Autonomous Vehicle Scenario Modeling System Requirements Specification

Version 1.0

11/26/2024

Document Control

Distribution List

The following list of people will receive a copy of this document every time a new version of this document becomes available:

Teaching assistants:

Clay Pate

Customer(s):

M. I. Akbas
AVVC

Project team members:

Isabella Acosta

Serena Conticello

Hannah Ramsden

William Reimer

Davian Rosario-Ortiz

Change Summary

The following table details changes made between versions of this document:

Version	Date	Modifier	Description
0.1	Oct-22-24	All team members	Initialization of the document and addition of rudimentary information.
0.2	Oct-24-24	All team members	Continuation of documentation
0.3	Oct-28-24	Davian, Isabella, and William	Continuing to work on document, divide sections and assign work between team
0.4	Oct-28-24	Hannah	Update TOC, updating grammar in section 1, updating references, comments, updating bibliography.
0.5	Oct-29-24	Isabella	Work on remaining parts of Section 2.
0.6	Oct-29-24	All team members	Worked on Sections 4-7.
1.0	Nov-24-24	All team members	Made small changes and updates based on Clay's comments from Version 0.1.

Table of Contents

1. Introduction

Purpose and Scope

PolyVerif is a simulation tool that is used to test the verification and validation of autonomous vehicles by performing multiple tests in different traffic scenarios. It is currently being used by researchers, developers, and engineers and it was developed by a team of individuals that saw the need for a testing platform for autonomous vehicles. PolyVerif provides itself as a noteworthy option in autonomous vehicle testing. However, seeing as preliminary documentation and software dates back as early as 2021, it is still a relatively new tool on the market. A lack of sufficient testing scenarios within PolyVerif's database leaves room for improvement, which is within the scope of this project's goals. To expand the testing within PolyVerif, Embry—Riddle Aeronautical University will be utilized as a testing ground. The campus provides a set of unique scenarios with varying traffic laws, object interactions, and traffic patterns. This should provide the necessary dynamic and customizable plane to bolster PolyVerif's capabilities.

Intended Audience and Reading Suggestions

This SRS includes all vital information required to properly understand the Autonomous Vehicle Project. For those unfamiliar with PolyVerif or simulation software, it is suggested to read through the entire document, focusing on background information first and then once a basic understanding is reached, continue reading through the document and focus on the more specific and technical sections. Below are some reading suggestions to fully understand the concepts gone over in this document:

Reading Suggestions:

For Those Unfamiliar with PolyVerif and Simulation Software:

- Public Transport System–Test and Validation framework for Autonomous Vehicles [1]
- Autonomous Vehicle Verification Consortium [2]

For those Familiar with PolyVerif and Simulation Software:

- PolyVerif Validation Report [3]

Document Conventions

The fonts throughout this document are consistent for each section and there is no highlighting of sections, each section is separated by its header in bold.

Project References

[1] D. R. Razdan, "PolvVerif_Format_v1.0," 5 5 2021. [Online]. Available: https://www.dropbox.com/scl/fi/dmdztn1lk4r2dg8rvk9zd/PolvVerif_Format_v1.0.pptx?rlkey=8muost983lvnrraxnod5e5957&e=3&st=218c5k5l&dl=0.

[2] "PolyVerif," Autonomous Vehicle Verification Consortium, [Online]. Available: https://www.avvc.net/.

[3] R. Sell, B. C. Baykara, E. Malayjerdi and I. Sell, "PolyVerif validation report," 2022. [Online]. Available: https://github.com/HKRamsden/ERAUPolyVerif/blob/main/Documentation/References/PolyVerif_Validation_Report.pdf.

[4] M. I. Akbaş, M. Menase, S. Verma and R. Razdan, "PolyFlows: Modular Test Framework Design for Autonomous Vehicles," *2024 IEEE International Conference on Mobility, Operations, Services and Technologies (MOST),* pp. 50-59, 2024.

