Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X Volume 4, Issue 3, 2026

FORMAL SPECIFICATION OF ENVIRONMENTAL REQUIREMENTS IN AUTOMATED DRIVING SYSTEMS: A FRET-DRIVEN APPROACH

[1]Faraz Ahmad, [*2]Hashim Ali, [3]Sana Amin, [4]Hazrat Wali
[1]*Abdul Wali Khan University Mardan, Pakistan*
[*2]*Abdul Wali Khan University Mardan, Pakistan.*
[3]*Abdul Wali Khan University Mardan, Pakistan.*
[4]*Abdul Wali Khan University Mardan, Pakistan*
[*2] hashimali@awkum.edu.pk

DOI: https://doi.org/10.5281/zenodo.21608457

Article History
Received on: 02 March, 2026
Accepted on: 17 March, 2026
Published on 20 March, 2026

Corresponding Author
Hashim Ali
hashimali@awkum.edu.pk

Abstract
The present thesis introduces a complete process that enables researchers to convert informal environmental requirements that apply to autonomous driving systems into formal specifications that can be analyzed and verified through official methods. The FRAV regulatory framework was used to extract 25 environmental requirements, which were processed into FRETish formal representation before being converted into diagrammatic semantics and temporal logic formats, which enable automated verification. The proposed workflow enables complete traceability throughout the process because it includes variable mapping and predicate abstraction and multiple backend export capabilities, which include LTL and SMT, thus supporting both design-time model checking and runtime monitoring. The study employed semantic and structural completeness, ambiguity and consistency metrics to assess FRET-based formalization, which demonstrated that its semantic comprehension and structural integrity improved compared to natural-language requirements, while it unveiled hidden assumptions that needed to be addressed before formal verification could proceed. The research demonstrates that diagrammatic semantics improve understanding for stakeholders, while formal encodings deliver exact specifications that machines can verify without altering the original purpose. The research demonstrates that combining requirements engineering with formal methods improves the safety-critical environmental requirements of autonomous systems via better quality and accurate traceability, and improved ability to verify requirements.

1. Introduction

Requirements engineering serves as the primary essential step for developing quality systems that need to establish safety requirements for their critical systems ([1]. A requirement establishes the expected system performance through its various measurable components, which people commonly call observables [2, 3]. A system achieves correct operation during its initial development phase when it successfully meets all defined requirements [4]. Most requirements tend to use natural language for their expression, which makes them understandable to people, but this choice creates uncertainty about their meaning and makes it difficult to analyze them through formal methods [5, 6]. The introduction of errors during this period results in mistakes that become impossible to detect while they continue through subsequent development stages until they cause system failures [7]. Early-stage defects are critical because systems can appear to meet their written requirements while their actual performance does not meet the intended operational standards.

The research evidence demonstrates that requirement defects arise from two primary sources, which include incomplete behavioral descriptions and the violation of quality attribute requirements [6]. The established guidelines, together with their corresponding tools, offer solutions to problems, but organizations continue to experience difficulties with traceability and verification processes that connect systems and applications [8]. The research evidence demonstrates that requirement defects arise from two primary sources, which include incomplete behavioral descriptions and the violation of quality attribute requirements [6]. The established guidelines, together with their corresponding tools, offer solutions to problems, but organizations continue to experience difficulties with traceability and verification processes that connect systems and applications [8].

The requirements for Environmental and safety standards that Automated Driving Systems (ADS) must meet create difficulties that become more severe. The environmental requirements need to document how the system will function under different road conditions and atmospheric conditions, and the presence of traffic participants and specific operational constraints. The regulatory requirements in FRAV require organizations to present their operational needs using informal natural language. This results in three main problems because of the following three aspects: (1) the presence of unclear and imprecise information, (2) the system behavior lacks strong connections to its main objectives, and (3) organizations have not established formal methods that allow them to prove their compliance with the requirements [9]. The existing limitations act as obstacles that prevent researchers from developing scientific methods that verify the accuracy of their findings in Automated Driving Systems (ADS) research.

[9] developed a complete list of environmental needs which they obtained from the FRAV framework that describes road characteristics, and situational factors, object detection methods, and speed-friction dynamics. Although the work establishes a solid regulatory base the requirements continue to exist as natural language text, which creates difficulties with understanding requirements and tracking, and testing requirements. The study uses NASA's Formal Requirements Elicitation Tool (FRET) [10], which transforms natural-language requirements into FRETish that structured system which, uses defined semantic restrictions. The system requires users to define specific boundaries and operating conditions, response times, and system behavior, which results in reduced uncertainty and enables formal assessment of system operation.

The research aims to develop an organized system that uses automated tools to transform environmental requirements into formal specifications, which researchers can use for verification purposes. The project involves:

(i) extracting and categorizing FRAV requirements,
(ii) formalizing them into FRETish representations,
(iii) generating diagrammatic semantics to support stakeholder understanding, and
(iv) translating the specifications into temporal logic for formal verification. The study presents a quantitative evaluation framework that assesses requirement quality enhancements through five criteria

that include semantic clarity and structural consistency and completeness, ambiguity reduction, and verification readiness.

The research uses four essential questions to direct its study because it investigates all of the following aspects, with its first question explaining how informal FRAV requirements can be transformed into FRETish through formalization processes. The second question investigates how diagrammatic semantics affect stakeholder comprehension. The third question examines how FRET-based specifications enable different user groups.

The two main contributions of this work provide its two primary outcomes. The research work shows how ADS regulatory environmental requirements can be converted into machine-readable specifications, which enable better traceability and formal verification. The research system produces a quantitative assessment method for requirement formalization through FRET, which identifies an existing research gap. The proposed framework establishes a reproducible pathway from informal regulatory text to formal verification artifacts and can be extended to other safety-critical domains facing similar challenges.

The remainder of this paper is organized as follows. Section 2 reviews background and related work in requirements engineering, formal methods, and FRET. Section 3 presents the proposed methodology for requirement extraction, formalization, and evaluation. Section 4 reports the results and analysis, followed by discussion and implications in Section 5. The paper concludes with Section 6, which also provides directions for future research.

2. BACKGROUND STUDIES AND RELATED WORK

a. Requirements Engineering Foundations

The system performance depends on its requirements because these requirements define what the system should accomplish and how it should operate. The stakeholders who need to understand these requirements prefer natural language because it provides them with better access to the information. The process of using this method creates three problems, which include ambiguity, incompleteness, and inconsistency [11, 12]. The problems that arise from these issues create two main problems, which include incorrect assumptions and ambiguous terminology, and imprecise interpretations that all affect developers, designers, and testers [13, 11]

The process of traceability presents another major problem because stakeholders need to create connections between their high-level objectives and system requirements and all subsequent system components. The system traceability weaknesses result in higher maintenance expenses and create challenges for the assurance operations. The development of traceability models has progressed, but researchers still find it difficult to create automated methods that can reliably retrieve trace links [14, 10].

