Translating Requirements into CARLA Executable Scripts: an LLM-Driven Automated Scenario Realization

1st Shiyang Guan
Waseda University
Tokyo, Japan
guanshiyang0509@toki.waseda.jp

2nd Yijun Lu
Waseda University
Tokyo, Japan
yijun@ruri.waseda.jp

3rd Jati H. Husen
Telkom University
Bandung, Indonesia
jatihusen@telkomuniversity.ac.id

4th Haowei Cheng
Waseda University
Tokyo, Japan
haowei.cheng@fuji.waseda.jp

5th Hironori Washizaki
Waseda University
Tokyo, Japan
washizaki@waseda.jp

6th Naoyasu Ubayashi
Waseda University
Tokyo, Japan
ubayashi@aoni.waseda.jp

7th Nobukazu Yoshioka
Waseda University
Tokyo, Japan
nobukazuy@acm.org

***Abstract*—Scenario-based testing is critical for ensuring the safety and robustness of autonomous driving (AD) systems, particularly in extreme scenarios such as heavy rain, pedestrian-involved crashes, and nighttime conditions. Our previous work integrated the CARLA simulator with the Multi-view Modeling Framework for ML Systems (M3S), facilitating scenario generation but still requiring substantial manual scripting. In this paper, we extend our framework by incorporating Large Language Models (LLMs) with M3S to automate scenario generation. Using a hierarchical prompt design, our approach extracts structured parameters from M3S descriptions into a JSON schema, which guides the LLM to generate accurate CARLA simulation scripts. Our evaluation demonstrates significant improvements in automation accuracy and efficiency, substantially reducing manual intervention and enhancing continuous testing cycles for AD systems.**

***Index Terms*—Autonomous Vehicles, CARLA Simulator, Large Language Models, Risk Analysis, Simulation-based Testing, Scenario Generation, Machine Learning, Model-based Systems Engineering.**

I. Introduction

Autonomous driving systems (ADS) are rapidly evolving due to significant advancements in artificial intelligence, sensor fusion, and control technologies [1], [4]. However, ensuring their safety and reliability remains challenging, particularly due to the inherent complexity and unpredictability of real-world scenarios, including adverse weather, dynamic urban intersections, and unpredictable pedestrian behaviors [2], [5]. To systematically address these challenges, researchers often utilize model-based risk analysis frameworks, such as the Multi-view Modeling Framework for ML Systems (M3S), to effectively identify and evaluate high-risk driving scenarios [6].

In our prior work, we developed an integrated workflow using the CARLA simulator within the M3S framework to systematically define safety-critical scenarios, generate multi-modal sensor datasets, and rigorously evaluate ADS through structured simulations [7]. While successful, this methodology heavily depended on manual efforts to translate M3S textual scenario descriptions into executable CARLA simulations, leading to considerable labor, errors, and inconsistencies.

Recent advancements in generative AI, particularly large language models (LLMs), present promising opportunities to automate this scenario translation process [3], [8], [9]. Leveraging these advancements, this paper extends our previous framework by integrating an LLM-based pipeline to automatically convert textual scenario descriptions from M3S into structured, executable Python scripts for CARLA. This automation significantly streamlines scenario realization, reduces manual intervention, minimizes errors, and accelerates scenario-based testing and data collection.

The primary contributions of this extended framework include: (1) proposing an innovative integration of LLMs with the M3S framework for automated scenario generation; (2) empirically demonstrating this method's effectiveness through practical evaluations; and (3) highlighting the improvements in accuracy, efficiency, and usability over our previous manual approach.

The remainder of this paper is structured as follows: Section 2 reviews related work on autonomous driving simulation, scenario generation, and generative AI applications. Section 3 outlines our methodology, detailing the workflow from textual scenario descriptions to executable CARLA scripts. Section 4 presents the experimental evaluations and results. Section 5 discusses key insights and implications of our findings. Section 6 concludes the paper and suggests future research directions. Finally, Section 7 includes the acknowledgment to the sponsor.

II. Background and Related Work

This section reviews existing research in autonomous driving system testing, scenario generation, and the emerging role of large language models in automating these processes.

