Milestone 7: Validation and Verification

Fatih Kurt

April 30, 2025

Contents

1 Introduction

Building upon the sensitivity and scenario analyses from Milestone 6, this milestone focuses on the critical task of validating and verifying the simulation model to ensure accuracy, correctness, and overall readiness for final submission. Validation ensures that the model accurately represents the real-world traffic system it is intended to simulate, while verification ensures that the simulation has been implemented correctly and operates as expected.

This milestone marks an important transition from experimentation and analysis to model refinement and finalization. The insights and results from Milestone 6 directly inform the validation process—particularly through sensitivity analysis follow-up—and provide a benchmark against which verification procedures can be compared.

Given that this stage precedes the final deliverable, time and effort were allocated efficiently to identify and resolve any inconsistencies, validate parameter behavior, and ensure structural correctness of the NetLogo-based traffic simulation model. The primary objective was to increase the confidence in the model's fidelity, transparency, and correctness before it is evaluated as a complete and functioning system.

2 Validation

2.1 Methods Used

The validation strategy for this project primarily relied on internal consistency checks and qualitative behavioral alignment with real-world traffic dynamics. The following validation approaches were considered:

- **Parameter Validation:** While the model parameters—such as acceleration, number of vehicles, and speed limits—were not directly based on real-world datasets, they were validated through intuitive behavioral responses. For instance, increased acceleration resulted in higher average speed but also more congestion, which aligns with traffic system expectations.

- **Cross-Model Validation:** Although no formal empirical dataset was used for direct comparison, the simulation results were qualitatively consistent with findings in the transportation modeling literature. In particular, aggressive driving behavior and higher vehicle densities are widely known to produce congestion, queueing, and stop-and-go dynamics, all of which emerged in the simulation under the pessimistic scenario.

- **Behavioral Face Validity:** Simulation outcomes were reviewed against domain expectations. The optimistic scenario showed minimal queueing and low wait times, whereas the pessimistic case showed increased congestion. These patterns align with typical behavior observed in real-world traffic systems and validate the behavioral realism of the agent rules.

2.2 Validation Results

No statistical anomalies or unrealistic traffic patterns were observed. Each parameter tested during the sensitivity and scenario analyses produced expected and explainable effects. For example:

- Lower acceleration values led to smoother traffic flow and shorter queues.

- Higher acceleration values increased the average speed, but at the cost of higher congestion.

- Increased vehicle count led to longer queues and higher wait times.

These trends reflect a consistent internal logic within the model, and results were verified across multiple simulation runs to ensure reproducibility.

2.3 Discrepancies and Their Resolution

No major discrepancies were observed that required model revision. The simulation behaved as expected in nearly all tested conditions. Any variability between runs was within reasonable stochastic limits and did not alter the overall trends of the metrics.

2.4 Acceptable Margins of Error

Given the exploratory and educational nature of the agent-based model, a variation of approximately ± 10–15% in key performance indicators (such as queue length or wait time) was considered acceptable. This accounts for stochastic factors, random turtle (car) placement, and the dynamic nature of autonomous movement across intersections. The consistency of trends across multiple runs further supports the validity of results within this margin.

3 Verification

3.1 Methods Used

To ensure correctness and reliability of the simulation code, the following verification methods were applied:

- **Unit Testing (Informal):** Although formal unit tests were not written for each procedure, functional testing was conducted. Built-in procedures such as `record-data`, `set-car-speed`, and `next-patch` were taken from trusted base NetLogo models and produced expected results throughout multiple milestone runs. Custom additions like the average queue length plot function were verified manually through live simulation behavior.

- **Integration Testing:** The interaction between turtles (cars), patch-based roads, and intersection light control was verified through controlled simulation setups. NetLogo's ability to run simulations tick-by-tick on small grids (e.g., 2x2) with a single car enabled fine-grained observation of how agent rules and environmental constraints worked together.

- **Regression Testing:** No breaking changes were made to the code following Milestone 6. Previously validated output patterns remained consistent in subsequent simulation runs, and milestone deliverables from earlier phases continued to function without modification. As such, functional regression testing was considered successful by continuity.

- **Code Review:** A formal peer review was not conducted. However, code structure and logic were evaluated iteratively during milestone development using assistance of generative AI tools and visual validation through the NetLogo interface. Discrepancies (such as the average travel time graph not functioning correctly) were identified and removed during earlier development stages.

