Requirement Validation for Embedded Systems in Automotive Industry Through Modeling

DANISH IQBAL[1], **ASSAD ABBAS**[1], **MAZHAR ALI**[2],
MUHAMMAD USMAN SHAHID KHAN[2], **AND**
RAHEEL NAWAZ[3], (Member, IEEE)

[1]Department of Computer Science, COMSATS University Islamabad, Islamabad Campus, Islamabad 45550, Pakistan
[2]Department of Computer Science, COMSATS University Islamabad, Abbottabad Campus, Abbottabad 22060, Pakistan
[3]Business School, Manchester Metropolitan University (MMU), Manchester M15 6BH, U.K.

Corresponding author: Assad Abbas (assadabbas@comsats.edu.pk)

ABSTRACT Requirement validation contributes significantly toward the success of software projects. Validating requirements is also essential to ensure the correctness of embedded systems in the auto industry. The auto industry emphasizes a lot on the verification of car designs and shapes. Invalid or erroneous requirements lead to inappropriate designs and degraded product quality. Considering the required expertise and time for requirement validation, significant attention is not devoted to verification and validation of requirements in the industry. Currently, the failure ratio of software projects is significantly higher and the key reason for that appears to be the inappropriate and invalidated requirements at the early stages in the projects. To that end, we propose a model-based approach that uses the existing V&V model. Through virtual prototyping, the proposed approach eliminates the need to validate the requirements after each stage of the project. Consequently, the model is validated after the design phase and the errors in requirements are detected at the earliest stage. In this research, we performed two different case studies for requirement validation in the auto industry by using a modeling-based approach and formal technique using Petri nets. A benefit of the proposed modeling-based approach is that the projects in the auto industry domain can be completed in less time due to effective requirements validation. Moreover, the modeling-based approach minimizes the development time, cost and increases productivity because the majority of the code is automatically generated using the approach.

INDEX TERMS Requirement engineering, requirement validation, formal verification, requirement modeling, embedded system, automotive industry.

I. INTRODUCTION

Requirement Engineering (RE) is a key activity during the software engineering process that plays a vital role in the success of software projects [1]. Every business wants to improve its productivity, minimize the cost of projects, and achieve maximum quality. Nowadays, the emphasis of most of the software-assisted industries is on the accuracy, reliability, and trustworthiness of the products. For this purpose, validating requirements at the earlier stage of the projects is essentially vital. A requirement is a specification of a need(s) and the term requirement denotes the functionality of the system [2].

In the RE process, initially, all the requirements are written in natural language and subsequently the experts analyze these requirements and develop specifications [3], [4].

The associate editor coordinating the review of this manuscript and approving it for publication was Resul Das.

A specification is the systematic form of requirements. When the requirements are collected, the validation and specification of these requirements is an issue since formalizing the requirements and extracting the appropriate word from the written document is a challenging task for requirement engineers [5]. Extracting the appropriate word from the natural language increases the understand-ability and clarifies the functional requirements of the system [6]. The goal of information extraction is to automatically distill and structure the unstructured user stories, to make it more facile to locate information of interest in sizable volumes of data [7]. Given the large amounts of data available in a digitized textual form that is collected from the user, it is paramount to provide mechanisms that sanction users to extract nuggets of relevant information from the ever-growing volumes of potentially paramount documents [8].
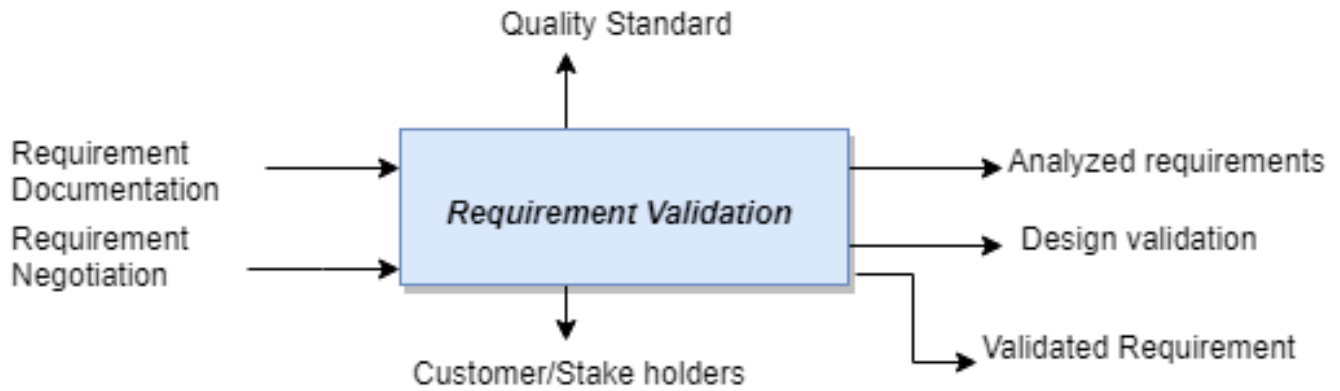
A large number of errors occur in the systems due to the poor requirements and several systems fail due to the informal requirements [9]. Requirement validation and specification ensure that all the requirements are correct, complete, consistent, and fulfill the customer needs [10]. For this purpose, requirement models can be created to ensure that all the requirements are correct and satisfy customer needs. Figure 1 shows the requirement validation process by specifying the inputs and outputs.

Applying the requirement validation at the early stage of the system has numerous benefits, for example, the issues arising due to imprecise requirements at later stages of the software development can be evaded. For a life-critical system, if the requirements are not validated at the start, then there might be substantial risks for human life if the problems appear later due to ambiguous requirements. Moreover, a benefit of validating requirements at an earlier stage is that it not only results in significant cost reduction but also minimizes threats to human lives in case of safety-critical systems [11].

In the past decades, several formal techniques were used to reduce the cost and to achieve quality in software products. The human involvement in validation and analysis of requirements increases the chances of errors. Requirement validation of software projects is a challenging but necessary step to ensure the success and quality of developed software. Validating requirements is a complex task because it requires a lot of effort, time, and cost to verify the requirements formally. Invalid or erroneous requirements result in the form of failure of the project.

A model-driven and rapid prototyping approach in agility increase the understandability of requirements. However, they are also costly, since it should not be used in large-scale and embedded projects [12].

Many software projects fail because the requirements are not validated in the initial stages of the projects and consequently, the requirements remain vague, incomplete, or inconsistent [13]. Such requirements result in problems at the later stages of the projects. However, validating the requirements at the earlier stages of the project will not only minimize the errors but also increases productivity.

Software modeling is an essential factor in project success. A model is a graphical visualization of factor that enhances the understanding of software requirements [14]. The benefit of modeling is that the requirement can be validated earlier, and the fault detection and correction is less expensive. Moreover, the model understanding is easier not only for requirement analysts but also for the customers.

Requirement validation is also important for the auto industry because it contributes toward better auto designs or shapes. For example, a better design or shape of racing cars requires the dragging force to be minimized. Considering the complexity of the auto manufacturing processes, the aforementioned requirement related to the auto industry can be easily validated using the modeling-based techniques in contrast to performing the reviews, inspection, prototyping, and formal verification. Moreover, a benefit of validation through modeling is that it not only reduces the development time and cost but also increases productivity. The above-mentioned benefits are also highly desired by the auto manufacturing industry.

A Petri net is a graphical and mathematical modeling tool applicable to many systems [15]. As a graphical tool, they can be used as visual communication aids. On the other hand, as a mathematical tool, it is possible to set up state equations, algebraic expressions, and other system behavior. The primary difference between Petri net and modeling is the presence of tokens, which are used for the simulation of system's concurrent and dynamic behavior. The Petri net focuses on the correctness of the system state while the modeling focuses on both the system's accuracy and design. Requirement validation through modeling is easier to understand for the user; however, the Petri nets are complicated for the user to understand and therefore, require high-level of expertise [16].

Requirement modeling is the most essential activity when we know about the culture of the organization but also plays a vital role in the initial stage of the project. In the first phase when the requirements are collected, the requirements are in the form of stories that need to be presented in a formal way that every stakeholder can understand easily. In this paper, we apply a modeling-based methodology which is a

combination of the formal techniques and mathematical modeling for requirement validation in embedded system design. A benefit of the proposed modeling-based approach is that it not only helps in ensuring quality during the requirement analysis process but also minimizes the costs and time for the project [17]. The proposed modeling-based requirements validation approach removes the ambiguities and inconsistencies from the requirements because every requirement is finalized at the start of the project. Moreover, the aforementioned approach increases the communication between different stakeholders that prevent the project from reworks, due to requirements fixed at the initial stage.

According to research by Carnegie Mellon University's Edge Case Research group, the problem in the auto industry is that the requirements are more complex and therefore, complete testing seems impractical due to mass-scale production. Therefore, the manufacturers' focus remains on the key functionality rather than drawing correlations among various factors, for example, how significantly dragging force affects the fuel consumption and speed of racing cars [18]. Contrary to the formal method techniques, the requirement validation through modeling does not emphasize on the expertise of a human. Instead, it auto-generates the code that increases productivity and minimizes the time and budget of the project. There are different modeling techniques, such as simulation, test case generation, inspections, and reviews. Improving the quality and productivity of the product is the main concern of every auto industry. Organizations employ different approaches to validate the requirements. However, for complex domains, such as the auto industry that require extensive testing, it is hard to validate the requirements completely through conventional requirement validation processes. In this paper, we performed simulation-based modeling that validates the requirements through modeling in the auto industry. A benefit of requirement validation through modeling is that it allows accommodating requirement changes at the start stage and fixes the problems as per customer needs thereby minimizing the chances of project failure [19].

To accomplish the requirement validation task in the auto industry through modeling, we used the SCADE tool. Through SCADE, we validate the requirements of embedded systems in the auto industry and focus on requirement validation for both the functional correctness as well as the attractiveness of designs or shapes of the racing cars. The key contributions of this research are:

- We present a model-based approach for requirement validation that minimizes the iterations performed after each phase in the V&V model.
- The approach detects the errors at the early stages of the project through virtual prototyping and consequently can minimize failure or risks in the later stages of the project.
- The proposed technique increases productivity through auto code generation because we do not have to spend a lot of time in project implementation.

- Through the virtual prototyping, we can reduce the production cost and improve the quality of the project.
- The proposed approach does not require human expertise for requirement validation because it validates the requirements through modeling and simulation.

The rest of the paper is organized as follows: Section 2 presents the related work whereas Section 3 discusses the motivation for the proposed work. The proposed model and validation through SCADE along with the proposed methodology are discussed in Section 4. Section 5 describes the simulation results of our proposed methodology whereas Section 6 concludes the paper.

II. RELATED WORK

For over a decade, industries have used different techniques for requirement validation for embedded systems. In [20], the authors validated the requirements through a formal approach for the European railway system. The authors formalized a set of 90 requirements and subdivided the requirements into different classes and subsequently converted the requirements from informal to formal way with user acceptance and scalability. Although the authors solve the specific problem, the technique does not have applicability for all embedded systems because it requires expertise to apply formal techniques and is, therefore, considered as expensive.

The authors in [3], applied the requirement validation on the new millennium program—a NASA's project with a focus on requirement validation for space flight system. The authors initially formalized the requirements for the embedded system and subsequently implemented the code and in the end, performed validation through the formal technique. Moreover, the authors negotiated the process activities and design sense making [21] and solved the issues of requirement stabilizing during project selection. However, a downside of such approaches is their applicability up-to abnormalities and failures observance only to ABS system. [22].

During the past decade, different formal method techniques have been proposed to reduce the cost of software and to improve quality. Zafar et al. [23] solved the problem of requirement validation through a formal technique where the requirement analyst had no expertise in informal techniques and also reduced the cost of testing. The authors in [24] used the Specification of Simple Control System (SCR) for requirement specification and validation. The SCR method automatically generates test cases and source code. A recent study [25] was conducted in which the researchers developed a tool that generates the test cases from the code. Through test cases the authors validated only functional requirements; however, the tool does not give any indication about the efficiency and correctness [26].

In the auto industries, it is impossible to accomplish exhaustive testing at the initial stage to guarantee the safety of the life-critical systems. The authors in [27] proposed an approach to improve validation safety and efficiency and also to increase the accuracy of Highly Automated Vehicles (HAVs). The approach improved the testing efficiency at the

phase level and also certified the techniques from ISO 26262. However, the proposed technique did not provide effective results on the system (complete testing) level testing due to its reliance on some assumptions.

Another related study conducted by [28] used a tool called T tool based on NuSMV for requirement specification. The authors checked the scalability through different techniques and specified the requirement at the early stages. The approach was able to find defects at the initial stage informally and efficiently but failed at large-scale and embedded systems projects. Moreover, the tool does not provide any graphical representation and is consequently hard to understand.

The authors in [4] employed a methodology which prescribes how different stakeholder use the same set of requirements virtually and validate those requirements. The authors solved the problem of communication and collaboration between client and requirement engineer. For this purpose, the authors developed a tool called TestMEReq through which requirement engineers effectively communicates the requirements with the customer to fix the issues at the initial stage. The Tool not only reduces the time and cost required to fix the requirements but also restricts the development organization form the incorrect implementation. TestMEReq tool facilitates the client and requirement engineer by expediting the process of requirement validation through face to face meetings. However, the proposed approach is not effective for large-scale projects because of the difficulty to facilitate face to face meetings of a large number of stakeholders and team members.

