2021 IEEE Intelligent Vehicles Symposium (IV)
July 11-17, 2021. Nagoya, Japan

Generation of Modular and Measurable Validation Scenarios for Autonomous Vehicles Using Accident Data

Quentin Goss, Yara AlRashidi and Mustafa İlhan Akbaş
Department of Electrical Engineering and Computer Science
Embry-Riddle Aeronautical University
Daytona Beach, Florida
{gossq,alrashy1}@my.erau.edu, akbasm@erau.edu

2021 IEEE Intelligent Vehicles Symposium (IV) | 978-1-7281-5394-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/IV48863.2021.9575506

***Abstract*—Autonomous vehicle (AV) technology is positioned to have a significant impact on various industries. Hence, artificial intelligence powered AVs and modern vehicles with advanced driver-assistance systems have been operated in street networks for real-life testing. As these tests become more frequent, accidents have been inevitable and there have been reported crashes. The data from these accidents are invaluable for generating edge case test scenarios and understanding accident-time behavior. In this paper, we use the existing AV accident data and identify the atomic blocks within each accident, which are modular and measurable scenario units. Our approach formulates each accident scenario using these atomic blocks and defines them in the Measurable Scenario Description Language (M-SDL). This approach produces modular scenario units with coverage analysis, provides a method to assist in the measurable analysis of accident-time AV behavior, identifies edge scenarios using AV assessment metrics.**

***Index Terms*—Autonomous vehicles, AV Crashes, Simulation, Validation, Scenario Generation**

I. INTRODUCTION

With the advances in sensory systems and artificial intelligence (AI) engines, advanced driver-assistance systems (ADAS) and autonomous features have been increasingly deployed in vehicles. As these features continue to be deployed and tested, there have been accidents reported for vehicles while operating in autonomous mode [1]. These accidents create public concern, which can be an obstacle for the deployment of autonomous vehicle (AV) technology.

The AVs are expected to be operating on street networks for long lifetimes, which increases the probability of every bug surfacing during operation. Therefore, even the bugs in corner cases must be methodologically found. Additionally, the test progress must be transparent for assessment. Hence, AV accidents provide invaluable data to analyze and use for testing and validation.

This paper presents a methodology for using the functional accident scenarios involving vehicles operating in autonomous mode to create both logical and concrete test scenarios for AV validation. Our approach takes the functional description of each accident scenario in crash reports and divides it into multiple abstract atomic blocks. Then, these blocks are used as modular units to analyze and build-up test scenarios. The methodology focuses on the scenario-based validation of decision-making in AVs. It provides test scenarios by recreating accidents and their variations using the Measurable Scenario Description Language (M-SDL) [2].

The atomic blocks used to formulate scenarios are modular and measurable scenario components. The small size and brevity of the atomic blocks provide reasonable testing complexity, while multiple atomic blocks can be linked together in serial and parallel to create complex scenarios. The coverage analyses for scenario parameters provide a measurable testing platform and enables progress tracking. Overall, the methodology assists in validation by recreating edge accident scenarios and their cousin scenarios, reducing the number of scenarios, and providing coverage analysis. Hence, the main contributions of this paper are as follows:

- We present the atomic block, a modular and measurable scenario unit in logical abstraction level for AV validation.
- We utilize accident report data in functional abstraction level to create atomic blocks in logical abstraction level and concrete scenarios in simulation.
- We create an evolving database of atomic blocks that can be combined to generate complex scenarios.
- We implement AV assessment processes to identify potential edge scenarios for future iterations and mutations of atomic blocks and complex scenarios.

The remainder of the paper is organized as follows. Related work is given in Section II. The methodology of our scenario-based approach in Section III. We present the implementation details and results in Section IV and conclude in Section V.

II. Related Work

Validation and verification of AI-controlled AVs has been a vexing issue, and considerable effort and funding has been directed towards providing safe autonomous systems [3].

There are studies on using process-based methods and setting scope requirements for an overarching safety case of AVs [4], [5]. This approach has the potential to provide standard safety practices and identify the aspects of the system to be validated. However, it must be complemented with measurability features and performance indicators in practical usage [6] to track progress. Without measurable performance indicators and precise metrics, the progress cannot be tracked, and the development cannot be adjusted for the progressive improvement of safety.

Another critical approach in the validation and verification of AVs has been the constructive bottom-up proof process [7] for testing. This approach has been successful in verification for transformation of requirements into design. However, it has limitations on validating whether the transformations actually fulfill the intended functionalities or not [3], [8].