A. Simulation-based Testing and M3S Framework

Developing safe autonomous driving systems requires rigorous testing across diverse scenarios, including edge cases such as extreme weather and complex urban environments [10], [11]. Real-world testing poses significant challenges including prohibitive costs, safety risks, and difficulty reproducing specific conditions [12]. Consequently, high-fidelity simulation environments like CARLA have become indispensable tools [4]. The Multi-view Modeling Framework for Machine Learning Systems (M3S) [13] provides a systematic approach to integrating ML components within larger systems while ensuring traceability and alignment with system requirements. Our previous research [7] extended M3S by integrating it with CARLA to systematically define safety-critical scenarios and generate multi-modal datasets. However, a key limitation was the substantial manual effort required to translate textual scenario descriptions from M3S into executable CARLA simulations.

B. LLM-based Scenario Generation

Recent advancements in large language models have created new opportunities for automating scenario generation. Cai et al. [8] introduced Text2Scenario, demonstrating how LLMs can translate natural language descriptions into executable driving scenarios. Similarly, Mei et al. [9] proposed a framework for generating collision scenarios through targeted prompt engineering. While these approaches show promise, several challenges remain: (1) lack of structured parameter extraction aligned with established modeling frameworks like M3S, (2) limited comprehensive pipelines from high-level descriptions to executable scripts without human intervention, and (3) insufficient assessment of scenario quality and reproducibility in safety-critical applications. Our work addresses these gaps by proposing a hierarchical prompt design that systematically extracts structured parameters from M3S descriptions into JSON schema, subsequently guiding LLMs to generate accurate CARLA simulation scripts. This approach combines structured system modeling benefits with LLM automation capabilities, enabling more efficient autonomous driving system testing.

III. Methodology

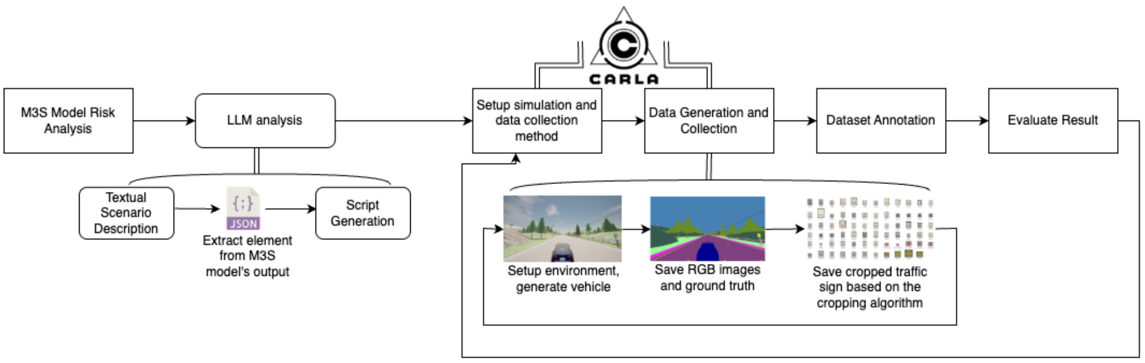
Our integrated simulation-based approach combines the Multi-view Modeling Framework for ML Systems (M3S) with the CARLA simulator to create a comprehensive pipeline for autonomous driving safety analysis. This methodology consists of four key components: (1) M3S Integration and Requirement Analysis, (2) Hierarchical Scenario Generation, (3) LLM Integration Details, (4) Multi-modal Dataset Collection, and (5) Evaluation Framework.

A. M3S Integration and Requirement Analysis

The foundation of our approach lies in leveraging the structured output from the M3S framework to systematically identify gaps in autonomous driving models. The M3S framework, originally proposed by Husen et al. [13], provides a multi-view perspective of ML systems that explicitly links high-level safety requirements with specific implementation concerns.

