Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 136 (2025) 564–569

www.elsevier.com/locate/procedia

35th CIRP Design 2025

Experimentable Digital Twin for Development of Electric Vehicles; A Case of Formula Society of Automotive Engineers

Guy Barker[a], Shiva Abdoli[a]*

[a]*School of Mechnical and Manufacturing Engineering, Univerity of New South Wales, Sydney, 2033, Australia*

* Corresponding author. Tel.: +61-209348 088 . *E-mail address:* a.abdoli@unsw.edu.au

Abstract

Developing electric vehicles (EV) pose challenges for engineering practices based on prior experience, so, it exists a need for a new approach that leverages the traditional Systems Engineering (SE), with current advancement in technologies such as simulation and Digital Twin as key technologies of industry 4.0. Existing Experimentable Digital Twin (EDT) approaches are domain specific, rely on specific software, and cannot easily be employed within SE. This research proposed an EDT development approach for integration of third-party simulation tools in EDTs of EVs. Application of this approach is demonstrated in development of a Formula-Society of Automotive Engineers EV.

Keywords: Keywords: Experimentable Digital Twin, Systems Engineering, Lifecycle Managemnet, Industry 4.0

1. Introduction

Systems engineering (SE) is a known approach to assist in the design of complex systems. The evolving terrain of current products, being more data-driven with more features, has forced adaptation of SE field along with advancements in others. Managing largescale data in modern products is a challenge and made designers from different disciplines to design them in silos [1]. This may lead to inconsistencies between different disciplines and failures. Modern products mainly achieve their overall functionality through dynamic interactions between their induvial components (subsystems), hence, it is crucial to assess the impact of design decisions on dynamic behaviour of a product at the system level. These issues emphasize on the importance of using methods such as simulation to simulate the behaviour of individual sub-systems and their dynamic interactions to assess the overall system performance of a design solution [2]. Developments in digital technologies have led to the Fourth Industrial Revolution or I. 4.0. Digital Twin (DT) is a technology of I.4.0 and aims to replicate a system's behavior. Experiment-able DT (EDT) and its application within SE can be a solution for explained challenges, allowing effective management of design data and utilizing simulation for development of complex products [3]. EDT is an executable (simulation capabilities) digital replica of a product that can support virtual validation of a design solution to assess its performance as a function of dynamic interactions among its subsystems. EDTs can be connected to a product to reflect its current state for further improvement.

1.1. Research Gap, Scope, and Objective

UNSW Redback Racing is the University of New South Wales's Formula Society of Automotive Engineers motorsport team, consists of over 100 students, who design, build, and race formula-style race cars. Redback Racing developed Internal Combustion racecars from Team's founding in 2000 until 2019. However, since 2019, Redback has transitioned to competing in EV category at Formula SAE Australasia (FSAE-A), and successfully competed with its

10.1016/j.procir.2025.08.097

first EV, RB21-E, at 2022 competition in Winton, Victoria. However, the development of RB21-E was fraught with challenges. There were a series of design flaws, requirements and rules legality errors, and manufacturing faults that were uncovered late into the vehicle's implementation and thus caused significant delays and budgetary impacts. It became apparent throughout the development process that the team's reliance on previous design and engineering judgement born of prior experience was no longer sufficient for the generation of an integrated and high performing vehicle, given the transition to the unknown of EVs. This research aimed at the application of EDTs in SE for development of Racing EV.

One challenge in EDT application is efficient integration of domain-specific simulation tools in developing an integrated EDT model [4]. Integration extent of third-party simulation tools into EDTs is currently quite limited [5]. Also, relying on the creation of custom simulation tools is a major barrier for organizations that lack capability to develop tailored solutions [6] [7]. There is a need for an EDT approach that can seamlessly integrate off-the-shelf domain analysis tools, including CAD, FEA, CFD, and Multibody Physics tools [8].