The pseudo-random test generation is also used in AV validation, and in some cases, this approach is combined with formal methods [9], [10]. This approach creates significant benefits in several areas, such as falsification or AI training. On the other hand, it must be complemented with abstraction for the system functionality. Most of the scenario generation work done today relies on manual design or minimal situational analysis with automation. Therefore, abstraction is critical for separating the validation concerns and ensuring the scalability for validation in complex systems [11].

There have been research initiatives to collect crash data for analysis [1], [12]. Gietelink and Verburg [13] gave an early example for crash scenario analyzers to create a tool for validation and design of pre-crash systems. The crash decisions are determined based on the velocity of the vehicle under test and the trajectory. The developed database is divided up into crash scenarios, near-collision scenarios, and normal driving scenarios. Wolschke et al. [14] suggest generating scenarios by observing the changes of input and output values. A new scenario is added to the proposed suite only if it can be categorized as different compared to existing scenarios. In another study, Xinxin et al. [15] used accident reports as a source for critical scenario generation.

We focus on abstract scenario testing, and the scenario information encoding is critical for this level of AV testing. The prominent scenario description languages include M-SDL [16], OpenSCENARIO [17], and Scenic by Fremont et al. [18]. OpenSCENARIO uses XML to describe the scenarios, whereas M-SDL and Scenic take a tabbed scenario description approach.

III. Scenario-Based Approach

We use a scenario-based approach for AV validation testing featuring our atomic blocks. To present our approach we use the six-part taxonomy introduced by Riedmaier et al. [19]. The framework of our contribution is presented in Fig. 1.

A. Sources for Scenarios

The first part of our framework is the source of functional scenarios. In this paper, our scenario sources are the AV accident reports, which are forms filled out after an AV is involved in an accident to record the “who”, “when”, “where”, “what”, and “how” of the accident. Some examples of the AV accident reports can be found in the “report of a traffic collision involving an autonomous vehicle” forms published by the California Department of Motor Vehicles [20]. A high-quality AV accident report will describe the actions of all vehicles and humans in the accident leading up to the time of the accident. An accident report is a human-readable version of an AV scenario, otherwise known as a functional scenario.

B. Scenario Generation/Extraction

In this part, functional scenarios in the form of AV accident reports are transformed into modular and measurable logical scenarios with ranges which we name ‘*atomic blocks*’. This is a two-step process of classification and then parameterization.

1) Classification: During classification, the “who/what” and “where” are identified. The ”who/what” are actors in a functional scenario. There are two types of actors, the AV, which we refer to as the Device Under Test (DUT), and the non-AV actor, which we refer to as the Non-Player Character (NPC). Its type describes the default function of an actor, and we use the terms in Equation 1 to define our actor type T.

$$T = \{s_{\min}, s_{\max}, a_{\min}, a_{\max}, \sigma\} \tag{1}$$

where T represents the actor type, s is actor speed, a is actor acceleration and σ represents adherence to speed and acceleration rules, where 0 is perfect and 1 is least adherent.

Three common examples of actor types are ‘car’, ‘pedestrian’ and ‘stationary object’, which are presented in Table I. We define the properties of these sample actors based on the vehicle types used in the SUMO traffic simulation [21]. Speed s is in kilometres per hour (kmh), and acceleration a is in kmh per second (kmh/s).

TABLE I: Example vehicle types T.

Name	$s_{\min}$	$s_{\max}$	$a_{\min}$	$a_{\max}$	σ
Car	0.00kmh	180.00kmh	−32.40kmh/s	10.44kmh/s	[0..1]
Pedestrian	0.00kmh	5.40kmh	−18.00kmh/s	5.40kmh/s	[0..1]
Stat. Obj	0.00kmh	0.00kmh	0.00kmh/s	0.00kmh/s	[0..1]

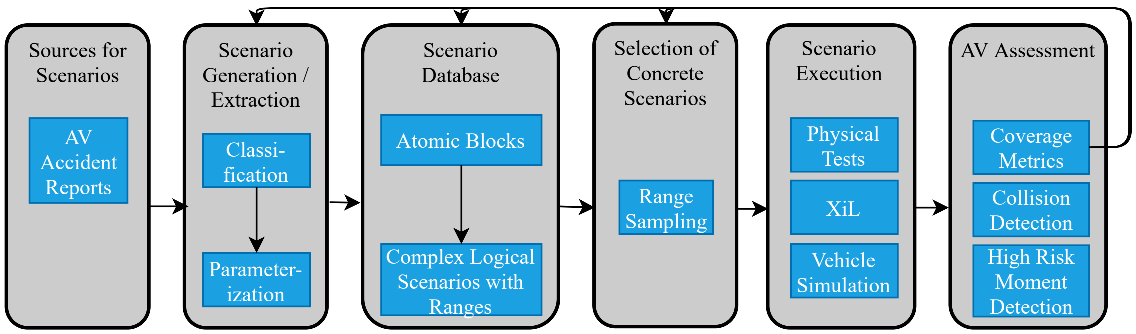


Fig. 1: The framework of our scenario-based approach, where the elements reflect our process from left to right.

