Dynamic Safety Assurance of Autonomous Cyber-Physical Systems

Shreyas Ramakrishna

PHD Defense Presentation

Vanderbilt University

July 5, 2022

ISIS
Tel (615) 343-7472 Fax (615) 343-7440
1025 16th Avenue South|Nashville, TN 37212
www.isis.vanderbilt.edu

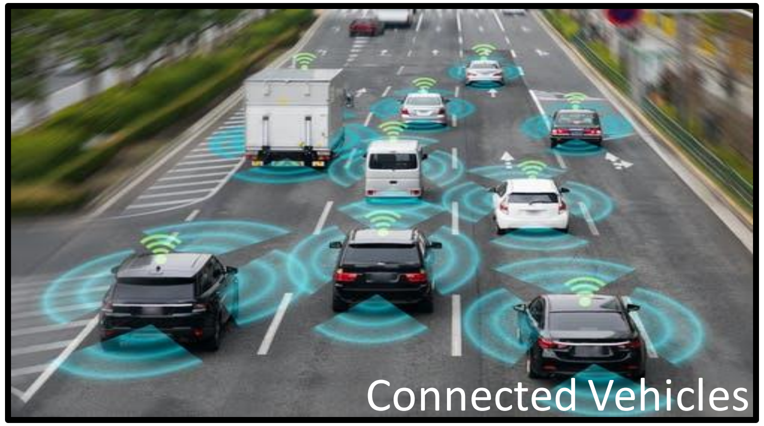
VANDERBILT UNIVERSITY

Outline

- Introduction
  - Cyber-Physical Systems and the Safety Problem
  - Safety Assurance Approaches and Limitations
- Our Approach: Dynamic Safety Assurance
  - Assurance Argument Development
  - Mitigation
  - Data Generation
- Summary & Questions

Introduction and Motivation

Cyber-Physical Systems



- Cyber-Physical Systems are "physical and engineered systems whose operations are monitored, controlled, and integrated by a computing device with communication capability"[1]

- Increased complexity and requirement to operate in dynamic and non-stationary environment have raised **safety concerns**

Toyota car crash (2007)

Columbia space shuttle disaster (2003)

Turkish airlines accident (2020)

1. Lee, Edward A. "Cyber physical systems: Design challenges." *2008 11th IEEE international symposium on object and component-oriented real-time distributed computing (ISORC)*. IEEE, 2008.

Safety Assurance Approaches

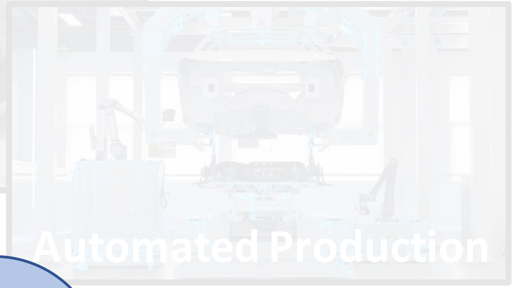
Conclusion with the assurance approaches

- Design-time approaches have created "a culture of **paper safety** at expense of **actual safety**" - Nimrod RAF accident[2]
- Despite runtime approaches, there is insufficient clarity on how to **evolve** the system's "safety reasoning" at runtime

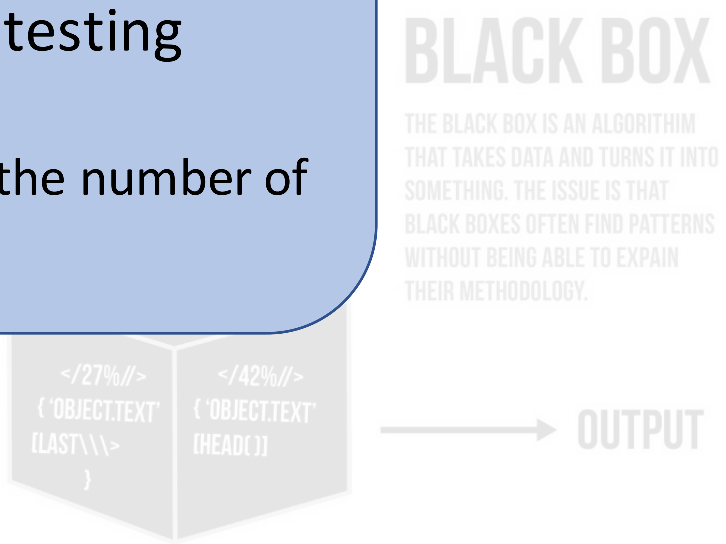
1. Leveson, Nancy G. "The role of software in recent aerospace accidents." *Proceedings of the 19th International System Safety Conference, System Safety Society: Unionville, VA*. 2001.
2. RAF Nimrod http://aerossurance.com/safety-management/nimrod-xv230-haddon-cave/

Safety Problems Exacerbated by LECs

- Learning Enabled Components* have revolutionized the field of CPS
  - **Simplified** the design of complex components
  - **Increased** the level of **autonomy**
- However, these
  increased CPS
  - **Safety** has b
- Reasons for th
  - Non-transp
  - Implicit ass
    Problem)
  - **Implicit biase**
    data



vehicle crashes were
ited States between
2022" - NHTSA



Effects on Assurance Approaches

- LECs bring in new **implicit assumptions** in designing the assurance arguments (e.g., out-of-distribution problem)
- LECs **further complicate** the verification and testing procedures
  - **Black-box** nature and **non-linearity** have limited the number of tools and techniques that are available

* Components trained from data using machine learning techniques

What Do We Need for Assurance of Autonomous CPS ?

A dynamic approach that combines both design-time and runtime assurance approaches to perform "through life safety assurance" of autonomous CPS

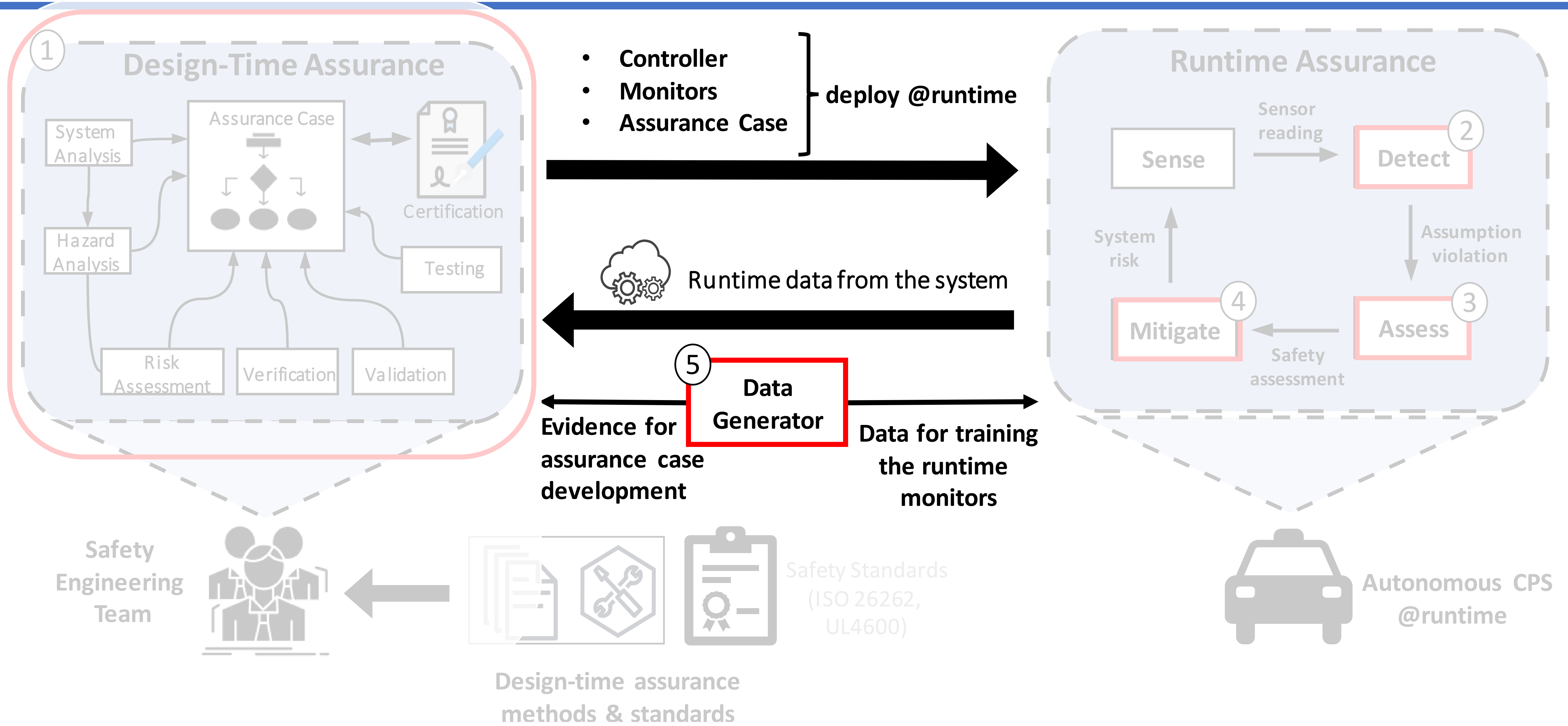
Overall Goal: should be to reduce risk of hazardous and catastrophic failures, while ensuring that the performance objectives are met

This requires: monitoring "how" and "when" the design-time assumptions get invalidated and assess its impact on the "system-level safety"

What's needed: a design-time assurance model with runtime detection, assessment, and mitigation strategies

Dynamic Safety Assurance

Dynamic Safety Assurance Framework



Research Questions with Dynamic Safety Assurance Components

Assurance Case Development and Evaluation

How do we automate the development and evaluation of an assurance case?

Out-of-Distribution Detection

How do we efficiently detect out-of-distribution data and identify the factor(s) responsible for the problem?

Mitigation

How do we mitigate the risk posed to the system, while still maintaining the performance objectives?

Risk Assessment

How do we quantify the risk posed to the system by the hazards and faults under varying operating conditions at runtime?

Data Generation

How do we automatically generate data for the dynamic assurance component, especially data from high-risk scenarios involving the system (e.g., faults and adverse weather conditions)?