Our extension of M3S follows a structured workflow:

1) **Requirement Extraction**: We analyze the formal requirement specifications produced by M3S, which identify potential safety hazards and edge cases in autonomous driving scenarios. These requirements are extracted in JSON format to facilitate automated processing.
2) **Gap Analysis**: By comparing the requirements against existing datasets and model capabilities, we identify specific domains where additional training data or specialized testing is needed. This process creates a structured representation of "data gaps" that must be addressed.



CARLA
M3S Model Risk Analysis
LLM analysis
Setup simulation and data collection method
Data Generation and Collection
Dataset Annotation
Evaluate Result
Textual Scenario Description
Extract element from M3S model's output
Script Generation
Setup environment, generate vehicle
Save RGB images and ground truth
Save cropped traffic sign based on the cropping algorithm

Fig. 1. Workflow of this research

3) **Scenario Classification**: Requirements are automatically classified into categories based on environment type (urban, highway, sidewalk), weather conditions, traffic density, and obstacle types. This classification enables targeted scenario generation.
4) **Prioritization**: Using risk assessment metrics from M3S (including severity, occurrence probability, and detection difficulty), we establish a prioritization mechanism that focuses simulation resources on the most critical scenarios.

This structured approach transforms qualitative safety requirements into quantifiable simulation parameters, enabling automated scenario generation within CARLA.

B. Hierarchical Scenario Generation

Building on the analyzed requirements, we implement a hierarchical scenario generation system that translates high-level safety concerns into concrete CARLA configurations. This hierarchical structure consists of four layers:

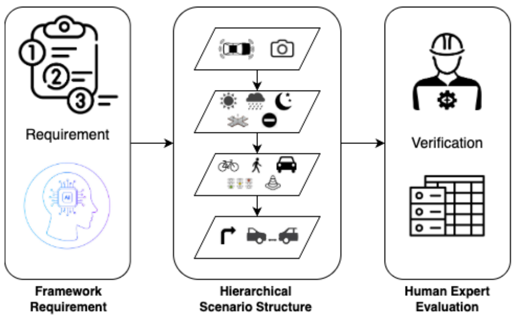


Fig. 2. Hierarchical structure for scenario generation in CARLA

1) **Vehicle Configuration Layer**: Defines the autonomous vehicle's properties, including sensor suite configuration, vehicle type (standard car or bicycle for sidewalk navigation), and physical parameters. For sidewalk scenarios, we predominantly use the `crossbike` blueprint to navigate narrow pedestrian zones.
2) **Environment Layer**: Establishes the simulation conditions, including weather parameters (precipitation, fog density, sun angle), time of day, and map selection. Environmental parameters are systematically varied based on the identified requirement gaps, with a focus on challenging conditions such as poor visibility and wet surfaces.
3) **Traffic Layer**: Controls the density and behavior of other road users, including vehicles and pedestrians. This layer incorporates stochastic behaviors for non-player characters to simulate the unpredictability of real-world traffic. For sidewalk scenarios, we implement specialized pedestrian behavior models that more accurately reflect human movement patterns in confined spaces.
4) **Scenario Logic Layer**: Coordinates the overall simulation flow, triggering specific events (such as pedestrian crossings or sudden traffic stops) and defining success/failure criteria. This layer implements the temporal aspects of the scenario, ensuring that events occur in a realistic sequence.

Our hierarchical approach enables efficient scenario generation by allowing individual components to be modified independently. For example, adverse weather conditions can be applied across multiple traffic configurations without requiring complete scenario redefinition. This modularity directly addresses the requirements identified in the M3S analysis.

1) Scenario Description Language: To bridge the gap between M3S requirements and CARLA execution, we developed a machine-readable scenario description language encoded in JSON. This representation serves as an intermediate format that:

```
{
  "scenario_id": "sidewalk_pedestrian_001",
  "environment": {
    "map": "Town03",
    "weather": {"rain": 80, "fog": 30}
  },
  "vehicle": {"type": "crossbike"},
  "actors": {
    "pedestrians": {"density": "high"},
    "vehicles": {"density": "low"}
  }
}
```

This structured representation enables automated translation from M3S requirements to executable CARLA scripts, creating a traceable link between safety requirements and test scenarios.