This paper aims to propose an approach for developing an integrated simulation model of a vehicle as its EDT to support virtual validation of a vehicle design and can be used as a test bed to assess its performance as a function of dynamic interactions between various vehicles elements. This EDT can be used for analyzing the impact of decisions on the vehicle performance at early stages while it can be used later when the physical vehicle is constructed for monitoring and real-time optimization. This research aims to integrate third-party tools into EDT for integrated use in the vehicle's lifecycle.

Although the scope of this research is the development of EDT for EVs, the proposed approach can be applied in other complex systems developments with needed adoptions. The outcomes of this research can support those development projects that have a repeating nature such as Redback Racing.

2. Methodology

To support the SE and design integration of the developing EV-RB23, an EDT was initiated by constructing a repository that centrally collated and integrated design data, requirement models, simulation models and physical vehicle data. This was done to reduce design integration/requirement fulfillment challenges experienced in the development of previous vehicles. The EDT of RB23 was composed of the following: SysML Models, High-Level, Vehicle Domain, and Low-Level Vehicle Models, and Vehicle Telemetry Data.

All elements of EDT were stored centrally on a Cloud-based Windows instance, and exchange of inputs and results between models was conducted by an Excel workbook, which stored shared input variables used by vehicle models, which were determined and provided by vehicle domain models.

2.1. Conceptual and Technological Modelling with SysML

The vehicle SE models were constructed in SysML using the Open-source Python-based modelling suite [9]. Gaphor was chosen due to its open-source nature allowing ease of access to small organizations, such as a Formula Student Team, and its Python-based construction allows customization or interfacing with other software and specific scripts. Within Gaphor, the vehicle system and its elements were modelled in a set of Requirements, Structure, and Behavior Diagrams. Diagrams were constructed at two detail levels: Conceptual and Technology levels corresponding to the maturity stage in the vehicle development.

Prior to EDT development commencement, requirements analysis documents were prepared for each element of the vehicle, containing a collation of all FSAE-A Rules, and functional and performance requirements derived from high-level vehicle functional and performance targets.

The high-level performance target for the vehicle was to finish in the Top 5 places at FSAE-A. Looking at previous competitions, an average of 698 points was required to place 5th, and a points goal of 750 points was chosen to ensure a small factor of safety. Accordingly, individual event point score targets were derived based on previous team performances and identified key areas for improvement.

Based on the dynamic event point targets: Acceleration, Skidpan, Autocross, Endurance and Efficiency, the required vehicle-level performance targets were defined (given in Table 1) by analysis of vehicle acceleration values required to score the point target in 2019 FSAE-A competition, since it was the most competitive recent competition and was a good basis for point score estimates. The vehicle performance targets are given in Table 1, based on 2019 FSAE-A competition scores and the relevant target values were calculated. From them, vehicle functional requirements and performance targets were derived, and modelled in SysML.

Table 1 Event Point Score Targets

Requirement	Target Value
Lateral Acceleration	1.4g
Longitudinal Acceleration	0.95g
Braking Deceleration	1.5g

Table 2 RB23 Performance Targets

	1	2	3	4	5	6	7	8	9
RB21-E	33	58	112	0	0	0	0	0	202
THB	86	71	130	77	67.5	123	265	95	914
2018	72	67	130	75	67.5	70	264	95	840
Goal	65	65	130	75	60	90	190	75	750

Theoretical Historic Best Results: THB, 1: cost (/100). 2: Presentation (/75), 3: Design (/150), 4: Acceleration (/100), 5: Skid pad (/75), 6: Autocross (/125), 7: Endurance (/275), 8: Efficiency (/100), 9: Overall (/1000)

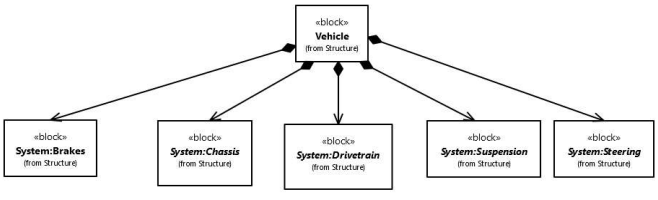

Fig. 1 Vehicle Concept Structure

Conceptual-level requirements were derived from the FSAE ruleset, functions required of the vehicle, and performance targets determined for the vehicle in the context of competition. The conceptual-level structure defined the constituent systems of vehicle as shown in Fig. 1.