Contributions of this Dissertation

Assurance Case Development

Contributions:

- A workflow for automatic synthesis of an assurance case
- Coverage metrics and an analysis report for evaluating the assurance case

Out-of-Distribution Detection

Contributions:

- Workflow for designing an **efficient detector** that performs detection on **low-dimensional space**
- **Bayesian Optimization heuristic** to design and train the detector
- OOD responsible **feature identification**

Mitigation

Contributions:

- A blended-simplex strategy called "**Weighted Simplex Strategy**" to overcome (a) conservatism of the decision logic, and (b) avoid instantaneous controller transition – **RL algorithm**
- The "**Dynamic Simplex Strategy**" with a non-myopic planner for reverse switching aimed to improve the system's performance without compromising on safety – **MCTS online heuristic**

Risk Assessment

Contributions:

- Proactive risk assessment framework called "**ReSonAte**"
- Combine design-time hazard rate with runtime system monitors to compute the system's operational risk

Data Generation

Contributions:

- Adversarial data generation framework "**ANTI-CARLA**"
- A scenario description language
- Two adversarial samplers

Today's Focus

Assurance Case Development

Contributions:

- Automated pattern selection
- Automated AC evaluation
- Integration with ACCELERATE assurance case generation tool

Out-of-Distribution Detection

Contributions:

- Workflow for designing an **efficient latent-space detector**
- Bayesian Optimization heuristic to train detector to generate a disentangled latent space
- OOD responsible **feature identification** in the latent space

Mitigation

Contributions:

- A blended-simplex strategy called "**Weighted Simplex Strategy**" to overcome (a) conservatism of the decision logic, and (b) avoid instantaneous controller transition – **RL algorithm**
- The "**Dynamic Simplex Strategy**" with a non-myopic planner for reverse switching aimed to improve the system's performance without compromising on safety – **MCTS online heuristic**

Risk Assessment

Contributions:

- Proactive risk assessment framework called "**ReSonAte**"
- Combine design-time hazard rate with runtime system monitors to compute the system's operational risk

Data Generation

Contributions:

- Adversarial data generation framework "**ANTI-CARLA**"
- A scenario description language
- Two adversarial samplers

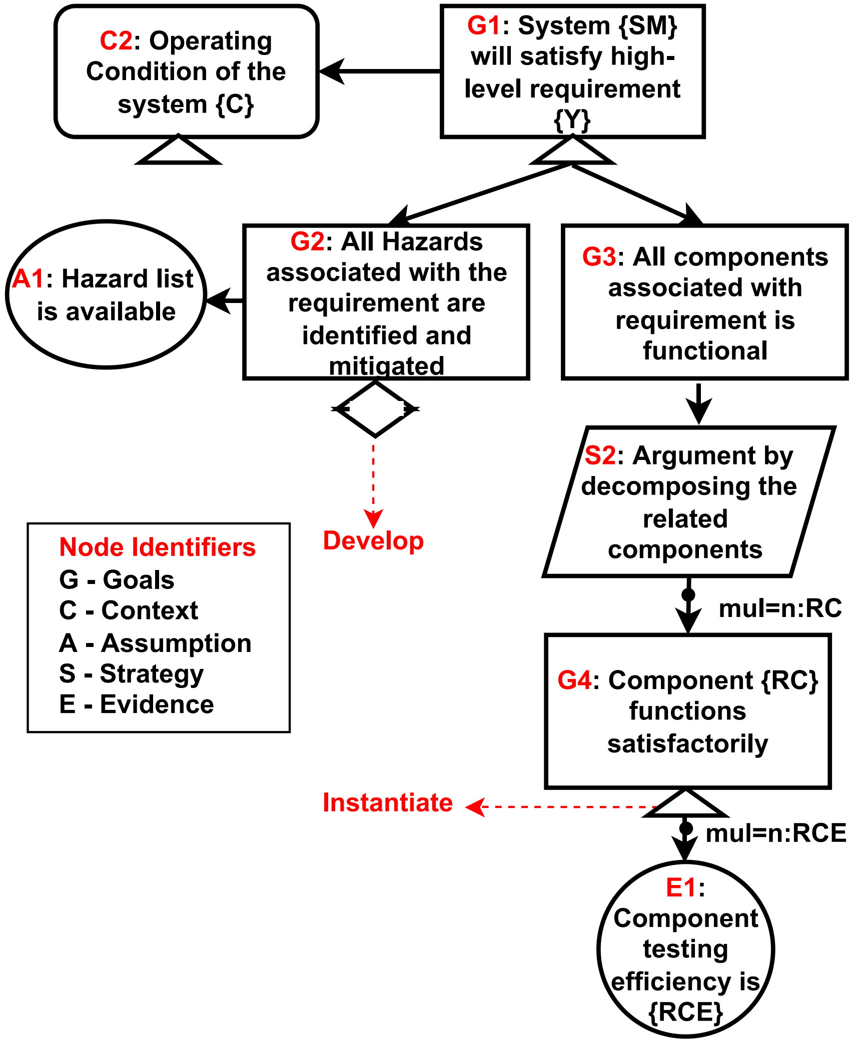
Publications

- **S. Ramakrishna**, H. Jin, A. Dubey, and A. Ramamurthy. “Automating Pattern Selection for Assurance Case Development of Cyber-Physical Systems”. 2022, Accepted, Pending Publication

Assurance Case Development and Evaluation

Assurance Case Patterns

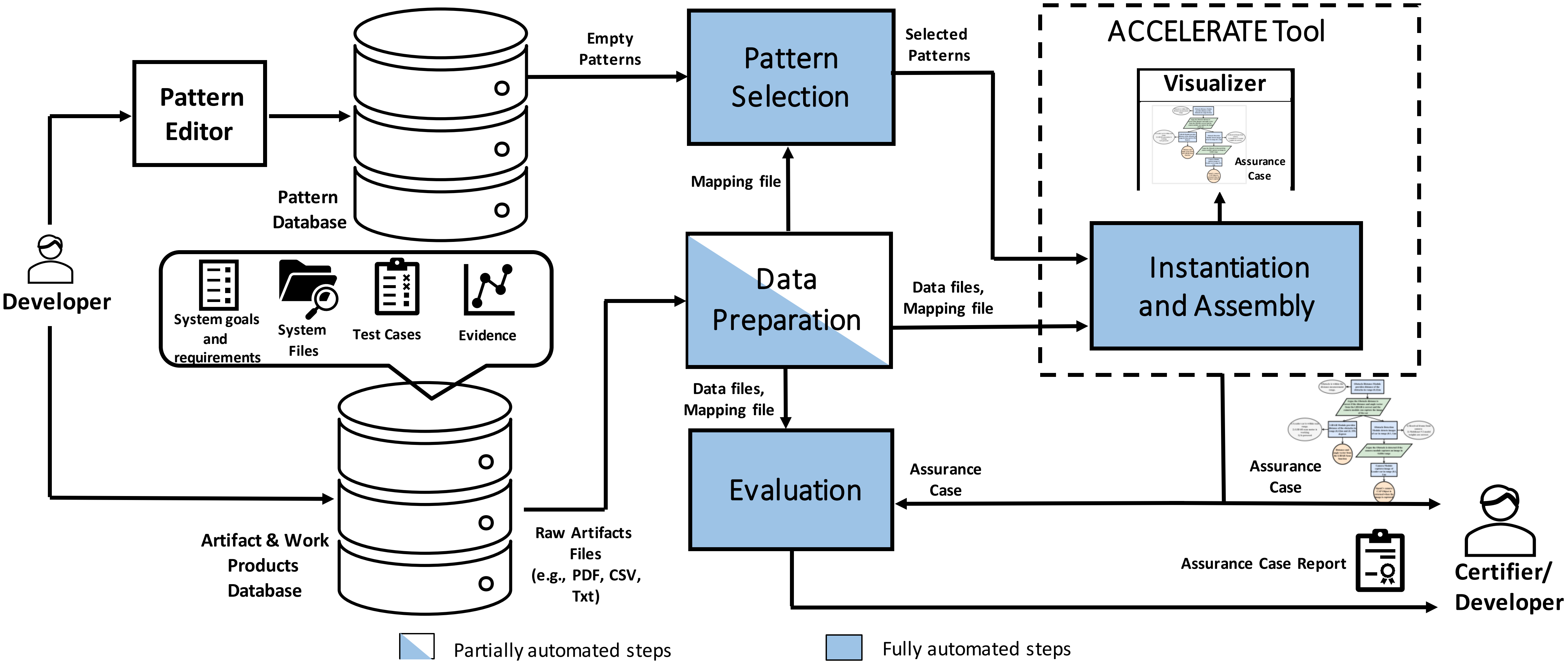
- **Divide and Conquer** strategy used to handle the growing complexity of assurance cases
  - Assurance fragments called **patterns** provide a partial argument for one aspect of the system
  - Patterns are **instantiated & assembled** into an assurance case
- Existing work aim at automating the instantiation and assembly algorithm



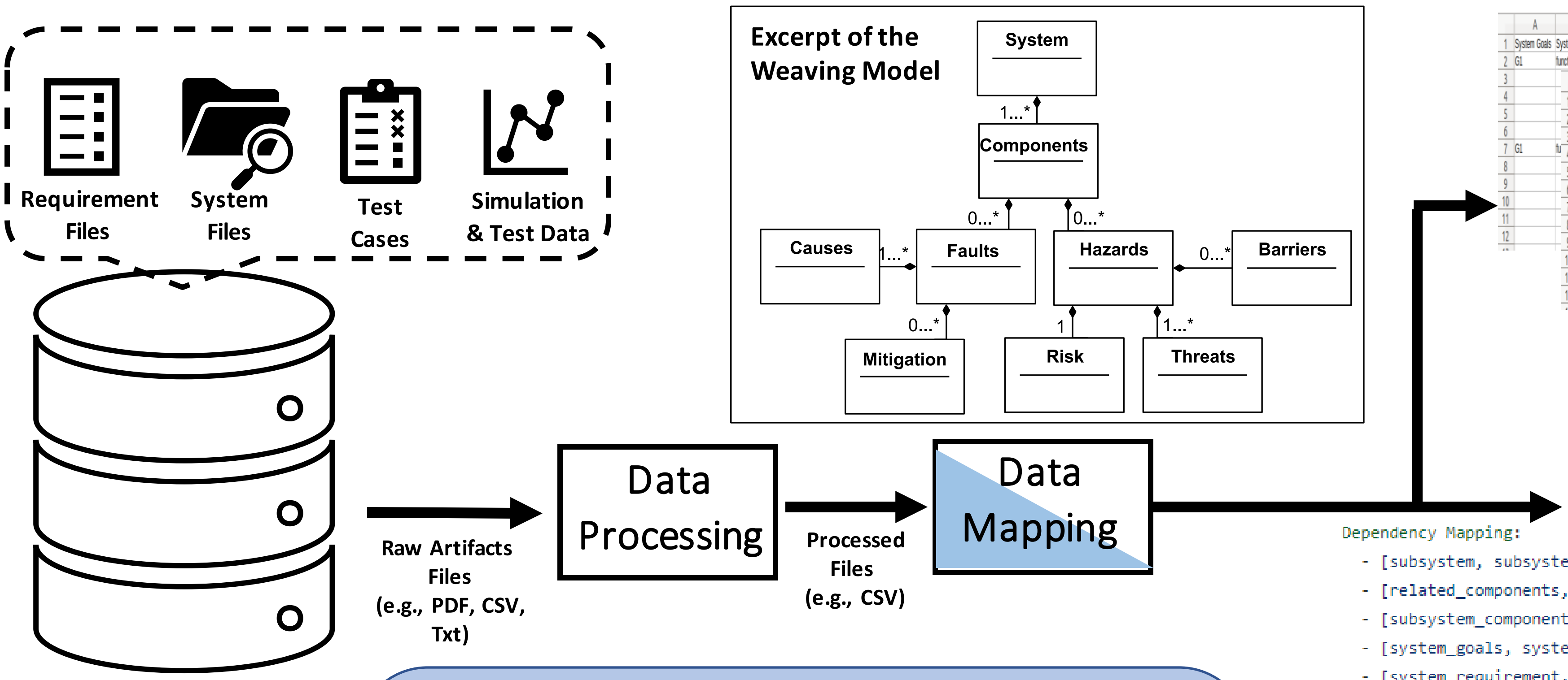
Problems with using patterns

- Selection has been shown to consume significant development time (14%)[3]
- Automating pattern selection could **reduce the construction time**