Kiran et al. [17] proposed a framework to define different requirement validation techniques for requirement validation based on the nature of projects. The proposed framework is useful for project managers because it helps them in selecting the requirement validation tools or techniques for different projects that eventually reduces the time and cost of the project. The techniques considered by the authors include prototyping, inspection, testing, viewpoint, model-based, and simulation. The approach mainly focuses on facilitating requirement engineers for the selection of appropriate requirement validation techniques. The proposed technique was applied to a project where designing is a key concern of the client. Therefore, based on the proposed framework's guidelines, the particular requirement validation technique is selected, and a prototype is generated. Prototyping, in this case, shows the design properties and functionality of the system but does not encompass the correctness of the system functionality. Through this framework, the quality of the project has improved; due to selecting the proper requirement validation techniques. The proposed framework guidelines are limited to specific projects and techniques.

The author in [12] proposed a framework to review the different approaches in agile methodology with model-driven development. The research combines different phases of agile with model-driven development or test-first development to build the software. The authors evaluated their work through a case study that is limited to a specific domain. The framework gives efficacious results in small scale project in agile methodology to utilize a model-driven approach. But their methodology has a limitation to utilize in astronomically immense scale project and very costly to utilize in embedded system.

The authors in [29] demonstrating the safety of automated driving requirement validation and testing. The level of safety assurance demanded embedded systems would require at least a billion hours of testing. In fact, this number should be much bigger to achieve statistical significance. This leads to the idea of the in-feasibility of complete testing.

Rapid changes in technology brought revolution in the auto industry to increase the safety of life in modern vehicles. In [30] the authors proposed a systematic approach to model-based supervisor design, which helps to increase the safety of traffic and to reduce the fuel consumption based on dragging force. The proposed study also helps in air pollution reduction. A model-based supervisory approach is effective for requirement validation of the auto industry in five stages namely, (i) behavior specification, (ii) requirement modeling, (iii) controller synthesis, (iv) simulation, and (v) testing. The proposed model helps to remove the ambiguity from the requirements. It generates codes and solves the problem of traffic automation with advanced vehicles. The authors solved the problem of automatic vehicles of Toyota and performed testing successfully on each phase. The integration testing of the proposed technique did not yield effective results due to inappropriate code generation for different modules which creates problems in integration.

The authors in [31] demonstrated the usage of Petri nets for developing an embedded system to check the unambiguity, completeness, correctness, and consistency in requirements. The study focuses on the specification of embedded system software with early validation of the model using Petri nets. The requirement validation through Petri nets concentrates on the correctness of the system rather than the design functionality of the system.

In the last years, the requirements engineering community is witnessing the emergence of research approaches based on the exploitation of huge amounts of data gathered from software repositories and system usage. These approaches tackle research questions such as identifying candidate features, predicting productivity and planning releases. The authors in [32] also present a novel, data-driven approach to providing automated support for project managers and other decision-makers in predicting delivery capability for an ongoing iteration.

The authors in [17] identified various requirement validation techniques like requirements testing, prototyping, reviews, writing user manual, preview, formal specification, viewpoint Based, simulation-based, throwaway prototyping, model checking, evolutionary processes, theorem proving, commenting, inspection process, test case driven inspection, walk-through, reading, ad-hoc based, checklist-based, perspective, defect based, and scenario-based. The techniques

are helpful to build defect-free and quality software products. The proposed study tells about the benefit of each technique, which helps us developing quality software. These techniques require an expert to select the appropriate technique for requirement validation which makes the project more costly and time-consuming. Table 1 presents a comparison of different requirement validation techniques, their features, and association among them.

III. MOTIVATION FOR THE PROPOSED WORK

According to the Standish research report, waterfall projects have three times the failure rate as compared to the agile projects [39]. The United States spends around $250 billion each year on IT applications and approximately 31.1% of projects fail due to the complex requirements [40].

Waterfall projects are not flexible enough to accommodate the change of requirements at each stage as compared to the agile methodology. The reason for the failure of most of the projects appears to be a failure to meet the: (i) schedule, (ii) budget of the project, (iii) needs of stakeholder, and (iv) project functional requirements and scope, because of the complexity and volatility of the requirements. The above-mentioned failure reasons are mainly overlooked in the RE Processes. Overseeing the mentioned factors negatively impacts the reputation, marketability, and customer satisfaction of the company's products.

Therefore, it is equally valid to state that the RE processes are related not only to the cost and time but also affect the reputation of the entire organization. Moreover, failure of IT projects is not only risky in terms of cost but also life-threatening due to poor requirement validation. Furthermore, the Standish report has proven that inappropriate RE practices can cause around 43% of project failure [39]. However, project failure can be minimized through appropriate understanding, negotiation, and validation of the requirements. To validate the requirements at the early stage of the project, we propose a model-based approach that deals with the complex and changing requirements to overcome the project failure, budget, and time.

IV. PROPOSED METHODOLOGY

Validation of requirements is one of the most important step for project success. Validating the requirements through formal techniques not only improves the quality and accuracy but also reduces the cost of the project [27]. In this paper, we have proposed a model-based approach for requirement validation in the automotive industry using the existing V&V model. The proposed approach validates requirements through the combination of formal and mathematical modeling. It, unlike the V&V process model, eliminates the need to validate the requirements at every phase of the project. This approach not only improves the quality of the project but also increases the accuracy of the requirements. It also reduces the burden of rework in situations where requirements change later on in the project. The proposed methodology consists of the following steps:

1) Identification of requirements validation steps using the V & V process model.
2) Elimination of steps from the V & V process model
3) Requirements Modeling Using Tool for V & V

Below is the description of all the steps involved:

A. IDENTIFICATION OF REQUIREMENTS VALIDATION STEPS USING THE V & V PROCESS MODEL

The V model is a model for the Software Development Life Cycle (SDLC) where the process executes sequentially. It is based on the association of a testing phase for each corresponding development stage. Testing activities are performed extensively at each development phase of the project that is extremely risky and time-consuming. This model is not suitable for projects where requirements frequently change. This also does not support the iteration of phases, which is time-consuming because of the testing process that is performed after each phase of development. Due to the above-stated issues, we have proposed a model that identifies different development phases and helps us in requirement designing and validation. The process of how different phases of the V model are identified for a requirements validation is shown in Figure 2.

The proposed approach identifies the requirements validation steps that helps us develop a quality project and fulfill customer needs. The model-based approach validates the requirement in seven steps.

- Step 1: The requirement analyst gathers the requirements from the customer that is called the customer needs.
- Step 2: Requirements written in natural language are analyzed and converted into technical jargon.
- Step 3: After analysis, the requirements are modeled to make them more understandable to all stakeholders.
- Step 4: After modeling, these models are converted into an architecture which is more meaningful than a model.
- Step 5: After completing the architecture, a design is created to validate the shape of the object (in this case a car) in the designing phase.
- Step 6: The modeling-based approach automatically generates codes based on design.
- Step 7: We perform the testing where we evaluate the main functionality of the system.

The proposed model solves the complex user needs very significantly to improve the accuracy and offer more accurate results without increasing the human resources. To improve the organization process, the proposed methodology helps the organization to amplify and improve the progress on lower cost. It also validates the requirements in the beginning and subsequently reduces the cost as well as the efforts of the project. We create a model from requirements and then validate it through a model-based approach to make equation and simulate it. In the beginning, it gives the output of our requirement by just simulating. In this way, we can get results without development. It avoids the repetition of verification again and again in the V&V model.

TABLE 1. Comparison of different requirement validation techniques.

Features	Formal Method [24]	Prototyping [33]	Machine learning [27]	Validation through Testing [34]	Review Based [17]	Inspection [17], [11]	Test cases [14]	Viewpoint [35]	Simulation [36]	Test-driven development [37]	Modeling through SCADE [38]
Correctness	✓		✓	✓			✓		✓	✓	✓
Designing		✓							✓		✓
Consistency		✓	✓							✓	✓
completeness				✓							✓
Formalism	✓										✓
Efficiency	✓						✓				✓
Accuracy	✓		✓							✓	✓
Trace functional requirement	✓	✓					✓	✓			✓
Check All possible Paths	✓							✓			✓
Requirement Traces Informally					✓	✓					
Requirement Scenario									✓		
Less Effort and Time required for Validation											✓
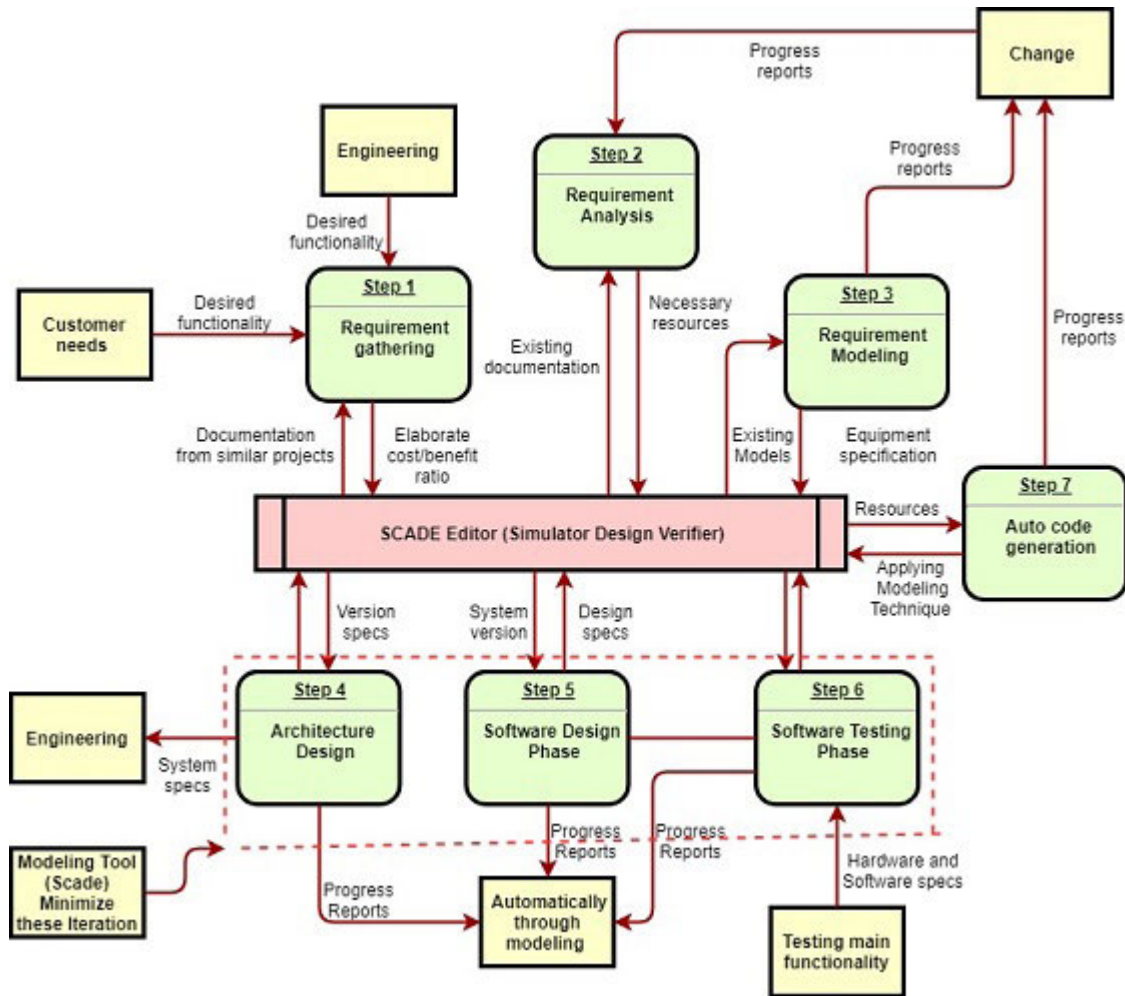
FIGURE 2. Proposed model of identification and validation of requirements for an embedded system in the auto industry.

B. ELIMINATION OF THE STEPS FROM THE V & V PROCESS MODEL

As the industry is evolved, the technologies have also become more complex and undergo rapid changes. To increase the productivity of the development organization, we have eliminated some steps form the V model that speeds up the development process without compromising on quality. Moreover, our proposed technique completely accommodates the IEC 61508 standard [38].

The model-based approach offers industrial solutions for developing safety-critical and embedded systems [41]. It supports the entire development and testing from the requirements analysis to the final product. As the proposed approach is used for the accurate construction of automatic code generation and formal verification, the modeling solution increases the productivity of team members and decreases the cost of the project. There is no gap between the model used by the design verifier for its analysis and the model that is executed at simulation time or on target.

Generally, if the errors in software are detected at the later stages, the cost of fixing the errors is much higher as compared to the situations when the errors are detected at the earlier stage, for example, requirement analysis. The model-based approach focus on finding those steps that are eliminated or minimized during the iterations of the V model, shown in Figure 3. It then eliminates the iterations of V process model that minimizes the cost and schedule of the project without compromising on its quality. In Figure 3, the V process model on the left side after requirements analysis represents the phases of requirements validation while on the right side avoids the repetition of requirements validation repeatedly. The proposed process supports the changes at each stage and minimizes the iteration of testing after each phase of the projects that is a big gap in the V model.