Formal methods create better requirements through their ability to conduct thorough verification analyses using model checking and theorem proving [15, 16]. The process only works properly when organizations provide high-quality vital specifications because system faults start to appear when organizations use vague or incomplete specifications. The process of pattern-based requirement expression through EARS and Property Specification Patterns (PSP) establishes boundaries that help reduce requirement expression errors. The process of tool-supported methods helps convert natural language into formats that researchers can analyze. The existing regulatory requirements still exist as free-text documents, which create a need for organizations to develop structured methods that use tools for the process of formalization.

b. Requirements Engineering Foundations

The set of formal specification methods includes mathematical methods which consist of model checking and theorem proving, and logic-based specification languages, according to [17], the primary function of model checking allows users to confirm temporal logic property compliance in system models while also assisting developers to find design flaws through counterexample analysis, according to [15]. The method requires precise abstract representations for effective operation, but its ability to handle complex systems presents operational challenges.

The designers of pattern-based approaches created their system to help users understand formal methods. Property Specification Patterns (PSP) provide reusable templates that allow engineers to express formal properties without requiring deep expertise in logic [16]. The Easy Approach to Requirements Syntax EARS provides structured templates that help users reduce uncertainty and enhance their understanding, according to [12]. The implementation of these methods creates uniformity while enabling organizations to adopt new technologies with reduced difficulties.

The restricted natural language methods enable developers to create precise documentation for free-text requirements, which can be tested automatically and tracked according to [16, 12, 18]. The current solutions need extensive domain knowledge to function properly, while their formal verification tools do not work together well with existing systems. The use of the FRET system becomes necessary because it enables users to combine controlled language with exact formal meaning and complete system connections.

c. FRET and FRETish Language

The NASA Formal Requirements Elicitation Tool FRET helps users create exact requirements that can be verified for use in safety-critical systems. The open-source platform FRET allows users to create and formalize requirements while they analyze a single workspace.

FRET uses FRETish, which is a controlled natural language for requirement structuring that includes five essential elements: scope, condition, component timing, and response. The component and response fields are mandatory, while the others serve to provide additional contextual information. The structured representation requires complete specification of all aspects related to requirement implementation, thus eliminating both ambiguity and hidden assumptions.

FRET automatically converts FRETish statements into formal semantic meaning, which usually takes the form of temporal logic formulas. The system provides users with quick feedback through its visual diagrams and machine-readable output. The system enables human readers to understand its content while maintaining the ability to conduct formal verification tests, which make FRET ideal for converting regulatory requirements into specifications that can be analyzed.

1 shows the FRET Editor, which includes features for formalization and explanation.

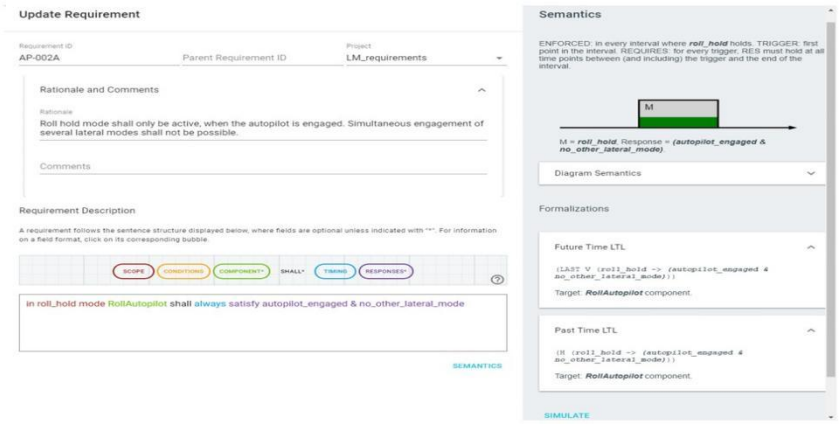

Fig. 1: FRET editor with formalizations and explanations in the help tab

The system increases traceability through the creation of distinct requirement identifiers, which can be exported to multiple formats, including JSON and tool-specific verification inputs. The system enables users to analyze realizability while they connect with formal verification processes, which enable them to transform informal requirements into methods for formal assessment.

d. Environmental Requirements in ADS and the FRAV Framework

The research study conducted by [9] and his coauthors in 2025 presents an extensive investigation of environmental requirements that govern ADS systems through their application of the FRAV framework 2. The research study identifies and classifies requirements that pertain to road characteristics and environmental conditions, vehicle velocity, and object recognition capabilities. The

requirements of the system emerge from regulatory documents which include WP.29/GRAV and they undergo testing to verify their compliance with SAE and ISO standards.

The study presents a detailed list of requirements, but they remain in natural language, which creates three problems. The requirements are fixed elements that apply only to certain road situations. The existing limitations demonstrate the essential need for formalization methods that require systematic implementation.

e. Related Work

The research community has investigated different methods for creating formal natural language requirements through their studies of controlled languages, pattern-matching methods, and software development environments.

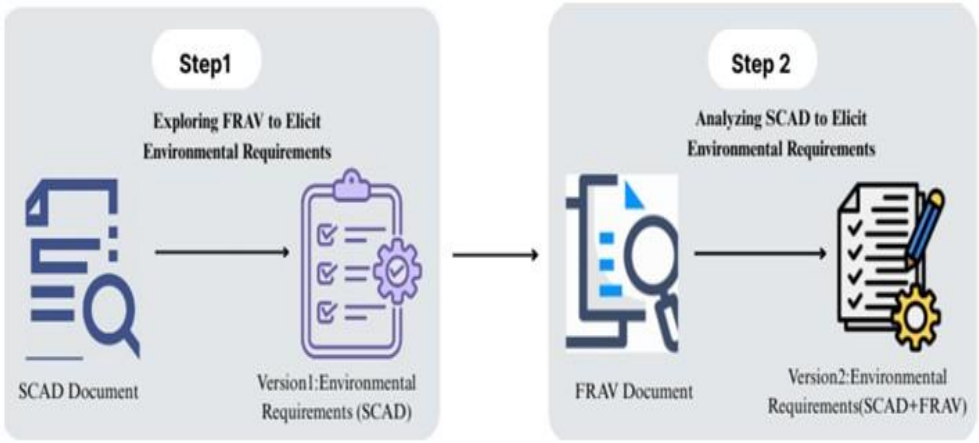

Fig. 2: Methodology of Environmental requirements for ADS from FRAV

FRET serves as a toolchain that enables users to create structured requirement specifications that produce machine-readable specifications according to [10]. The system uses templates to create formal procedures that help users understand things while testing formal verification procedures. The research presents descriptive content about the study, but it fails to provide numerical measurements for assessing the requirement standard.