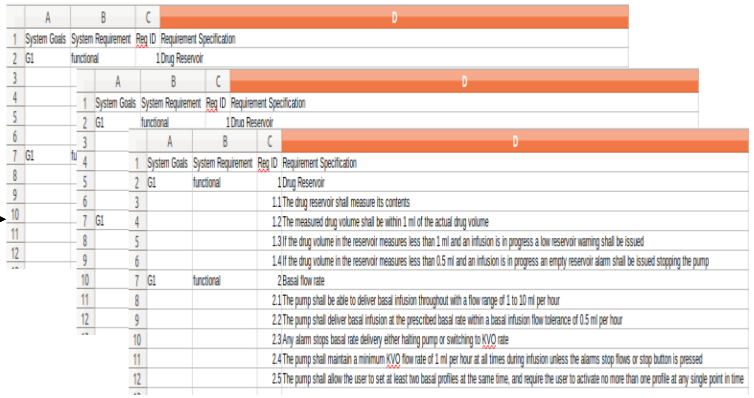
Automated Assurance Case Development Workflow



Step1: Data Processing and Mapping



Artifact Database



Data Files

Mapping File

```
Dependency Mapping:
  - [subsystem, subsystem_table, system]
  - [related_components, related_components_table, sub_requirement]
  - [subsystem_components_redundancy, subsystem_components_redundancy_table, subsystem_components]
  - [system_goals, system_goals_table, system]
  - [system_requirement, system_requirement_table, system_goals]
  - [failure_mitigation, failure_mitigation_table, failure_modes]
  - [failure_modes, failure_modes_table, subsystem_components]
  - [subsystem_components, subsystem_components_table, subsystem]
  - [mitigation, mitigation_table, cause]
  - [mitigation, mitigation_table, hazard]
  - [sub_requirement, sub_requirement_table, highlevel_requirement]
  - [highlevel_requirement, highlevel_requirement_table, system_requirement]
  - [subsystem_components_efficiency, subsystem_components_efficiency_table, subsystem_components]
  - [failure_modes_detectors, failure_modes_detectors_table, failure_modes]
```

Data preparation

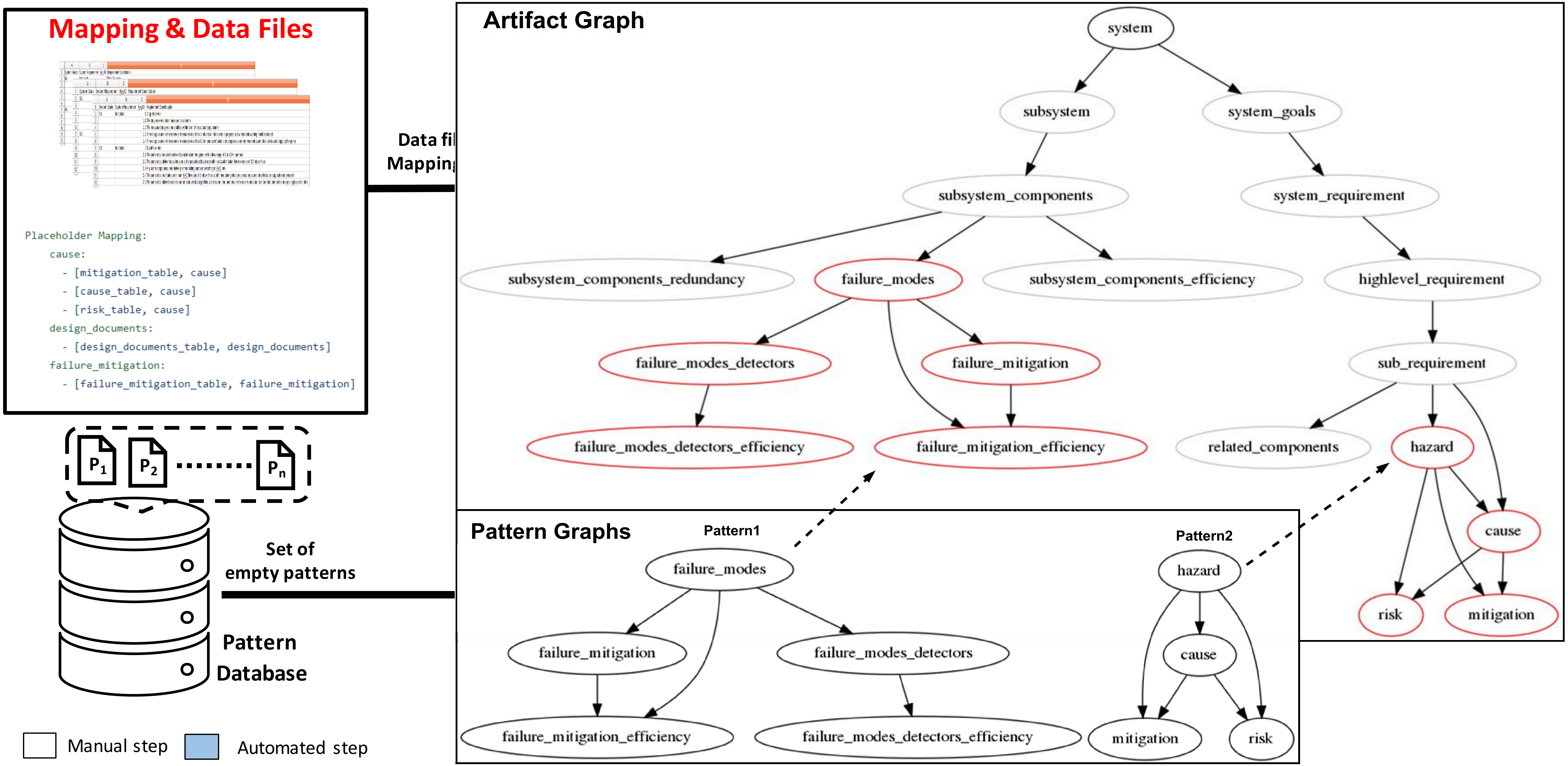
- Manual processing and checking for completeness and relevance
- A **weaving model*** is used to perform data mapping
- The model **captures** fine-grained **relationships** of the different artifacts

Manual step

Partially automated step

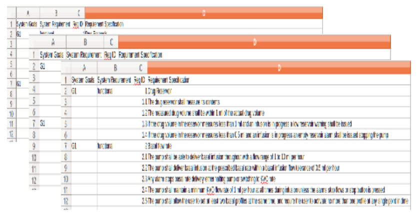
*Hawkins, Richard, et al. "Weaving an assurance case from design: a model-based approach." *2015 IEEE 16th International Symposium on High Assurance Systems Engineering*. IEEE, 2015.

Step2: Pattern Selection

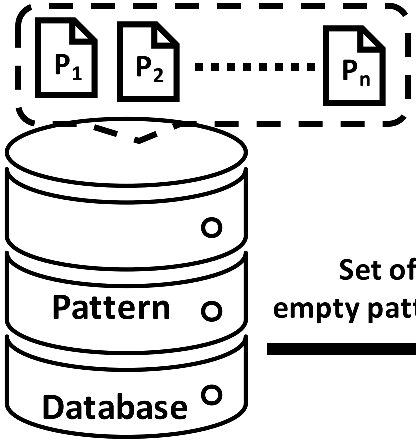


Step3: Assurance Case Development and Evaluation

Mapping & Data Files



```
Placeholder Mapping:
    cause:
      - [mitigation_table, cause]
      - [cause_table, cause]
      - [risk_table, cause]
    design_documents:
      - [design_documents_table, design_documents]
    failure_mitigation:
      - [failure_mitigation_table, failure_mitigation]
```



Set of empty patterns

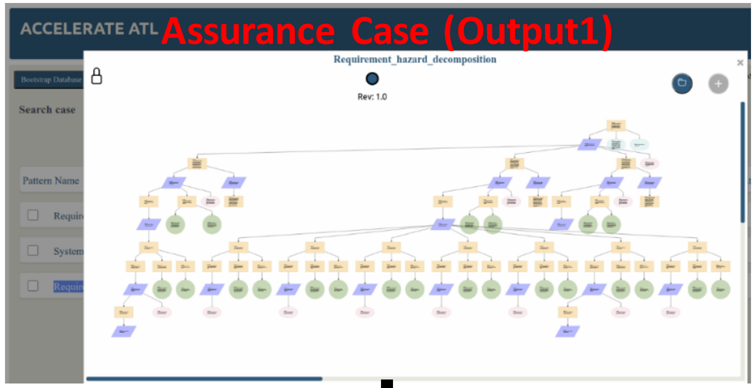
Data files, Mapping file

Pattern Selection

Selected patterns

Instantiation and Assembly

Assurance Case (Output1)



Evaluation

Evaluation Report (Output2)

```
Selected Patterns: [
    Component_Dependency,
    Hazard_Decomposition,
    System_Goals,
    System_Architecture ],
Unused Patterns: [
    Functional_Decomposition,
    Risk_Alarp,
    Safety_Compliance ],
Unused Artifacts: [
    System_model_documents,
    Requirement_documents,
    Failure_modes,
    Failure_mode_mitigation,
    Mitigation_efficiency ],
Assurance Score: [
Confidence_Score: 60%
Artifact_Coverage: 76%
Safety_Coverage: [
    Hazard_information: 60%
    Failure_mode_information: 100% ] ]
```

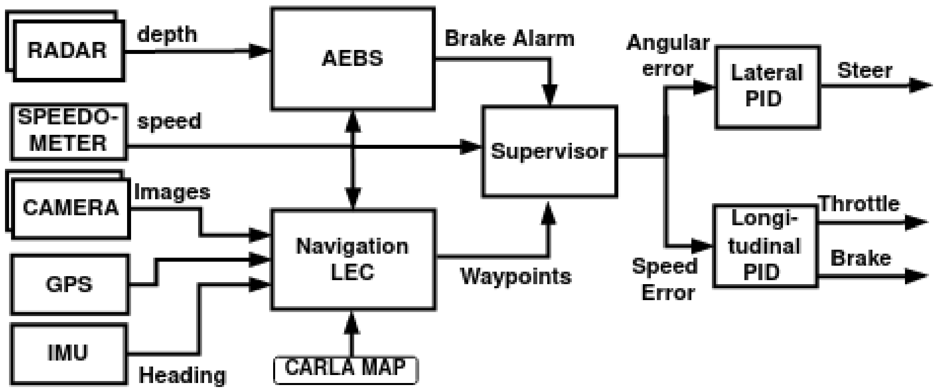
Automated AC development and evaluation

- Automated Instantiation with **ACCELERATE* tool**
- **Strongly typed language** that defines the syntax for designing patterns
- We evaluate the assurance case for coverage
    - **Artifact coverage, Problem coverage**
- We generate an evaluation report

* ACCELERATE is being built by JHU and Siemens as efforts towards DARPA ARCOS

Demonstration Platform

An autonomous vehicle operating in CARLA[1] simulation under varying weather and sensor faults.