Invalidated requirements can cost human lives and finances as well. These modeling techniques are used to overcome the problems related to the ambiguity of requirements.

C. REQUIREMENTS MODELING USING V & V PROCESSES

Modeling is the key process for requirements validation in the auto industry [18]. The proposed methodology validates the shape of a car after the design phase through the modeling approach and detects the errors at an earlier stage. A benefit of employing modeling approach in the proposed scenario is
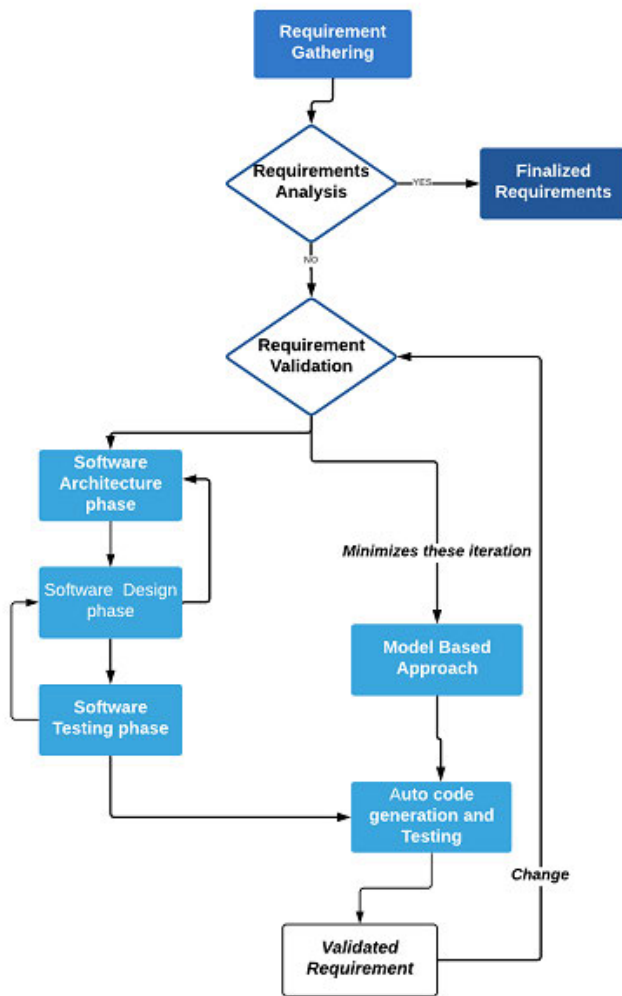
FIGURE 3. SCADE Life-Cycle with minimization of V & V model steps.

that it not only generates effective designs or shapes of the racing cars through virtual prototyping but also increases the productivity and decreases the cost and time of the project, as the major part of the code is auto-generated. Initially, the requirements are elicited by the users and a model specification is generated based on the requirements. Subsequently, the test oracles are prepared for the sake of system analysis and the system is visualized and the code is auto-generated for the system by using a Model-based approach. A benefit of using the Model-Based Approach is that different activities are performed in a single step and the clients easily see the virtual prototype at the initial phase of the projects and change it according to their needs.

The Model-based approach also minimizes the simulation time of embedded products by factors of 16 that reduces the time from 72 hours to 4.5 hours [42]. The model shows that after gathering the requirements from the client we can validate the requirements through modeling at the initial level. The whole process is described in Figure 4.

1) MODELING TOOL

Several tools are being used for designing and validating the requirements. For requirement validation, the proposed approach employs the Safety Critical Application Development Environment (SCADE) tool by the Esterel technology. Different steps are used to validate the requirements: First of all the requirement analyst gathers the requirements from the customer and then creates the geometry out of those requirements. After creating geometry parameters are assigned for the shape of the car, such as velocity, pressure, volume, etc. Subsequently, the parameter is finalized according to the customer needs and simulation is started against the shape. Finally, the results are matched with the requirements and the dynamic behavior of the car is found. The above-stated steps are illustrated in Figure 5.

2) VERIFICATION AND VALIDATION OF A SIMULATION TOOL

The requirement verification and validation process through simulation tool using the SCADE are shown in Figure 6.

As discussed previously, the requirements of the customer are written in natural language. The following methodology validates the requirements by modeling the requirements written in natural language. After finalizing the requirements provided by various stakeholders, a model is created based on the finalized requirements. After modeling, the geometry of that car model based on customer needs is created that gives information about the main functionality of the car. The shape and main feature of the car can also be modified at this stage. After fulfilling the customer needs by geometry, we move to the next step where we perform the simulations. If the geometry is not according to the customer needs then we need to go for a change and redesign the geometry. The proposed approach that uses a modeling tool reduces the cost of coding and testing of an embedded system by auto testing and code generation. It optimizes the overall performance of the system and enables a virtual prototype of the requirements. The tool is based on a formal method that reduces cost and improves accuracy. The model-based approach shows how to reduce the cost of the project and improve the accuracy of the software.

3) VERIFICATION AND VALIDATION LIFE-CYCLE OF THE MODEL-BASED APPROACH

The model-based approach is used for requirement verification and validation of embedded systems. Figure 7 shows the process of requirements validation life cycle through modeling. The requirement validation life cycle model gives information about the different activities of requirement validation through a model-based approach for embedded systems.

The needs of a user are considered, and analysis is performed over those requirements. After analysis, a structure or shape is designed to make it more understandable for the user. After designing we create a model of a car against the user requirements.
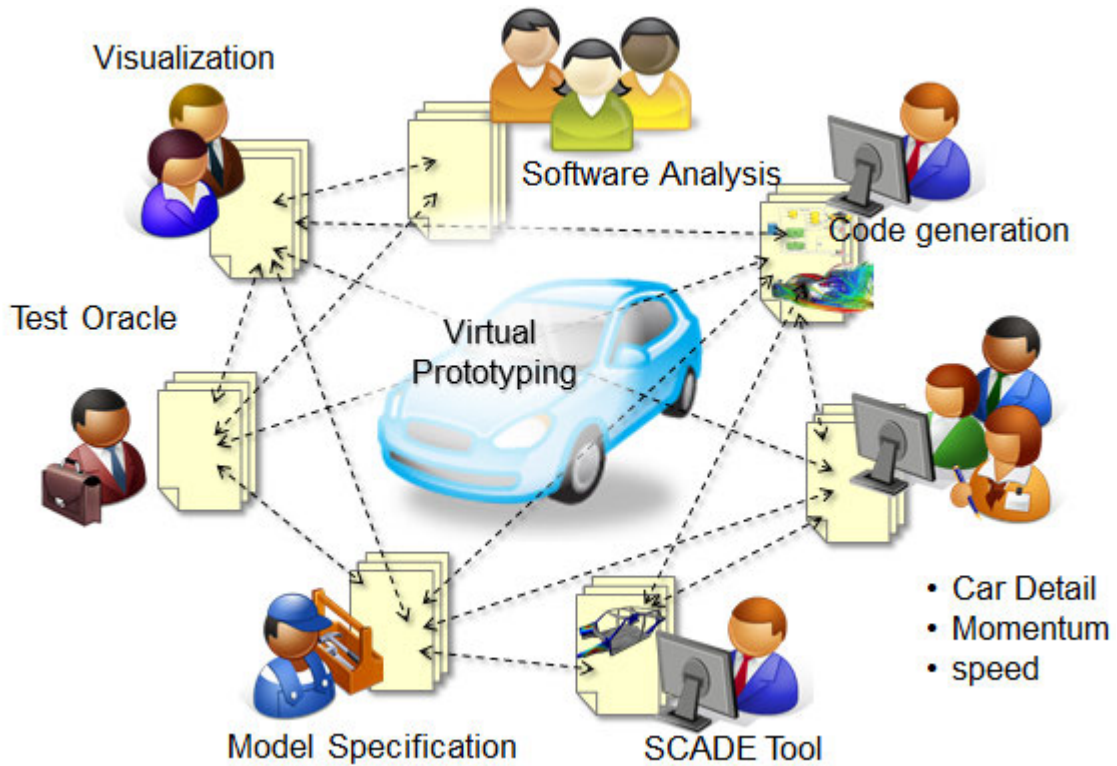
FIGURE 4. Requirement validation and verification process using a model-based approach.
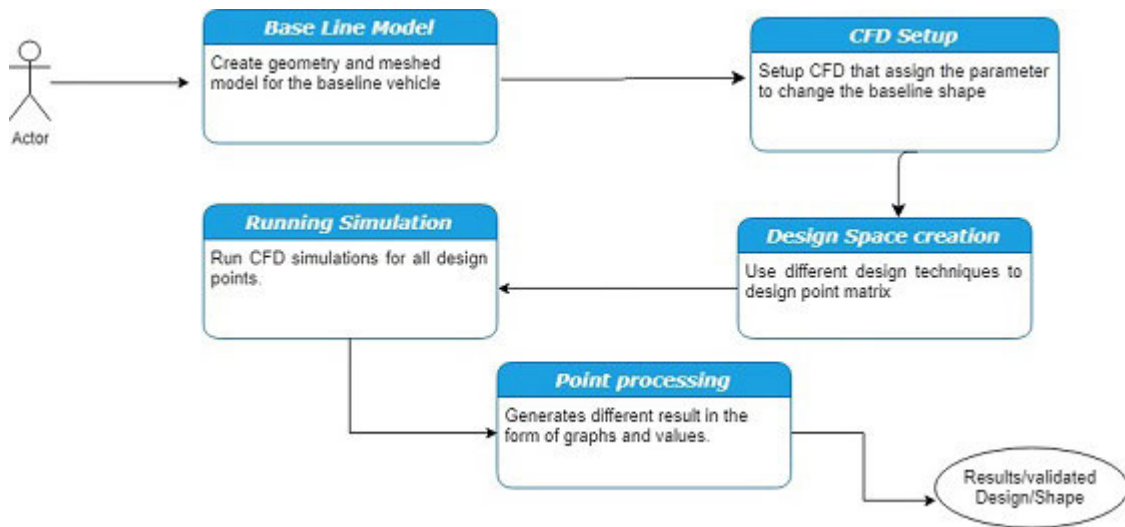

FIGURE 5. Basic steps of SCADE model-based validation tool.

For example, suppose the requirement of the user is fuel efficiency on a specific car model. We used the requirement validation life cycle model to validate the user requirement. Also, by using this model we have minimized the drag force on a specific car model to validate the customer needs that make a model more fuel-efficient. We calculate the drag force of a vehicle to increase the fuel efficiency based on geometry, which shows us the 3D version of the car and drags force is

calculated in Eq. 1 [43]. The basic formula for the drag force is given below.

$$D = c_d \times A_f \times q \qquad (1)$$

where q $= 1/2\rho V^2$

In the above equation D denotes the drag force, cd indicates the coefficient of drag force on the body of a car, Af denotes the Area of a car size of that model and q indicates the
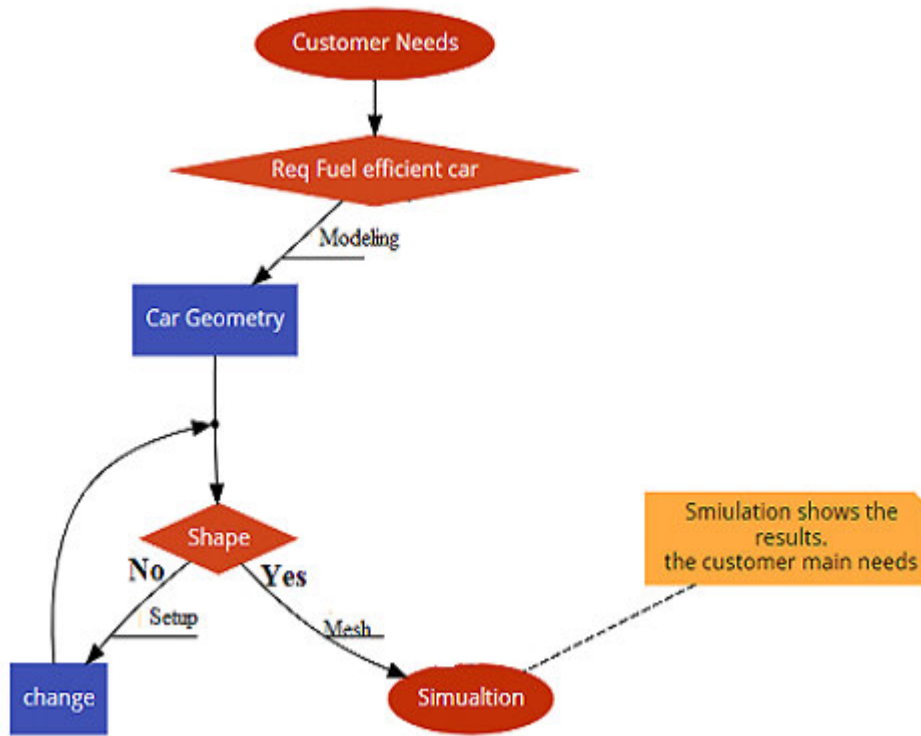
FIGURE 6. Verification and validation of simulation tool.
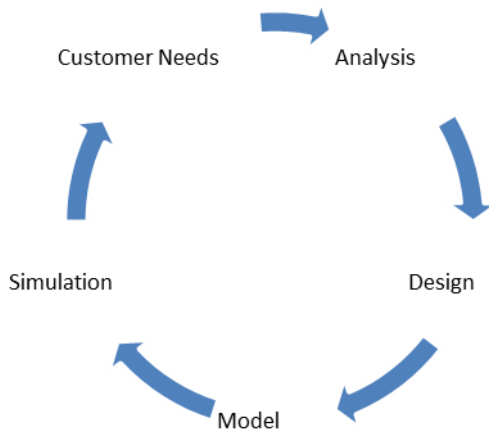
FIGURE 7. Verification and validation lifecycle of the proposed model-based approach.

pressure against that model of vehicle. The values of q can be put in Eq. 1, to identify the velocity, pressure, kinetic energy, and dragging force of the car [44] to use the formula in Eq. 2.

$$D = \frac{c_d \times \rho \times v^2 \times A}{2} \quad (2)$$

In the above equation D is the drag force which shows the resistance of air or water [44] against the body. These motion and flow are called fluid dynamics. Through Eq.3 we have

found out the right side dragging force against that body. The force that resists against the body is called the drag force and is always in the opposite direction of the body [45]. Its velocity is expressed in Newton and is denoted by (N).

$$Where C_D = \frac{2F}{\rho A V^2} \quad (3)$$

Equation 4 for the coefficient of lift-drag force [45] is given below.

$$Where C_L = \frac{2F}{\rho A V^2} \quad (4)$$

where F is the force ρ is the density of air kg/m3 and v is the velocity which is m/s.

