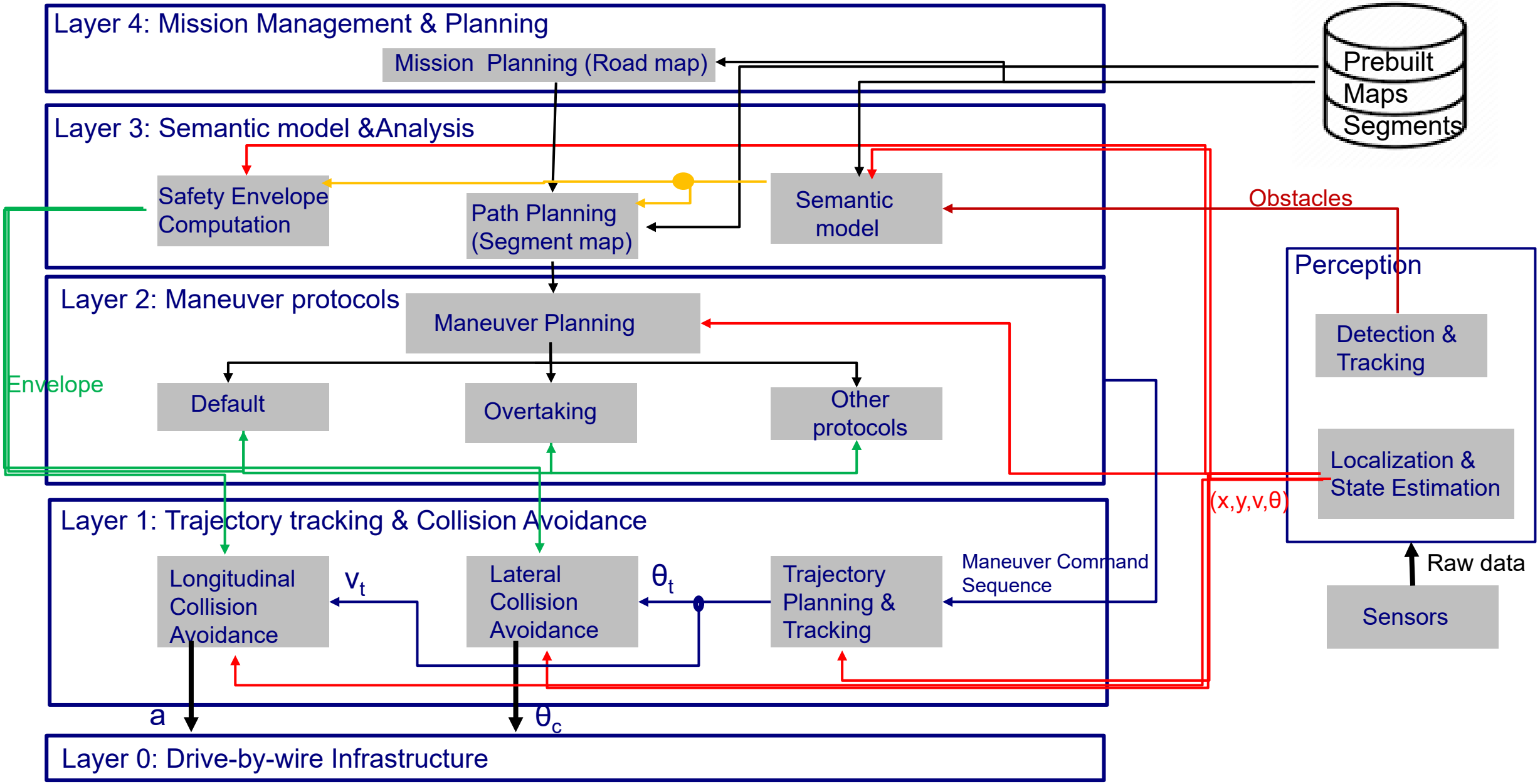
Maneuver sequences

$(lane\_change, s_{Rk,1}, s_{Rk,4})$

Trajectory



Autopilot Design – Hierarchical Autopilot Architecture



- Autonomous Systems
- Autopilot Design
- Global ADS Validation
- Discussion

OVERVIEW

Global ADS Validation – When an ADS is Safe Enough?