To classify "where", we utilized a real AV accident dataset, which includes 97 functional scenarios aggregated from the AV accident reports from Florida, Texas, and California. We surveyed the street networks where accidents took place and classified them into five location descriptions, which we attribute with a letter and define as a path requirement.

The first letter we used in location descriptions is S, chosen for its resemblance to an S-curve. Scenarios classified by the path requirement S consist of one or more connected street network edges. This may include traversing an intersection as long as all actors traverse the intersection in the same way.Another letter we use is T, chosen for its resemblance to a T-intersection. Scenarios classified by the path requirement T include a 3-way intersection. We use X for its resemblance to two segments crossing. Scenarios classified by the path requirement X include a 4-way intersection. The last letter we used in this study is M, chosen for the five vertices of the four edges that make up the capital letter M. Scenarios classified by the path requirement M include a 5-way intersection.

2) Parameterization: The parameterization of a functional scenario is a tackled with the "atomic block" concept. An atomic block is a modular and measurable scenario that can function alone or be combined with other atomic blocks to create a complex scenario. During parameterization, an atomic block is perceived as a functional scenario.

We surveyed our dataset and identified 62 functional atomic blocks.To create a naming convention for the blocks, we devised a classification method that comprises the number of actors excluding the DUT, a path requirement, whether a pedestrian is or is not involved, and an increasing index starting from 1. Scenarios that include a pedestrian actor include the designation P. The number of non-DUT actors in a scenario currently ranges [1..3].

As an example, bl_2TP3 is broken down as follows: The prefix "bl_" identifies this item as an atomic block. "2" means there are two non-DUT actors, "T" means the scenario occurs at a 3-way intersection, "P" means a pedestrian actor is included and "3" means that it is the third unique scenario belonging to the classification of 2TP. The actors of the scenario bl_2TP3 are DUT, NPC_1, NPC_2, and NPC_3 where NPC_3 is a pedestrian.

By the end of parameterization, a functional scenario will be translated into one or more atomic blocks, such as the template for the atomic block B in Equation 2.

$$\begin{aligned} B &= \{\text{DUT}, \text{NPC}, \ldots\} \\ \text{DUT} &= \{T, s_{\text{start}}, s_{\text{end}}, p_{\text{start}}, p_{\text{end}}\} \\ \text{NPC} &= \{T, s_{\text{start}}, s_{\text{end}}, p_{\text{start}}, p_{\text{end}}\} \end{aligned} \tag{2}$$

To understand the translation, we explain the logical version of the atomic block in detail. Every atomic block has at least two actors, the AV, referred to as a device under test (DUT), and one more actor, referred to as a non-player character (NPC). The $(\ldots)$ represent additional actors and constraints such as a second NPC or lane change maneuvers. We expose the actor type T, and the starting and ending speed s and position p. If T, s, and p are not modified, the actor will perform the default behavior. The DUT is the anchor of the block, and we refrain from constraining the DUT when possible since the DUT is the AV that is being validated.The NPC actions are described with respect to the DUT.

Equation 2 is a template for an atomic block. If a simulation was constructed from Equation 2, it would consist of a DUT and an NPC operating within the constraints of their respective actor types. The NPC may change speed or position unpredictably. Practically, constraints are necessary to create a meaningful atomic block for describing scenarios.

To explain the atomic block, we use two scenarios with a DUT actor, an NPC actor, no pedestrian, and path requirement of S. We will work through an example of creating two atomic blocks from a functional scenario described by an AV accident report. We will use the following scenario:*"An AV was involved in a collision while operating in autonomous mode, traveling westbound on McAllister St. at the intersection of Polk Street. Another vehicle attempted to pass the AV* *on the right but crossed into its lane, scraping the AV's right sensor. There were no injuries, police were not called."* [20].

From the functional scenario, we abstract that the DUT is the "AV", and the NPC is "Another vehicle". In Fig. 2 we visualize the functional scenario where we infer one of many possible point-of-contacts with the knowledge that the DUT's right sensor was scraped in a non-serious collision. We plot the trajectory of the NPC in the functional scenario and split the trajectory into two. We choose to split at this specific point since there are also other functional scenarios in our dataset,, which consist of this "cut in from the side" maneuver, without the "approach on the side" maneuver. Therefore, we consider this difference as an interesting trajectory.