[5] Q. Goss, W. C. Pate and M. İ. Akbaş, "An Integrated Scenario-Based Testing and Explanation Framework for Autonomous Vehicles," *IEEE,* May 2024.

[6] M. I. Akbas, "Testing and Validation Framework for Autonomous Aerial Vehicles," Jan 2021.

[7] D. K. E. D. T. e. a. Fremont, "Scenic: a language for scenario specification and data generation," *Machine Learning,* vol. 112, p. 3805–3849, February 2022.

[8] "Simulation of Urban MObility," SUMO, [Online]. Available: https://eclipse.dev/sumo/.

[9] K. Chavan, " PolyVerifFramework/Polyverif: Polyverif AV validation suite," GitHub, 28 September 2022. [Online]. Available: https://github.com/PolyVerifFramework/PolyVerif.

[10] R. Razdan, M. İ. Akbaş, R. Sell, M. Bellone, M. Menase and M. Malayjerdi, "PolyVerif: An Open-Source Environment for Autonomous Vehicle Validation and Verification Research Acceleration," *IEEE ,* vol. 11, pp. 28343-28354, 2023.

[11] "Verification and validation in software engineering," GeeksforGeeks, 19 June 2024. [Online]. Available: https://www.geeksforgeeks.org/software-engineering-verification-and-validation/.

[12] "What is an autonomous vehicle?," TWI, 2024. [Online]. Available: https://www.twi-global.com/technical-knowledge/faqs/what-is-an-autonomous-vehicle.

Definitions, Acronyms, and Abbreviations

Definitions

This section lists terms used in this document and their associated definitions.

Table 1: <Definitions>

Term	Definition
PolyVerif	An open-source validation and verification framework for autonomous vehicles.
Autonomous Vehicle Testing	An autonomous vehicle utilizes a fully automated driving system to allow the vehicle to respond to external conditions that a human driver would manage.
Verification	The process of checking that software achieves its goal without any bugs
Validation	The process of checking whether the software product is up to the mark or in other words product has high-level requirements
Scenic	A language for scenario specification and data generation

Acronyms

This section lists the acronyms used in this document and their associated definitions.

Table 2: <Acronyms>

Term	Definition
AVVC	Autonomous Vehicle Verification Consortium
SUMO	Simulation of Urban Mobility

Abbreviations

This section lists the abbreviations used in this document and their associated definitions.

This does not apply to our project right now; we do not currently have any abbreviations.

Table 3: <Caption>

Term	Definition
e.g.	