Waymo has now driven 10 billion autonomous miles in simulation

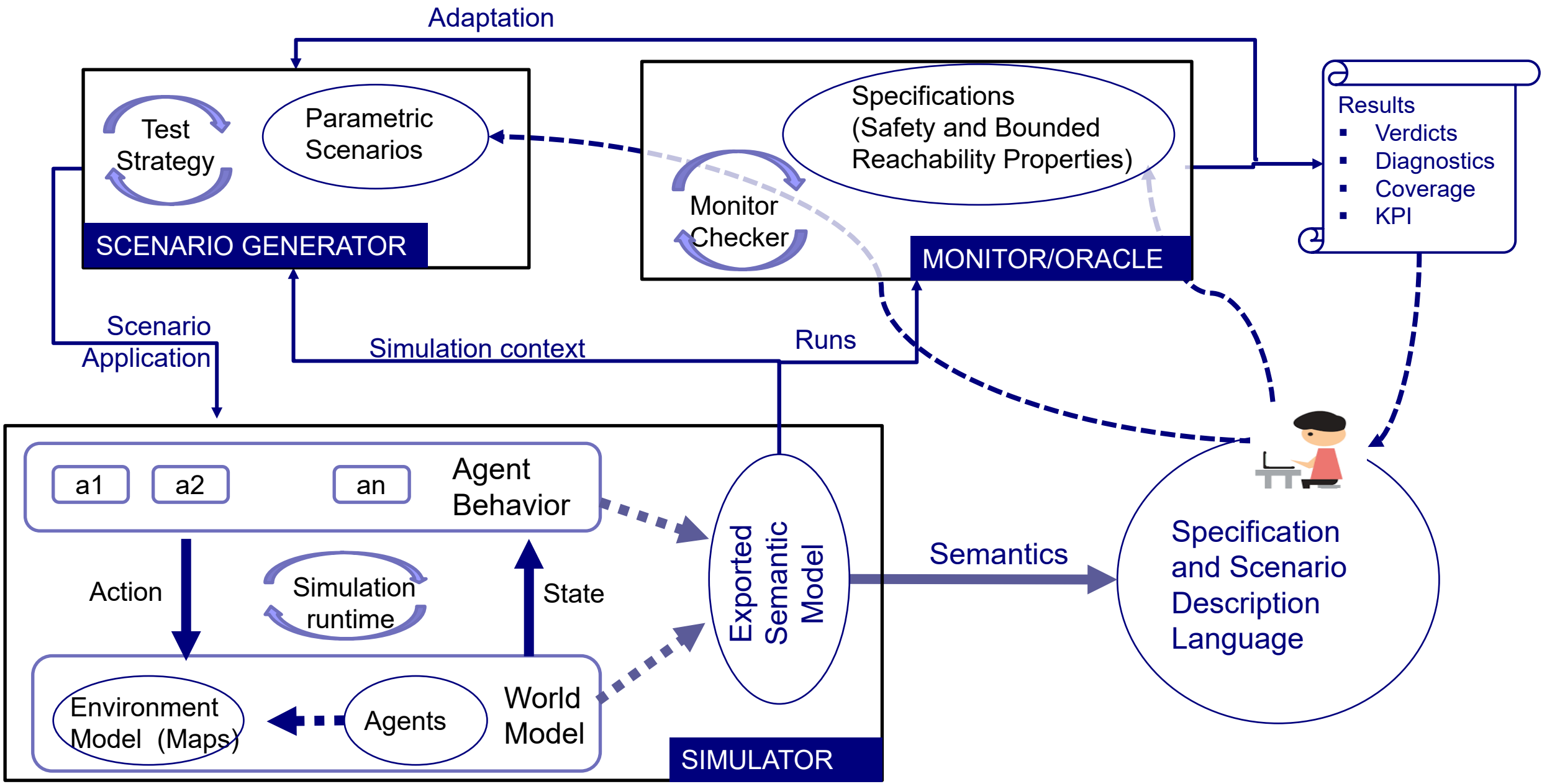
Darrell Etherington @etherington / 11:17 pm CEST • July 10, 2019