The research studies three areas, which include proof of concept assessment and systems development, because FRET can assess requirement validation while detecting hidden requirements. The research studies face limitations because their studies focus on restricted domains and show limited ability to expand. The research uses FRET on a regulatory dataset, which contains FRAV data, and assesses research results through quantitative measurement methods. [19] studied how lightweight toolchain integration functions for constrained systems enable them to verify their processes and export the data. Their proposed methodology achieved successful results, but the functions only work within particular system limits, failing to meet wider regulatory standards. [20] provides a cross-domain survey of FRET applications across aerospace, medical, and robotics domains. The research presents practical difficulties together with implementation findings, yet it depends on qualitative data instead of using structured assessment methods. The study by [20] represents how FRET technology can be used in mobile robotics field for its observation purposes and its mission execution requirements. The research focuses on diagrammatic semantics together with stakeholder interaction, yet it maintains its boundary as a particular field of study, which does not consider environmental impacts or regulatory requirements. [21] in their study and [22] both provide complete formal verification processes, which apply to safety-critical systems. The research demonstrates how formal methods achieve successful outcomes, yet it requires formal requirements to exist because the process of transforming natural language remains unaddressed.

TABLE 1: *ELICITED REQUIREMENTS RELATED TO ROADS (FRAV CASE STUDY)*

REQ. ID	REQUIREMENTS
ROD.01	FRAV incorporates detailed physical infrastructure elements such as lane markings, crosswalks, and signalized/nonsignalized junctions.
ROD.02	FRAV includes stop, yield, speed-limit, crosswalk, railroadcrossing, and school-zone signs to ensure compliance with road regulations.
ROD.03	FRAV supports multiple lane types, including cycling lanes and pedestrian pathways, to enhance road-user safety.
ROD.04	FRAV addresses both motorways/highways and urban roads, including scenarios with unprotected turns.

III. METHODOLOGY

The section shows the research methods that create formal specifications that can be verified through testing based on the environmental needs of Automated Driving Systems (ADS). The project uses a multi-stage process that defines requirements through traceable pathways while handling testing procedures in a structured manner. The process starts with requirement extraction, which leads to formal verification and ends with quality assessment through each stage of the process.

a. Requirement Collection (Data Source)

The FRAV case study [9] serves as the primary data source for this research. The document delivers complete environmental requirements, which include road conditions, situational factors, speed limits, friction behavior, and object detection requirements. The requirements come from regulatory standards which establish actual operational conditions for ADS.

The data preparation begins with the systematic extraction of individual requirements in their original natural language form. Each requirement is assigned a unique identifier (e.g., ROD.01) within its respective category to preserve traceability and maintain semantic integrity. The process of formalization will not change the original intent, which requires maintenance before this stage.

The FRAV framework provides 1, which displays sample natural language requirements that researchers used to process further. The remaining 25 requirements in natural language are provided in the Appendix

b. Requirement Formalization

The extracted natural language requirements are transformed into FRETish, a structured and controlled representation that supports both human readability and machine processing. The transformation produces enhanced clarity together with reduced ambiguity while establishing a standardized framework for conducting formal evaluations.

FRET organizes requirements into six fields:scope, condition, component, shall, timing, and response, as illustrated in 3. The fields require precise identification of all contextual elements together with behavioral components and temporal details that pertain to each requirement.

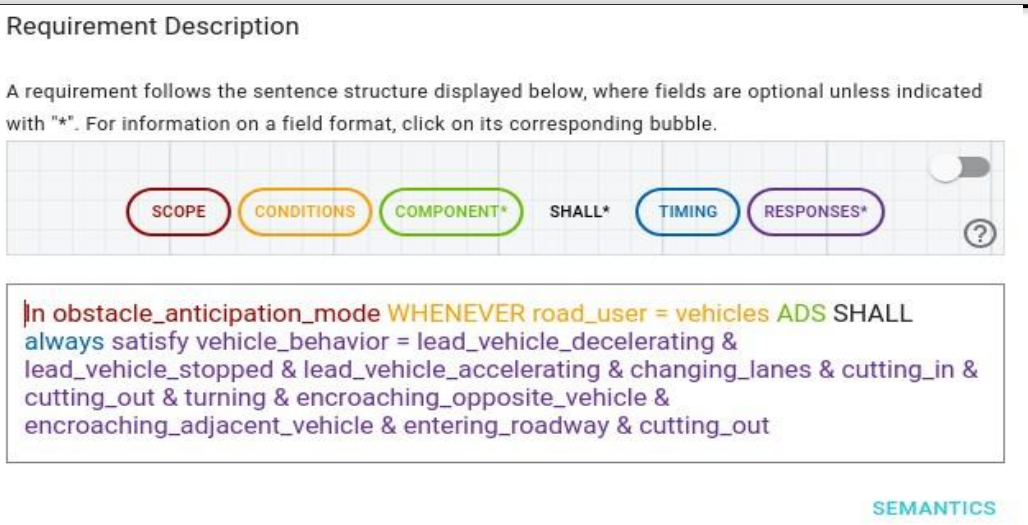

Fig. 3: Six Compulsory Fields of the FRET

c. Diagrammatic Semantics

FRET creates visual diagrams that show formal requirements through their logical and temporal relationships. The diagrams create a validation process that connects textual specifications with formal verification methods.

FRETish elements get converted into matching visual elements. The condition shows when the system will activate while timing shows the time connections, and response describes how the system operates within its established boundaries. The component shows which system element has the duty to execute the task. The visualizations help stakeholders to comprehend information better because they show hidden time and logical connections, which make the content easier to understand and validate through testing.

d. Temporal Logic and External Tool Integration

The FRETish requirements, which were defined through a formalization process, are converted into Linear Temporal Logic (LTL) for the purpose of performing automated verification. The translation process uses a semantics-driven method which applies pattern matching and template substitution according to PSP.