Key Results

- Manual construction of a small assurance case (500 GSN nodes) took one engineer **8 hours**
- Automated pattern instantiation and assurance construction took **less than 5 minutes**
- We observed slight increase in construction times (~10 minutes) for larger assurance cases (1500-3000 GSN nodes)

1. https://carla.org/

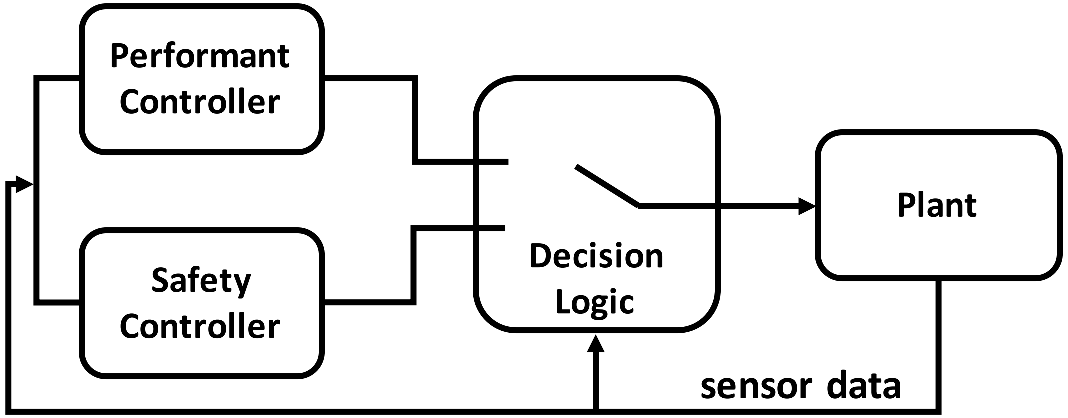
Experiments available on our GitHub: https://github.com/scope-lab-vu/AV-Assurance

Publications

- **S. Ramakrishna**, B. Luo, C. Kuhn, A. Mukhopadhyay, G. Karsai, and A. Dubey. “Dynamic Simplex Strategy for Autonomous Cyber-Physical Systems”. 2022, Awaiting submission
- **S. Ramakrishna**, C. Hartsell, M. Burruss, G. Karsai, and A. Dubey. “Dynamic-weighted simplex strategy for learning enabled cyber physical systems.“ 2019, Journal of systems architecture

Mitigation

Simplex Architecture for Mitigation



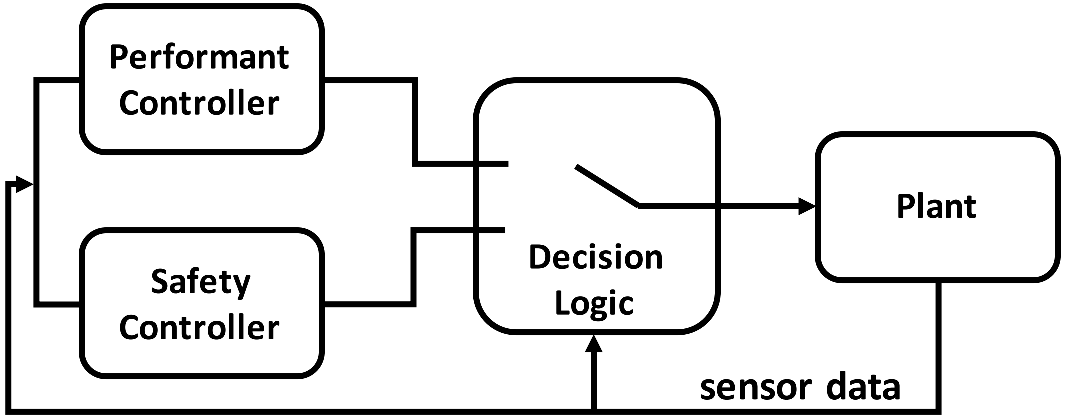
Conventional simplex architecture

Simplex Architecture[1]

- The framework augments a safety controller and a **decision logic** to CPS with unverified high-performance controller
- On sensing that the system is entering into a bad state, the logic performs a **forward switch**
  - Performant controller ➞ Safety controller

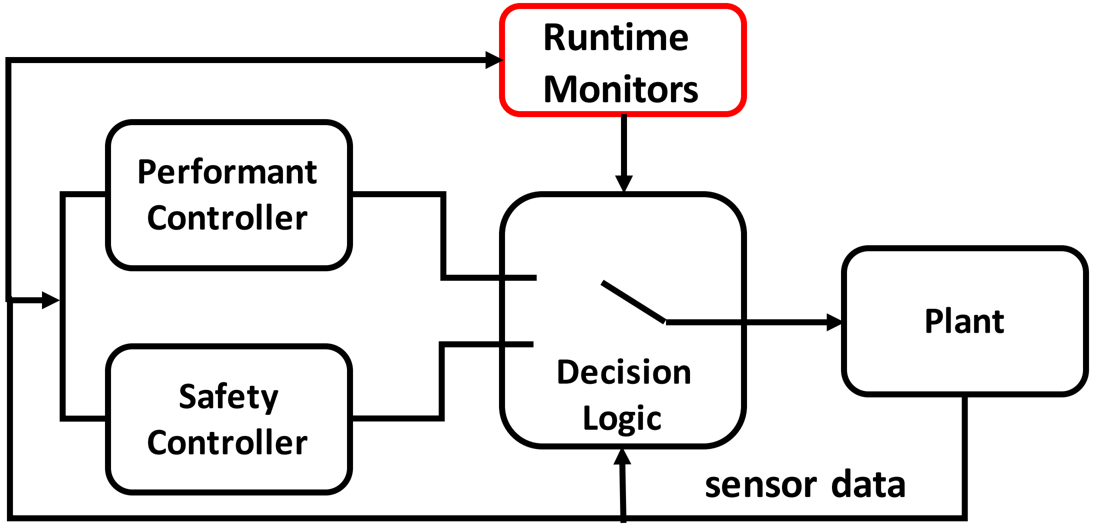
1.Lui Sha, et al. *"A Software Architecture for Dependable and Evolvable Industrial Computing Systems"*. Carnegie-Mellon University Pittsburgh Software Engineering Inst, 1995.

Problems with the Conventional Simplex Strategy



Problems with the architecture

- Designed and **trained offline**
- **Too conservative** (instantly switches to the safety controller)
- Do not perform a **reverse switch**
- **Instantaneous switching** hurts the system

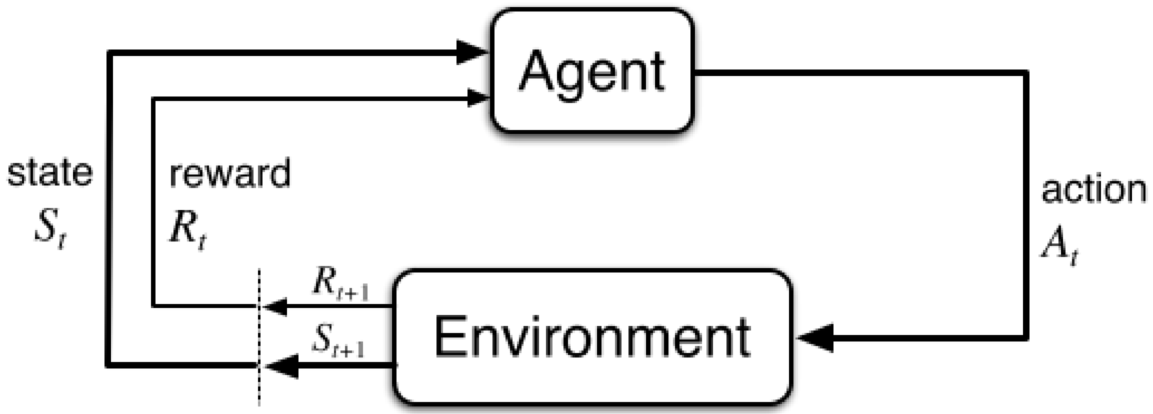


Dynamic Simplex Strategy

- Goal is to **not compromise on safety** but **increase performance** of the system
- **Non-myopic** optimal reverse switch to improve performance
- Switching routine to avoid **instantaneous switch**

System Model – Semi-Markov Decision Process

A natural modeling framework for control problems in which a decision agent interacts with an uncertain environment and actions have future consequences is the *(Semi)-Markov Decision Process* (SMDP)*



State: System location, weather condition and the current controller, sensor failure
Action: Switch/ not switch
Transitions: Environment evolution depends on travel times of the system, weather conditions, and sensor failures
Reward: Performance and safety scores

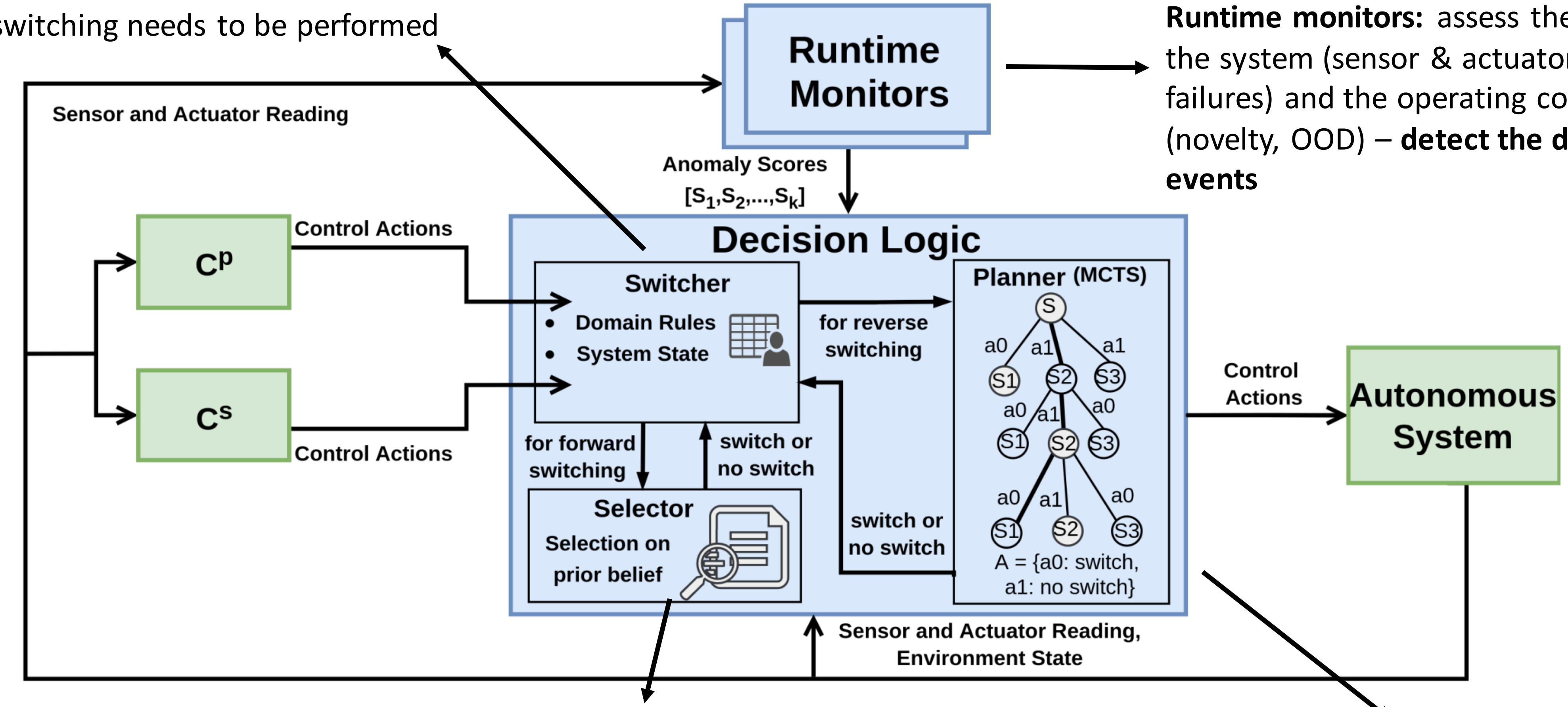
Decision-maker triggered by three events: (1) when location changes, (2) when the weather changes, and (3) when a sensor on the system fails