C. LLM Integration Details

Our framework utilizes GPT-4 (gpt-4-0613) with a context window of 8,192 tokens for parameter extraction and script generation. The model processes M3S textual descriptions through our hierarchical prompt design, which consists of a system prompt (450 tokens) defining the JSON schema structure and a user prompt containing the scenario description (typically 200-400 tokens).

D. Multi-Modal Dataset Collection

Once scenarios are generated, our system collects comprehensive multi-modal datasets that serve both training and evaluation purposes. The data collection process is implemented through specialized CARLA Python scripts that:

1) **Synchronized Sensor Capture**: Ensure temporal alignment between different sensor modalities (RGB, depth, LiDAR, and semantic segmentation) by synchronizing sensor callbacks within the simulation tick mechanism.
2) **Automatic Annotation**: Generate ground truth annotations for object detection, semantic segmentation, and instance segmentation without manual labeling, leveraging CARLA's semantic tagging system.
3) **Dataset Structuring**: Organize collected data according to established formats (e.g., COCO for object detection, CityScapes for semantic segmentation) to ensure compatibility with existing model training pipelines.

For traffic sign recognition specifically, we implement a specialized pipeline that:

1) Uses LiDAR data to detect traffic signs within the environment.
2) Processes semantic segmentation data to create precise masks around detected signs.
3) Crops and categorizes the signs to create a GTSRB-compatible dataset with accurate bounding boxes and class labels.

Similarly, for sidewalk navigation, our collection process captures sequential data with an emphasis on pedestrian interactions, sidewalk boundaries, and obstacle detection.

E. Evaluation Framework

To systematically assess our LLM-integrated approach and close the iterative cycle between M3S requirements and simulation results, we implemented a comprehensive evaluation framework. This framework consists of three interconnected components:

1) **Requirement-to-Metric Mapping**: Each M3S requirement is explicitly linked to specific quantitative metrics that measure fulfillment. For example, pedestrian avoidance requirements are associated with minimum distance metrics, while lane-keeping requirements correlate with lateral deviation measurements.
2) **Parameter Extraction Validation**: We conducted systematic validation of the LLM's ability to extract accurate scenario parameters from textual descriptions. For this analysis, we selected a test set of 125 distinct scenario descriptions across our three scenario categories (urban intersections, adverse weather, and sidewalk navigation). Each generated JSON schema was manually evaluated against expert-defined ground truth by two autonomous driving researchers.
3) **Execution Success Analysis**: Generated CARLA scripts were evaluated based on their ability to execute without runtime errors and produce the intended scenario dynamics. This execution validation provides a practical measure of the end-to-end pipeline effectiveness.

For parameter extraction validation, we conducted 3 generation attempts per scenario category, analyzing a total of 375 LLM-generated JSON files. Each JSON parameter was evaluated against five key criteria: environment configuration, vehicle configuration, actor behavior parameters, event triggers, and success criteria. This methodology is comparable to that employed by Cai et al. [8], who evaluated 368 scenarios in their Text2Scenario framework, although our approach places greater emphasis on structured parameter extraction aligned with the M3S framework.

IV. Experimental Evaluation

We present an evaluation of our LLM-integrated M3S framework across four dimensions: (1) accuracy of LLM-generated scenario descriptions, (2) execution success rate of generated scripts, and (3) coverage of safety-critical scenarios.

A. Experimental Setup

1) Scenario Categories: We evaluated our framework on three distinct categories:

1) **Urban Intersection Scenarios**: Complex multi-actor situations with traffic lights, pedestrian crossings, and multiple vehicle interactions.
2) **Adverse Weather Conditions**: Reduced visibility scenarios that challenge sensor perception capabilities.
3) **Sidewalk Navigation**: Scenarios where the ego vehicle traverses sidewalks while interacting with pedestrians in confined spaces.

2) Dataset and Methodology: We selected a test set of 125 distinct scenario descriptions across our three categories. For parameter extraction validation, we conducted 3 generation attempts per scenario category (375 total LLM-generated JSON files), comparable to the methodology employed by Cai et al. [8], who evaluated 368 scenarios.