The LTL specifications enable future-time reasoning and past-time reasoning, which allows their application in model checking and runtime monitoring. FRET enables users to export data in multiple formats which include JSON, XML, and SMT-LIB for smooth operation with external verification tools. The current stage establishes that requirements have been both formally defined and transformed into verification pipeline components.

e. Quality Evaluation

The study compares natural language requirements (RiNL) with their FRETish counterparts (RiFRET) to assess the effectiveness of the formalization process. The evaluation uses five essential quality attributes which are semantic clarity, structural quality, completeness, ambiguity and consistency, together with the Final Quality Index (FQI), which aggregates these attributes [7, 23, 24].

The metrics enable an organized quantitative evaluation that measures advancement in requirements. The study uses pre- and post-formalization comparisons to show that structured representation improves both requirement quality and verification readiness.

4 shows the complete process of the methodological workflow which transforms FRAV-derived natural language requirements through its various stages into formal specifications that can be verified. The process of extraction and formalization leads to the creation of semantic visualizations, which enable the encoding of temporal logic and quality assessment procedures.

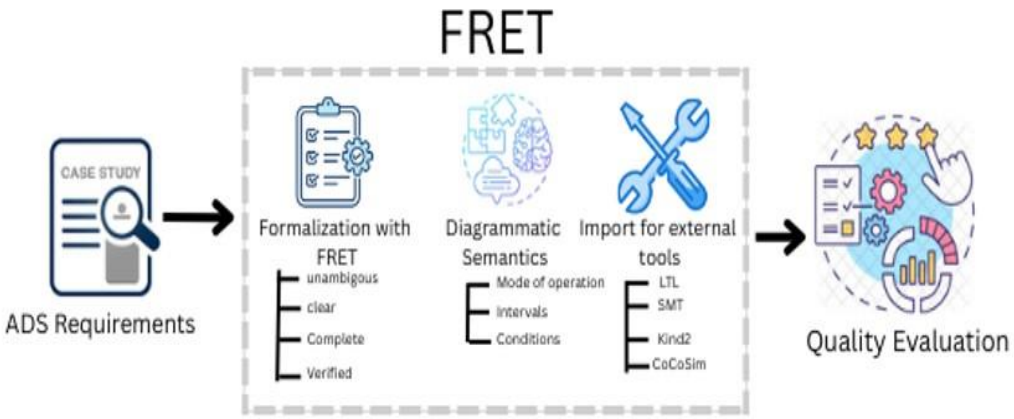

Fig. 4: Formalized Environmental Requirements for Automated Driving Systems: A FRET-Base Approach

f. Other general recommendations

Define properly each of the acronyms used, describing their meaning the first time they appear in the text. For example, "such a relationship is known as the mass ratio (RGM)". Once defined, always use the acronym instead of the full term. Remember to define each of the symbols that appear in the mathematical expressions and do not forget to clarify the notation used when using special or unusual mathematical operators. The use of capital letters should be made according to the general conventions, that is, at the beginning of a sentence after a period, in proper names, in acronyms, etc., and in titles according to what is defined in the work format. If in doubt, contact ajceam@gmail.com.

IV. RESULTS

This section presents the results of applying the proposed pipeline to the FRAV dataset (Kundi et al., 2025). The outcomes move through different stages, which include formalization, semantic representation, verification readiness, and quality evaluation. The study demonstrates improvements from natural language to formal specifications through dedicated examples and quantitative assessment.

a. RQ1: Formalization of FRAV Environmental Requirements into FRETish

The research question RQ1 examines how FRAV requirements transform from their original natural language form into FRETish. The process defines 28 requirements, which they organized into four distinct categories that include road requirements, situational requirements, speed/friction requirements, and object detection requirements. The defined requirements are then converted into a formal structure for requirements transformation Ri, which are defined as: Ri =⟨ scope, condition, component, shall, timing, and response⟩ .

As a result, successful requirement parsing in FRET was achieved through minimal changes, as shown in the Table 2. It includes their work on implicit timing, responses, and conditions to attain syntactic correctness and semantic consistency.

b. Diagrammatic Semantics Representation of Formalized Requirements

The research question RQ2 evaluates FRET diagrammatic semantics because they enable better stakeholder comprehension than natural language descriptions, which FRET provides. The visual representations demonstrate three specific aspects of requirements, which include requirement structure, temporal behavior, and logical dependencies.

FRET diagrams use node-link structures together with timeline-based semantics and a brief textual description. The main components of this system include five elements, which are scope (context of applicability), condition (triggering events), timing (temporal constraints), response (expected behavior), and modal operators (e.g., "shall"). The additional cues, which include color coding, legends, and

metadata, help users read and trace content more effectively. The combined elements establish timing and causality as implicit aspects that become accessible for comprehension through their existence.

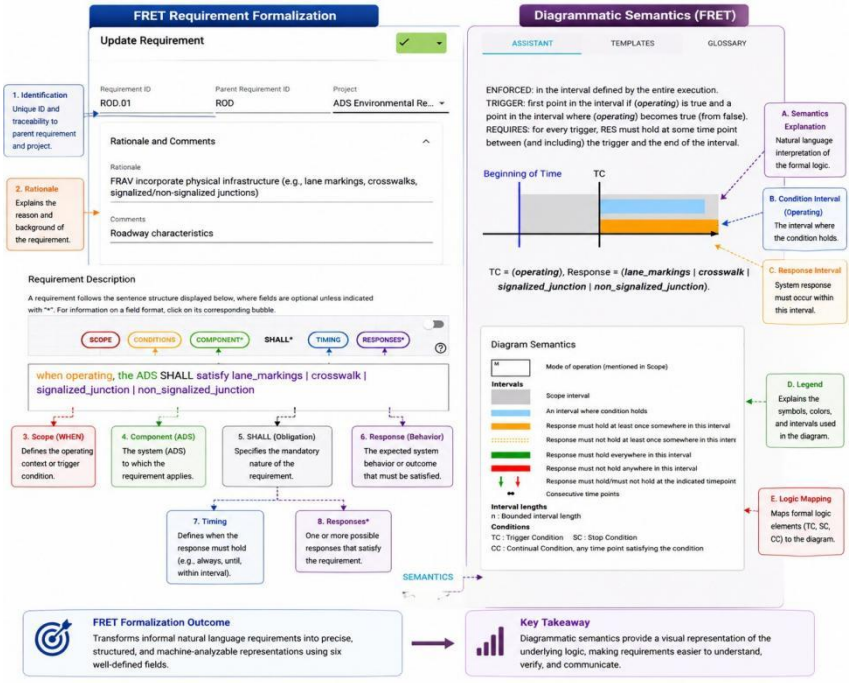

Fig. 5: Diagrammatic Representation of FRETish Requirements

The diagrammatic semantics for requirement ROD.01 in the figure 5 have detailed information related to each aspect the diagrammatic view provides. The timeline visually separates scope, condition, and response intervals. The trigger condition (TC) indicates the specific moment when the system needs to respond, and the response interval establishes the time period when the required behavior must be maintained. The semantic summary explicitly links the condition ("operating") to the required response (infrastructure elements). The diagram establishes precise times and specific situations in which the requirement should be implemented because it uses natural language to remove uncertainties.