Comment

- The inability to build global system models limits system validation to simulation and testing.
  - Simple simulation is not enough - how a simulated mile is related to a "real mile" ?
  - We need evidence, based on coverage criteria, that the simulation deals fairly with the many different situations, e.g., different road types, traffic conditions, weather conditions, etc.

- Testing theory to calculate, on the basis of statistical analysis, confidence levels for given properties.
  - Sampling theory: methods for building sample scenarios that adequately cover real-life situations
  - Repeatability: for two samples of scenarios with the same degree of coverage, the estimated confidence levels are approximately the same.

Global ADS Validation – The Big Picture



Global ADS Validation – Simulation Key Issues

- Whatever design approach is taken, simulation is of paramount importance for validation – and raises a large variety of problems from purely technical to theoretical ones.
- Not only the appearance should be realistic but also it should be real: the execution mechanism should rely on a semantic model of the environment consistent with laws of Geometry and Physics.
- Note that realism and consistency with reality are hard to reconcile - simulation environments built on top of game engines lack semantic awareness.

1. Realism: agent behavior and environment look real in a way that is accurate or true to life.
2. Expressiveness: supports rigorous modeling language e.g. DSL for
   - component-based description of mobile agents and their dynamic coordination;
   - modelling of physical environment in which the agents operate (maps).
3. Semantic awareness: the simulated system dynamics is rooted in transition system semantics.
   - Notion of state allowing repeatability of experiments.
   - Distinguishing between controllable and uncontrollable actions.
   - Multiscale multigrain modeling of time scales and of their correlation with space scales.
4. Performance: run-time infrastructure federating simulation engines e.g. HLA, FMI.