In Figure 8 the above equation is applied which show that how dragging force effect on a car body.

V. RESULTS
We validate the proposed model using two different case studies. The first case study is related to minimizing the fuel consumption of the Ford Motor car 2012 model. The company focuses on customer satisfaction and the improvement of the required quality. In the first case study, we applied the proposed model to validate the requirements in a real-time scenario where a customer requires fuel efficiency on a specific model. For this purpose, we calculated the drag force based on which, the speed, velocity, and pressure are calculated on a virtual prototype of that specific car model.
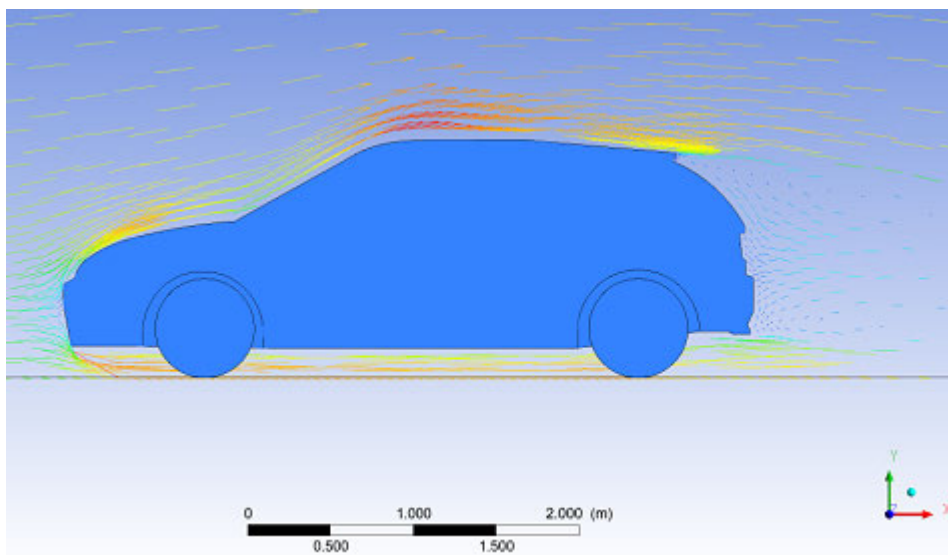
FIGURE 8. Air resistance showed on the 3D model of car.

The second case study is related to the Cruise Control System with distance warning to validate the proposed model using two different techniques. The first technique used is requirement validation through Petri nets and the second technique used is the validation through modeling that is applied to the car model of the BMW 2018 model.

The proposed model validated the requirements through the dynamic behavior of a car. Our proposed approach uses virtual prototyping in the auto industry rather than traditional techniques, as it validates the requirements at the design level based on vehicle shape and verify its features at the initial level. Due to automatic testing and auto code generation, the requirement engineers do not need any sort of testing that eventually increases productivity, minimizes the time, and saves the cost. We have considered the real needs of the customer in Table 2 that they want a specific functionality in a car that the company provides for its customers. It also minimizes the rework in the project at later stages. The customer requirement validation in the first case study on Ford Motor Company is presented below:

A. CUSTOMER SPECIFIC REQUIREMENT VALIDATION (A CASE STUDY OF FORD MOTOR COMPANY)

The customer-specific requirement is documented along with the model of the car. The customer selected the 2012 Model of Ford Motor Company. The reason to select this model is that the geometric dimensions and 3D design for this model are freely available online. The car specifications for the selected model is shown in Table 3 and the car model [46] is shown in Figure 9.

The body of the car, besides the precise windscreen angle, wasn't compromised. We assumed that the side mirror was neglected to make the design simple.

FIGURE 9. Ford 2012 model.

1) GEOMETRY
After finalizing and modeling the requirements in solid works from the customer we have imported the model design to SCADE Tool that simulates the feature of that car at initial phase before the manufacturing of that model of car in the industry. We imported the model from Solid work to design modeler where we introduced the geometry around, which is shown in Figure 10. The specification of the geometry was defined as the inlet to be placed at a space of 5cm car length, 10cm car lengths outlets and 10cm height of car from top and 10cm width of car from all sides and the effect of windscreen angle was also checked on fuel consumption. All of the work was done to confirm that the boundaries of the area is not altering the final result and inlet is made sure to be at a equitable distance to make sure the flow is reaching at the front of geometry, which is streamlined already and is totally free from any kind of eddies or disturbances.

TABLE 2. Customer-Specific needs.

Req. ID	User Requirement Name	Description	Justification/OR Comments	User Need Ref.
UR-ID-001	Fuel efficient car	To minimize the fuel efficiency on the 2012 model of ford company	To calculate the drag force for fuel efficiency	UN-002

TABLE 3. Ford focus specifications.

Specification of Model(2012 Ford)		Customer-specific Requirements
Power	350bhp	214 bhp
Top Speed	155mph	144 mph
Torque	310Nm,228ft-lb	309 Nm, 226 ft-lb
Fuel consumption	32mph	28 mpg
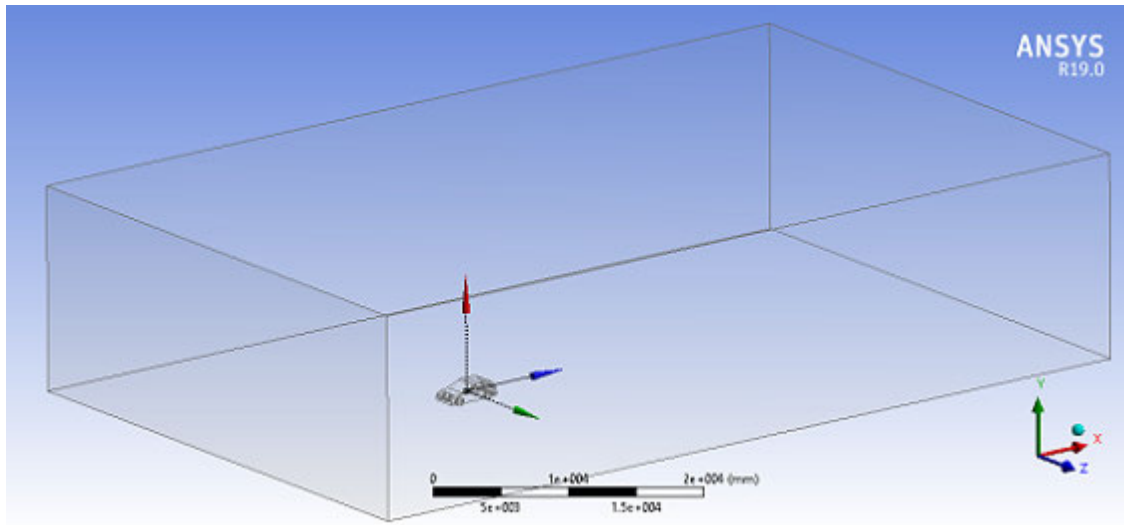
FIGURE 10. Geometry through SCADE tool.

The final boundary of this enclosure was quite big. So, it could take much time in setting up and generating result. So we used the XY plane to slice whole model into a symmetry that is shown in Figure 11. This won't change the results, but ANSYS will realize the manipulation and generation of accurate results with less time consumption taken for computing.

2) MESH GENERATION

For an effective mesh, the preferences of control were set to Computational fluid dynamic Fluent (CFD). The Sizing of mesh was assigned the function of ''Proximity Curvature'' to establish the elements in an effective way with contours. Windscreen and the front end of the car were a assigned a refinement function of level 2 shown in Figure 12, so the mesh density was increased around the important areas.

3) SETUP AND MODEL GENERATION

In the Fluent Module, we chose the steady-state and pressure based settings. We then chose the Viscous-Turbulent (k-e)

model with a realizable option. As we expected the diverse pressure so we chose a non-equilibrium turbulent flow model.

The velocity of a car which is 70mph after converting the unit, the velocity was reached to 31.3 m/s which makes the model more fuel-efficient. Turbulent behavior was shown by air at this amount of velocity so we had to the defined hydraulic diameter and turbulent intensity in equation 5 [47]. Turbulent intensity [48] in equation 7 was calculated by using Reynolds number [49] and the hydraulic diameter was found with the help of the area of inlet and inlet parameters defined in equation 6 [50].

$$HydraulicDia. = \frac{4 \times InletArea}{InletPerimeter} \quad (5)$$

$$ReynoldsNumber = R_e = \frac{pU_eD_h}{\mu} \quad (6)$$

$$TurbulentIntensity = I \approx 0.16(R_e)^{(-\frac{1}{8})} \quad (7)$$

4) AUTO CODE GENERATION

Automatic code generation helps to increase the effectiveness of complex software production by reducing the cost and

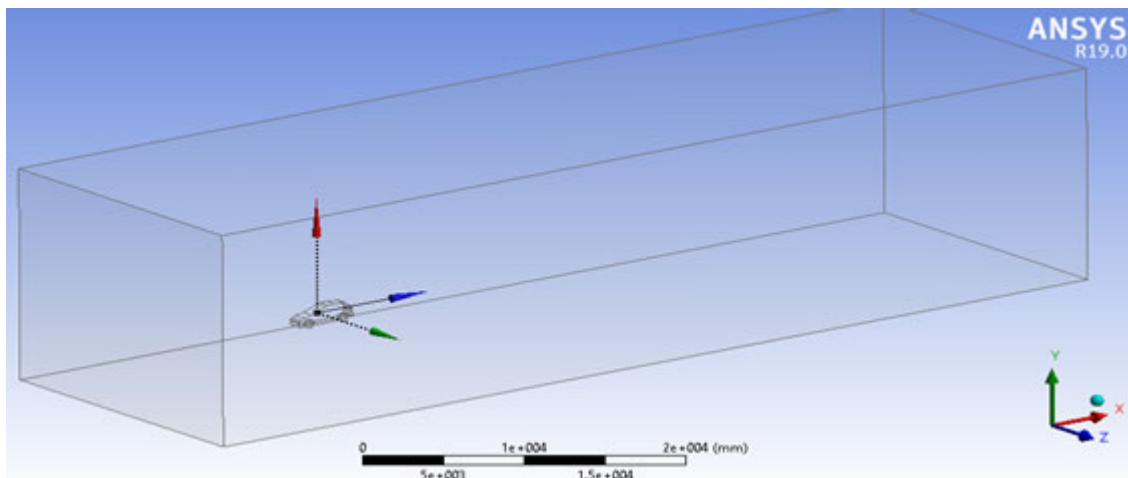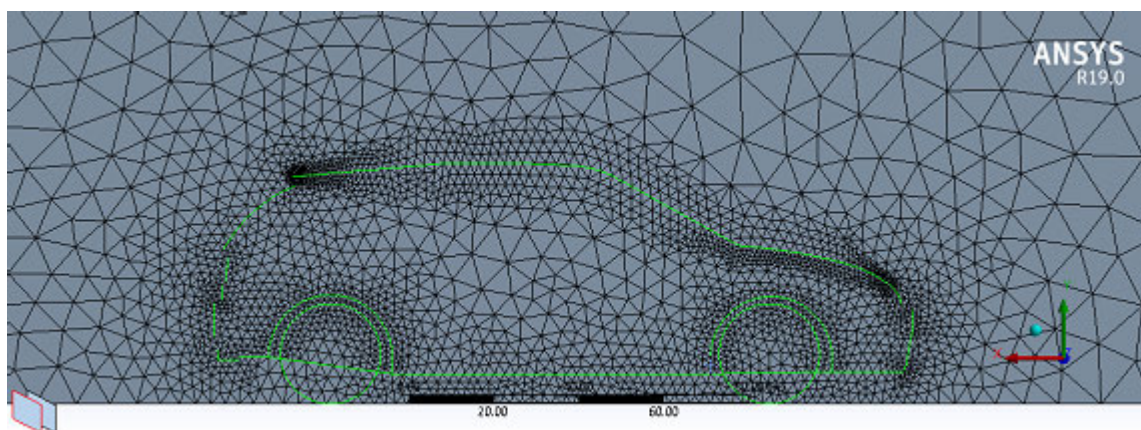**FIGURE 11.** 3D Symmetry through SCADE Tool.