Figure 6 presents a more complex example involving quantitative constraints. The diagram shows a continuous condition (road condition and vehicle type) and a corresponding response constraint of the speed-related requirement (R01). The visual representation of the information shows how time continuity and parameter dependencies work together, which the original text format does not show. The alignment of condition and response intervals clarifies that the speed constraint must hold throughout the valid condition period.

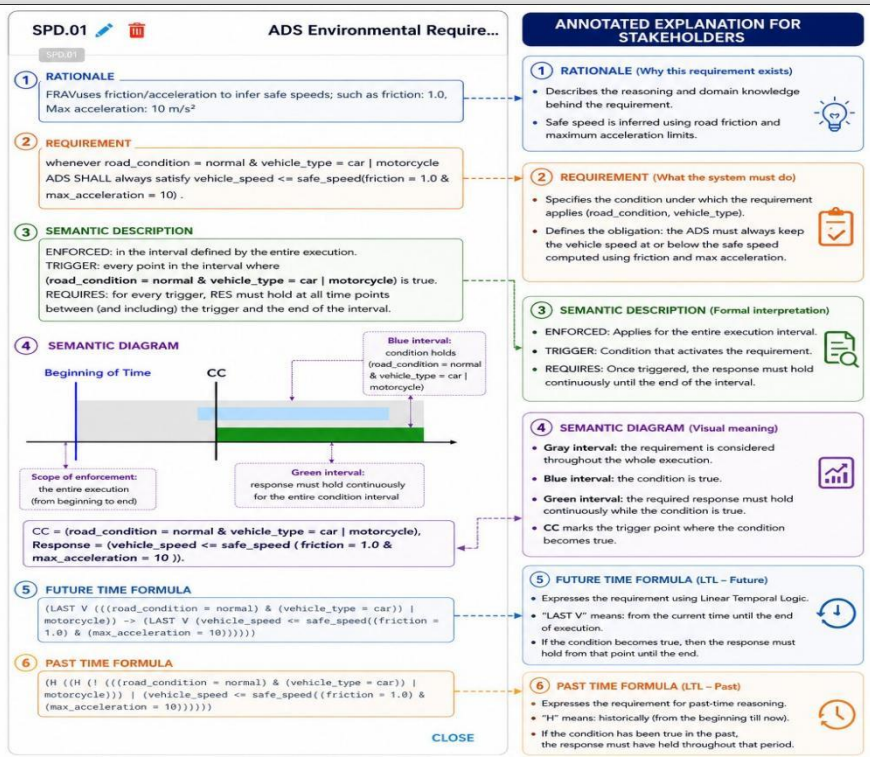

Fig. 6: Diagrammatic Representation of SPD.01 Requirement in FRET

The results demonstrate that diagrammatic semantics enhance interpretability through their ability to present temporal relationships and triggering conditions, and system responses to viewers. The system helps stakeholders to understand information faster, while it eliminates confusion about who needs to know about hidden requirements in complex requirements, which include many details.

TABLE 2: *COMPARISON OF NL REQUIREMENTS AND FRETISH FORMALIZATION*

ID	FRAV (Original NL)	FRETish (Formalization)
ROD.01	FRAV incorporates physical infrastructure (e.g., lane markings, crosswalks, signalized/non-signalized junctions)	When operating ADS SHALL satisfy lane_markings | crosswalk | signalized_junction | non_signalized_junction
SCR.02	FRAV includes vulnerable road users (e.g., pedestrians and cyclists) to enhance safety	When traffic_composition = pedestrian ∧ cyclist ADS SHALL always satisfy safety
SPD.03	FRAV uses friction/acceleration to infer safe speeds (e.g., friction: 1.0, max acceleration: 10 m/s^2)	Whenever road_condition = normal ∧ vehicle_type = car | motorcycle ADS SHALLalways satisfy vehicle_speed ≤ safe_speed(friction = 1.0 ∧ max_acceleration = 10)
OBJ.04	Lead vehicle decelerating, stopped, accelerating, changing lanes, cutting in/out, turning, encroaching opposite/adjacent vehicle, entering roadway	In obstacle_anticipation_mode WHENEVER road_user = vehicles ADS SHALL always satisfy vehicle_behavior = lead_vehicle_decelerating ∧ lead_vehicle_stopped ∧ lead_vehicle_accelerating ∧ changing_lanes ∧ cutting_in ∧ cutting_out ∧ turning ∧ encroaching_opposite_vehicle ∧ encroaching_adjacent_vehicle ∧ entering_roadway

c. Representation of Natural-Language Requirements in Temporal Logic

The research question 3 investigates how FRETish requirements that exist in formalized form can be converted into Linear Temporal Logic (LTL), which will then be used for formal verification. Natural-language requirements show intuitive value, but they lack explicit temporal semantics, which makes direct verification impossible. The limitation is solved by converting structured FRETish specifications into temporal logic formulas, which enable verification of the specifications.

The Table 3 presents representative examples across all requirement categories, showing the progression from natural language to FRETish and finally to Future-Time LTL. (Full detailed Future and Past-Time LTL formulations are provided separately in Appendix/Index.)

Researchers developed predicates that serve as understandable representations of complex environmental conditions. The condition C which defines both environmental and vehicle limitations, and speed-ok which indicates safe speed requirements:

1. Future-Time Interpretation

The system uses a general representation, which it defines as: G(C→speed-ok)

The formula defines an operational safety requirement that must be fulfilled whenever condition C activates. FRET semantics require conditions and responses to exist at the same time for their implementation.

2. Past-Time Interpretation

The requirement for trace-based analysis can be shown through past-time logic, which expresses the requirement as: H(C→speed-ok)

The formulation verifies that all past requirements have been fulfilled, which makes it appropriate for running system checks and evaluating completed processes.

The research findings demonstrate that FRET-based logic successfully models both trigger-based system behavior and continuous system operation. The requirements present in SPD.03 impose ongoing restrictions, which include speed limits, whereas other requirements define performance responses according to specific events. The generated formulas possess architectural complexity, yet they provide a detailed representation of temporal relationships between events and the required system behavior.