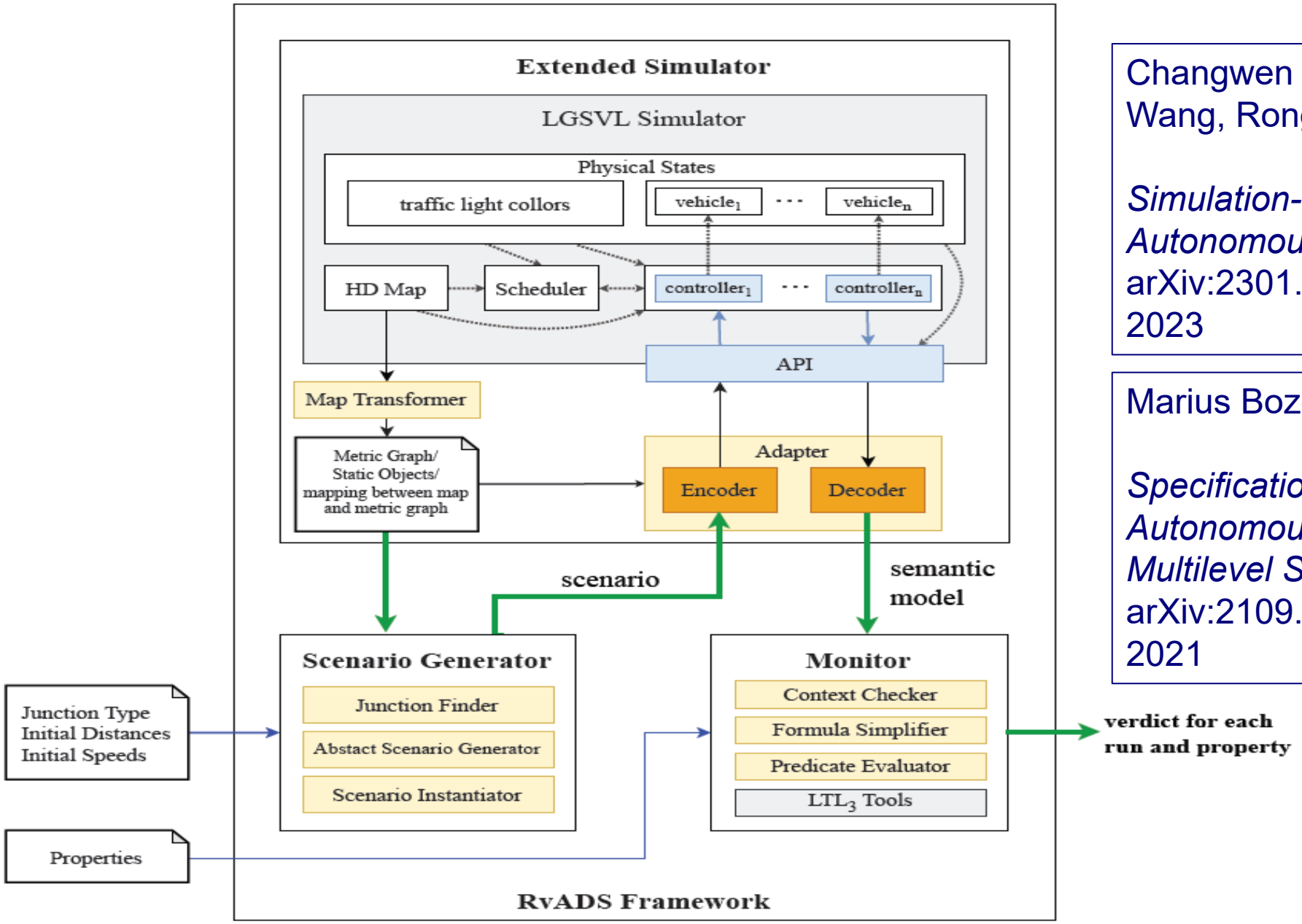
Global ADS Validation – Gaps in the State of the Art

- Validation should rely on model-based criteria defined on an implicit or an explicit system model.
  - Superficial quantitative criteria such as simulation hours, miles travelled, do not provide sufficient evidence of trustworthiness.
  - Any technically sound safety evaluation should be model-based providing evidence that simulation covers a good deal of the many and diverse situations specified by the system properties to be validated e.g. different types of roads, traffic conditions, weather conditions.

- Property specification languages supporting genericity (types of objects) and parametricity (quantification over domains) e.g. first or higher order temporal logics and associated runtime verification techniques.

- Scenario description languages for the controlled simulation of agents so as to explore situations based on
  - Coverage criteria measuring the degree to which relevant system configurations have been explored, as for structural testing of software systems;
  - Functional criteria to explore/detect corner cases and high risk situations, exactly as for functional testing software systems;
  - Metamorphic relations that define similarity relations between scenarios used by the Scenario Generator to cope with complexity of their space – similar scenarios should produce close enough responses;
  - Verdicts and diagnostics about the relationship between failures and various risk factors e.g. road structure, congestion level, weather and violations of traffic regulations.