Technological requirements were derived from FSAE ruleset, functional, or performance targets. Unlike conceptual level, the requirements were expressed as specific design features, technology components, or functions. Requirements for many branching subsystems in vehicle were modelled in separate diagrams using <deriveReqt> relationships. This allowed for traceability of requirement derivations, while streamlining the representation of complex collections of requirements for parent functions. For structure and behavior diagrams at the technology level, subsystems were modelled as specific technological features. The Power Transmission Behavior Diagram is shown in Fig. as example.

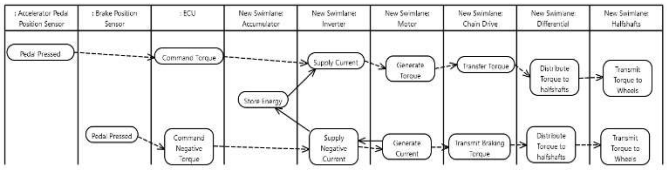

Fig. 2 Power Transmission Behaviour Diagram

2.2. Vehicle Modeling

For use at the vehicle conceptual design stage, the High-Level Vehicle Model was programmed in MATLAB based on OpenLap open-source lap simulator project and Redback Racing-Developed models [38]. The simulation workflow, shown in Fig. 3, consisted of modelling the Vehicle performance envelope based on vehicle parameters, configuring the racetrack or maneuver based on either collected GPS data or manually entered coordinates, and then the simulation of lap time and vehicle behavior.

Fig. 3 High-Level Model Workflow

The vehicle was modelled as a point-mass, with a simple tire model featuring lateral/longitudinal friction coefficients and cornering stiffnesses, and a simulated steering system. Powertrain model included a configurable torque map, modifiable gear ratio, and configurable driveline efficiencies. Aerodynamic model was customized to model simulated pitch and cornering sensitivity and provision for the simulation of powered ground-effect aerodynamics.

Brake model was of significant detail, including hydraulic cylinder sizes and brake pad friction coefficient and radius. While the model lacked fidelity and accuracy, it provided sufficient detail for use in conceptual design stage of vehicle development process. Although, the model's fidelity was low but tailored to focus on key functional requirements enabling early-stage decision-making with no need for detailed simulations. The skidpan lap time simulation output is demonstrated in Fig 4.