The process of transforming the original requirements into machine-verifiable form maintains their original meaning. The three elements of FRETish structure, diagrammatic semantics (RQ2), and LTL encoding show consistent alignment across different representation methods.

The study results show that FRAV requirements can be converted into future-time and past-time temporal logic, which provides a path to connect with formal verification methods used in model checkers and runtime monitors.

d. Requirement Quality Evaluation and Empirical Validation

This section evaluates the impact of FRET-based formalization on requirement quality by comparing natural-language (NL) and FRETish representations across five dimensions: semantic clarity, structural quality, completeness, ambiguity, and consistency. The results are presented per FRAV category to ensure systematic assessment.

1. Road-Related Requirements

Tables 4 present the comparative results of the Road-related requirement. After formalization, semantic scores increase significantly (0.9) due to explicit scoping, conditions, and responses.

A slight reduction in completeness is observed, reflecting the decomposition of implicit assumptions into explicit elements required for formal verification. Overall, the FQI improves (0.84 → 0.89), indicating enhanced clarity and verification readiness. The heatmap 7 confirms consistent gains in semantic quality and reduced ambiguity while maintaining structural stability.

TABLE 3: *PROGRESSIVE FORMALIZATION FROM NL TO FRETISH AND TEMPORAL LOGIC*

ID	FRAV (Original NL)	FRETish (Formalization)	Linear Temporal Logic (LTL)
ROD.01	FRAV incorporates physical infrastructure (lane markings, crosswalks, junctions)	When operating ADS SHALL satisfy lane_markings | crosswalk | junctions	G(operating → F(lane_markings ∨ crosswalk ∨ junctions))
SCR.02	FRAVincludesvulnerable (pedestrians, cyclists) for safety	When traffic_composition = pedestrian ∧ cyclist ADS SHALL always satisfy safety	G((pedestrian∧cyclist) → safety)
SPD.03	FRAV uses friction/acceleration to infer safe speed	Whenever road_condition = normal ∧ vehicle_type = car | motorcycle ADS SHALL always satisfy vehicle_speed ≤ safe_speed	G((normal∧vehicle) → (speed ≤ safe_speed))
OBJ.04	Vehicle behaviors: decelerating, stopping, lane change, turning, etc.	In obstacle_anticipation_mode WHENEVER road_user = vehicles ADS SHALL always satisfy safe vehicle_behavior	G((mode∧vehicles) → F(safe_behavior))

TABLE 4: *QUALITY ASSESSMENT OF ROAD-RELATED REQUIREMENTS AFTER FRET FORMALIZATION*

Index	Req_ID	Semantic	Structural	Completeness	Ambiguity	Consistency	FQI
0	R1	0.897	1.0	0.700	0.909	1.0	0.901
1	R2	0.875	1.0	0.700	0.909	1.0	0.895
2	R3	0.888	1.0	0.512	0.923	0.8	0.833
3	R4	0.876	1.0	0.571	1.000	1.0	0.883
4	R5	0.914	1.0	0.600	0.929	1.0	0.888
5	R6	0.904	1.0	0.600	0.944	1.0	0.888
6	R7	0.899	1.0	0.827	1.000	1.0	0.940
7	R8	0.909	1.0	0.767	1.000	1.0	0.931
8	R9	0.775	1.0	0.626	0.889	1.0	0.852

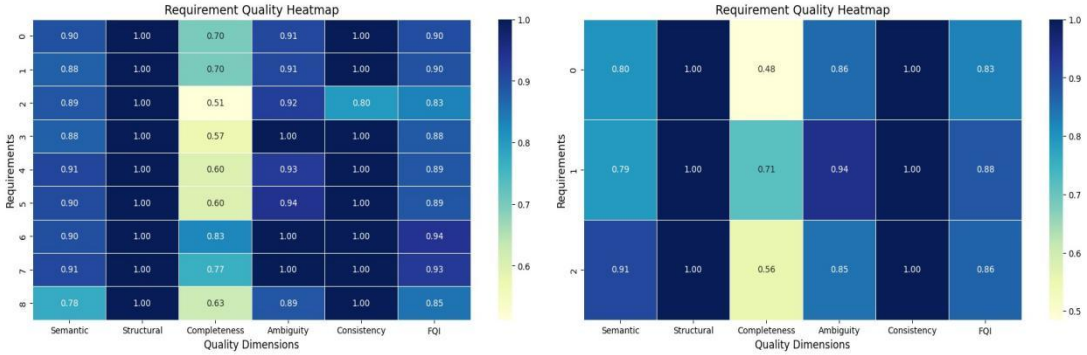

Fig. 7: Requirement Quality Heatmap of Road-Related Requirements

2. Quality assessment of Conditional Situation Requirements

The table 5 summarize the results of the comparative analysis of the situational condition requirements of NL and FRETished in the FRET tool.

The requirements for natural language demonstration show average understanding because they contain hidden time-based requirements. FRET formalization makes these explicit, resulting in improved semantic scores (0.79–0.90) and preserved structural consistency (1.0). The explicit separation of condition–response elements leads to a minor reduction in completeness.

The first statement shows that FQI values increased from 0.81 to 0.88, showing better performance in both interpretability and precision. The heatmap 8 displays condition-related information with greater clarity while showing less uncertain elements.

3. Speed-Related Requirements Quality Evaluation

This category exhibits the most notable improvement after formalization. The requirements from the Speed-related table 6, after they were compared with NL, show great improvements.

NL requirements suffer from low structural quality, high ambiguity, and weak completeness due to vague quantitative constraints. After FRET formalization, semantic clarity becomes consistent (0.89–0.90), and structural quality reaches 1.0 across all requirements. Completeness remains moderate as hidden parameters become explicit.

The FQI improves significantly (0.55–0.77 → 0.80–0.87). The heatmap 9 displays substantial improvements in both semantic and structural aspects, which shows that the formalization process effectively enhances constraint-driven environmental requirements.