In Fig. 3a, we expand the "approach" trajectory by extruding the lane, speed, and relative distance of the NPC from the DUT. The result is a logical scenario, i.e. the first atomic block (bl_1). Atomic block bl_1 is an "approach on the side" logical scenario, described by Equation 3.

$$\begin{aligned} B_{\text{bl\_1}} &= \{\text{DUT}, \text{NPC}, \ldots\} \\ \text{DUT} &= \{T, s_{\text{start}}, s_{\text{end}}, p_{\text{start}}, p_{\text{end}}\} \\ \text{NPC} &= \{T, s_{\text{start}}, s_{\text{end}}, p_{\text{start}}, p_{\text{end}}\} \\ \text{NPC}.s_{\text{start}} &= [0..80.467]\text{kph} + \text{DUT}.s_{\text{start}} \\ \text{NPC}.p_{\text{start}} &= [-30.480..3.048]\text{m} + \text{DUT}.p_{\text{start}} \\ \text{NPC}.p_{\text{end}} &= [-3.048..3.048]\text{m} + \text{DUT}.p_{\text{end}} \end{aligned} \tag{3}$$

Here we use "." as the notation for "belonging to". The NPC starting speed is 0 to 80.467kmh faster than the DUT speed. The NPC position at the start is up to 30.480m behind and up to 3.048m ahead of the DUT. The NPC position at the end is within 3.048m ahead or behind the DUT. Throughout the scenario, the NPC lane, NPC.l, is the lane to the side of the DUT. There is no constraint on the DUT.

In Fig. 3b, we expand the "cut in" trajectory by extruding the starting lanes of the NPC while keeping the ending lane same as the DUT. Since the accident description describes an "approach" leading into a "cut-in", the range of starting distance of the NPC from the DUT at the beginning of the "cut in" is the same as the range of ending distance of the NPC from the DUT at the end of the approach. The result is the second atomic block (bl_2), which is a cut in from the side logical scenario described by Equation 4.

$$\begin{aligned} B_{\text{bl\_2}} &= \{\text{DUT}, \text{NPC}, \ldots\} \\ \text{DUT} &= \{T, s_{\text{start}}, s_{\text{end}}, p_{\text{start}}, p_{\text{end}}\} \\ \text{NPC} &= \{T, s_{\text{start}}, s_{\text{end}}, p_{\text{start}}, p_{\text{end}}\} \\ \text{NPC}.s_{\text{start}} &= [0..80.467]\text{kph} + \text{DUT}.s_{\text{start}} \\ \text{NPC}.s_{\text{end}} &= \text{NPC}.s_{\text{start}} \\ \text{NPC}.p_{\text{start}} &= [-3.048..3.048]\text{m} + \text{DUT}.p_{\text{start}} \\ \text{NPC}.p_{\text{end}} &= [-3.048..9.144]\text{m} + \text{DUT}.p_{\text{end}} \\ \text{NPC}.l_{\text{start}} &= \text{Side of DUT.} \\ \text{NPC}.l_{\text{end}} &= \text{DUT}.l_{\text{end}} \end{aligned} \tag{4}$$

In bl_2, the NPC starting speed is 0 to 80.467kmh faster than the DUT starting speed. The NPC position at the start is within 3.048m ahead or behind the DUT. The NPC position at the end is within 9.144m ahead or 3.048m behind the DUT. The NPC starts in the lane to the side of the DUT and ends in same lane as the DUT. There is no constraint on the DUT.

As a result, a functional scenario is processed into bl_1 and bl_2 in this example. The atomic blocks are logical scenarios that are the outputs of the second stage in our framework.

C. Scenario Database

In part three of the framework, scenarios are stored in a scenario database. There are two groups of scenarios that can be produced using this database. The first group includes atomic blocks generated during part two of the framework and stored in the database. The second group includes the complex logical scenarios created by concatenating atomic blocks rather than generating them during the scenario generation/extraction part of the framework. For example, bl_1

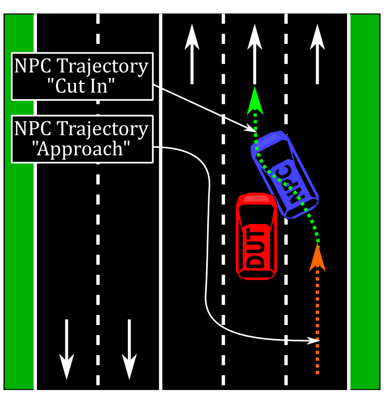
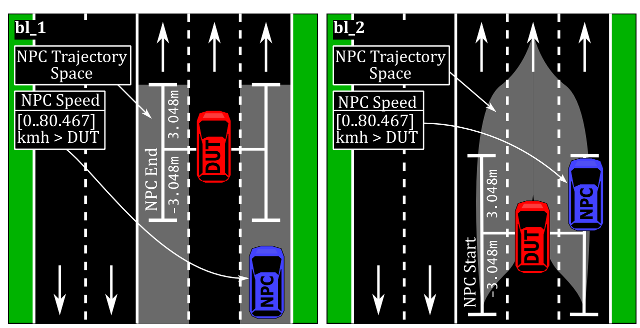


Fig. 2: The trajectory of an AV functional scenario.