* *Semi-Markov because process evolves in continuous time and transitions are not memoryless*

Dynamic Simplex Strategy: Overall Architecture

Switcher: Decides "if" and "when" the switching needs to be performed

Runtime monitors: assess the state of the system (sensor & actuator failures) and the operating conditions (novelty, OOD) – **detect the decision events**



Selector: decision-maker during the forward switch

Planner: decision-makes during the reverse-switch

Dynamic Simplex Strategy: Switcher

Switcher: Decides if and when the switching needs to be performed

Constantly **monitor for decision events** using the runtime monitors

Forward switch: performed by the **selector**

Reverse switch: performed by the **planner**

Controller transition: performed by a **switching routine to avoid instantaneous transitions**

- **System state (e.g., reduce speed)**
- **Domain rules:** location and system state under which switching can be performed

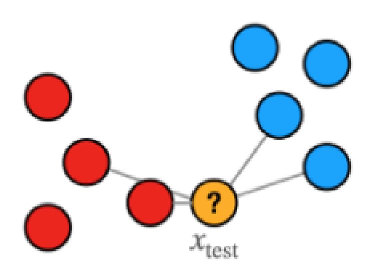
Switching	Domain rules
Acceptable	Main road, freeway, overpass
Restricted	Intersection, roundabout
	Pedestrian crossing the road
	Turns
	Lane changes

Dynamic Simplex Strategy: Forward Switch

At each decision event, select an optimal action based on the **current state**

Selector: Decision-maker for forward switching

- Uses historical data to decide if a forward switch is required
- Decision made only on the current state – **safety concerns**



K-NN based Search

A K nearest neighbor algorithm-based search heuristic that finds which controller performed better in the given state

Dynamic Simplex Strategy: Reverse Switch

At each decision event, evaluate potential actions by estimating their future trajectories using generative models

Find a *policy:* a general mapping from states to actions

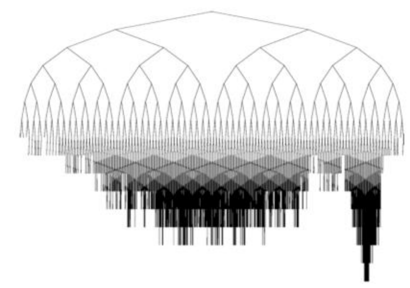
Methods: Reinforcement Learning

Barriers:

- Long time to learn a policy for large state-action space
- Must re-learn models to account for non-stationary environment
- Not resilient to failures and unexpected environmental shifts such as weather, sensor failures.

Planner – Decision Maker for reverse switching

- Focuses computation on one relevant state
- Adaptability – if environment changes or there is a failure, simply update the underlying models

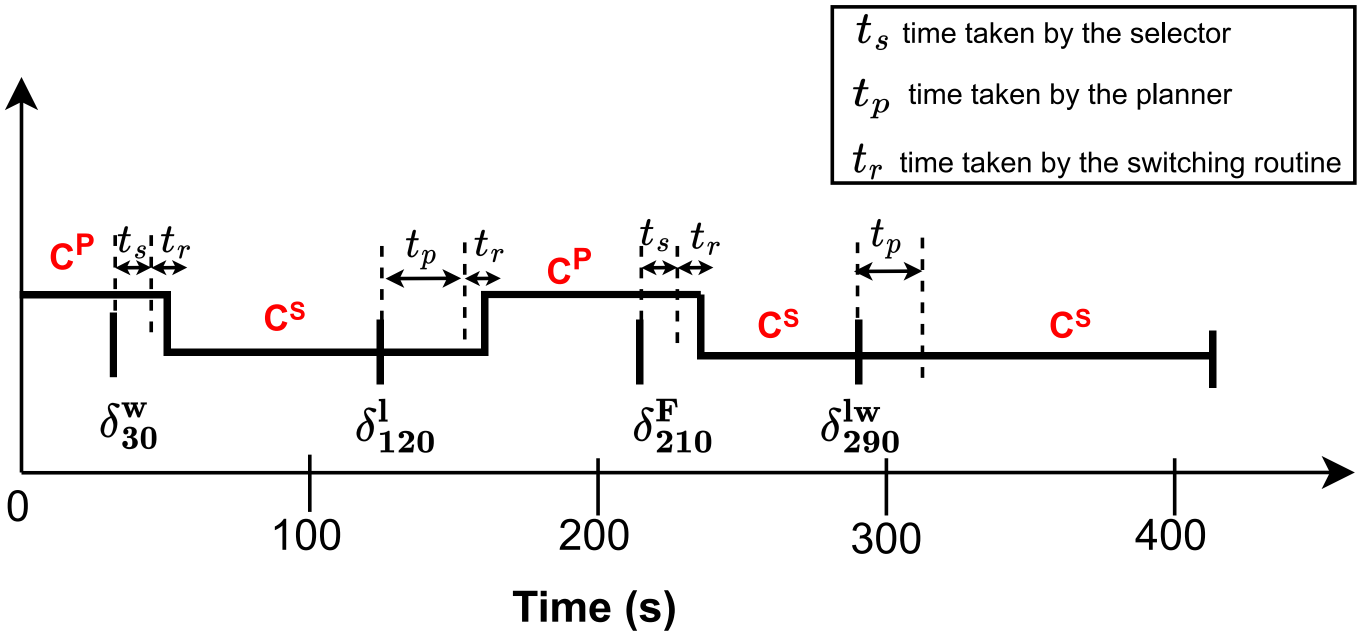


Monte Carlo Tree Search

Heuristic planning algorithm which balances exploration and exploitation to efficiently navigate a decision tree, focusing computation on promising action trajectories

1. Lenz, David, et al. "Tactical cooperative planning for autonomous highway driving using Monte-Carlo Tree Search." 2016 IEEE Intelligent Vehicles Symposium (IV).
2. Hoel, Carl-Johan, et al. "Combining planning and deep reinforcement learning in tactical decision making for autonomous driving." *IEEE transactions on intelligent vehicles* 5.2 (2019): 294-305.

How the approach works



Controller transition with the proposed solution approach

- The switcher picks one decision-maker based on the controller driving the system
- The selector requires t_s **seconds** and planner requires t_p **seconds** to decide an action
- The switching routine will take t_r **seconds** to perform the controller transition

Evaluation with CARLA Simulator

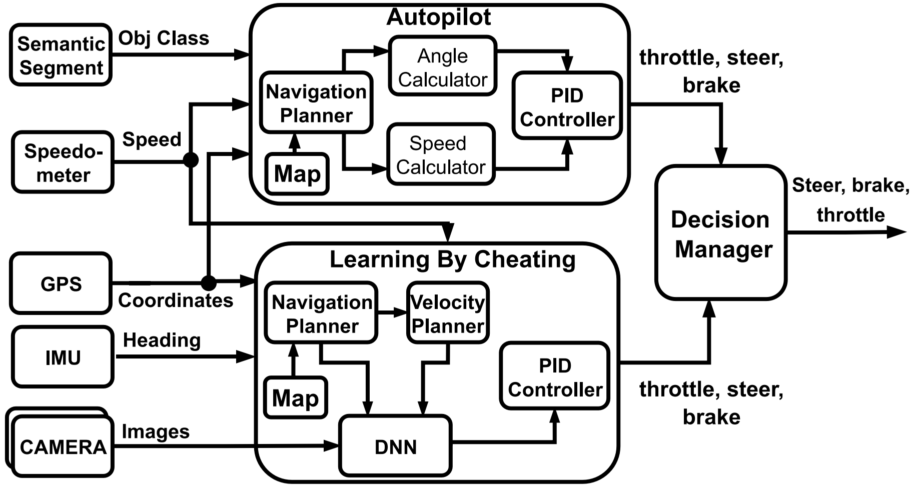
Experimental Setup

Experimental Setup

- **Safety controller**: Autopilot controller
- **Performant Controller**: Learning By Cheating LEC[1]
- **10 training** tracks and **4 testing** tracks
- A lookup table of **12500** for designing the logic
- System monitors: Collision likelihood estimator, Novelty detector, and Camera failure detectors

Baselines

- Performant controller (LBC), Safety controller (AP), Simplex strategy (SA), and Simplex strategy with reverse switch (SA^R)