TABLE 5: *FRET FORMALIZED SITUATIONAL CONDITION REQUIREMENTS EVALUATION*

Index	Req_ID	Semantic	Structural	Completeness	Ambiguity	Consistency	FQI
0	R1	0.800	1.0	0.485	0.857	1.0	0.826
1	R2	0.792	1.0	0.715	0.944	1.0	0.883
2	R3	0.905	1.0	0.555	0.846	1.0	0.864

TABLE 6: *QUALITY ASSESSMENT OF SPEED-RELATED REQUIREMENTS AFTER FRET FORMALIZATION*

Index	Req_ID	Semantic	Structural	Completeness	Ambiguity	Consistency	FQI
0	R1	0.893	1.0	0.500	0.808	0.900	0.829
1	R2	0.891	1.0	0.538	0.750	1.000	0.843
2	R3	0.889	1.0	0.444	0.818	0.778	0.800
3	R4	0.896	1.0	0.647	0.864	0.857	0.862
4	R5	0.895	1.0	0.615	0.857	0.750	0.838
5	R6	0.899	1.0	0.474	0.960	1.000	0.864
6	R7	0.901	1.0	0.571	0.905	1.000	0.875

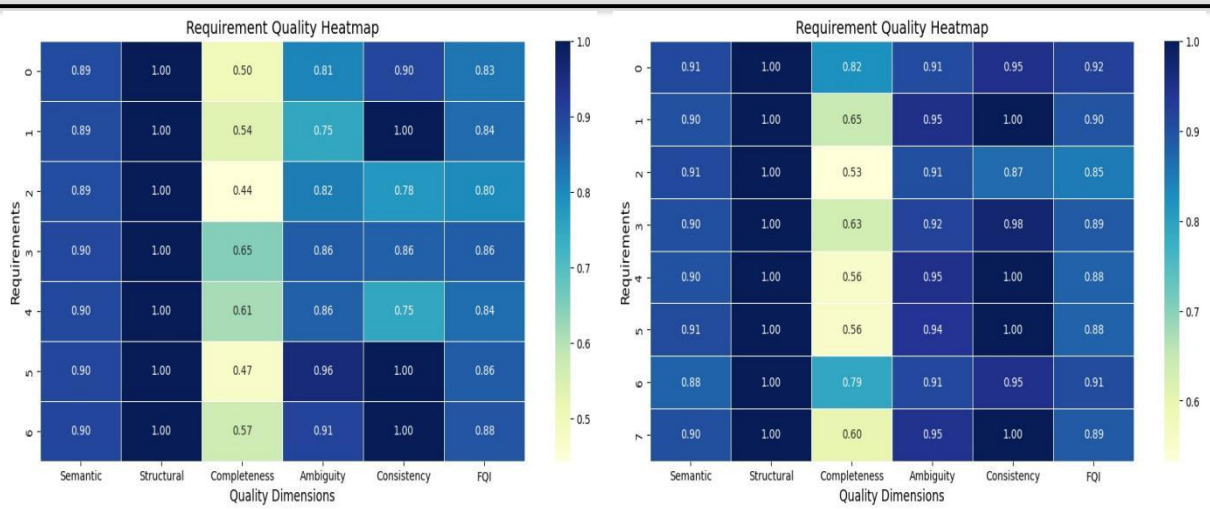

Fig. 9: Heatmap of Formalized Speed-related Requirements

4. Object Detection Related Requirements Quality Evaluation

The Tables 7 present comparative results of the final object detection requirements of ADS while perceiving the environment. The semantic clarity of NL requirements demonstrates significant fluctuations because their behavioral descriptions are complicated. After formalization, semantic scores converge (0.90), and structural quality becomes uniform (1.0). The system shows improved consistency between (0.95–1.0) complete system analysis and its ability to measure complex behaviors through distinct components.

The FQI stabilizes within a high range (0.85–0.92). The heatmap 10 shows that quality improvements have become more stable while requirement testing now shows less variability.

V. DISCUSSION

The chapter interprets its empirical findings according to research objectives, which demonstrate how FRET-based formalization enhances requirement quality, together with interpretation capabilities and autonomous driving system verification readiness. The discussion synthesizes key findings while it explains observed patterns and describes their practical implications, and acknowledges existing limitations.

1. Key Findings

The entire process from requirement extraction to FRETish formalization and diagrammatic semantics, temporal logic generation, and quality assessment demonstrates continuous improvement throughout its various stages.

The formalization process resulted in major enhancements to both semantic understanding and structural uniformity of the content. The FRET tuple
(Scope-Condition-Timing-Response) system that was implemented to establish requirements showed complete precision in standardizing requirement presentation through its requirement definition process. The system achieved better verification capabilities because the team succeeded in converting all project requirements into formal verification materials, which included LTL, SMT-LIB, and JSON/XML documents.

The use of diagrammatic semantics improved understandability because it provided clear explanations of temporal relationships and condition-response connections, which helped both technical experts and non-experts understand the requirements. The study showed that researchers made a strategic decision to achieve controlled completion, which led to both complete research results and incomplete research results. The requirement scores decreased because the assessment discovered hidden assumptions, which resulted in better analysis and verification readiness for the requirements.

TABLE 7: *QUALITY METRICS FOR OBJECT-RELATED REQUIREMENTS AFTER FRET FORMALIZATION, SHOWING THE IMPACT OF STRUCTURED SEMANTICS AND PREDICATE BINDING ON THE SAME QUALITY DIMENSIONS*

Index	Req_ID	Semantic	Structural	Completeness	Ambiguity	Consistency	FQI
0	R1	0.912	1.0	0.824	0.909	0.950	0.922
1	R2	0.902	1.0	0.645	0.952	1.000	0.897
2	R3	0.911	1.0	0.533	0.905	0.867	0.850
3	R4	0.904	1.0	0.633	0.920	0.978	0.887
4	R5	0.898	1.0	0.555	0.950	1.000	0.878
5	R6	0.907	1.0	0.555	0.941	1.000	0.879
6	R7	0.877	1.0	0.792	0.913	0.950	0.907
7	R8	0.904	1.0	0.600	0.947	1.000	0.888

2. Interpretation of Results

Semantic and Structural Gains

The observed improvements stem from explicit encoding of operational context and system behavior. FRET controls expression standards through structured fields because that approach provides requirements that people can understand and machines can process. The standardization process provides essential support for both automated processes and formal verification activities.

Completeness vs. Verifiability

The reduction in completeness shows that natural-language requirements. FRET exposes these gaps, which force engineers to create essential requirements by defining missing parameters. The process results in better requirements.

Diagrammatic Semantics

The timeline-based diagrams function as a cognitive layer that displays all temporal constraints and dependencies through visual representation. The system enables faster requirement validation through reduced interpretation work while also supporting time-dependent ADS scenario validation.

Temporal Logic Duality

The combination of future-time and past-time temporal logic enables two different methods of verification. The design process uses future-time logic for model checking while past-time logic allows both runtime monitoring and trace validation to improve the overall assurance process.

Toolchain Practicality

The pipeline demonstrates strong interoperability with verification tools, requiring minimal manual adjustments. This confirms that FRET can bridge the gap between informal requirements and formal verification workflows in practical settings.