(a) NPC approaches DUT. (b) NPC cuts into DUT's lane.

Fig. 3: Two "atomic blocks" generated by using the trajectories in a functional scenario.

and bl_2 may be combined to form a more complex scenario. We will use "+" to represent an atomic block starting right after the previous atomic block. Possibilities are bl_1 + bl_2, and bl_2 + bl_1, and bl_1 + bl_1 + bl_2 + . . . + bl_2, and so forth. Both atomic blocks and their combinations can be used as input to the next part of the framework.

D. Selection of Concrete Scenarios

In part four of the framework, concrete scenarios are generated using the logical scenarios in "Scenario Database." Our approach uses range sampling, which means that a value is picked from each range in the logical scenario until a concrete value has been chosen for each range. In addition, parameters with no default value in the scenario may be set, such as the seed for the pseudo-random number generator or the street network where the concrete simulation will execute. Lastly, a default parameter may be overloaded to allow for polymorphism, as required when tuning a scenario or mutating an existing scenario for a unique use case.

To demonstrate an example, we combine blocks bl_1 and bl_2 to form a complex scenario called "bl_1 + bl_2" and show the ranges that we wish to have sampled in Table II.

TABLE II: Example of range sampling for the complex scenario bl_1 + bl_2.

Parameter	Range	Sample
seed	Unique value.	42
street network	circ_fix	circ_fix
First atomic block: bl_1		
NPC.$s_{\text{start}} \geq$ DUT.s_{start}	[32.187..96.561]kph*	56kph
NPC.$p_{\text{start}} \geq$ DUT.p_{start}	[-30.480..3.048]m	-28m
NPC.$p_{\text{end}} \geq$ DUT.p_{end}	[-3.048..3.048]m	1m†
Second atomic block: bl_2		
NPC.$s_{\text{start}} \geq$ DUT.s_{start}	[0..80.467]kph	80kph
NPC.$p_{\text{start}} \geq$ DUT.p_{start}	[-3.048..3.048]m	1m†
NPC.$p_{\text{start}} \geq$ DUT.p_{start}	[-3.048..9.144]m	7m
NPC.l_{start}	Side of DUT.	Left of DUT.

*Default range overloaded, †Samples are linked.

In the parameter column, we have parameters that belong to bl_1 and bl_2. We have parameters that belong to the complex scenario bl_1 + bl_2, which is the seed for the pseudo-random number generator and the street network. These two parameters will be inherited by atomic blocks bl_1 and bl_2. The range column is the set of valid items to be sampled, and the sample column is the choice of a sampling algorithm. The range NPC.$s_{\text{start}} \geq$ DUT.s_{start} of bl_1 has been overload from [0..80.467]kmh to [32.187..96.561]kmh.

A notable feature of this range sampling is the linking of two samples. The samples NPC.$p_{\text{end}} \geq$ DUT.p_{end} of bl_1 and NPC.$p_{\text{start}} \geq$ DUT.p_{start} of bl_2 are linked since bl_2 starts right after bl_1, and the actor positions at the end of bl_1 will be the actor positions at the start of bl_2. A linked parameter is sampled once, and the value is distributed to both parameters. When a value is chosen for each range, a concrete scenario is generated.

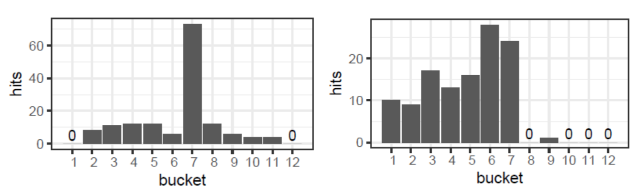
E. Scenario Execution

In the fifth part of the framework, a concrete scenario is executed. Depending on multiple factors such as scenario risk, fiscal budget, or scope, the role of simulation in scenario execution would change. An option for scenario execution may be physical testing on a test track or on a public street network. Another option is a form of in-the-loop simulation (XiL) where parts of the DUT are connected to a simulator. The scenario can also be run entirely in simulation, which may take several different forms depending on the desired fidelity level and the test dimension. In this paper, we focus on the abstract scenarios in simulation and the validation of decision-making systems of AVs.