System model of the AV

Track 1 - Downtown

Track 2 - Suburb

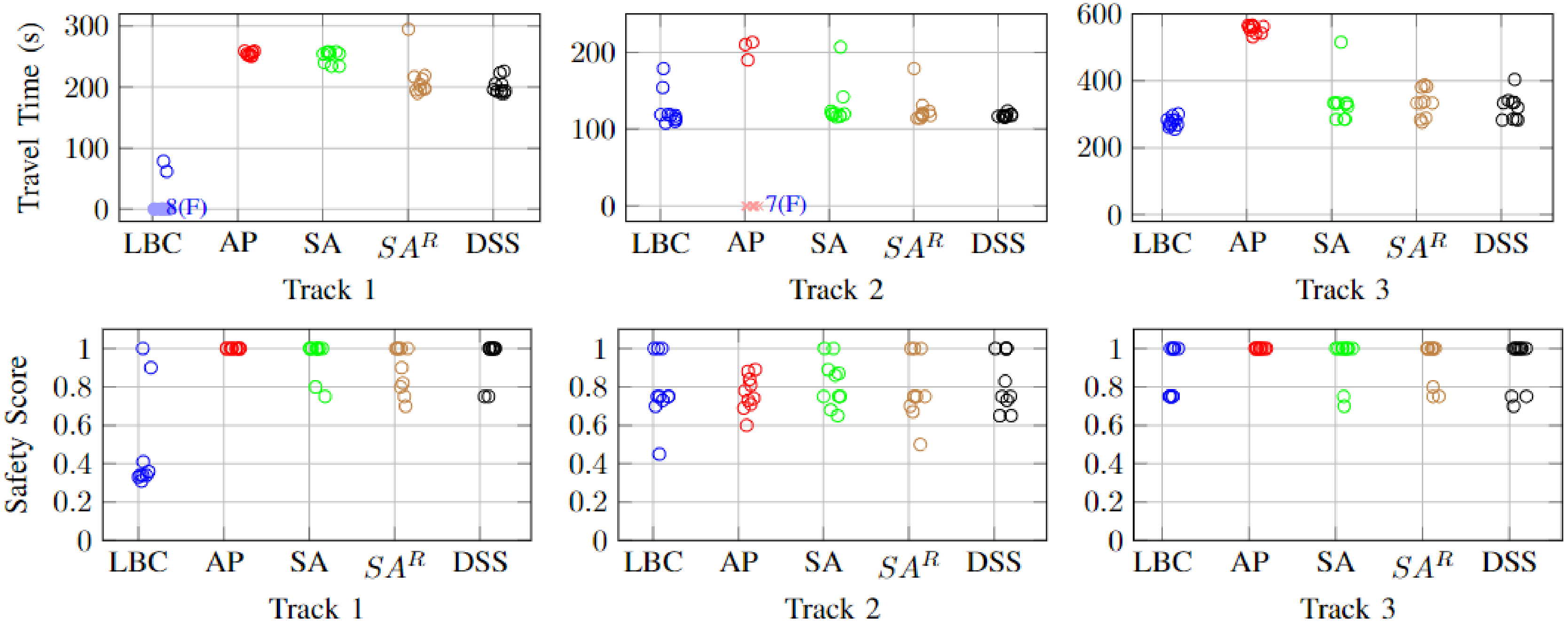
Track 3 - Freeway

Track 4 - Tunnel

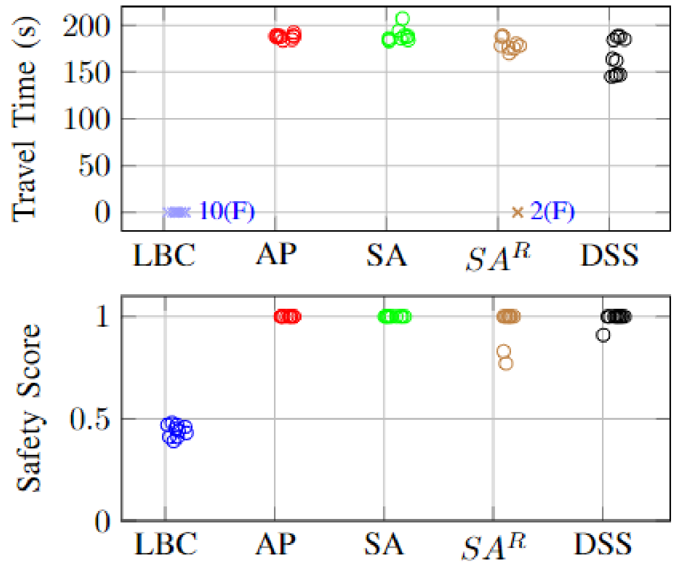
1. Chen, Dian, et al. "Learning by cheating." *Conference on Robot Learning*. PMLR, 2020.

Experimental Results

Experiment1: AV with no sensor failure



Experiment2: AV with sensor failure



Key Results

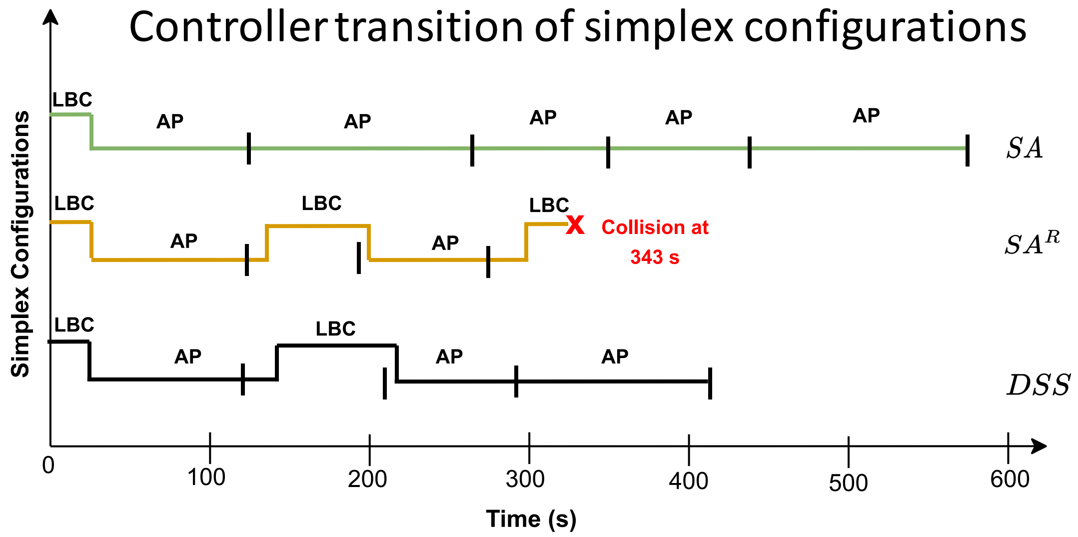
- The proposed strategy overcomes the safety problem of the LEC
- It has **shorter travel times** and a **high safety score** compared to baselines
- Performance gains primarily because of the **planner-based reverse switching**

Performance measured using travel times around the track

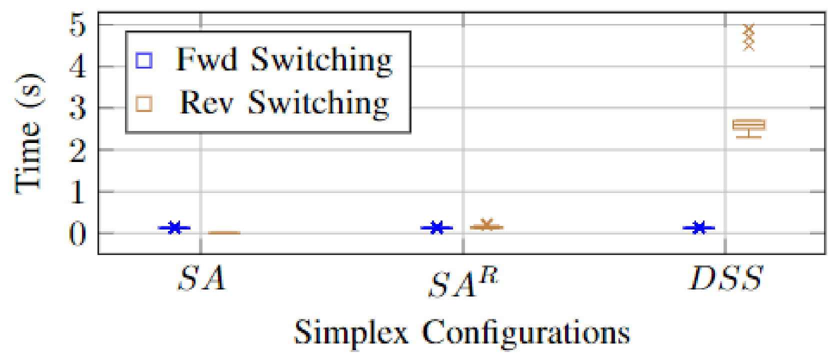
Safety measured using a combined infraction score (ideal safety score is 1.0)

Experiment Results

Controller transition of simplex configurations



Execution times of the decision-makers



Key Results

- The planner does not reverse switch anticipating the future consequence
- The **planner** has an average execution time of **2.5 seconds**, compared to **0.13 seconds** of the **selector**
- Our approach consumes **similar resources** as the other simplex configurations

Controller Configuration	CPU (%)		GPU (%)		Inference Times (s)
	Util	Mem	Util	Mem	
LBC	27.7	21.8	5.21	7.98	0.031
AP	28.4	23.4	0	0	0.215
SA	24.1	19.9	5.44	8.15	0.075
SA^R	22.1	18.8	5.35	8.15	0.088
DSS	32.2	24.6	5.21	8.14	0.090

Resource and execution time comparison

Publications

- **S. Ramakrishna**, B. Luo, Y. Barve, G. Karsai, and, A. Dubey. "Risk-Aware Scene Sampling for Dynamic Assurance of Autonomous Systems". 2022, ICAA

- **S. Ramakrishna**, B. Luo, C. Kuhn, G. Karsai, and, A. Dubey. "ANTI-CARLA: An Adversarial Testing Framework for Autonomous Vehicles in CARLA". 2022, Accepted at ITSC, Pending Publication

Data Generation

Scene Generation

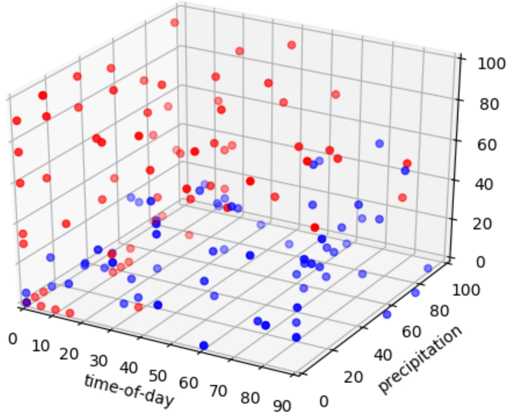
Track 1 - Downtown

Track 2 - Suburb

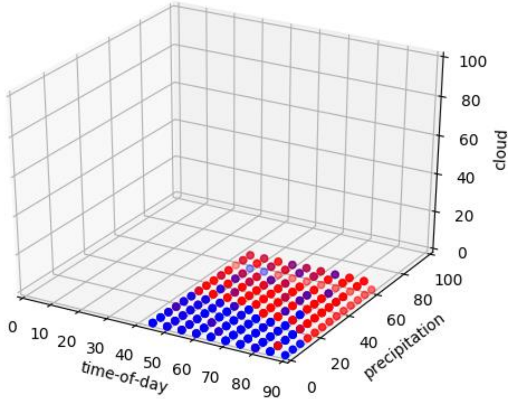
Track 3 - Freeway

Track 4 - Tunnel

Random Search



Grid Search



Scene Generation for Autonomous CPS

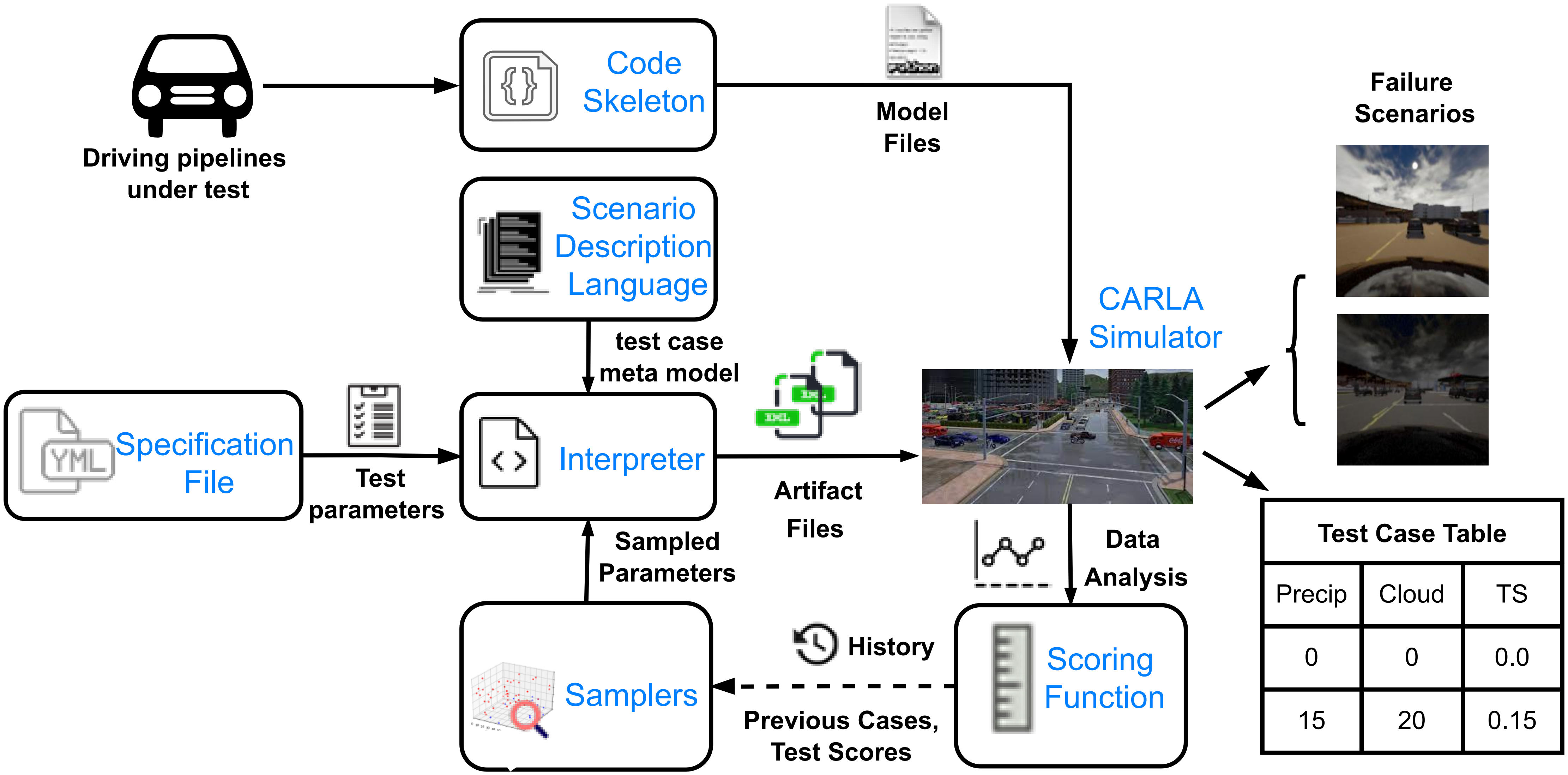
- Scenario Description languages like **Scenic**[1] and **MSDL**[2] are available for automotive domain
- They sample scenes using **passive samplers (no-feedback)** like random and grid search