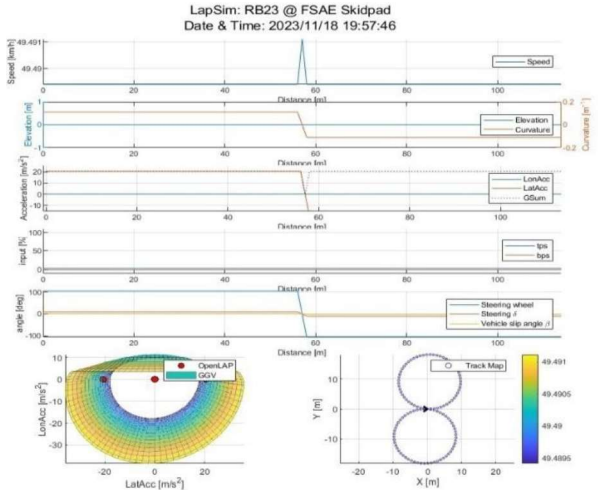

Fig 4 RB23 Skidpan Lap time Simulation Output

2.3. Vehicle's Models integration

The developed domain specific models of the vehicle vary in detail and platform, which complicate data management and traceability of requirement satisfaction. To overcome this issue, the various vehicle system models were stored and run on the central cloud instance, and their inputs and outputs were centrally managed. As an example, the developed drivetrain and braking system model is shown in Fig. 5. The model could get accelerator and braking input signal manually defined or as the logged driver inputs to simulate the vehicle longitudinal motion, brake line pressures, brake disc temperatures, powertrain current draw, cell state of charge, and battery pack temperature. This model was used in virtual validation of vehicle, simulation of its behavior under extreme conditions, and for diagnostics, as the physical outputs could be compared to simulated outputs for a given driver inputs.

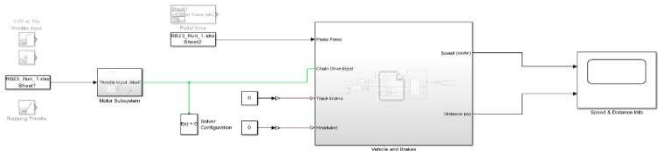

Fig. 5 Vehicle Drivetrain and Brakes Model

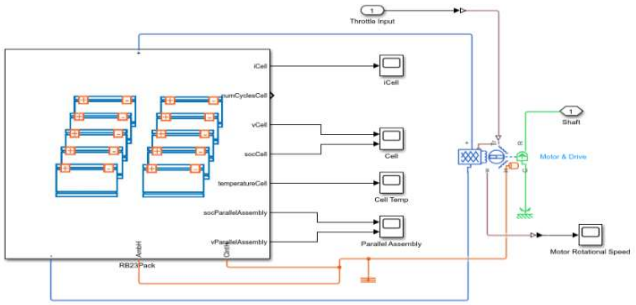

Fig. 6 Battery and Motor Subsystem Model

The powertrain electrical was modeled using MATLAB Simscape Battery Builder Application (see in Fig. 6), which consisted of a battery pack model, a 144S6P configuration of Sony VTC6 cells, arranged in 12 segments, and a combined

motor and drive system model. The driveline model consisted of a non-deal chain drive and limited slip differential. The brakes system model included mathematical functions to convert pedal force to pad friction and resulting brake torque.

The drivetrain and brakes models were connected to a longitudinal vehicle model to return Weight transfer and transient response of vehicle in longitudinal motion. This model, Fig. 7, used a Magic-Formula tire model which was tuned to experimental tire data possessed by the team [10].

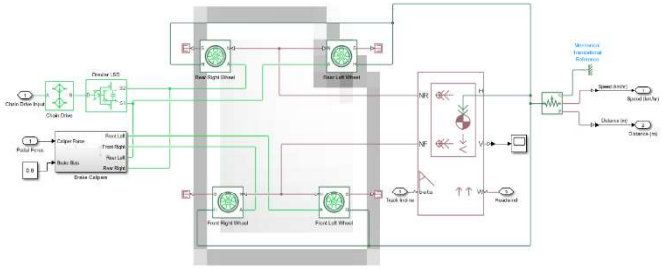

Fig. 7 Vehicle and Driveline Model

Other specific models, such as vehicle aerodynamic CFD and chassis torsional stiffness FEA models, were configured to output key values determined from probes or defined output values to centralized simulation parameter storage, which were then gave input to vehicle models, see in Fig. 8.

Fig. 8 Vehicle Aerodynamic Model Workflow

2.4. Vehicle Detailed Modelling

The low-level vehicle twin model was constructed using Simscape Multibody Vehicle Modelling Templates available in Mathworks Racing Lounge for Student Teams [39]. These templates provided subsystem multibody and solver models for use in the assembly and configuration of full-vehicle models. Accordingly, a full-vehicle RB23 model was configured to closely match specifications of the physical vehicle. The model had high details, with suspension model being of full multibody form and featuring a Pacejka-based Magic Formula Tyre model, transient damper, spring/anti-roll bar models with programmable force curves, joint friction, and structural compliance modelling [40]. The suspension model calculated individual member forces and spring and damper displacements, which combined with member dimensions, could simulate structural stresses in service.