Global ADS Validation – Simulation and Testing Environment



Changwen Li, Joseph Sifakis, Qiang Wang, Rongjie Yan and Jian Zhang

Simulation-based Validation for Autonomous Driving Systems, arXiv:2301.03941v1 [cs.SE], 10 Jan. 2023

Marius Bozga and Joseph Sifakis

Specification and Validation of Autonomous Driving Systems: A Multilevel Semantic Framework arXiv:2109.06478v1 [cs.MA] 14 Sep 2021

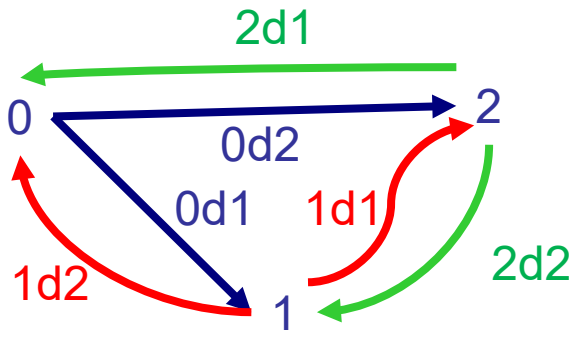
Global ADS Validation – Property Specification for Junctions (Traffic Rules)

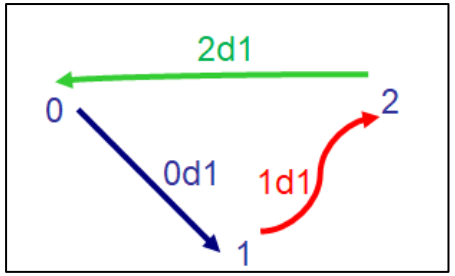
Properties for a stop junction j:
p_1: If a vehicle is in the junction, then no other vehicle can be in the junction:
$\forall a.\forall a'.\,\square[[a@j \wedge a \neq a'] \rightarrow \neg a'@j]$
p_2: If a vehicle arrives at the same time as another vehicle, the vehicle on the right has the right-of-way:
$\forall j.en.\forall j.en'.\forall a.\forall a'.\,\square[[a@j.en \wedge a'@j.en' \wedge a.wt = a.wt' \wedge j.en \texttt{ right-of } j.en'] \rightarrow [[\mathrm{N}\ a'@j.en']\ \mathrm{U}\ a@j]]$
p_3: The vehicle that arrives first at the entrance will pass before other vehicles:
$\forall j.en.\forall j.en'.\forall a.\forall a'.\square[[a@j.en \wedge a'@j.en' \wedge a.wt < a'.wt] \rightarrow [[\mathrm{N}\ a@j.en]\ \mathrm{U}\ a'@j]]$
Properties for a traffic light junction j:
p_4: Any vehicle facing a red light must stop until the traffic light turns green, unless the vehicle is turning right.
$\forall j.en.\forall a.\forall lt.\,\square[[a@j.en \wedge lt@j.en \wedge lt.cl = red \wedge \neg take(a, j.en, \texttt{right})] \rightarrow [a@j.en\ \mathrm{U}\ lt.cl = green]]$
p_5: If a vehicle facing a red light is turning right, then the vehicle should wait until there is no vehicle on the left.
$\forall j.en.\forall j.en'.\forall a.\forall a'.\forall lt.\,\square[[a@j.en \wedge a'@j.en' \wedge (j.en \texttt{ right-of } j.en') \wedge lt@j.en \wedge (lt.cl = red) \wedge take(a, j.en, \texttt{right})] \rightarrow [[\mathrm{N}\ a@j.en]\ \mathrm{U}\ a'@j]]$
p_6: If two vehicles arrive at the entrances of a junction opposite each other and the traffic lights are green, the vehicle turning left must give way to the other.
$\forall j.en.\forall j.en'.\forall a.\forall a'.\square[[a@j.en \wedge a'@j.en' \wedge j.en \texttt{ opposite } j.en' \wedge take(a, j.en, \texttt{left}) \wedge \neg take(a', j.en', \texttt{left})] \rightarrow [[\mathrm{N}\ a@j.en]\ \mathrm{U}\ a'@j]]$

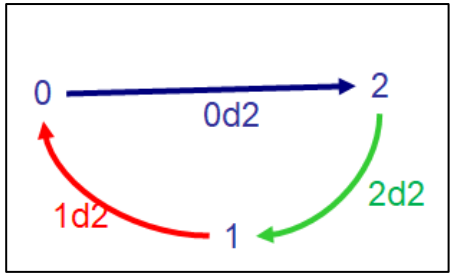
Table 2: Traffic rules and their formal specifications in linear temporal logic