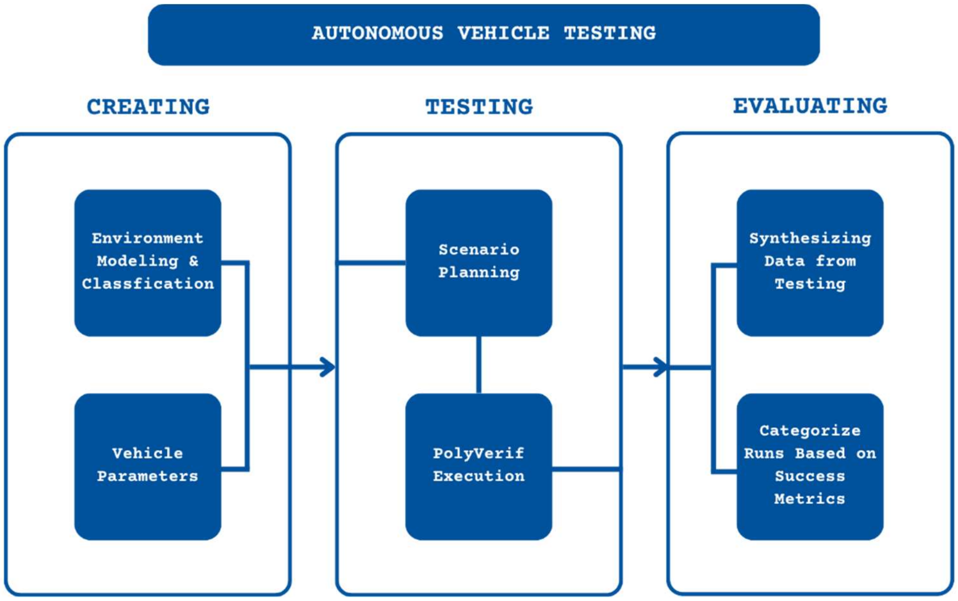
2. General Description

Product Perspective

This product is intended to expand upon pre-existing test simulations found in the PolyVerif GitHub. Currently, 4 distinct maps exist, with several different traffic scenarios involving each map. The product adds another map to these test simulations, the ERAU campus, and a new test scenario (the number of scenarios is subject to change depending on time). The structure and function of pre-existing code or products will not be altered with this product's introduction, rather, this project will utilize the software currently available, and add plugins, APIs, and supporting code, if necessary, but only to support the functionality of the ERAU map and scenario(s). The existing code allows us to build on the verification and validation of our initial scenario that we will be testing, as well as any other scenarios that we may create in the future.

These additional scenarios will be valuable to the overall PolyVerif system because the amount of raw data to pull from is lacking. The overall goal of PolyVerif is to assist in the creation of autonomous vehicles, and to have a successful autonomous vehicle, it needs to access data created from simulations to address as many different scenarios as possible. PolyVerif does have a library of scenarios to choose from, however, more scenarios to draw from means an objectively safer autonomous vehicle. The addition of the map and scenarios our product provides will contribute to the overall performance of PolyVerif when applied to real-world autonomous vehicles.

Product Features


AUTONOMOUS VEHICLE TESTING
CREATING
TESTING
EVALUATING
Environment Modeling & Classfication
Vehicle Parameters
Scenario Planning
PolyVerif Execution
Synthesizing Data from Testing
Categorize Runs Based on Success Metrics

[i]

Figure 1: Polyverif Framework and Functionality

Both Environment Modeling and Classification and Vehicle Parameters are responsible for creating the landscape and environment for the digital twin vehicle to operate. Vehicle Parameters gives the vehicle instructions on how to operate (e.g, how to move forward, turn, stop, etc.)

Scenario planning is responsible for creating the problem needing to be tested by the simulation user. If the user wants to test an autonomous vehicle stopping at a red light, the user will select that scenario. Additionally, this scenario planning gives instruction to the vehicle parameters on how to operate, to move in accordance with the selected scenario.

PolyVerif execution is responsible for running the simulation according to the selected scenario.

Synthesizing Data from Testing and Categorize Runs Based on Success Metrics are both generated when the simulation successfully runs, and a report is generated that contains simulation data in an organized and efficient manner.

User Classes and Characteristics

<< This section should identify each type of user of the system (by function, location, type of device), the number in each group, and the nature of their use of the system. State any user characteristics that may be influential or significant in the structure of the product. Certain requirements may pertain only to certain user classes. Include possibly important characteristics such as educational level, experience level, privilege levels, and technical expertise as necessary.

This diagram is coming soon (use case diagram of the system).

Actors

This section presents the actors in the system.

- Simulation User
  - A user that runs the simulation and collects the data when the simulation ends
  - Can also use this data for external products, such as implementing the gathered data into an autonomous vehicle.
- GitHub Contributor
  - A person that contributes updates to the PolyVerif product as a whole.
  - A simulation user can make changes to PolyVerif locally, as PolyVerif is open source, however, these changes will not be reflected in the PolyVerif GitHub unless a GitHub Contributor approves these changes

Use Cases

This section presents the Use Cases, developed for the system.