FIGURE 12. Detailed mesh representation.

time associated with the coding effort. The generated code faithfully reflects the model from which it is produced. In an embedded system, SCADE generates a 60% auto code, which means around 50 % reduction in cycle time and 5X reduction in errors. The auto code generation sample from SCADE is illustrated in Figure 13.

5) BOUNDARY CONDITIONS

- We assigned an enclosure inlet; velocity inlet was assigned to air at 31.3m/s.
- We assigned Pressure outlet to the outlet.
- We designated XY-plane as the plane of symmetry.
- The Walls were assigned automatically as walls.
- We set the boundary condition at the start of simulation experiments. The boundary condition for the different parameters is depicted in Figure 14.
- Velocity vector visualization or virtual prototyping is shown in Figure 15 and the dynamic simulation, path visualization in Figure 16.

6) MASS AND MOMENTUM EFFECT ON DRAG FORCE

The simulation graph shows that the mass and shape of the car affect the dragging force. The greater the dragging force on

```
LIBRARY:
  MATERIAL: Air at 25 C
    Material Description = Air at 25 C and 1 atm (dry)
    Material Group = Air Data, Constant Property Gases
    Option = Pure Substance
    Thermodynamic State = Gas
  PROPERTIES:
    Option = General Material
    EQUATION OF STATE:
      Density = 1.185 [kg m^-3]
      Molar Mass = 28.96 [kg kmol^-1]
      Option = Value
    END
    SPECIFIC HEAT CAPACITY:
      Option = Value
      Specific Heat Capacity = 1.0044E+03 [J kg^-1 K^-1]
      Specific Heat Type = Constant Pressure
    END
    REFERENCE STATE:
      Option = Specified Point
      Reference Pressure = 1 [atm]
      Reference Specific Enthalpy = 0. [J/kg]
      Reference Specific Entropy = 0. [J/kg/K]
      Reference Temperature = 25 [C]
```

FIGURE 13. A sample of auto code generation.

a body of a car, the more is the power required for moving that car. Moreover, when more power is required, the fuel will be consumed more. Figure 17 shows that the momentum

TABLE 4. Validated requirements results (fuel consumption table).

Features	Velocity	Drag Force	Power	Fuel Consumption
Units	(m/s)	(N/m2)	(W/m2)	Mpg
Validated Requirements	31.3	259.70	8376.92	28

TABLE 5. Dragging Force Table.

Version	Dragging force at a Different Angles.
Iterations	90
Drag (N)	259.70057
Drag Coefficient	0.54595726
Lift (N)	-23.927045
Lift coefficient	-0.045207412

FIGURE 14. Variable range information.

which denotes by p of a body is equal to the mass of the body and velocity in which the equation is p = mv. Therefore, the momentum is directly related to the velocity of the body and the greater the mass and velocity, the greater will be the momentum. When greater the is the momentum the less will be the velocity of that car. In the above-mentioned case study, we minimized the momentum to increase the velocity of the car and minimize the drag force.

7) CALCULATE DRAG FORCE

From the coefficient of the equation, we have calculated the drag force. Eq. 8 shows that if greater is the surface area of the body then more force is required for moving that object [51]. In the coefficient equation, U is the flow velocity and p is the mass density regarding that object.

$$Coefficient = \frac{2 * Force}{U^2 A} \tag{8}$$

Table 4 shows information about validated requirements regarding fuel consumption. We calculated the drag force in Table 5 for fuel consumption. We calculated drag force on each angle and subsequently found the required power of that model of car. We also calculated the velocity required for that power and afterward, validated the requirements using a model-based approach based on virtual prototyping. Through the model-based approach, we calculated the drag force and fuel consumption ratio based on the shape of a car at the initial level. As we made validation on initial stages based on virtual prototyping, it resulted in fewer resources required, cost, and time consumption.

B. CUSTOMER SPECIFIC REQUIREMENT VALIDATION (A CASE STUDY OF BMW 3 SERIES 2018 MODEL)

Table 6 represents the real needs of the customers that the company provides for its particular customer. Now we validate the customer requirements using a case study of BMW Model 2018.

The specification of the case study is presented through an example [52]. The pictorial representation is provided in Figure 18.

We consider the example of the cruise control system, having the following functionality:

At the initial stage, the system would remain in the OFF state. As soon as the car has reached the speed of 60km/h, the cruise controller automatically turns to ON state. Now, the velocity of the car is stored in a (+) button. The (+) button stores velocity and controls the car at a constant speed.
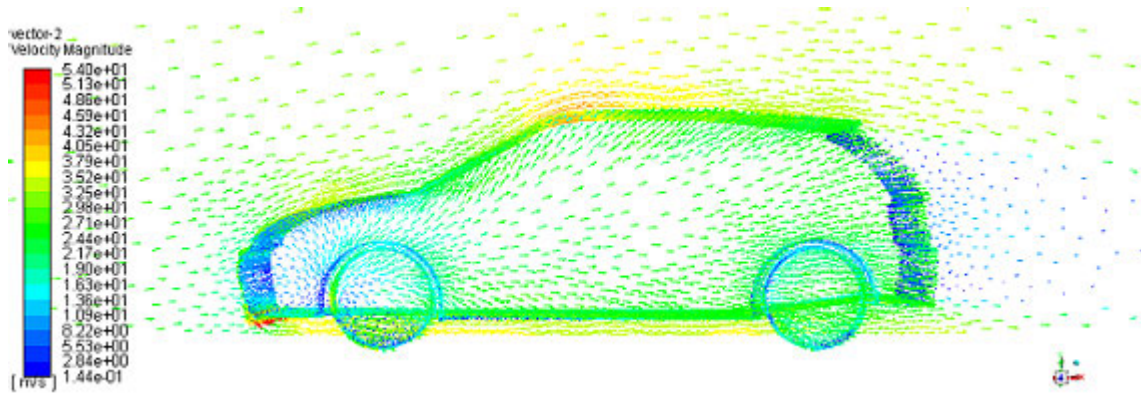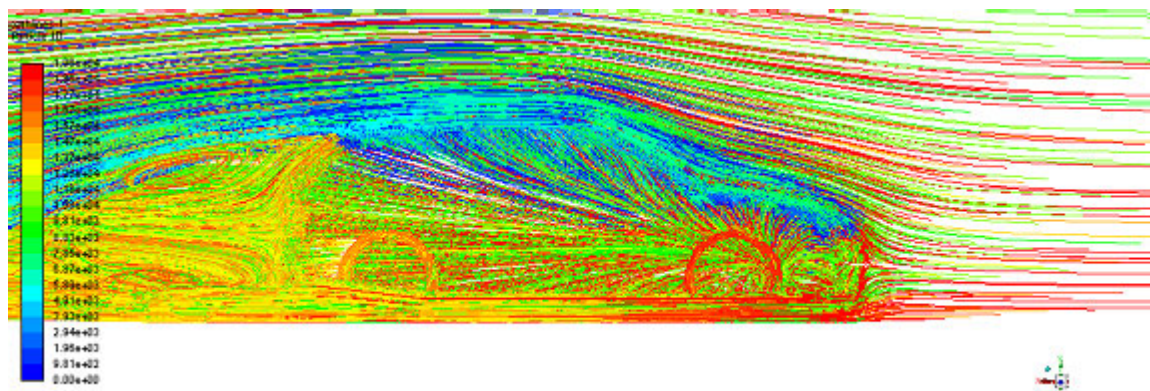
FIGURE 15. Velocity vector visualization.

FIGURE 16. Particle path visualization.

TABLE 6. Customer specification for cruise control system.

Req. ID	User Requirement Name	Description	Justification/OR Comments	User Need Ref.
UR-ID-002	Cruise controller with distance warning	To minimize the fuel efficiency using the cruise controller with a with speed limit reach to 60 km/h the cruise control system automatically	To calculate the increment and decrement velocity with a ratio 4km/h	UN-002

Using the (+) button again increases the velocity with 4km/h. The button (−) is used for the decrement of velocity with a 4km/h. If the driver uses the brake the velocity goes to a suspended state. After removing braking, the car goes again to a moving state. In a suspended state, it matches the actual velocity with a stored velocity. If the car matches the actual velocity with stored velocity, then again cruise controller goes to ON state. The cruise control system turns off by pushing the button to OFF.

The case study that is explained through an example is based on customers' needs. Now we validate the customer needs through the proposed model for requirement validation. For the evaluation of a proposed model, we compare the proposed modeling-based techniques with the formal technique using Petri nets.

1) THE CASE STUDY EVALUATION THROUGH PETRI NETS

Let us reconsider the example of the cruise control system with the customer specification in Table 6. The cruise control system with distance warning is considered initially off. The p1 state we have considered the OFF state of the cruise control system. When the speed of the car reaches 60km/h, then the cruise control system goes to the ON state. Therefore, different transition states and their actions are defined as P1 (OFF state), T1 (Turn ON), p2 (Ready state), T2 (store speed), T3 (Store velocity), p5 (cruise controller system), and T8 (Brake). Consequently, the transition connectivity becomes as p1–>T1–>p2–>T3, respectively. The remaining states and their transitions are shown in Figure 19.
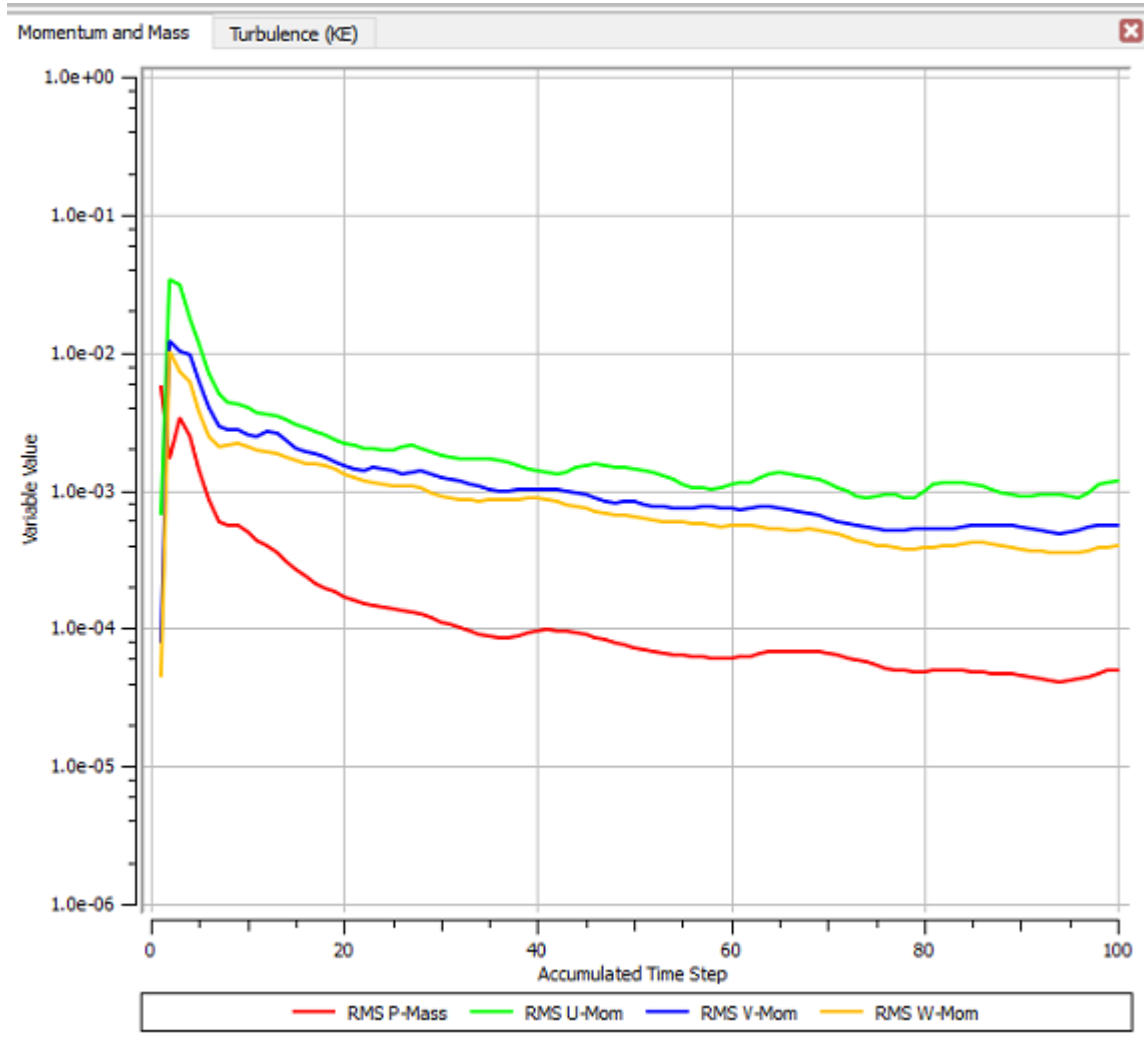
FIGURE 17. Mass and momentum graph.