VI. CONCLUSION AND FUTURE WORK

The research establishes a methodical workflow that converts autonomous driving environmental requirements from their natural language form to officially validated specifications. The research team used the FRAV dataset to create formal requirements, which they expressed in FRETish and showed through diagrammatic semantics, and converted into temporal logic representations. The study results demonstrate that formalization processes create better semantic understanding, which establishes consistent organizational patterns and permits direct verification tool compatibility through LTL and other related elements.

The research results demonstrate that FRET-based formalization improves verification capabilities across all requirement categories while maintaining the original requirements. Diagrammatic semantics enable stakeholders to better understand content by showing both temporal and causal connections in an explicit manner. The completeness scores show a decline because the study reveals hidden assumptions, but this outcome strengthens the requirement specifications through detailed and traceable documentation. The method creates an effective solution which connects informal requirements with formal verification processes.

Future research should work to develop better automated systems that can manage increased workload demands for pipeline operations. The project requires three particular enhancements, which include creating standardized parameter models that apply to specific domains, developing automatic translation capabilities for backend systems, and implementing native past-time temporal logic into verification toolchains. The research will use wider empirical testing across various datasets and controlled user studies, which involve stakeholders to measure the system's ability to function effectively in diverse engineering situations.

References:

S.-C. Necula, F. Dumitriu, and V. Greavu-S, erban, "A systematic literature review on using natural language processing in software requirements engineering," Electronics, vol. 13, no. 11, p. 2055, 2024.

M. Alenezi and M. Akour, "Ai-driven innovations in software engineering: a review of current practices and future directions," Applied Sciences, vol. 15, no. 3, p. 1344, 2025.

C. Ponsard, G. Dallons, and M. Philippe, "From rigorous requirements engineering to formal system design of safety-critical systems," ERCIM News, no. 75, pp. 22–23, 2008.

A. Abdullah, R. A. Bakar, K. Gunaratnam, F. Hujainah, and M. F. A. Farid, "Safety property attributes in critical systems for requirement specification: A review," in 2023 IEEE 8th International Conference On Software Engineering and Computer Systems (ICSECS). IEEE, 2023, pp. 481–486.

S. Frank, A. Hakamian, D. Zahariev, and A. van Hoorn, "Verifying transient behavior specifications in chaos engineering using metric temporal logic and property specification patterns," in Companion of the 2023 ACM/SPEC International Conference on Performance Engineering, 2023, pp. 319–326.

A. Rasheed, B. Zafar, T. Shehryar, N. A. Aslam, M. Sajid, N. Ali, S. H. Dar, and S. Khalid, "Requirement engineering challenges in agile software development," Mathematical Problems in Engineering, vol. 2021, no. 1, p. 6696695, 2021.

L. Montgomery, D. Fucci, A. Bouraffa, L. Scholz, and W. Maalej, "Empirical research on requirements quality: a systematic mapping study," Requirements Engineering, vol. 27, no. 2, pp. 183–209, 2022.

R. Silhavy, P. Silhavy, and Z. Prokopova, "Analysis and selection of a regression model for the use case points method using a stepwise approach," Journal of Systems and Software, vol. 125, pp. 1–14, 2017.

M. Kundi, F. Ahmad, and R. Monahan, "Refining environmental requirements for autonomous driving systems: Leveraging the frav framework," in 2025 55th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSNW). IEEE, 2025, pp. 15–22.

D. Giannakopoulou, A. Mavridou, J. Rhein, T. Pressburger, J. Schumann, and N. Shi, "Formal requirements elicitation with fret," in International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ-2020), no. ARC-E-DAATN77785, 2020.

F. de Bruijn and H. L. Dekkers, "Ambiguity in natural language software requirements: A case study," in International Working Conference on Requirements Engineering: Foundation for Software Quality. Springer, 2010, pp. 233–247.

A. Mavin, P. Wilkinson, A. Harwood, and M. Novak, "Easy approach to requirements syntax (ears)," in 2009 17th IEEE international requirements engineering conference. IEEE, 2009, pp. 317–322.

E. Kamsties and B. Peach, "Taming ambiguity in natural language requirements," in Proceedings of the Thirteenth international conference on Software and Systems Engineering and Applications, vol. 1315, 2000.

B. Ramesh and M. Jarke, "Toward reference models for requirements traceability," IEEE Transactions on Software Engineering, vol. 27, no. 1, pp. 58–93, 2002.

E. Clarke, O. Grumberg, and D. A. Peled, "Model checking the mit press," Cambridge, Massachusetts, London, UK, vol. 988, 1999.

M. B. Dwyer, G. S. Avrunin, and J. C. Corbett, "Patterns in property specifications for finite-state verification," in Proceedings of the 21st international conference on Software engineering, 1999, pp. 411–420.

H. Garavel, M. H. Ter Beek, and J. Van De Pol, "The 2020 expert survey on formal methods," in International Conference on Formal Methods for Industrial Critical Systems. Springer, 2020, pp. 3–69.

R. Lorch, B. Meng, K. Siu, A. Moitra, M. Durling, S. Paul, S. C. Varanasi, and C. McMillan, "Formal methods in requirements engineering: Survey and future directions," in Proceedings of the 2024 IEEE/ACM 12th International Conference on Formal Methods in Software Engineering (FormaliSE), 2024, pp. 88–99.

C. M. de Ferro, A. Mavridou, M. Dille, and F. Martins, "Simplifying requirements formalization for resource-constrained mission-critical software," in 2023 53rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W). IEEE, 2023, pp. 263–266.

M. Farrell, M. Luckcuck, R. Monahan, C. Reynolds, and O. Sheridan, "Fretting and formal modelling: A mechanical lung ventilator," in International Conference on Rigorous State-Based Methods. Springer, 2024, pp. 360–383.

M. Luckcuck, M. Farrell, O. Sheridan, and R. Monahan, "A methodology for developing a verifiable aircraft engine controller from formal requirements," in 2022 IEEE Aerospace Conference (AERO). IEEE, 2022, pp. 1–12.

M. Farrell, M. Luckcuck, R. Monahan, C. Reynolds, and O. Sheridan, "Adventures in fret and specification," in International Symposium on Leveraging Applications of Formal Methods. Springer, 2024, pp. 106–123.

R. J. Costello and D.-B. Liu, "Metrics for requirements engineering," Journal of Systems and Software, vol. 29, no. 1, pp. 39–63, 1995.

R. J. Halligan, "Requirements metrics: the basis of informed requirements engineering management," in COMPLEX SYSTEMS ENGINEERING AND ASSESSMENT TECHNOLOGY WORKSHOP,1993