- Running a simulation
  - A simulation user uses the system to collect data on various traffic scenarios. If the user needs to collect data on an autonomous vehicle response when at a red light and a car passing through th intersection, the user will run that corresponding scenario and collect the data from that simulation.
- Generating a report
  - When the simulation successfully runs, the user can generate a report of the simulation's results and corresponding data

2.3.3. Scenarios

Scenario 1: Running Simulation

Description: The user runs a simulation using the PolyVerif framework

Actors: Simulation User

Preconditions:

- PolyVerif has successfully been installed onto a machine with Linux Ubuntu OS

Trigger Condition: The user chooses to create an account.

Steps:

1. PolyVerif successfully loads, and the user is prompted to select a map for the simulation to run in. (ALT 1)
2. The user is prompted to select a type of scenario to run.
3. The user is prompted to run the simulation.
4. The simulation runs to completion. (ALT 2)

(ALT 1)

1.1. PolyVerif fails to load correctly.

(ALT 2)

4.1. The simulation fails to fully run to completion.

Scenario 2: Generating a report

Description: The user generates a report from a simulation

Actors: Simulation User

Preconditions:

- PolyVerif has successfully been installed onto a machine with Linux Ubuntu OS
- A simulation has successfully run to completion

Trigger Condition: The simulation is successfully run to completion

Steps:

1. The user is prompted to generate a report of the simulation results. (ALT 1)
2. The report loads, and the user has access to the report generated from the simulation.

(ALT 1)

1.1. The user is not prompted to generate a report
Note: This prompt should occur immediately after the simulation runs, if not, simulation must be re

(ALT 2)

2.1. Report fails to generate successfully due to errors in simulation.
2.2. Report generates successfully, but has faulty data present.

General Constraints

The CPU that is running PolyVerif must have a minimum 8-minimum core. The machine must be running on the Operating System of Ubuntu V.18.04 or V.20.04, it cannot run otherwise. The simulation must be run in the terminal, it will not run otherwise. Currently, the only computer usable is in the Micaplex MP 224(WiDe Lab), we are working on getting PolyVerif installed on Serena's computer. The reason that it can only run on this computer is because it needs a GPU to run and none of our team members have one on their computers, except for Serena. The computer that is running the simulation must have a NVIDIA GTX 1080 (8GB or higher). It also must have Python 3.8 running on the machine. To download it on another Linux machine, the machine must be connected to Wi-Fi to download it and run the simulations as well as meeting all the requirements stated above.

Operating Environment

Micaplex MP 224 WiDe Lab Computer

OS: Linux

Ubuntu: Version 18.04 or later

User Documentation

Research Papers:

- PolyVerif: An Open-Source Environment for Autonomous Vehicle Validation and Verification Research Acceleration [1]
- PolyFlows: Modular Test Framework Design for Autonomous Vehicles [4]
- An Integrated Scenario-Based Testing and Explanation Framework for Autonomous Vehicles [5]
- Testing and Validation Framework for Autonomous Aerial Vehicles [6]
- Scenic: a language for scenario specification and data generation [7]

Websites:

- SUMO Download - https://eclipse.dev/sumo/ [8]

Git Hub Repositories:

- PolyVerif GitHub - https://github.com/PolyVerifFramework/PolyVerif [9]

Assumptions and Dependencies

Assumptions:

The assumptions that can be made are that to download PolyVerif must be connected to Wi-Fi, ensuring a successful install. It must also be connected to Wi-Fi to run the simulation. The user has a list of test cases that are pre–determined, as well as adding their own test simulations (the goal of our project).

Dependencies:

Our only dependency is that the simulation is built in PolyVerif.

3. External Interface Requirements

User Interfaces

Hardware Interfaces

This is not applicable to our project.

Software Interfaces

[REQ-BEHAV.1] PolyVerif
[REQ-BEHAV.1.1] The system shall define simulation scenarios as two components. [REQ-BEHAV.1.2] The system shall have the first simulation interface be visual with OSSDC-SIM.
[REQ-BEHAV.1.3] The system shall have a second simulation interface for Lidar calculation interface.