3.2 Testing Results and Resolutions

Most custom procedures behaved as expected with no unresolved errors. The average travel time plot, which initially returned incorrect values, was removed from the final implementation. The remaining plots and data collection mechanisms were visually and statistically verified across tests.

3.3 Version Control Practices

Version control was managed using a GitHub repository. While not all changes to the source code were committed at each development step, all milestone reports and supporting documentation were uploaded to the `docs` branch. This ensured traceability of progress and facilitated reproducibility of simulation configurations.

4 Validation and Verification Checklist

The following checklist summarizes the completeness and rigor of the validation and verification efforts conducted for this simulation project. Items are marked with appropriate commentary based on project scope and implementation.

4.1 Validation Checks

- **Compared simulation outputs with real-world data:** No direct comparison was made, as the model is conceptual and based on the Traffic Grid Goal example in NetLogo.
- **Model parameters within reasonable ranges:** Yes. Although not empirically calibrated, all parameters were bounded within realistic and justifiable values for educational traffic models.
- **Model behavior aligns with expert expectations:** Yes. The results, including the impact of acceleration and car density, were consistent with documented traffic patterns in transportation literature.
- **Validation results documented with visuals:** No. While output graphs were analyzed internally, no external data overlays were used.
- **Performed historical data validation:** No.
- **Conducted cross-model validation:** No formal cross-model analysis was performed, though the behavior matched general expectations from agent-based traffic models.

4.2 Verification Checks

- **Tested with boundary/edge conditions:** No. Edge-case simulations were not formally tested but can be explored in future iterations.
- **Verified input parameter handling:** Yes. NetLogo's GUI enforces bounds on inputs (e.g., sliders), preventing invalid states.
- **Checked mathematical/logical consistency:** Yes. The structure and sequence of procedures produced expected outputs, and known bugs (e.g., broken average travel time plot) were identified and removed.

- **Reasonable behavior across time scales and scenarios:** Yes. Behavior scaled as expected with varying car counts and acceleration values.
- **Testing results systematically documented:** Yes. All scenario and sensitivity test results were recorded, analyzed, and presented with accompanying discussion.
- **Performed unit, integration, and regression testing:** Not formally. Functional correctness was confirmed via live simulations and visual inspection.
- **Used version control best practices:** Partial. GitHub was used to document milestone progression, though not all code versions were committed at each step.

4.3 Documentation Checks

- **Documentation up to date:** No. While milestone reports were maintained, in-code comments and comprehensive documentation were limited.
- **Assumptions and limitations documented:** No, though this report contains relevant discussion.
- **Test cases and results recorded:** Yes. Test output CSVs were labeled clearly and reflected in milestone analyses.

4.4 Version Control Checks

- **Changes committed with descriptive messages:** Yes.
- **Tagged versions appropriately:** No.
- **Repository synchronized with remote:** No, except for documentation branches.

4.5 Peer Review Checks

- **Code reviewed by peers or mentors:** No formal peer review was conducted.
- **Feedback incorporated:** Yes. Iterative improvements were made based on simulated behavior and AI-assisted analysis.

4.6 Data Validation Completeness

- **Validation data complete and reliable:** Yes. CSV outputs were consistently structured and representative of simulation behavior.
- **Checked for data inconsistencies or anomalies:** Yes. Graphical results were manually verified, and problematic metrics (e.g., average travel time) were excluded.

5 Common Pitfalls

5.1 Validation Mistakes

Several common validation mistakes were intentionally avoided throughout the development process. No unexplained discrepancies were observed in the simulation outputs; the behavior remained consistent and interpretable across different tests. Critical parameters—especially acceleration and vehicle count—were thoroughly explored through both sensitivity and scenario analyses, ensuring parameter relevance was not overlooked.

While the model was not validated using real-world datasets or extensive cross-model comparisons, its behavior aligned with general expectations found in traffic modeling literature. However, over-reliance on behavioral face validity and internal consistency may limit the depth of validation insights. This is acknowledged as a limitation, and future work could incorporate historical traffic datasets for a more robust validation framework.

5.2 Verification Oversights

All simulation runs—including those that revealed issues like the broken average travel time plot—were documented, and invalid results were either debugged or excluded from final reporting. While no formal unit or regression testing framework was implemented, test coverage was decent within the context of agent-based simulation.

Basic edge cases, such as simulations with only a single vehicle, were tested, though large-scale grid scenarios or combinations of extreme values were not exhaustively evaluated. This is noted as an area for future verification improvements.