Problems with the scene generation approach and samplers

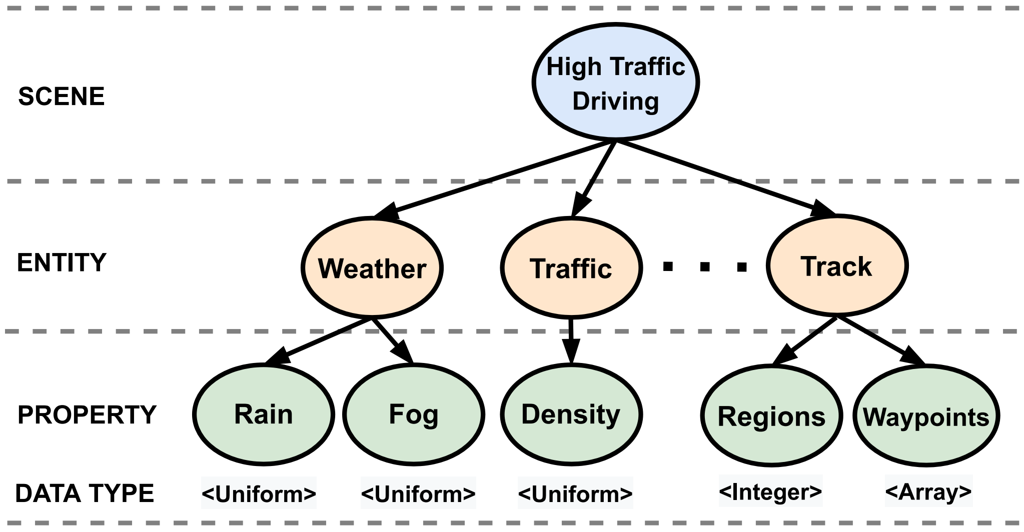
- These languages have a **steep learning curve** and difficult to transfer to other domains
- The passive samplers have problems:
  - Do not perform directed search **(no-feedback)**
  - Do not consider **parameter constraints** and correlations in sampling
  - Do not **optimally balance** the exploration vs. exploitation tradeoff

1. Fremont, Daniel, et al. "Scenic: Language-based scene generation." *arXiv preprint arXiv:1809.09310* (2018).
2. Measurable Scenario Description Language https://www.foretellix.com/wp-content/uploads/2020/07/M-SDL_LRM_OS.pdf

A data generation framework for Autonomous CPS



Specification Files and Scenario Description Language



Meta-model of our SDL*

```
Scenario Description{
    town: 5 //Available towns 3 and 5
    track: 1 // 1 track available for each town
       regions: 5 //Each town has 5 regions
    weather:  //Weather parameters and distribution range
      cloudiness: [0,100]
      precipitation: [0,100]
      time-of-day: [-90,90]
    pedestrian_density: [0,3]
    traffic_density: [0,10]
    Constraints: //A constraint on the rate of change in
         parameter values
        weather_delta: 2
        traffic_delta: 2
        pedestrian_delta: 1
    Infraction Metrics: //Infraction metrics to be
         recorded
        Infraction Penalty: true
        Off-road Driving: true
        Route Deviation: false
    Record Frequency: 5Hz } //Frequency of data recording
```

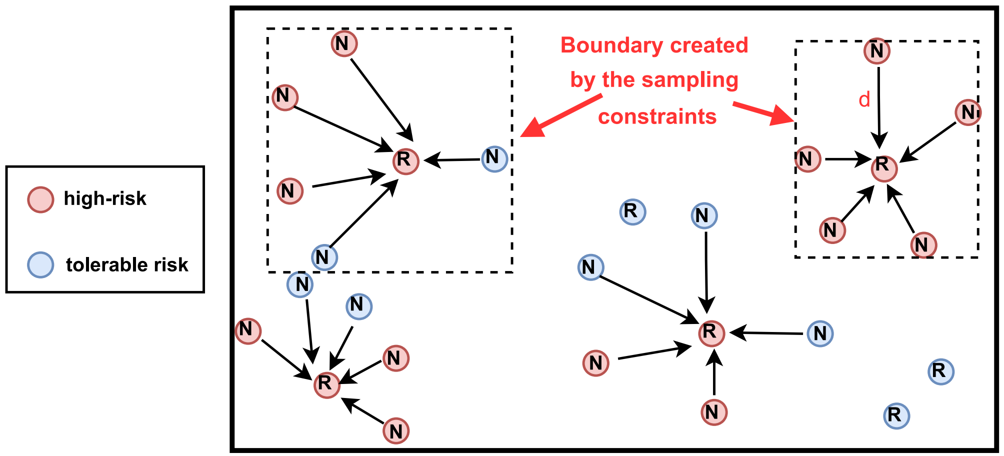
Excerpt of scene specification file

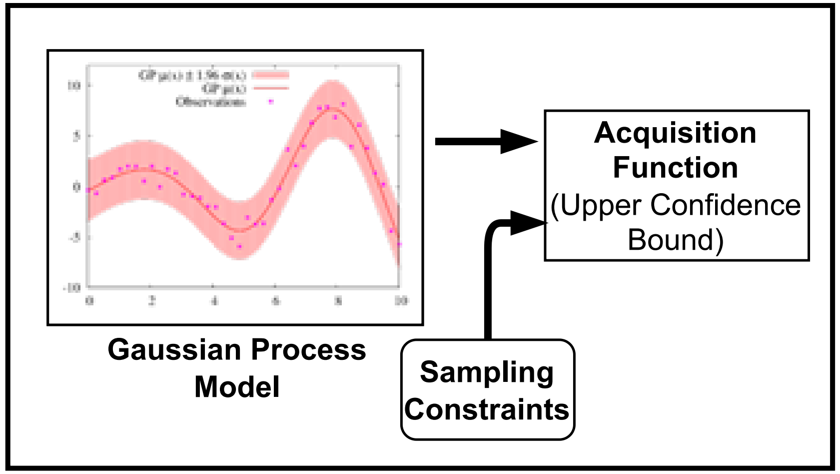
- Scenario description language models an operational scene and its contents
    - Includes grammar and a meta-model
- Specification files allow users set the parameters for:
    - Operating conditions (e.g., weather) and Agent configurations (e.g., sensors, inference rate)
- An interpreter connects the specification file, SDL, and the samplers

* SDL written in textX modeling language

Dejanović, Igor, et al. "TextX: a Python tool for domain-specific languages implementation." *Knowledge-Based Systems* 115 (2017): 1-4.

Adversarial Samplers





Random Neighborhood Search

- Extends the conventional random search to perform exploitation using the K-NN algorithm
- Overall idea is to randomly generate a scene and in case it is of high risk, then **exploit** the nearby scenes

Guided Bayesian optimization Search

- Extends the conventional Bayesian optimization algorithm with sampling constraints
- **GP model** is fit across the previously explored points
- Then, the constraints bound the **acquisition function** to look into smaller search space for future sampling
- Exploration vs. exploitation controlled by **Upper Confidence Bound** function

Demonstration Platform – CARLA Simulator


(a) Town5, t=1, c=10%, p=5%, d= 8°

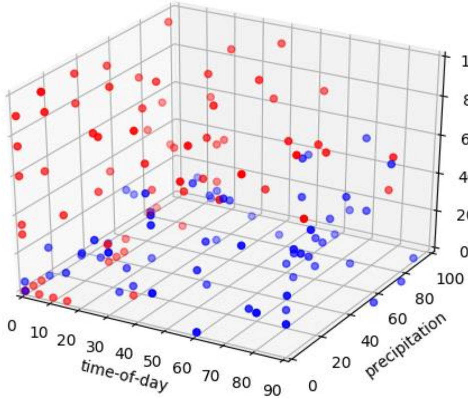

(b) Town5, t=10, c=70%, p=60%, d=24°


(c) Town3, t=4, c=20%, p=0%, d=60°

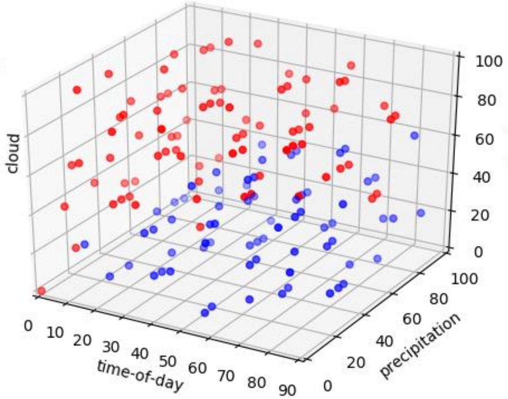

(d) Town3, t=2, c=40%, p=0%, d=0°

- Scenes with different weather conditions, road segments and traffic density
- AV driven with Learning By Cheating[1] LEC