FIGURE 18. Interface of Cruise controller system of BMW [53].

a: SPECIFICATION OF PETRI NETS SIMULATION THROUGH CPN TOOL

Through CPN tool the Petri nets are simulated with different runs as:

- Run1: (The cruise control system is an OFF state)
- Run2: (we assume that the speed of the car reached to 60km/h the cruise control system is turned ON state)
- Run3: (In turned ON state cruise controller store the current velocity.)
- Run4: (increment velocity by 4km/h and decrement velocity by 4km/h through cruise controller)
- Run5: (cruise controller goes to OFF state by clicking brake.)
- Run6: (clicking on the brake the velocity goes to suspended state)

These are the different runs through Petri nets using the modeling simulation tool CPN, which shows the dynamic behavior of the model.

b: REACH-ABILITY GRAPH ANALYSIS

For the net simulation validation, we used mathematical analysis. Net simulation can be validated through three different techniques namely, (i) linear invariant analysis,
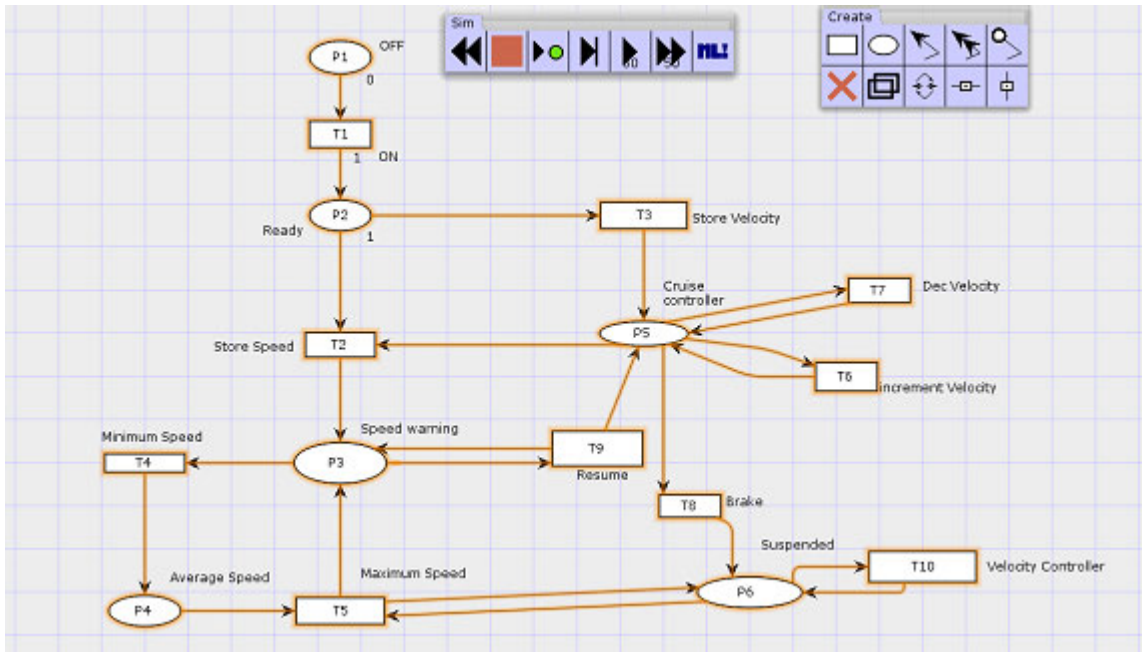
FIGURE 19. Petri nets for the cruise control system using CPN tool.

(ii) analysis with reach-ability graph, and (iii) model checking. Therefore, we confirmed our simulation results with reach-ability graph analysis. The reach-ability analysis graph is shown in Figure 20, and markings are written as a row vector.

Hence, the net is deadlock-free, and the reachability graph has at least one outgoing edge.

2) THE CASE STUDY EVALUATION THROUGH MODELING

In this section, we evaluate the case study through the modeling process. Using Scade2Nu to validate the safety-critical requirements of the cruise control system with distance warning. Scade2Nu is a simulation tool that is created by Esterel technologies [54]. Using the simulation tool for modeling shows the dynamic behavior of requirements at the initial stage. The cruise control system maintains the speed of a car at a constant velocity. Figure 21 shows the machine state of the cruise control system that is designed through the circuit lab.

Based on these machine states, we set the boundaries and written the codes in Scade2Nu. After coding completion, we have translated the code of the cruise control system. After the translated code, it generates some auto code with a file extension.NuSMV. With the running of. NuSMV file, we have found out the dynamic aspects of the architecture. Through Scade2Nu, we have verified the different sets of safety-critical requirements of the cruise control system, which are listed below.

- Requirement 1: The cruise control system is initially at the OFF state. It goes automatically to ON state when the speed reaches to 60km/h.

- Requirement 2: When the driver clicks on the button (+), its speed increases with a speed of 4km/h. On clicking the button (−), its speed decreases by 4km/h.
- Requirement 3: When the driver pushes the brake, the cruise control system goes automatically to the OFF state.
- Requirement 4: All the states of the system verify the requirements of customers.

a: BOUNDARY CONDITIONS
We have set the boundary condition in coding and saved it with the .text file in a simulation tool, which is shown in Figure 22.

b: Scade2Nu CODE FOR CRUISE CONTROL SYSTEM
The code written Scade2Nu tool for the cruise control system to fulfill different cases of the requirements and their screen short are given in Figure 23.

c: AUTO CODE GENERATION IN Scade2Nu
The translation of Scade file code generated automatically.NuSMV file. This is the auto-generated code of the Scade. It created the model of the targeted requirements. The auto code is shown in Figure 24.

d: EVALUATION OF THE REQUIREMENTS OF THE CRUISE CONTROL SYSTEM
Scade2Nu shows traces in a more friendly manner than the other simple formal techniques, Petri nets, or other techniques. Scade2Nu is a combination of mathematical modeling, formal techniques, and simulation modeling. Therefore, it can debug the SCADE model by analyzing the
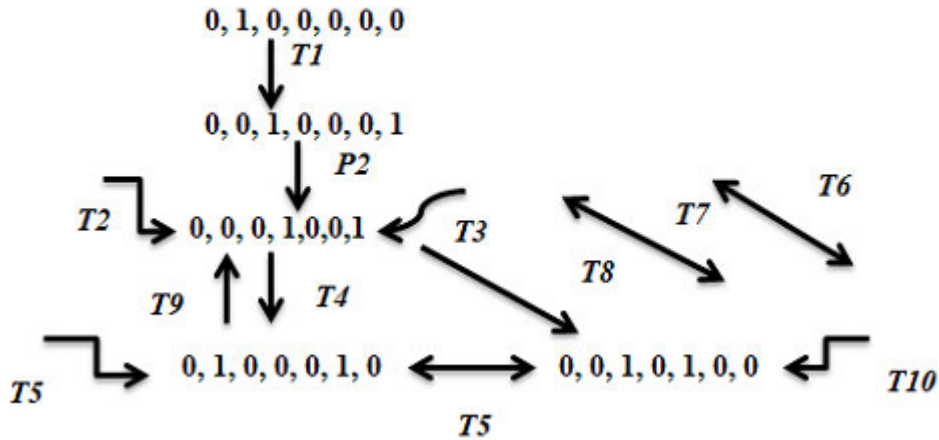
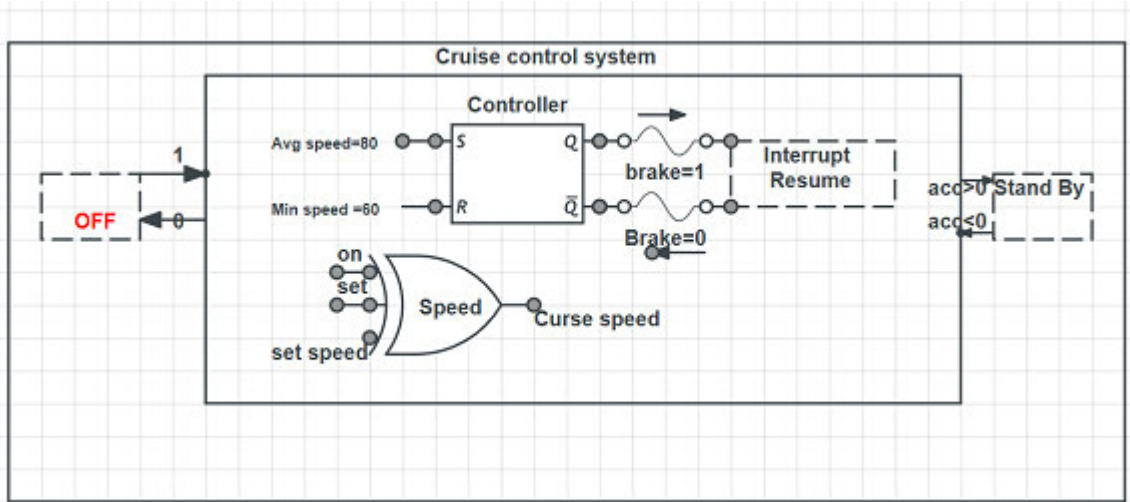**FIGURE 20.** Reachability graph of the cruise control system.

FIGURE 21. Cruise control system architecture for customer requirement validation through modeling using a simulation Scade2Nu.

detail results. In our case, NuSMV reports that requirement 1, requirement 2, and requirement 3 are valid and hence gives true results. Requirement 4 is false sometimes, because, in some cases, it is not verifying all the condition. When the driver pushes the brake, it sometimes goes into interrupt state and sometimes to standby. Their different traces are shown in Figure 25.

C. HOW TO MINIMIZE THE EFFORT, COST, TIME AND INCREASE THE PRODUCTIVITY

As stated earlier, the proposed technique requirement validation through modeling increases the productivity and minimize the effort, cost and time of the project. To prove our claim, we are considered the same case study or the same set of requirements to validate with two different techniques. The first scenario is based on requirement validation through Petri nets and the second scenario is the requirement validation through modeling. We calculated the effort time and cost

using COCOMO II Post Architectural Model and find out which method has minimized the time, effort and cost.

1) SCENARIO 1 USING A FORMAL TECHNIQUE

We developed the case study of the cruise control system or the same set of requirements that are explained in Table 6. For the case study, we developed the net architecture through the CPN tool and generated the token among the different transition graphs and their results are proved through reach-ability graph analysis. After the evaluation of the net based on experimental judgment, we assumed that the size of the code for that net will be 4500 lines of code. Based on Size, we calculated the effort. The effort is calculated based on the formula of COCOMO II in Equations 9, 10, 11 [55]. We performed calculation directly and put their answer in boxes which are given below:

$$Effort = Asize^B M \qquad (9)$$
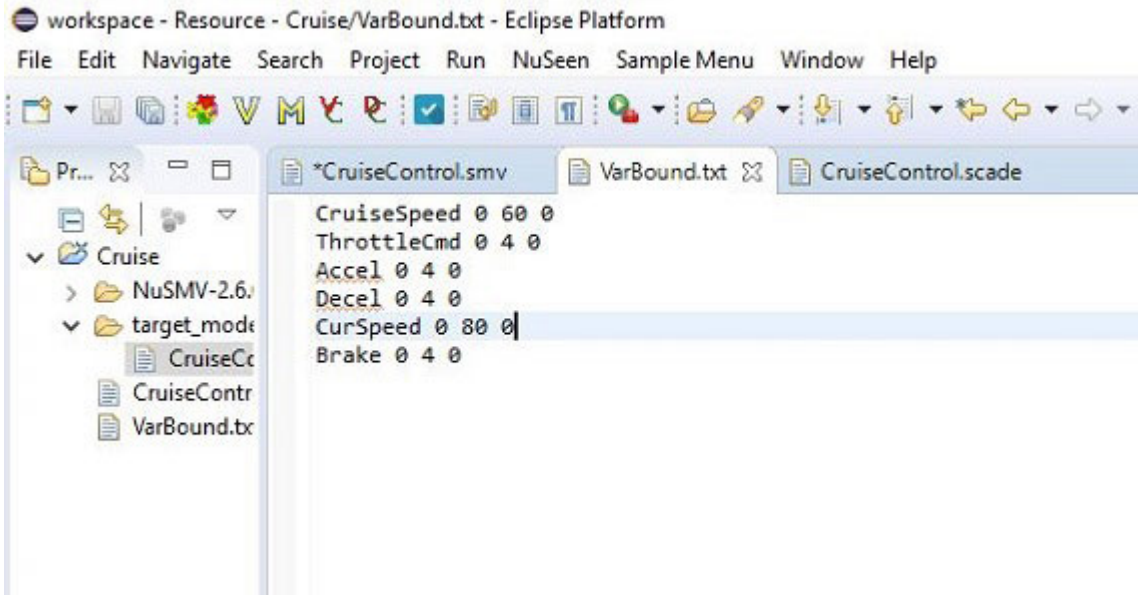$$Time = CEffort^f \qquad (10)$$