Chassis model included torsional and bending rigidity to model chassis as a torsional and bending spring. The chassis model considered rotational inertia in all freedom degrees.

The brakes model returned the required pedal force from the required braking torque used by the dynamic solver, based on vehicle parameters, including pedal mechanical advantage, hydraulic cylinder sizes and numbers, brake pad radii and pad coefficient of friction. The brake model, however, did not model thermal effects, and thus has limited utility.

The aerodynamic and powertrain models had low details, with the aerodynamic model consisting of CL, CD and Frontal Area constants, and the powertrain model utilizing an ideal motor and battery model.

The driver model used for simulation was configurable as open or closed loop, with limited ability to simulate extreme performance. The driver model provided by Mathworks functions, aimed to model a target path and velocity, which were defined in maneuver configuration. Hence, to simulate extreme performance, the lap time must be optimized using an algorithm that could increase the target velocity during the maneuver until the vehicle was not capable of achieving the target. This method eventually yielded a relatively indicative representation of vehicle; yet it was less efficient than driver models employed by other software packages.

The vehicle model is alternatively able to receive manually defined inputs or logged driver inputs from the physical vehicle in the form of CSVs. The vehicle Chassis and Suspension Model is shown in Fig. 9 as an example.

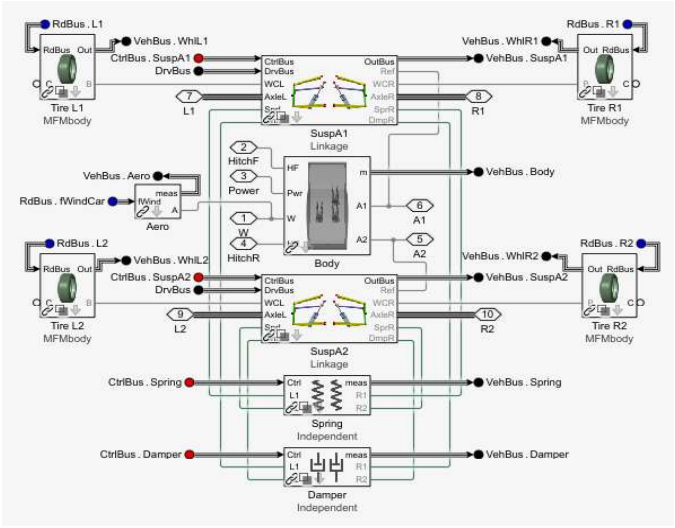

Fig. 9 Chassis and Suspension Model

2.5. Vehicle Telemetry Methodology

The vehicle data acquisition and cloud-based telemetry system included a Raspberry Pi mounted on the vehicle, and intercepted vehicle sensor and control system signals as CAN signals, timestamped and encoded them into binary-serialized format and forwarded them to the AWS cloud-based telemetry system via an onboard Wifi router. As the CAN data was sent to web server, it was decoded from CAN protocol to individual signals and simultaneously logged and displayed live on a web-hosted telemetry front-end.

The logged signal data was stored alongside the vehicle configurations entered via a setup web application. This way, all vehicle data was linked to its related setup parameters at running time. The signal and vehicle configuration logs were exported in CSV format once logging was concluded. The logs could be viewed on the telemetry web service through a data analysis front-end, or input to any of vehicle models, where logged driver inputs may be used as model inputs, and the simulated vehicle outputs compared to logged outputs.

2.6. Data integration

All vehicle data, including CAD models, simulation models, analysis results and vehicle telemetry data, was stored centrally on Team's Amazon Web Services cloud server. This

allowed all data to be accessed by all team members with required access permissions, while retaining a common authoritative source of truth for all vehicle models and data.