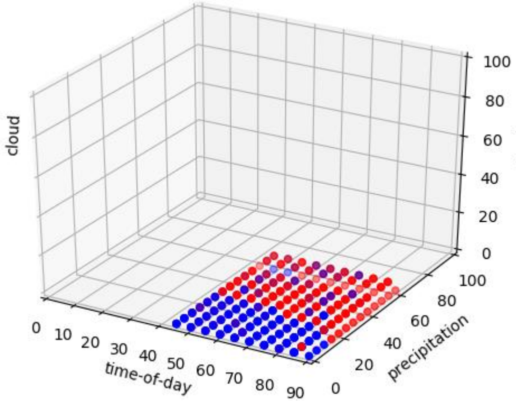
Random



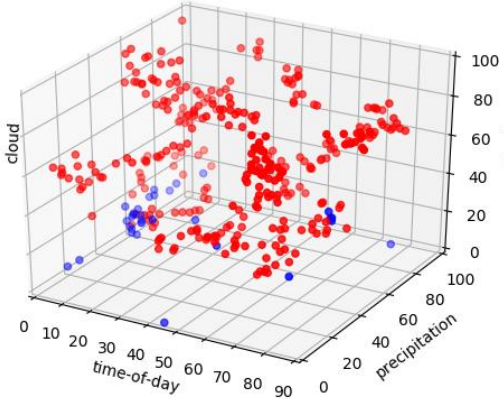
Halton



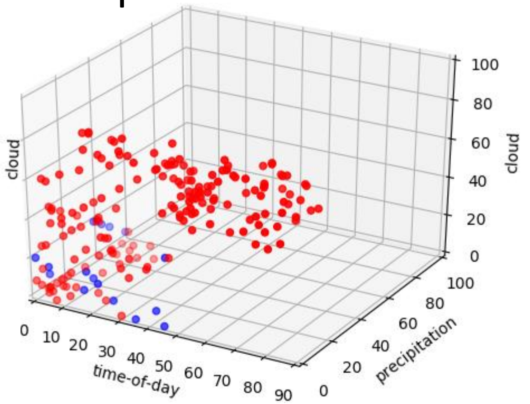
Grid Search



Random Neighborhood Search



Guided Bayesian Optimization



Scenes with Tolerable Risk

High-Risk Scenes

Key Results

- We sampled 250 test cases using each of these samplers
- Our samplers better balance the exploration vs. exploitation

1. Chen, Dian, et al. "Learning by cheating." *Conference on Robot Learning*. PMLR, 2020.

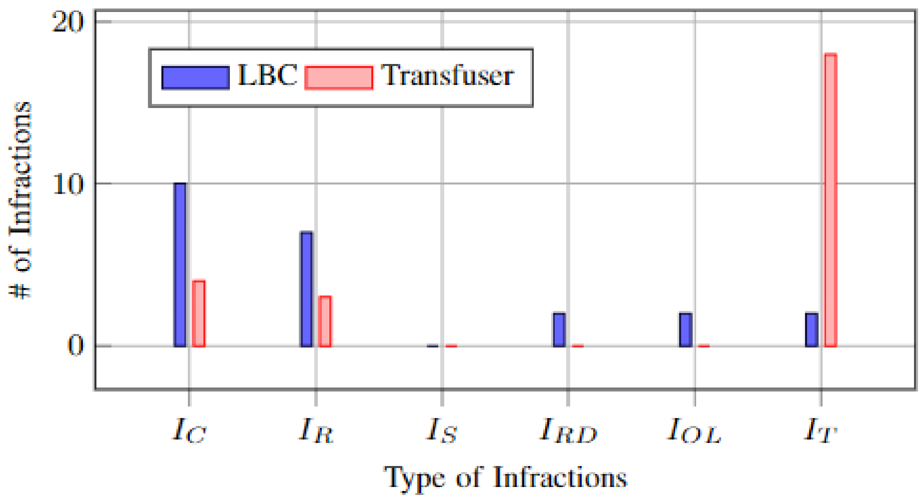
Github - https://github.com/scope-lab-vu/ANTI-CARLA

Quantitative Comparison of Samplers

Quantitative comparison of samplers across 250 simulations

Sampler	Total risk scenes (%)	Diversity		Search Time (min)
		# of clusters	Silhouette score	
Random	66	3	0.34	323
Halton	71	2	0.27	315
Grid	56	2	0.71	309
RNS	83	6	0.56	332
GBO	92	4	0.62	897

Comparing infractions of different LECs[1,2]



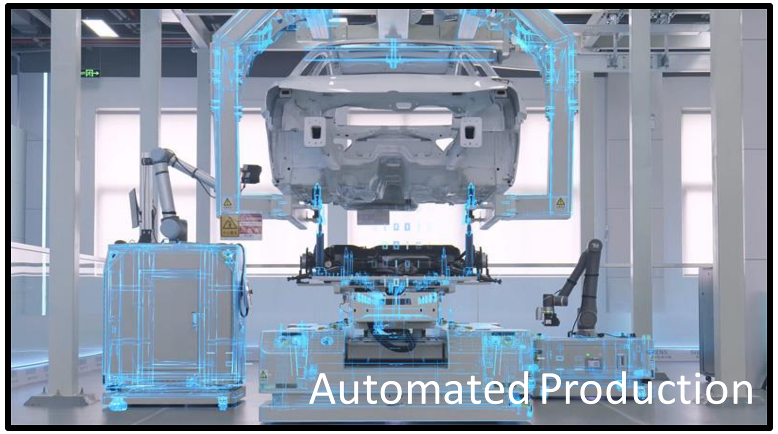
Key Results

- Our Samplers generated higher total high-risk scene percentage compared to the baselines, across different starting conditions
- We also tested different controllers across different towns in the simulator

1. Chen, Dian, et al. "Learning by cheating." *Conference on Robot Learning*. PMLR, 2020.
2. Prakash, Aditya, et al. "Multi-modal fusion transformer for end-to-end autonomous driving." *Conference on Computer Vision and Pattern Recognition*. 2021.

Conclusion

Safety Assurance of Autonomous CPS



- Safety Assurance has always been a **known problem** of CPS
- This problem has been **further exacerbated** with the use of **LECs**
- **New assurance approach** is needed to incorporate:
  - Changing operational nature of CPSs
  - New components (e.g., LECs)

Contributions of this Dissertation

Assurance Case Development

Contributions:

- A workflow for automatic synthesis of an assurance case
- Coverage metrics and an analysis report for evaluating the assurance case

Out-of-Distribution Detection

Contributions:

- Workflow for designing an **efficient detector** that performs detection on **low-**dimensional space
- **Bayesian Optimization heuristic** to design and train the detector
- OOD responsible **feature identification**

Mitigation

Contributions:

- A blended-simplex strategy called "**Weighted Simplex Strategy**" to overcome (a) conservatism of the decision logic, and (b) avoid instantaneous controller transition – **RL algorithm**
- The "**Dynamic Simplex Strategy**" with a non-myopic planner for reverse switching aimed to improve the system's performance without compromising on safety – **MCTS online heuristic**

Risk Assessment

Contributions:

- Proactive risk assessment framework called "**ReSonAte**"
- Combine design-time hazard rate with runtime system monitors to compute the system's operational risk

Data Generation

Contributions:

- Adversarial data generation framework "**ANTI-CARLA**"
- A scenario description language
- Two adversarial samplers

Major Publications

Assurance Case Development

Ramakrishna, S., Jin, H., Dubey, A. & Ramamurthy, A. “Automating Pattern Selection for Assurance Case Development of Cyber-Physical Systems”. Accepted at SafeComp 2022, Pending Publication.

Out-of-Distribution detection

- **Ramakrishna, S.**, Rahiminasab, Z., Karsai, G., Easwaran, A., & Dubey, A. (2021). "Efficient Out-of-Distribution Detection Using Latent Space of β-VAE for Cyber-Physical Systems." in TCPS 2020
- **Ramakrishna, S.**, Rahiminasab, Z., Easwaran, A., & Dubey, A. (2020, September). "Efficient Multi-Class Out-of-Distribution Reasoning for Perception Based Networks: Work-in-Progress." In *2020 International Conference on Embedded Software (EMSOFT)*

Risk Assessment

Hartsell, C.*, **Ramakrishna, S.***, Dubey, A., Stojcsics, D., Mahadevan, N., & Karsai, G. (2021). "ReSonAte: A Runtime Risk Assessment Framework for Autonomous Systems". In SEAMS 2021

Adaptive Mitigation

- **Ramakrishna, S.**, Harstell, C., Burruss, M. P., Karsai, G., & Dubey, A. (2020). "Dynamic-weighted simplex strategy for learning enabled cyber physical systems." *Journal of systems architecture*
- **Ramakrishna**, S., Luo B., Kuhn, C., Mukhopadhyay, A., Karsai, G., and Dubey, A. "Dynamic Simplex Strategy for autonomous CPS." pending submission

Data Generation

- **Ramakrishna, S.**, Luo, B., Barve, Y., Karsai, G., & Dubey, A. (2021). "Risk-Aware Scene Sampling for Dynamic Assurance of Autonomous Systems“. In ICAA 2021
- **Ramakrishna, S.***, Luo, B.*, Kuhn, C. , Karsai, G., & Dubey, A. “ANTI-CARLA: An Adversarial Testing Framework for Autonomous Vehicles in CARLA”. Accepted at ITSC 2022, Pending Publication

Other Publications

- Sundar, V. K., **Ramakrishna, S.**, Rahiminasab, Z., Easwaran, A., & Dubey, A. (2020, May). "Out-of-distribution detection in multi-label datasets using latent space of β-VAE". In *2020 IEEE Security and Privacy Workshops (SPW)*
- **Ramakrishna, S.**, Hartsell, C., Dubey, A., Pal, P., & Karsai, G. (2020). "A Methodology for Automating Assurance Case Generation". In TMCE 2020
- **Ramakrishna, S.**, Dubey, A., Burruss, M. P., Hartsell, C., Mahadevan, N., Nannapaneni, S., & Karsai, G. (2019, May). "Augmenting learning components for safety in resource constrained autonomous robots." In *2019 IEEE 22nd International Symposium on Real-Time Distributed Computing (ISORC)*
- Burruss, M. P., **Ramakrishna, S.**, Karsai, G., & Dubey, A. (2019, May). "Deepnncar: A testbed for deploying and testing middleware frameworks for autonomous robots." In *2019 IEEE 22nd International Symposium on Real-Time Distributed Computing (ISORC)*
- Hartsell, C., Mahadevan, N., **Ramakrishna, S.**, Dubey, A., Bapty, T., Johnson, T., Koutsoukos, X., Sztipanovits, J. and Karsai, G., 2019, October. Cps design with learning-enabled components: A case study. In *Proceedings of the 30th International Workshop on Rapid System Prototyping (RSP'19)* (pp. 57-63).
- Hartsell, C., Mahadevan, N., **Ramakrishna, S.**, Dubey, A., Bapty, T., Johnson, T., Koutsoukos, X., Sztipanovits, J. and Karsai, G., 2019, April. Model-based design for CPS with learning-enabled components. In *Proceedings of the Workshop on Design Automation for CPS and IoT* (pp. 1-9).
- Burruss, M., **Ramakrishna, S.** and Dubey, A., 2021, August. Deep-rbf networks for anomaly detection in automotive cyber-physical systems. In *2021 IEEE International Conference on Smart Computing (SMARTCOMP)* (pp. 55-60). IEEE.
- Hartsell, C., Mahadevan, N., **Ramakrishna, S.,** Dubey, A., Bapty, T. and Karsai, G., 2019. Demo Abstract: A CPS Toolchain for Learning-based Systems.

Thank you!

Committee

- Abhishek Dubey
- Janos Sztipanovits
- Gabor Karsai
- Xenofon Koutsoukos
- Arun Ramamurthy
- Ayan Mukhopadhyay

Collaborators and mentors

- Ted Bapty
- Arvind Easwaran
- Nagabhushan Mahadevan
- Daniel Stojcsics
- Baiting Luo
- Christopher Kuhn
- Charles Hartsell
- Patrick Masau
- Zahra Rahiminasab
- Vijaya Kumar Sundar

DARPA

Institute for Software Integrated Systems

Scope Lab: Scott Eisele, Geoffrey Pettet, Michael Wilbur, Rishav Sen, Jose Paolo Talusan