FIGURE 22. Boundary condition according to requirements using Scade2Nu tool.
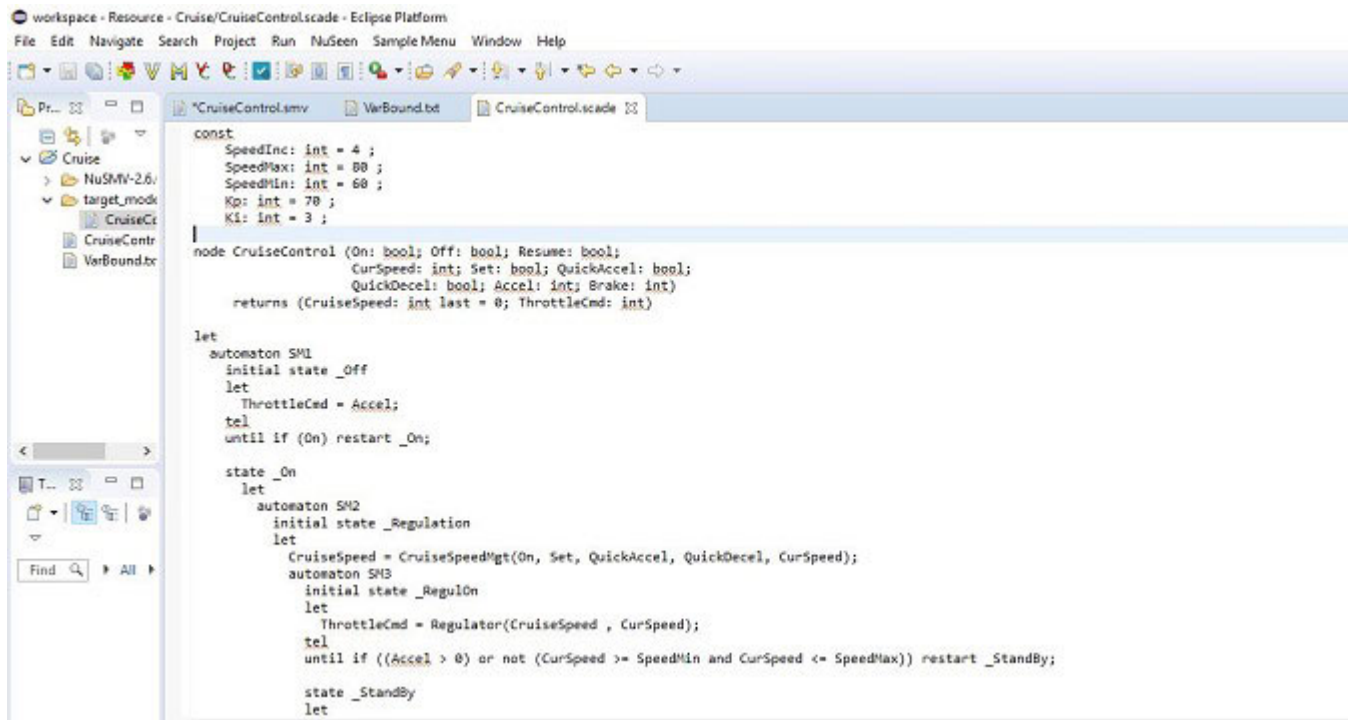

FIGURE 23. Cruise control system codes using modeling and simulation tool Scade2Nu.

$$Staff = \frac{Effort}{Time} \quad (11)$$

So using the above formulas, we have calculated the Effort = 17.72PM, Time = 6.2Weeks, Staff = 2.8Person, and cost = 9000$ for scenario 1.

2) SCENARIO 2 USING MODELING
Reconsidering the first scenario to calculate the effort, time and cost. we are using the same formula but changed the developing environment from formal technique to modeling. We assumed the line of code is the same for both scenarios is 4500LOC whereas the SCADE tool generates 60% auto code for the project. So the code which is written to the programmer for the same project and the same case study is 2700LOC. Now putting the value in the formula, so the results are: Effort = 9.6PM, Time = 5.1Weeks, Staff = 1.5person and Cost = 5500$.
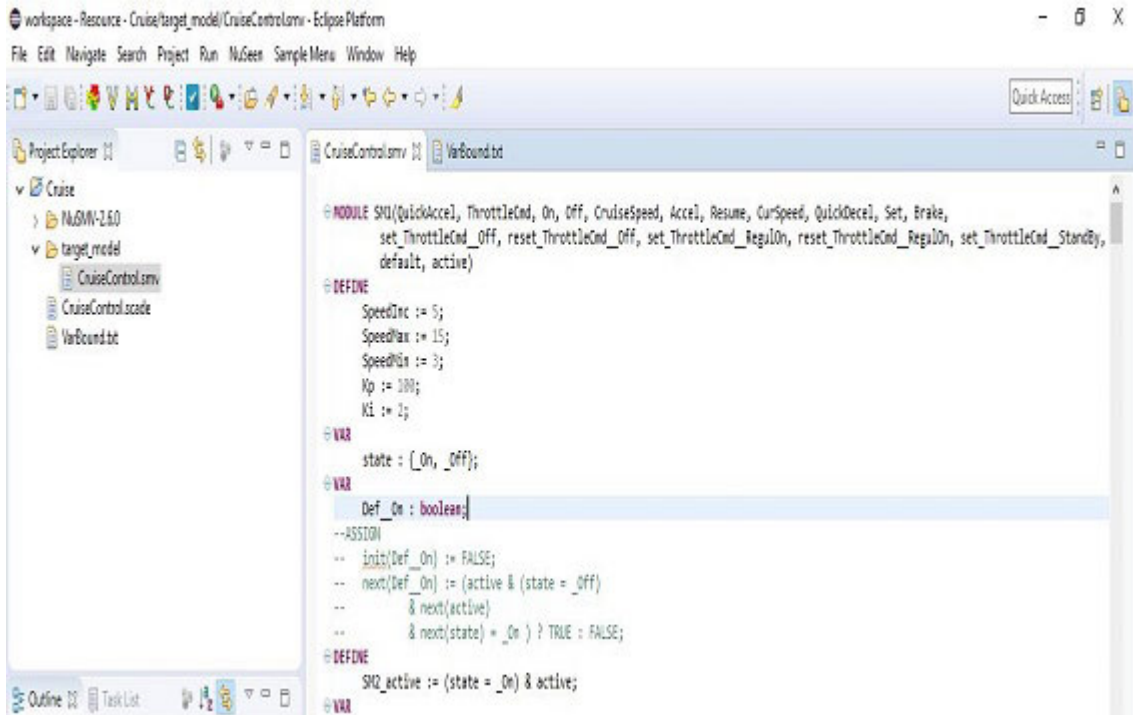
FIGURE 24. Auto code generation for the cruise control system using Scade2NU.
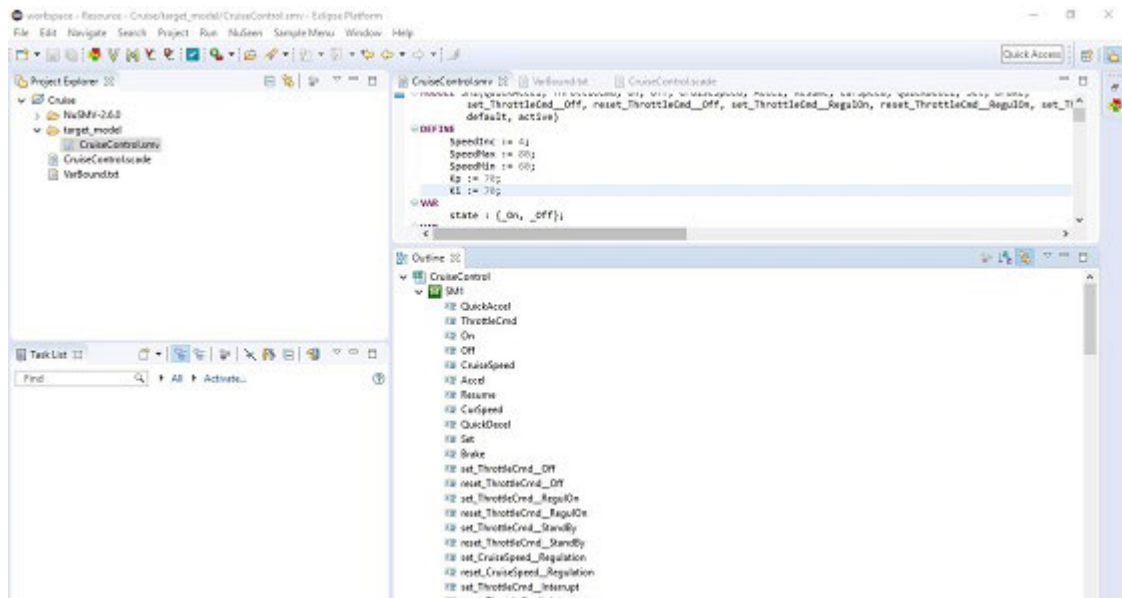
FIGURE 25. Simulation results of a cruise control system with customer specification through modeling.

D. COMPARISON ANALYSIS OF REQUIREMENTS VALIDATION THROUGH PETRI NETS AND MODELING

There are different methods in the industry which are used for requirement validation. In our case, we have validated the case study through two techniques in which one is a formal technique using Petri nets and the other is simulation through modeling. Comparing these two methods of requirement validation namely validation through Petri nets and validation through modeling can be noted in Table 7.

E. DISCUSSION AND OUTCOME OF THE ANALYSIS

Through modeling, and intensive simulation of different case studies with the different techniques, we have found that

TABLE 7. Comparative analysis of requirement validation techniques.

Factors	Validation Through Formal technique using Petri nets	Validation Through Modeling
Correctness	High	High
Understanding by the customer	Low	High
Accuracy	High	High
productivity	Low	High
Effort	High	Low
Time consumption	High	Low
Cost	High	Low
Testing	Manual Testing	Auto Testing
Coding	Manual coding	60% Auto code
Expert	Required	Not required
Design validity	Shows the system behavior	Virtual prototyping

the cost and time of the project are reduced, and it is also proved through COCOMO II. Due to the prototype, the company and client both understand the final product at an initial level [56]. Through modeling, we can understand the behavior of the project and the specification of the final product.

1) REDUCTION IN PRODUCTION COST AND TIME
We estimated the production cost at the initial level by developing a virtual prototype. It helps the requirement analyst to improve the feasibility of the project by determining the cost at the initial level. It reduces the cost as well as speeds up the process by using the auto code generation and auto testing features. Modeling is a way of effective designing and functionality of the system that can also find the correctness of the system.

2) REDUCTION IN MODELING COST
We have found out that modeling has a high initial cost because in the initial stage it requires a lot of effort to simulate the problem. But in several ways, the initial cost of the project can be minimized. To minimize the initial cost of the model, we have used the SCADE tool which simulates real-time requirements.

3) CUSTOMER SATISFACTION
The comparison analysis of validation through two different techniques proved that customer easily understands the end product using the modeling rather than validation through formal technique using Petri nets. The proposed approach validates the requirement at an early stage, and the required changes can easily be handled through modeling at the initial level. The customers are more satisfied through modeling because they have validated their requirements through the virtual prototype.

4) DESIGN VALIDITY
The proposed approach validated the customer specific needs and verified the design through modeling according to case study. It is always easier for the user or client to get the picture or shape and functionality of the product through the modeling at the early stage of the project.

5) IMPROVEMENT IN REQUIREMENT ACCURACY
At the end stage of the project, there are fewer chances to evolve the requirements. But modeling increased the requirement accuracy because it validates the requirement at early stages.

6) MINIMIZE THE RISK
The risks are minimized through modeling technique because the customer validates the requirement at an early stage by their own choice and fixes them.

VI. CONCLUSION AND FUTURE RESEARCH WORK
The requirement validation through modeling with the V processes in the auto industry gives us a significant result. However, every industry wants to improve their processes and quality of the product in the auto industry. The proposed approach also improves the productivity of a developer, which resultantly, gives us a better-quality product. The proposed model minimized the cost and effort of requirement validation through modeling using auto code generation and simulation through modeling. The errors which come due to requirements validation in final product using model based approach for validation were minimized with the help of the virtual prototyping and auto testing. The virtual prototyping in auto industries also has minimized the risk for manufacturing industries, as all of the specifications are finalized in a virtual prototype. The modeling approach improves the quality and design of the car. This approach also evaluated the customer-specific requirements applied to the Ford Company Motor model and BMW model 2018 for the cruise

control system. The methodology evaluated the real problem of the auto industry to make a fuel-efficient car. The dragging force of the Ford company motor 2012 model is calculated to minimize the consumption of fuel. The dragging force on a shape or model is also minimized to improve the speed of auto to make a car more efficient those benefits for the auto industry in many ways. The second case study is related to the cruise control system which validated through two different techniques, one is validation through a formal technique using Petri nets and the second is validation through modeling. At last, we compared them and found some analysis.