Vehicle operational data was exported from telemetry web service in CSV form and linked with vehicle's parameters at running time, as entered by lead race engineer.

The simulation parameters were organized into a central source of truth using a common Excel Workbook, including specific sheets for each vehicle system. Each sheet was tiered in order of design maturity, with system requirements, high-

level vehicle model parameters and low-level vehicle model parameters tiered accordingly. Full-vehicle CFD or chassis FEA could return specific values directly to linked cells in the related sheets. The high and low-level vehicle models could draw simulation parameters directly from linked cells in the workbook, allowing vehicle models to be modified by only updating stored parameters in workbook. Upon completion of vehicle simulation, system requirements linked to each sheet were automatically verified based on the simulation results.

Excel workbooks were chosen as the structuring element for simulation parameters as usability and accessibility was important for general application of EDT, particularly one unfamiliar with the methodology and technology.

3. Virtual Validation Application

The key use case of full vehicle model was the continuous virtual validation of full-vehicle performance requirements. Thus, the vehicle model, configured in detailed design phase, was simulated on acceleration, skid pad/autocross circuits, and in other open-loop maneuvers to validate performance requirements satisfaction as given from analysis of previous years of FSAE-A EV category results, given in Table 3.

Table 3 RB23 Dynamic Targets

Requirement	Target Value
Lateral Acceleration	1.4g
Longitudinal Acceleration	0.95g
Braking Deceleration	1.5g

To validate as-designed of vehicle could achieve dynamic requirements, its performance was simulated by low-level vehicle model and Wide-Open Throttle-(WOT) Braking and Steady State Cornering Maneuvers (SSCM). During WOT Braking Maneuver, simulated vehicle accelerated at limit for 80 meters, to simulate FSAE-A acceleration event, and then braked at limit until a stop. This allowed for combined validation of longitudinal/braking acceleration requirements. SSCM simulated vehicle travelling around a circular path of 18.25m diameter at traction limit to simulate maximum lateral acceleration of vehicle. Plots of longitudinal acceleration in WOT-Braking and SSCM are shown on Fig. 10.

Vehicle achieved Braking longitudinal acceleration target with significant factor of safety, yet could not sustain the longitudinal acceleration requirement, excluding the initial launch, a peak longitudinal acceleration of only 8.298 m/s^2, or 0.846g. The vehicle could achieve a sustained lateral acceleration of 15.77 m/s^2 or 1.61g, satisfying dynamic performance requirement for lateral acceleration.

During initial physical testing, the vehicle recorded a skid pad time of 4.81s corresponded to average lateral acceleration of 15.57 m/s^2. This lateral acceleration was achieved using tires with significant wear and likely lost significant grip. Hence, it was assumed that the vehicle could achieve the simulated 15.77 m/s^2lateral acceleration.

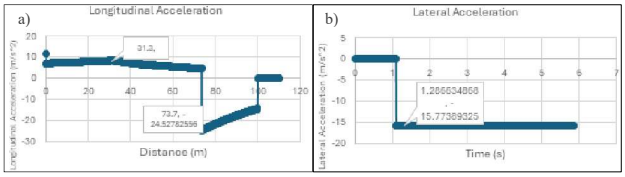

Fig. 10 a) Braking Longitudinal Acceleration b) SSCM Lateral Acceleration

4. Discussion

4.1. Integration of EDT in Vehicle Project Lifecycle

Unlike many previous EDT applications, RB23 EDT was constructed in a tiered fashion, growing in fidelity in design process. This allowed for incremental increase in modelling fidelity as the design progressed, ensuring that initial stages focused on high-level, system-level analysis, while later stages incorporated more precise data to refine performance predictions. Like this study [11], the proposed EDT consisted of multiple models with decreasing levels of abstraction [11]

Requirement analysis: SysML modelling of EDT coincided with requirements analysis stage of design process. The requirements were modelled diagrammatically in SysML requirement models at conceptual and technology levels. Various systems and subsystems could be defined at the end of this stage, determining the structure of next stage models.