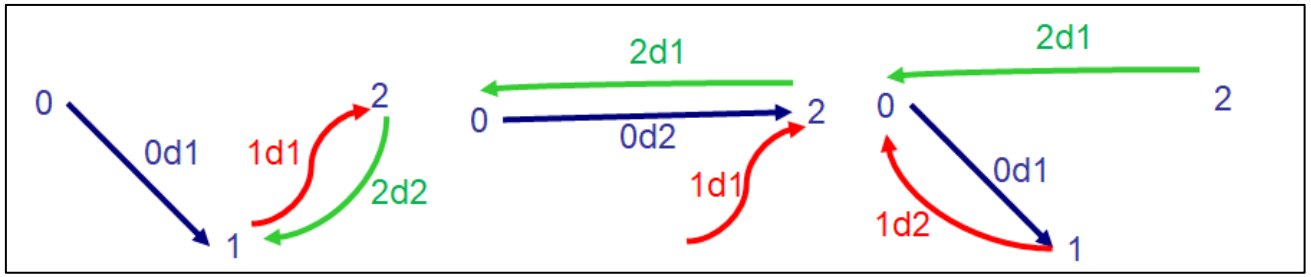
Global ADS Validation – Scenario-based Coverage of Junctions

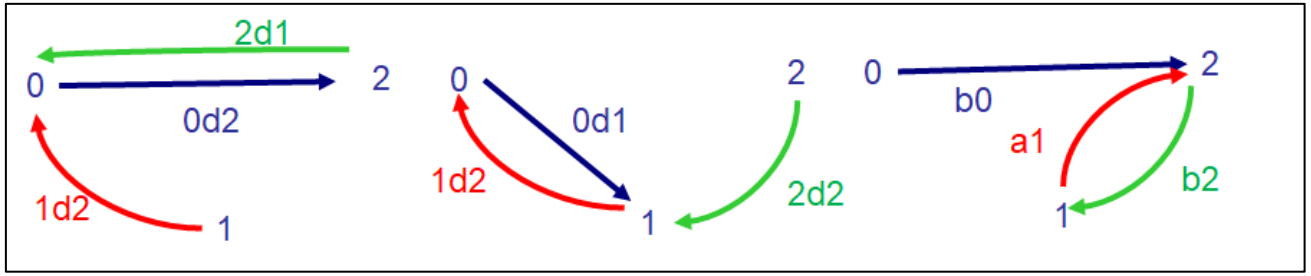
- Given a set of agents A ={a1,..,an}, a map G and a set of properties P to validate
  - a scenario sc= {<it1, v1>, …, <itn,vn>} and an abstract scenario asc={it1, .., itn}
  - sc1, sc2 are equivalent wrt P (sc1~$_P$ sc2) if the corresponding runs rn1, rn2: rn1|=p iff rn2|= p $\forall p \in P$
  - asc1, asc2 are equivalent wrt P if any extension by the same speed context gives sc1, sc2: sc1~$_P$ sc2











There are 2^3 different abstract scenarios in a 3-way junction obtained as the Cartesian product of