F. AV Assessment

The last part of the framework is where the results of the concrete scenarios are evaluated. We employ three assessments: 1) coverage metrics, 2) collision detection, and 3) high-risk moment detection.

1) Coverage Metrics: Coverage metrics are used to measure completeness or whether all cases within a certain granularity are met. For instance, consider that we ran various tests of a complex scenario bl_1 + bl_2 using the ranges in Table II and it is decided that every case of DUT.p_{end} − NPC.p_{end} in bl_1 must be in the range of [-3.048..3.048]m with a granularity of 0.610m. To check for edge cases, we expand the range to be [-3.658..3.658]m. During each test at the end of bl_1 DUT.p_{end} − NPC.p_{end} is evaluated. That value is considered a hit, and the value of bucket of [-3.658..-3.048]m, or [-3.048..-2.438]m, ... , or [3.048..3.658]m is incremented by one until all tests are complete. After all, tests are finished and hits are collected into buckets, a histogram is generated, such as the histograms in Fig. 4, where the x-axis is the bucket index. If the histogram looks like Figure 4(a), where the edge cases (buckets 1 and 12) have zero hits, and the buckets between (buckets 2-11) have one or more hits, then



(a) Complete (b) Incomplete

Fig. 4: Examples of complete and incomplete coverage.

all case are confidently met. If the histogram looks like Figure 4(b), where the histogram is skewed to one side, and there are gaps in middle buckets such as bucket 8, which would be [1.219..1.829]m, that means that not all cases are met. In this event, it necessary to adjust the coverage, execute more concrete scenarios, or check for defects in the atomic blocks or scenario generation/extraction process.

G. Collision Detection

A collision happens when the DUT of a scenario crashes an NPC. Information about each actor in the scenario is collected at the instant of a collision, such as the actor's speeds and positions on the street network.

H. High Risk Moment Detection

A high-risk moment is a moment in time during concrete scenario execution when the DUT has a high probability of performing unfavorably. In addition, this metric may be used to detect edge case scenarios where the DUT performs unexpectedly when the scenario is mutated with injected faults, change of DUT logic, change in the scenario execution environment, etc. High-risk moment detection is a binary classification described by Equation 5.

$$\psi = \begin{cases} \text{KPI} \leq \tau & \text{Fail} \\ \text{KPI} > \tau & \text{Pass} \end{cases} \tag{5}$$

A high-risk moment is found when ψ has measured a point in time and evaluates to a "fail". A key performance indicator (KPI) is compared against a threshold τ. As τ moves towards ∞, the likelihood of a high-risk moment increases. A KPI metric may be any type of metric such as temporal or distance variants in recent scenarios based tests for AVs [22].

IV. Implementation

This section presents an implementation example to demonstrate how our modular and measurable atomic blocks are helpful in a scenario-based approach to AV validation. We construct an experiment with three scenarios from our scenario database: atomic blocks bl_1 and bl_2, and the complex logical scenario bl_1 + bl_2. The ranges are left at the defaults as they are in Equation 3 for bl_1 and Equation 4 for bl_2. The logical scenarios are constructed in the scenario description language M-SDL [16]. We use Foretify™ [23] by Foretellix for concrete scenario generation and SUMO for vehicle simulation during scenario execution.

For each of the three logical scenarios, we generate 50 concrete scenarios on three street networks. That is, scenarios 1-50 happen on the street network long_single_road_2, which is a long bending road, scenarios 51-100 happen on the street network circ_fix, a figure-eight, Furthermore, scenarios 101-150 happen on the street network Town04, a grid city. Foretify automatically provides coverage metrics to tune the atomic blocks bl_1 and bl_2 defaults.

To perform collision detection and high-risk moment detection, we create an extension to the logical scenarios to trace the status of the DUT and NPC at 20ms intervals during scenario execution. This trace captures the unique concrete scenario id, seed, street network, actor type (i.e., DUT or NPC), x and y position values, speed, the time elapsed since the start of the simulation, and if a collision is happening at the moment, i.e. (1 or 0).

A. Collision Detection

In Table III, we record collisions which occurred during the concrete scenario simulations.