[REQ-BEHAV.2] Blender 3.5
[REQ-BEHAV.3] MapsModelImporter Plugin 0.6.2
[REQ-BEHAV.4] RenderDoc 1.26
[REQ-BEHAV.5] LGSVL 2020.06

Communications Interfaces

This is not applicable to our project.

4. Behavioral Requirements

Same Class of User

This is not applicable to our project, as all users will have the same permissions throughout the system.

4.2. Related Real-world Objects

[REQ-BEHAV.1] The ERAU DB Campus shall be modeled using the Open Street Map (OSM) program.

[REQ-BEHAV.2] This model will include traffic intersections, vehicles, and environmental factors that are relevant to each scenario for our simulation(s).

[REQ-BEHAV.3] All intersections throughout the ERAU Campus will be modeled.

[REQ-BEHAV.4] The intersection that will be focused on is the intersection between Aerospace Blvd and Clyde Morris.

[REQ-BEHAV.5] This model will then be implemented into PolyVerif.

[REQ-BEHAV.6] This shall allow for customization of things such as speed and position throughout our simulation.

[REQ-BEHAV.7] Within PolyVerif, the simulation will begin with the vehicle at a pre-determined starting point

[REQ-BEHAV.8] The user will be able to begin the simulation, pause it, and stop it.

4.3. Stimulus

The PolyVerif simulation will not have any dynamic stimuli, with two main scenarios being simulated: (0) the autonomous vehicle will stop on time when approaching a red light, or (1) the autonomous vehicle will not stop on time and run a red light. As such there are no stimuli requirements.

4.4. Related Features

[REQ-FUNCT.1] To be expanded on once system has been utilized more.

4.5. Functional

[REQ-FUNCT.1] To be expanded on once system has been utilized more.

5. Non-behavioral Requirements

Performance Requirements

[REQ-PER.1] The system shall not accept more simulations than can be accurately simulated.

[REQ-PER.2] The system shall only run one scenario at any moment.

[REQ-PER.3] The system shall not terminate a simulation to begin a new simulation.

[REQ-PER.4] The system shall accept file inputs of any size.

Safety Requirements

[REQ-SAFE.1] The system shall only be read in python scripts for validation test cases.

[REQ-SAFE.2] The system shall accept python scripts exclusively for simulation test cases.

[REQ-SAFE.3] The system shall print result reports in separate text files.

[REQ-SAFE.4] The system shall validate location data for each simulation.

[REQ-SAFE.5] The system shall validate decision data for each simulation.

[REQ-SAFE.6] The system shall validate control data for each simulation.

Qualitative Requirements

[REQ-AVAIL.1] To be expanded on once system has been utilized more.

Security

[REQ-SEC.1] To be expanded on once system has been utilized more.

Maintainability

[REQ-MAINT.1] To be expanded on once system has been utilized more.

Portability

[REQ-PORT.1] To be expanded on once system has been utilized more.

Design and Implementation Constraints

[REQ-CONST.1] The system shall operate using PolyVerif.

[REQ-CONST.2] The system shall run on Ubuntu V.18.04 or V.20.04.

6. Other Requirements

Database Requirements

This is not applicable to this project.

Operations

[REQ-OPS.1] The system shall save all map render information locally.

[REQ-OPS.2] The system shall save all scenario information locally.

7. Analysis Models

Data Flow Model

Data Sources

This diagram will be coming soon.

Data Sinks

This diagram will be coming soon.

Data Dictionary

Name	*Description*	*Structure*	*Range*
	To be filled in.		

This diagram will be coming soon.

Context Diagram (Level 0 Data Flow Diagram)

This diagram will be coming soon.

Level 1 Data Flow Diagram

This diagram will be coming soon.

Level 2 Data Flow Diagram

This diagram will be coming soon.

Class Model

This diagram will be coming soon.

State Model

This diagram will be coming soon.

8. To Be Determined List

- PolyVerif may change to a different, unknown simulator.

[i]