Conceptual design: the high-level vehicle model can be used for early conceptual trade studies and virtual validation, allowing the indication of early vehicle concept's satisfaction of performance requirements. The high-level vehicle model was linked to the SysML conceptual requirement blocks within Gaphor, allowing automatic requirement satisfaction check based on output parameter values, such as vehicle accelerations, top speed or energy consumption.

Detail design: various vehicle models can be created in detailed design stage (e.g., CFD, DES, FEA, …) and be connected to central parameter repository by giving output to and take input from relevant parameter. Vehicle models can be linked to SysML technology-level requirement blocks to automatically trigger requirement satisfaction traceability. As models are modified, resulting in changes to whole-system performance parameters, the corresponding high-level vehicle model parameters are modified in turn, so maintaining an up-to-date representation of vehicle at a high-level, including high-level requirement satisfaction.

Commissioning: vehicle twin models can be utilized to aid troubleshooting and virtually validate the performance of components or systems. Diagnostics are facilitated by comparing simulated with measured system behavior. Logged driver inputs were used as input to vehicle models, then simulated outputs compared to measured ones to identify the cause of undesired behavior.

Operation: Integration of telemetry data with simulation models streamlined vehicle optimization. By facilitating the comparison of historical data to simulated performance, setup modifications investigated and tested by simulation, reducing dependence on physical testing to optimize the performance.

4.2. Simulation

The common approach in integration of simulation within SE has been using simulateable parametric equations [12] [13]. Existing approaches do not fully integrate third-party domain simulation tools. The proposed approach allows integrating third-party domain simulation tools (e.g., FEA, CFD) into model of the full vehicle, and consequently enables increasing vehicle modelling fidelity, improving design process integration by integrating domain simulation results. This integration enhances the accuracy of the overall vehicle model by enabling detailed, real-time feedback from simulations, ensuring that the full vehicle's performance is iteratively refined with high-fidelity data in design process. The proposed approach did lack full co-simulation between models. For example, low-level full vehicle models lacked a detailed constituent powertrain or aerodynamic model. The decision to abstract some vehicle models was made due to computational or resource limitations. In-built simulation of aerodynamic behavior within the full-vehicle model would be prohibitively computationally intensive, for example.

5. Future Work and Conclusion

Major design integration issues in previous vehicles were physical interference and geometric incompatibility between components in CAD. While during RB23 development, the development of a method of integrating automated collision checking and validation of geometric requirements into EDT and greater integration of CAD model properties into simulation models greatly streamlined the integration efforts required. For example, one geometric requirement of the vehicle was the High Voltage battery must have a clearance of >25mm from all non-crushable items in its vicinity. This requirement necessitated manual confirmation of satisfaction in design, yet its traceability was improved by automation. Yet, EDT can be improved by improvement of model fidelity and validation of all model features. The low-level vehicle model lacked fidelity in some key areas. Addition of aerodynamic maps of behavior in varying cases of roll, pitch, heave and yaw, or the inclusion of aerodynamic co-simulation, and the integration of a detailed electrical and thermal powertrain model, would greatly improve the utility of the full vehicle model. Further, the physical validation of the vehicle and low-level vehicle model behavior was also required to further improve the reliability and utility of EDT.

During the project, the powertrain and low-level vehicle model were configured for vehicles featuring a single electric motor and rear-wheel drive, however, the team preferred to develop a four-wheel drive electric vehicle in future. Hence, the vehicle model required significant reconfiguration, which posed an obstacle to the continued adoption of EDT approach in projects with repeating nature as was the case for RB23. Therefore, a future improvement would be the addition of a configuration application front-end, which would abstract the model configuration process, streamlining user experience.