{0d1, 0d2}
{1d1, 1d2}
{2d1,2d2}

Global ADS Validation – Four-Way Stop Junction

#	class of abstract scenarios	distance (0.01,0.01,0.01,0.01) A: speed (0,0,0,0)			distance (0.3,0.3,0.3,0.3) B: speed (0,0,0,0)			distance (20,20,20,20) C: speed (0,0,0,0)			distance (20,20,20,20) D: speed (10,10,10,10)			distance (0.3,0.3,20,20) E: speed (0,0,0,0)		
		p_1	p_2	p_3	p_1	p_2	p_3	p_1	p_2	p_3	p_1	p_2	p_3	p_1	p_2	p_3
1	$0_{d_2}1_{d_1}2_{d_1}3_{d_1}$	Fail	Fail	NA	Fail	Fail	NA	Fail	Fail	NA	Fail	Fail	NA	Fail	Fail	Pass
	$0_{d_1}1_{d_2}2_{d_1}3_{d_1}$	Fail	Fail	NA	Fail	Fail	NA	Fail	Fail	NA	Fail	Fail	NA	Fail	Fail	Pass
	$0_{d_1}1_{d_1}2_{d_2}3_{d_1}$	Fail	Fail	NA	Fail	Fail	NA	Pass	Fail	NA	Pass	Fail	NA	Pass	Fail	Pass
	$0_{d_1}1_{d_1}2_{d_1}3_{d_2}$	Fail	Fail	NA	Fail	Fail	NA	Fail	Fail	NA	Fail	Fail	NA	Fail	Fail	Pass
2	$0_{d_3}1_{d_1}2_{d_1}3_{d_1}$	Fail	Fail	NA	Fail	Fail	NA	Fail	Fail	NA	Fail	Fail	NA	Fail	Fail	Pass
	$0_{d_1}1_{d_3}2_{d_1}3_{d_1}$	Fail	Fail	NA	Fail	Fail	NA	Fail	Fail	NA	Fail	Fail	NA	Fail	Fail	Pass
	$0_{d_1}1_{d_1}2_{d_3}3_{d_1}$	Fail	Fail	NA	Fail	Fail	NA	Pass	Fail	NA	Pass	Fail	NA	Fail	Fail	Pass
	$0_{d_1}1_{d_1}2_{d_1}3_{d_3}$	Fail	Fail	NA	Fail	Fail	NA	Fail	Fail	NA	Fail	Fail	NA	Fail	Fail	Pass

Table 3: Experimental results for a 4-way stop junction

Deficiency ID	Explanation
I_1	Lack of controllability for short distances. During initialization, the controller always randomly assigns a rate for vehicle acceleration without taking into account the safe braking distance ahead. When the distance to go is small, the vehicle may not be able to brake safely. This may happen even for a single vehicle.
I_2	Hidden guidance for control. Vehicle control uses a boundary zone for junctions that cannot be obtained by sensing the junction environment delineated by entrances/exits and traffic signs.
I_3	No consideration of priorities between different itineraries with the same waiting time. The Scheduler sets priorities according to the creation order of vehicles.

- Autonomous Systems
- Autopilot Design
- Global ADS Validation
- Discussion

OVERVIEW

Trustworthy ADS – Systems Engineering meets AI

- Traditional systems engineering is being disrupted by new trends resulting from the economic and technical challenges of ADS:
  - adopting ML-based end-to-end solutions that do not provide trustworthiness guarantees;
  - allowing "self-certification“, in the absence of standards;
  - allowing regular updates of critical software - trustworthiness cannot be guaranteed at design time as required by standards - systems will be evolvable, with no end point in their evolution.

- Hybrid design leveraging on a solid body of knowledge for safe and efficient decision making.
  - Building trusted systems from untrusted components – Non-explainable AI will remain a wide open problem!
  - Linking symbolic and non-symbolic knowledge e.g. sensory information and models used for decision-making
- System validation marked by the shift from rationalism e.g. verification to empiricism e.g. testing.
  - Simple simulation is not enough - Statistics-based estimation of confidence levels;
  - Weaker trustworthiness guarantees!

- Bridging the gap between Automation and Autonomy
  - There is a big gap between automated and autonomous systems – the transition from ADAS to ADS cannot be progressive!
  - Nonetheless, autonomic complexity drastically scales down for enhanced situation awareness (perception) and environment predictability, e.g. “geofence autonomy”.

To reach the full autonomy vision we need to develop a new scientific and engineering foundation. And this will take some time.

THANK YOU