TABLE III: Collisions

Street Network	bl_1	bl_2	bl_1 + bl_2
long_single_road_2	0 (0%)	44 (88%)	33 (66%)
circ_fix	1 (2%)	48 (95%)	43 (86%)
Town04	0 (0%)	44 (88%)	31 (62%)
Total	1 ($0.\overline{6}$%)	136 ($90.\overline{6}$%)	107 ($71.\overline{3}$%)

We observe that bl_1 + bl_2 has fewer collisions than only bl_2 by 18.7%. By combining the two atomic blocks to form a complex logical scenario, we created a unique scenario. This is a result of linked samples during range sampling. The results in Table III show that our framework can be used to analyze scenarios in different street networks for collisions.

B. High Risk Moment Detection

We use two example KPIs to show the high risk moment detection feature. It must be noted that other KPI metrics are also applicable [22]. The first KPI (KPI_1) is distance and evaluated as follows:

$$\text{KPI\_1} = \sqrt{(\text{DUT}.x - \text{NPC}.x)^2 + (\text{DUT}.y - \text{NPC}.y)^2}$$

where $.x$ and $.y$ are x and y positions of their respecting actor. The second KPI (KPI_2) is time to collision (TTC) [24]. We evaluate KPI_2 as follows:

$$\text{KPI\_2} = \frac{\sqrt{(\text{DUT}.x - \text{NPC}.x)^2 + (\text{DUT}.y - \text{NPC}.y)^2}}{|\text{DUT}.s - \text{NPC}.s|}$$

At every 20ms intervals, we evaluate KPIs with our classification in Equation 5 with a range of τ values. In Table IV, we count the total number of concrete scenarios where the high-risk moment classification equation resulted in a "fail" at one or more moments in time. For example, for KPI_1 and $\tau = 2.438$ft on the street network Town04 for the scenario bl_2, 16 of 50 concrete scenarios were classified as containing at least one high-risk moment in time. These results show that our system can extract high-risk moments using any KPI and threshold τ defined in the system.

TABLE IV: High risk moment detection using KPIs.

Scenario	KPI_1 (Distance) $\tau = 2.438\text{m}$	$\tau = 3.048\text{m}$	$\tau = 3.658\text{m}$	KPI_2 (TTC) $\tau = 700\text{ms}$	$\tau = 1000\text{ms}$	$\tau = 1500\text{ms}$
Street network: long_single_road_2						
bl_1	0 (0%)	0 (0%)	37 (74%)	1 (2%)	9 (18%)	17 (36%)
bl_2	15 (30%)	37 (74%)	50 (100%)	0 (0%)	1 (2%)	5 (10%)
bl_1 + bl_2	11 (22%)	20 (80%)	50 (100%)	3 (6%)	13 (26%)	24 (48%)
Street network: circ_fix						
bl_1	0 (0%)	0 (0%)	35 (70%)	0 (0%)	6 (12%)	10 (20%)
bl_2	29 (58%)	36 (72%)	49 (98%)	0 (0%)	2 (4%)	5 (10%)
bl_1 + bl_2	30 (60%)	40 (80%)	50 (100%)	5 (10%)	10 (20%)	20 (40%)
Street network: Town04						
bl_1	0 (0%)	0 (0%)	41 (82%)	4 (8%)	8 (16%)	11 (22%)
bl_2	16 (32%)	28 (56%)	45 (90%)	0 (0%)	2 (4%)	11 (22%)
bl_1 + bl_2	19 (38%)	29 (58%)	49 (98%)	7 (14%)	13 (26%)	26 (52%)

V. Conclusion

This paper provides a methodology to construct modular and measurable scenarios that may be combined and reused to construct complex logical scenarios from real AV accident data. Then it explains how to execute the scenarios, utilize assessment metrics to improve them, and identify potential edge scenarios.We also construct an experiment with two atomic blocks and a complex scenario to demonstrate the framework's capabilities. As future work, we plan to automate the scenario generation and automatically translate new AV accident reports into the existing database.

VI. Acknowledgements

This work was partially supported by the "Modeling of Autonomous Vehicle Crashes Using Open Measurable Scenario Description Language" project with Foretellix Ltd.

References

[1] F. Favarò, S. Eurich, and N. Nader, "Autonomous vehicles' disengagements: Trends, triggers, and regulatory limitations," *Accident Analysis & Prevention*, vol. 110, pp. 136–148, 2018.

[2] Foretellix, "ASAM OpenSCENARIO: Version 2.0.0 Concepts ASAM P2019-02 Association for Standardization of Automation and Measuring Systems." https://releases.asam.net/OpenSCENARIO/2.0-concepts/ASAM_OpenSCENARIO_2-0_Concept_Paper.html, 2020.

[3] R. Razdan, M. İ. Akbaş, A. Sargolzaei, A. J. Alnaser, S. Sahawneh, S. Alsweiss, and J. Vargas, "Unsettled Technology Areas in Autonomous Vehicle Test and Validation," *Society of Automotive Engineers (SAE) EDGE^{TM} Research Report EPR2019001*, June 2019.