5.3 Avoiding Overfitting in Validation

The model's structure remained unchanged across all test scenarios, and performance varied appropriately with changes in input parameters. This consistency across a diverse set of configurations (e.g., 40 to 60 cars, low to high acceleration) suggests the model was not overfitted to any one test case. Additionally, sensitivity analysis reinforced the model's robustness by confirming expected trends across controlled parameter variations.

5.4 Tips for Efficient Testing Strategies

Efficient testing was supported by maintaining well-organized simulation outputs and structured CSV datasets. These records proved valuable in identifying and correcting test setup mistakes—such as misconfigured acceleration values—by enabling cross-checks with screenshots and logs. While formal automated testing was not employed, visual checks and reproducible NetLogo runs supported a systematic development process.

Future improvements may include the adoption of lightweight automated testing tools or structured templates for scenario setup and execution tracking.

6 Documentation

6.1 Assumptions Made During Validation and Verification

Several assumptions were made throughout the V&V process. First, the core logic and structure of the simulation were largely inherited from official NetLogo models (specifically, the Traffic Grid Goal model), which were assumed to be functionally correct. These preexisting components were not rewritten or deeply revalidated, though their continued behavior was monitored during test runs.

Second, built-in NetLogo mechanisms were assumed to function reliably, such as sliders, tick-based updates, patch interactions, and turtle movement. Although not all of these were explicitly tested, behaviors such as speed remaining below the defined maximum and turtles following road paths as expected provided indirect validation.

6.2 Format for Reporting Test Cases

Test cases were systematically reported using clearly labeled CSV exports and summarized through structured tables in the LaTeX report. Additionally, bar charts were used to visualize key performance indicators (KPIs) across test configurations. A consistent naming convention (`Test 1`, `Test 2`, etc.) was used throughout the sensitivity and scenario analyses to maintain clarity.

6.3 Presentation of Comparison Results

Simulation outcomes were compared primarily through bar charts showing average values of metrics such as queue length, speed, wait time, and stopped cars. Each chart included labeled axes, titles, and consistent ordering for accurate cross-test analysis. These visuals were paired with interpretive commentary to highlight trends and validate model expectations.

6.4 Model Limitations

The model has several known limitations:

- **Static Grid Layout:** The simulation operates on a fixed-size grid with unchanging intersection locations and road structure.

- **Lack of Lane Behavior:** Cars cannot overtake or share lanes. All vehicles follow a single-lane logic with no explicit traffic rules like right-of-way or priority.

- **No Pedestrian or Environmental Interactions:** The model includes only vehicle agents, with no simulation of pedestrians, weather, or other environmental variables.

- **No Real-World Calibration:** Parameter values such as acceleration and speed limits are conceptually reasonable but not tied to empirical traffic data.

- **Agent Cognition Simplification:** Although cars follow goal-based routing, their decisions are based solely on nearest-goal patch distance rather than intelligent navigation algorithms.

Development was also constrained by the author's limited familiarity with NetLogo at the project's outset. While progress was made in extending the base model and adding new metrics, decisions were intentionally conservative to maintain stability. More advanced features (e.g., adaptive signal control or reinforcement learning agents) were out of scope for this iteration but may be viable in future work.

7 Conclusion

The validation and verification (V&V) process provided an essential checkpoint in the development of the traffic simulation model. Through reflective validation, the model's behaviors were evaluated against conceptual expectations, and the simulation consistently demonstrated logical and interpretable results. While no empirical data was used for calibration, behavioral trends—such as increased congestion with higher car counts or aggressive acceleration—were consistent with real-world traffic phenomena.

Verification activities focused on ensuring that the simulation code and outputs were functioning as intended. Testing was conducted across a range of parameter settings and scenarios, and outputs were systematically recorded and compared. Although formal regression and unit testing were not implemented, simulation runs were closely monitored and effectively debugged when discrepancies occurred.

The documentation process—including consistent test naming, structured data exports, visual summaries, and detailed milestone reports—ensured transparency and reproducibility of all tests. Assumptions and limitations have been clearly stated, providing a realistic perspective on the model's current capabilities.

With these efforts, the simulation is now in a strong position for final refinement and presentation. The groundwork laid by the V&V milestone ensures confidence in the model's stability, and it provides a solid foundation for drawing conclusions and proposing future improvements in the final deliverable.