3) Evaluation Metrics: We used the following metrics:

- **JSON Parameter Correctness**: Percentage of parameters correctly extracted into the JSON schema.
- **Script Execution Success Rate**: Percentage of generated scripts that execute without errors.
- **Requirement Coverage**: Percentage of M3S-identified safety requirements successfully translated into testable scenarios.

B. Results and Analysis

The hierarchical prompt design consistently achieves over 91% overall accuracy (Table I). LLMs perform better on static environmental parameters (95.8-98.5%) than on dynamic elements like event triggers (87.8-91.2%). Analysis of extraction errors showed parameter omissions (41%) as the most common issue, followed by value-range errors (32%), parameter misclassification (18%), and syntax errors (9%).

TABLE I
Accuracy of LLM-extracted JSON parameters (%)

Parameter Category	**Urban**	**Weather**	**Sidewalk**
Environment Configuration	97.2	98.5	95.8
Vehicle Configuration	95.4	94.7	93.2
Actor Behavior	91.3	88.9	90.6
Event Triggers	89.5	91.2	87.8
Success Criteria	93.7	92.9	90.4
Overall	**93.4**	**93.2**	**91.6**

TABLE II
Execution success rate and time efficiency comparison

Scenario Type	**Success Rate (%)**		**Avg. Time (min)**	
	Manual	LLM	Manual	LLM
Urban Intersection	95.2	92.8	45.3	8.2
Adverse Weather	91.5	89.3	38.7	7.1
Sidewalk Navigation	88.7	85.2	52.4	9.8
Average	**91.8**	**89.1**	**45.5**	**8.4**

For script execution (Table II), LLM-generated scripts achieve an average success rate of 89. 1%, close to manual coding (91. 8%). The narrow 2.7% gap between manual and LLM-integrated approaches demonstrates near-human reliability. The overall efficiency has been significantly improved with the help of LLM. The scenario generation time for all three types has been reduced by around 81.5%.

C. Discussion

Our evaluation demonstrates that integrating LLMs with the M3S framework creates an effective solution for the generation of autonomous driving scenarios. Key insights include:

- **Bridging Requirement and Simulation**: Our hierarchical prompt design effectively bridges the gap between text descriptions and simulation parameters, achieving over 91% accuracy.
- **Reducing Expertise Barriers**: High execution success rates (89.1% average) demonstrate that LLMs can generate reliable simulation code without requiring specialized CARLA programming expertise.
- **Areas for Improvement**: The performance gap in complex actor behaviors and event triggers highlights opportunities for further refinement, particularly for temporal event sequences and multi-actor interactions.

These results provide strong evidence that LLMs can effectively automate the translation from risk analysis to executable simulation, addressing a critical bottleneck in autonomous driving development while maintaining high accuracy and execution success rates.

V. Limitations and Future Work

Despite our framework's effectiveness, several limitations warrant consideration. First, generated scenario quality depends on initial M3S textual descriptions, with ambiguous descriptions potentially leading to suboptimal parameter extraction. Second, while our approach significantly reduces manual intervention, human verification remains necessary for safety-critical scenarios.

For future work, we aim to create a fully closed-loop system that analyzes simulation outcomes, where LLMs would:

1) Process simulation logs and sensor data from failed test cases
2) Generate explanations identifying causal factors
3) Automatically formulate refined scenario descriptions targeting identified weaknesses
4) Feed these descriptions back into the scenario generation pipeline

VI. Conclusion

This paper presented an integrated framework that leverages Large Language Models to bridge the gap between risk analysis and executable simulation for autonomous driving systems. By automating the translation of textual scenario descriptions into structured parameters and executable scripts, our approach significantly reduces scenario creation time while maintaining high execution success rates and comprehensive safety requirement coverage. The hierarchical prompt design effectively bridges natural language requirements and simulation code, achieving over 91% parameter extraction accuracy and 89.1% script execution success rate. The integration of LLMs with established modeling frameworks like M3S represents a promising direction for autonomous driving validation, enabling faster iteration cycles and more comprehensive testing across diverse operational conditions.