[4] P. Koopman, U. Ferrell, F. Fratrik, and M. Wagner, "A safety standard approach for fully autonomous vehicles," in *Conference on Computer Safety, Reliability, and Security*, pp. 326–332, Springer, 2019.

[5] I. ISO, "PAS 21448-Road Vehicles-Safety of the Intended Functionality," *International Organization for Standardization*, 2019.

[6] J. Burden and J. Hernández-Orallo, "Exploring ai safety in degrees: Generality, capability and control," in *SafeAI@ AAAI*, pp. 36–40, 2020.

[7] I. ISO, "26262: Road vehicles-functional safety," *International Standard ISO/FDIS*, vol. 26262, 2011.

[8] P. Koopman and M. Wagner, "Challenges in autonomous vehicle testing and validation," *SAE International Journal of Transportation Safety*, vol. 4, no. 1, pp. 15–24, 2016.

[9] S. Hallerbach, Y. Xia, U. Eberle, and F. Koester, "Simulation-based Identification of Critical Scenarios for Cooperative and Automated Vehicles," tech. rep., SAE Technical Paper: 01-1066, 2018.

[10] D. J. Fremont, E. Kim, Y. V. Pant, S. A. Seshia, A. Acharya, X. Bruso, P. Wells, S. Lemke, Q. Lu, and S. Mehta, "Formal scenario-based testing of autonomous vehicles: From simulation to the real world," in *IEEE ITSC*, pp. 1–8, IEEE, 2020.

[11] T. Menzel, G. Bagschik, and M. Maurer, "Scenarios for development, test and validation of automated vehicles," in *IEEE Intelligent Vehicles Symposium (IV)*, pp. 1821–1827, IEEE, 2018.

[12] M. Aydin and M. İ. Akbaş, "Identification of Test Scenarios for Autonomous Vehicles Using Fatal Accident Data," *SAE Journal of Connected and Automated Vehicles (JCAV)*, vol. 4, no. 1, 2021.

[13] O. Gietelink, D. Verburg, K. Labibes, and A. Oostendorp, "Pre-crash system validation with prescan and vehil," in *Intelligent Vehicles Symposium*, pp. 913–918, IEEE, 2004.

[14] C. Wolschke, T. Kuhn, D. Rombach, and P. Liggesmeyer, "Observation based creation of minimal test suites for autonomous vehicles," in *IEEE ISSREW*, pp. 294–301, 2017.

[15] Z. Xinxin, L. Fei, and W. Xiangbin, "Csg: Critical scenario generation from real traffic accidents," in *IEEE Intelligent Vehicles Symposium (IV)*, pp. 1330–1336, 2020.

[16] Foretellix Inc., "Measurable Scenario Description Language (M-SDL)," 9 2019. Version 0.9.

[17] T. Menzel, G. Bagschik, L. Isensee, A. Schomburg, and M. Maurer, "Detailing a Keyword Based Scenario Description for Execution in a Simulation Environment Using the Example of Scenarios on German Highways," in *Workshop Fahrerassistenzsysteme und automatisiertes Fahren*, vol. 12, pp. 15–26, 2018.

[18] D. J. Fremont, T. Dreossi, S. Ghosh, X. Yue, A. L. Sangiovanni-Vincentelli, and S. A. Seshia, "Scenic: a language for scenario specification and scene generation," in *Proceedings of the ACM SIGPLAN Conference on PLDI*, pp. 63–78, 2019.

[19] S. Riedmaier, T. Ponn, D. Ludwig, B. Schick, and F. Diermeyer, "Survey on scenario-based safety assessment of automated vehicles," *IEEE Access*, vol. 8, pp. 87456–87477, 2020.

[20] California Department of Motor Vehicles, "Autonomous vehicle collision reports." https://www.dmv.ca.gov/portal/vehicle-industry-services/autonomous-vehicles/autonomous-vehicle-collision-reports/, 2020. [Online; accessed 5-June-2020].

[21] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of SUMO - Simulation of Urban MObility," *International Journal On Advances in Systems and Measurements*, vol. 5, pp. 128–138, December 2012.

[22] D. Nalic, T. Mihalj, M. Bäumler, M. Lehmann, A. Eichberger, and S. Bernsteiner, "Scenario based testing of automated driving systems: A literature survey," in *FISITA Web Congress*, 2020.

[23] Foretellix, "Technology - Foretellix." https://www.foretellix.com/technology/, 2020.

[24] J. C. Hayward, "Near-miss determination through use of a scale of danger," *Highway Research Record*, 1972.