This paper validated the requirements of embedded system in the auto industry through modeling and formal technique using Petri nets. However, we plan to enhance the current model to increase the safety of a vehicle based on the V process certified from ISO 912626. We also want to validate the other requirements related to the anti-lock braking system, traction control, emission control, airbags, and drive by wire, etc. are also very important and we plan to validate our model for these requirements in future. This will also improve the quality of vehicles and customer satisfaction.

REFERENCES

[1] M. Morandini, L. Penserini, A. Perini, and A. Marchetto, "Engineering requirements for adaptive systems," *Requirements Eng*, vol. 22, no. 1, pp. 77–103, Mar. 2017.

[2] B. Nuseibeh and S. Easterbrook, "Requirements engineering: A roadmap," in *Proc. Conf. Future Softw. Eng. (ICSE)*, 2000, pp. 35–46.

[3] H. Leopold, J. Mendling, and A. Polyvyanyy, "Supporting process model validation through natural language generation," *IEEE Trans. Softw. Eng.*, vol. 40, no. 8, pp. 818–840, Aug. 2014.

[4] N. A. Moketar, M. Kamalrudin, S. Sidek, M. Robinson, and J. Grundy, "An automated collaborative requirements engineering tool for better validation of requirements," in *Proc. 31st IEEE/ACM Int. Conf. Automat. Softw. Eng. (ASE)*, Sep. 2016, pp. 864–869.

[5] K. Khalid, H. Afzal, F. Moqaddas, N. Iltaf, A. M. Sheri, and R. Nawaz, "Extension of semantic based urdu linguistic resources using natural language processing," in *Proc. IEEE 15th Int. Conf. Dependable, Autonomic Secure Comput., 15th Int. Conf. Pervas. Intell. Comput., 3rd Int. Conf. Big Data Intell. Comput. Cyber Sci. Technol. Congr. (DASC/PiCom/DataCom/CyberSciTech)*, Nov. 2017, pp. 1322–1325.

[6] M. Shardlow and R. Nawaz, "Neural text simplification of clinical letters with a domain specific phrase table," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, Jul. 2019.

[7] M. Shardlow, R. Batista-Navarro, P. Thompson, R. Nawaz, J. McNaught, and S. Ananiadou, "Identification of research hypotheses and new knowledge from scientific literature," *BMC Med. Inform. Decis. Making*, vol. 18, no. 1, pp. 1–13, 2018.

[8] P. Thompson, R. Nawaz, J. Mcnaught, and S. Ananiadou, "Enriching news events with meta-knowledge information," *Lang. Resour. Eval.*, vol. 51, no. 2, pp. 409–438, Jun. 2017.

[9] P. Rempel and P. Mader, "Preventing defects: The impact of requirements traceability completeness on software quality," *IEEE Trans. Softw. Eng.*, vol. 43, no. 8, pp. 777–797, Aug. 2017.

[10] S. Maalem and N. Zarour, "Challenge of validation in requirements engineering," *J. Innov. Digit. Ecosyst.*, vol. 3, no. 1, pp. 15–21, Jun. 2016.

[11] E. Stachtiari, A. Mavridou, P. Katsaros, S. Bliudze, and J. Sifakis, "Early validation of system requirements and design through correctness-by-construction," *J. Syst. Softw.*, vol. 145, pp. 52–78, Nov. 2018.

[12] H. Alfraihi, "Towards improving agility in model-driven development," in *Proc. CEUR Workshop*, vol. 1675, 2016, pp. 2–10.

[13] I. Inayat, S. S. Salim, S. Marczak, M. Daneva, and S. Shamshirband, "A systematic literature review on agile requirements engineering practices and challenges," *Comput. Hum. Behav.*, vol. 51, pp. 915–929, Oct. 2015.

[14] G. Philip, M. Dsouza, and V. P. Abidha, "Model based safety analysis: Automatic generation of safety validation test cases," in *Proc. IEEE/AIAA 36th Digit. Avionics Syst. Conf. (DASC)*, Sep. 2017.

[15] S. Kumanan and K. R. Raja, "Modeling and simulation of projects with Petri nets," *Amer. J. Appl. Sci.*, vol. 5, no. 12, pp. 1742–1749, 2008.

[16] Ö. Başak and Y. E. Albayrak, "Petri net based decision system modeling in real-time scheduling and control of flexible automotive manufacturing systems," *Comput. Ind. Eng.*, vol. 86, pp. 116–126, Aug. 2015.

[17] T. Kiran, S. Farhan, H. Tauseef, and M. A. Fahiem, "Requirement validation model for virtual distributed system," *Int. J. Multidiscip. Sci. Eng.*, vol. 4, no. 8, pp. 34–42, 2013.

[18] R. G. Cooper and A. F. Sommer, "The agile-stage-gate hybrid model: A promising new approach and a new research opportunity," *J. Product Innov. Manage.*, vol. 33, no. 5, pp. 513–526, Sep. 2016.

[19] A. Goknil, I. Kurtev, and K. Van Den Berg, "Generation and validation of traces between requirements and architecture based on formal trace semantics," *J. Syst. Softw.*, vol. 88, no. 1, pp. 112–137, Feb. 2014.

[20] E. S. Kim, D. A. Lee, S. Jung, J. Yoo, J. G. Choi, and J. S. Lee, "NuDE 2.0: A formal method-based software development, verification and safety analysis environment for digital I&Cs in NPPs," *J. Comput. Sci. Eng.*, vol. 11, no. 1, pp. 9–23, 2017.

[21] A. Miyazawa, P. Ribeiro, W. Li, A. Cavalcanti, J. Timmis, and J. Woodcock, "RoboChart: Modelling and verification of the functional behaviour of robotic applications," *Softw. Syst. Model.*, vol. 18, no. 5, pp. 3097–3149, Oct. 2019.

[22] R. Magargle, L. Johnson, P. Mandloi, P. Davoudabadi, O. Kesarkar, S. Krishnaswamy, J. Batteh, and A. Pitchaikani, "A simulation-based digital twin for model-driven health monitoring and predictive maintenance of an automotive braking system," in *Proc. 12th Int. Modelica Conf.*, Prague, Czech Republic, Jul. 2017, pp. 35–46.

[23] S. Zafar, N. Farooq-Khan, and M. Ahmed, "Requirements simulation for early validation using behavior trees and datalog," *Inf. Softw. Technol.*, vol. 61, pp. 52–70, May 2015.

[24] C. L. Heitmeyer, "Formal methods for specifying, validating, and verifying requirements," *J. Universal Comput. Sci.*, vol. 13, no. 5, pp. 601–618, 2007.

[25] Y. Wiseman, "Autonomous vehicles," in *Encyclopedia of Organizational Knowledge, Administration, and Technologies*, 1st ed., 2019

[26] *Recent Trends in Formal Validation and Verification of Autonomous Robots Software*, Standard HAL-01968265, 2019.

[27] P. Koopman and M. Wagner, "Toward a framework for highly automated vehicle safety validation," in *Proc. SAE Tech. Paper Ser.*, Apr. 2018, pp. 1–13.

[28] A. Fuxman, L. Liu, J. Mylopoulos, M. Pistore, M. Roveri, and P. Traverso, "Specifying and analyzing early requirements in Tropos," *Requirements Eng.*, vol. 9, no. 2, pp. 132–150, May 2004.

[29] P. Koopman and M. Wagner, "Challenges in autonomous vehicle testing and validation," *SAE Int. J. Trans. Saf.*, vol. 4, no. 1, pp. 15–24, Apr. 2016.

[30] T. Korssen, V. Dolk, J. van de Mortel-Fronczak, M. Reniers, and M. Heemels, "Systematic model-based design and implementation of supervisors for advanced driver assistance systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 2, pp. 533–544, Feb. 2018.

[31] A. K. Bharadwaj and V. Agrawal, "Improvement in requirement specifications using Petri nets," in *Proc. Int. Conf. Comput., Electr. Commun. Eng. (ICCECE)*, Dec. 2017, pp. 1–8.

[32] M. Choetkiertikul, H. K. Dam, T. Tran, A. Ghose, and J. Grundy, "Predicting delivery capability in iterative software development," *IEEE Trans. Softw. Eng.*, vol. 44, no. 6, pp. 551–573, Jun. 2018.

[33] N. M. Devadiga, "Tailoring architecture centric design method with rapid prototyping," in *Proc. 2nd Int. Conf. Communication Electron. Syst. (ICCES)*, Oct. 2017, pp. 924–930.

[34] V. Garousi and M. V. Mäntylä, "A systematic literature review of literature reviews in software testing," *Inf. Softw. Technol.*, vol. 80, pp. 195–216, Dec. 2016.

[35] M. Mori, A. Ceccarelli, P. Lollini, B. Frömel, F. Brancati, and A. Bondavalli, "Systems-of-systems modeling using a comprehensive viewpoint-based SysML profile," *J. Softw., Evol. Process*, vol. 30, no. 3, Mar. 2018, Art. no. e1878.

[36] W. C. Baldwin, B. Sauser, and R. Cloutier, "Simulation approaches for system of systems: Events-based versus agent based modeling," *Procedia Comput. Sci.*, vol. 44, pp. 363–372, Jan. 2015.

[37] A. Tosun, M. Ahmed, B. Turhan, and N. Juristo, "On the effectiveness of unit tests in test-driven development," in *Proc. Int. Conf. Softw. Syst. Process (ICSSP)*, 2018, pp. 113–122.

[38] A. B. Hocking, J. C. Knight, M. A. Aiello, and S. Shiraishi, "Formal verification in model based development," in *Proc. SAE Tech. Paper Ser.*, vol. 1, Apr. 2015.

[39] A. Hussain, E. O. C. Mkpojiogu, and F. M. Kamal, "The role of requirements in the success or failure of software projects," *Int. Rev. Manage. Marketing*, vol. 6, no. S7, pp. 11–13, 2016.

[40] *Standish Group CHAOS Report 2015*, Standish Group, West Yarmouth, MA, USA, 2015.

[41] M. J. Musa, S. Sudin, and Z. Mohamed, "Aerodynamic disturbance on vehicle's dynamic parameters," *J. Eng. Sci. Technol.*, vol. 13, no. 1, pp. 69–82, 2018.

[42] F. Hensler and C. Reeves, "Excellence in engineering simulation executive editor," *Ansys Advant.*, vol. 6, no. 3, pp. 8–9 and 21, 2012.

[43] L. Patidar and S. R. Bhamidipati, "Parametric study of drag force on a formula student electric race car using CFD," *Appl. Mech. Mater.*, vol. 575, pp. 300–305, Jun. 2014.

[44] S. Saad and M. F. Hamid, "Numerical study of aerodynamic drag force on student formula car," *ARPN J. Eng. Appl. Sci.*, vol. 11, no. 20, pp. 11902–11906, 2016.

[45] *The Measurement and Analysis of Road Vehicle Drag Forces*, Passmore MA, Herzogenaurach, Germany, 1990.

[46] Ford Motor Comapany. Accessed: Mar. 2019. [Online]. Available: https://www.ford.com/

[47] C. Bayındırlı, Y. E. Akansu, and M. S. Salman, "The Determination of aerodynamic drag coefficient of truck and trailer model by wind tunnel tests," *Int. J. Automot. Eng. Technol.*, vol. 5, no. 2, p. 53, Jul. 2016.

[48] V. Kshirsagar and J. V. Chopade, "Aerodynamics of high performance vehicles," in *Proc. Aerodyn. High Perform. Vehicles*, 2018, pp. 2182–2186.

[49] M. Ghiasi, E. Ahmadinia, M. Lariche, H. Zarrabi, and R. Simoes, "A new spinning reserve requirement prediction with hybrid model," *Smart Sci.*, vol. 6, no. 3, pp. 212–221, 2018.

[50] S. Hetawal, M. Gophane, B. Ajay, and Y. Mukkamala, "Aerodynamic study of formula SAE car," *Procedia Eng.*, vol. 97, pp. 1198–1207, Jan. 2014.

[51] S. Sarkar, K. Thummar, N. Shah, and V. Vagrecha, "A review paper on aerodynamic drag reduction and CFD analysis of vehicles," in *Proc. A Rev. Paper Aerodyn. Drag Reduction CFD Anal. Vehicles*, Jan. 2019, pp. 231–235.

[52] W. Reisig, "Petri nets in software engineering," in *Petri Nets: Applications and Relationships to Other Models of Concurrency* (Lecture Notes in Computer Science: Lecture Notes Artificial Intelligence Lecture Notes Bioinformatics), vol. 255. 1987, pp. 63–96.

[53] *BMW Website*. Accessed: Nov. 2019. [Online]. Available: https://www.bmw.com/en/automotive-life/autonomous-driving.html

[54] J. Shi, J. Shi, Y. Huang, J. Xiong, and Q. She, "Scade2Nu: A tool for verifying safety requirements of SCADE models with temporal specifications," in *Proc. CEUR Workshop*, vol. 2376, 2019.

[55] T. N. Sharma, "Analysis of software cost estimation using COCOMO II," *Int. J. Sci. Eng. Res.*, vol. 2, no. 6, pp. 1–5, 2011.

[56] E.-M. Schön, J. Thomaschewski, and M. J. Escalona, "Agile requirements engineering: A systematic literature review," *Comput. Standards Interfaces*, vol. 49, pp. 79–91, Jan. 2017.

• • •