VII. Acknowledgment

This work was supported in part by the Japan Society for the Promotion of Science (JSPS) KAKENHI Program under Grant 21KK0179, Grant 23K18470 and Grant 25H01125, and in part by the Japan Science and Technology Agency (JST)-Mirai Program under Grant JPMJMI20B8.

References

[1] J. Li, M. Zhang, N. Li, D. Weyns, Z. Jin, and K. Tei, "Generative AI for Self-Adaptive Systems: State of the Art and Research Roadmap," *ACM Transactions on Autonomous and Adaptive Systems*, vol. 19, no. 3, Art. 13, pp. 1–60, 2024.9, doi:10.1145/3686803.

[2] Q. Chen, J. Li, and K. Tei, "Attention-guiding Takeover Requests for Situation Awareness in Semi-autonomous Driving," in *Proc. 18th ACM/IEEE Int. Conf. on Human-Robot Interaction (HRI), Late-Breaking Work*, pp. 416–421, 2023.3.

[3] J. Ling, J. Li, K. Tei, and S. Honiden, "Towards Personalized Autonomous Driving: An Emotion Preference Style Adaptation Framework," in *Proc. 5th IEEE Int. Conf. on Agents (ICA)*, pp. 47–52, 2021.12.

[4] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An Open Urban Driving Simulator," in *Proceedings of the 1st Conference on Robot Learning (CoRL)*, 2017, pp. 1–16.

[5] B.-K. Ruan, H.-T. Tsui, Y.-H. Li, and H.-H. Shuai, "Traffic Scene Generation from Natural Language Description for Autonomous Vehicles with Large Language Model," *arXiv preprint arXiv:2409.09575*, 2024.

[6] J. H. Husen, D. Mendez, and K. Wnuk, "Metamodel-Based Multi-View Modeling Framework for Machine Learning Systems," in *Proceedings of the 11th International Conference on Model-Based Software and Systems Engineering (MODELSWARD)*, 2023, pp. 200–211.

[7] S. Guan, Y. Lu, J. H. Husen, H. Cheng, H. Washizaki, N. Ubayashi, and N. Yoshioka, "Enhancing Safety in Autonomous Driving: Integrating CARLA for Multi-Sensor Dataset Generation and Advanced Scenario Testing," *IPSJ SIG Technical Report*, vol. 2025-SE-219, no. 23, Information Processing Society of Japan, Mar. 4, 2025.

[8] X. Cai, X. Bai, Z. Cui, D. Xie, D. Fu, H. Yu, and Y. Ren, "Text2Scenario: Text-Driven Scenario Generation for Autonomous Driving Test," *arXiv preprint arXiv:2503.02911*, 2024.

[9] Y. Mei, T. Nie, J. Sun, and Y. Tian, "Seeking to Collide: Online Safety-Critical Scenario Generation for Autonomous Driving with Retrieval Augmented Large Language Models," *arXiv preprint arXiv:2505.00972*, 2025.

[10] C. Li, J. Sifakis, R. Yan, and J. Zhang, "Rigorous Simulation-based Testing for Autonomous Driving Systems–Targeting the Achilles' Heel of Four Open Autopilots," *arXiv preprint arXiv:2405.16914*, 2024.

[11] Y. Zhou, M. Simon, Z. Peng, S. Mo, H. Zhu, M. Guo, and B. Zhou, "SimGen: Simulator-conditioned Driving Scene Generation," *arXiv preprint arXiv:2406.09386*, 2024.

[12] M. Biagiola, A. Stocco, V. Riccio, and P. Tonella, "Two is better than one: digital siblings to improve autonomous driving testing," *Empirical Software Engineering*, vol. 29, no. 4, pp. 1–33, 2024, Springer.

[13] J. H. Husen, H. Washizaki, J. Runpakprakun, N. Yoshioka, H. T. Tun, Y. Fukazawa, and H. Takeuchi, "Integrated multi-view modeling for reliable machine learning-intensive software engineering," *Software Quality Journal*, vol. 32, no. 3, pp. 1239–1285, 2024, Springer.