The EDT can incorporate fault scenarios, such as bearing faults, to assess their impact on system performance in future developments. The EDT used in development of the Redback Racing Electric Vehicle, RB23, successfully contributed to the systems integration of the vehicle while addressing key limitations of previously documented projects. The vehicle achieved its goals and finished in second place overall.

6. References

[1] Shiva Abdoli, Sami Kara, "A modelling framework to design executable logical architecture of engineering systems," *Modern Applied Science ,* vol. 11, no. 9, p. 75, 2017.

[2] Abdoli, S., Kara, S, " A modelling framework to support design of complex engineering systems in early design stages," *Research in Engineering Design,* vol. 31, pp. 25-52, 2020.

[3] S. Abdoli, "Experimentable digital twin for virtual validation of manufacturing systems," in *In Proceedings of the 2023, 10th international conference on industrial engineering and applications*, 2023, January. .

[4] Kaslow, D.; Soremekun, G.; Kim, H.; Spangelo, S., "Integrated Model-Based Systems Engineering (MBSE) Applied to the Simulation of a CubeSat Mission," in *2014 IEEE Aerospace Conference*, Big Sky, MIT, USA, 2014.

[5] Abdoli, S., Ye, D., "An Investigation on Available Technologies and Approaches for Integration of Control Systems with the Digital twin in Cyber-Physical Production Systems,"*In Proceedings of the 2023 10th International Conference on Industrial Engineering and Applications* , 2023, January.

[6] C. Nigischer, S. Bougain, R. Riegler, S. H. P. and M. Grafinger, "Multi-domain Simulation Utilizing SysML: State of The Art and Future Perspectives," *31st CIRP Design Conference*, Twente, Netherlands, 2021.

[7] A. M. Madni, C. C. Madni and S. D. Lucero, "Leveraging Digital Twin Technology in Model-Based Systems Engineering," *Systems ,* vol. 7, no. 1, 2019.

[8] Ji, X., Abdoli, S., "Challenges and Opportunities in Product Life Cycle Management in the Context of Industry 4.0.," *Procedia CIRP,* vol. 119, pp. 29-34, 2023.

[9] A. Molenaar and D. Yeaw, "Modelling for Everyone | Gaphor," Gaphor. [Online]. Available: https://gaphor.org/en/. [Accessed 15 July 2023].

[10] "MagicDraw," Dassault Systemes, 2023. [Online]. Available: https://www.3ds.com/products-services/catia/products/no-magic/magicdraw/. [Accessed April 2023].

[11] M. Maio and e. al, "Closed-Loop Systems Engineering (CLOSE): Integrating Experimentable Digital Twins with the Model-Driven Engineering Process," in *2018 IEEE International Systems Engineering Symposium (ISSE)*, Rome, Italy, 2018.

[12] G. H. Fisher, "Model-Based Systems Engineering of Automotive Systems," in *AIAA/IEEE/SAE Digital Avionics Systems Conference Proceedings*, Bellevue, WA, USA, 1998.

[13] C. Dumitrescu, R. Mazo, C. Salinesi and A. Dauron, "Bridging the gap between product lines and systems Engineering: An experience in Variability Management for Automotive Model-based Systems Engineering," in *Proceedings of the 17th International Software Product Line Conference*, Tokyo, Japan, 2013.

[14] H. Kim, D. Fried, P. Menegay, G. Soremekun and C. Oster, "Application of Integrated Modeling and Analysis to Development and Analysis of Complex Systems," in *Conference on Systems Engineering Research 2013*, Atlanta, GA, USA, 2013.

[15] S. Boschert and C. H. R. Rosen, "Next Generation Digital Twin," in *Proceedings of TMCE 2018*, Las Palmas de Gran Canaria, Spain, 2018.

[16] E. J. Tuegel, A. R. Ingraffea, T. G. Eason and S. M. Spottswood, "Reengineering Aircraft Structural Life Prediction Using A Digital Twin," *International Journal of Aerospace Engineering,* vol. 2011.