SysML Based Modelling of Gear Shifting Strategy and Drivability for Automatic Transmission

Ankush Kumar1,* and Sachin Dhanwate1

1KPIT Technologies, Pune, 511047, Maharashtra, India

*Corresponding Author: ankushjmi@gmail.com

Abstract. In this project SysML modelling of gear shifting strategy and drivability are done for the development of the transmission control system. Gear shifting is the complex system operation and drivability is also important for driving pleasure. The work starts by developing system functions to capture system-level requirements such as grade compensator, hot mode manager, shift scheduler, torque converter state selection, transmission speed calculator, manual mode manager, etc. The next phase includes designing and developing structure models, behavioural models, and the analysis of system requirements. Thus, the functionality of the design patterns is verified in the earlier stage of the product development life cycle.

1. Introduction

Model-based systems engineering (MBSE) is the modelling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases [1]. System modelling language (SysML) represents one of the most used modelling languages to support MBSE approaches. SysML is intended to unify all these modelling languages, techniques and tools and serves as standard modelling language for systems engineering as in the case of unified modelling language (UML) for software development [2]. SysML extends the characteristics of UML and replaces the classes and objects by modelling blocks of System Engineering [5]. The SysML diagrams are shown in figure 1, each diagram is dedicated to the representation of a concept of the system. These nine SysML diagrams are organized into four categories known as the four pillars of SysML as shown in figure 2. These categories represent the system structure, behavior, requirements and constraints.

Systems engineers communicate with a diverse group of customers including different engineering disciplines, managers, organizational and business roles. The document-based analysis is not suitable for the complex system [3]. SysML modelling gives a common basis for the analysis and design of the entire model [4]. This project is concentrated on the Shift strategy and drivability of automatic transmission. Shift strategy is a very complex system operation and drivability is necessary for driving pleasure.
This system includes general aspects of the gear proposal for an automatic transmission by adjusting the shift points under different driving conditions, allowing optimal driving performance, low fuel consumption, low exhaust emission, etc. Basic scheduling is realized with maps in terms of engine load and vehicle speed. It uses a set of maps, covering different driving situations for optimizing performance. Shift point selection is the function carried out based on several inputs from the driver (drive modes, accelerator pedal position, etc.) and active driving situations. These shift points are updated during vehicle operation based on a set of conditions and then used by the Transmission control system for ratio calculation. The current driving situation is also the input information for a ratio calculation.
these inputs, it will calculate a gear proposal. This system also includes the requirement for gear limitation based on the multiple parameters like oil temperature, thermal de-rate, friction clutch temperature, etc. These gear limitations should be done to protect the engine and transmission components and to provide user safety.

Another part of this system covers general aspects of torque converter clutch state determination, which depends on the driver selected mode or an active driving condition (fuel shut off, multi-displacement system, engine braking, etc.). The state of the torque converter clutch is selected for providing torque multiplication, for maximum torque transfer from the engine into the transmission or to provide slip between the engine and the transmission to mitigate torque disturbance, when engine features fuel shut off, multi-displacement system or variable valve lift control are active, optimal power to the transmission. There is a separate shift scheduling map for the torque converter clutch which will accommodate the desired state selection of the torque converter clutch.

2. Description of System operation
We divided the overall proposed system into 4 Components.

2.1. Gears shifter
This component considers vehicle grade estimator, vehicle mass estimator, grade and mass compensator, altitude compensation manager, hot mode management, kick down detection, after treatment warm-up management, engine over-speed detection, transmission over-speed detection, curve detection, engine under-speed detection, brake on coast down management, shift scheduler, gear arbitration and desired gear selection as different functions. These functions are responsible for selecting the shift maps and adjusting the shift points in the selection of desired gear in different driving conditions.

2.2. Torque converter manager
This component considers torque ratio calculation, select torque converter clutch interpolated lock point, torque converter clutch management as different functions. These functions are responsible for selecting a shift scheduling map for the torque converter clutch in different driving conditions.

2.3. Transmission shaft speed calculator
This component considers transmission speed calculation as a function. This function determines input shaft speed and output shaft speed by acquiring their raw value from reading their sensors connected to the transmission input and output shafts.

2.4. Manual mode calculator
This component provides driver intervention and control over automatic transmission using manual shifting.

3. System Modelling
The project is implemented in SysML using IBM Rhapsody. This modelling method comprises some important phases that need to be followed to perform the modelling process. The first phase is the analysing of system requirements where the environmental actors and assumptions are captured and developing system behavioural model, the second phase is the designing of the logical architecture and structural model, the third phase is the validation the system functional requirements to check correctness and completeness of the system.

3.1. Requirement analysis
Analysed and refined system requirements and gave the relationship with one requirement to another requirement and use case diagram considers the different actors involved in the system to describe the system in terms of usages by actors. The use case diagrams further described using textual representation and visually using a sequence diagram and activity diagram. Figure 3 shown a use case diagram for the system.
3.2. Designing the logical architecture

Developed system logical architecture with block definition diagram, internal block diagram, and state machine diagram. We have taken actors, ECUs, sensors and another sub-system of the transmission control system into consideration. Figure 4 shows the block definition diagram for the system.

![Diagram](image)

**Figure 3. Use Case Diagram**

![Diagram](image)

**Figure 4. Block definition diagram for (a) ECUs, Sensors, and actors (b) gear shifting and drivability management**

The internal block diagram represents interconnection and interface between the parts of a block by using ports and interface blocks. It refers to the communication and flow of items between the block. Figure 5 shows the internal block diagram for the system.
Figure 5. Internal block diagram for (a) ECUs and Sensors (b) transmission control module and subsystems and (c) gear shifter and drivability management.

The state machine diagram specifies the sequence of events that an object goes through during its lifeline in response to the event. Each block contains state machines according to the requirement and C++ programming logic is developed. The purpose of the state machine diagram is to trigger the states, to perform the action while entering and exiting the states, to generate the event and to accept an event upon execution of the model. Some state machine diagrams are shown below in figure 6.
3.3. System validation
To check correctness and completeness of the system is validated the model functional behaviour by using panel diagram and animated state chart. Panel diagrams are used to simulate the model functional behaviour by binding with the value properties and events to the panel elements according to the developed C++ programming logic in the state machine diagram. Rhapsody has an inbuilt option to generate executable code from model to simulate panel diagrams. Three panel diagrams are developed for this system namely transmission panel, controller panel and diagnostics panel. Rhapsody also enables the animated state charts by generating events to check the behaviour of the component. The bright coloured state shows the current state of the system. One executed simulation panel and animated state chart are shown below to validate a system requirement.

3.3.1. An example requirement
The transmission control system shall detect uphill and downhill gradients that the vehicle is traversing using the longitudinal acceleration information received from the brake control system and calculated vehicle acceleration from vehicle speed as determined by using the output shaft speed sensor.
Figure 7. Panel diagram for example requirement (a) transmission panel (b) control panel and (c) diagnostics panel

Figure 8. Animated start chart for example requirement (a) brake control module and (b) gear shifter
4. Result and Discussion
The functionality of every component in the design pattern validated to build an error-free and complete system architecture. This modelling allows us to identify interactions, the flow of data and control between parts of the system. As shown for example in section 3.3.1 each requirement of the system is checked for validation. It is identified that out of 125 requirements, 38 are ambiguous and hence modification in requirement is needed. The present work also found 8 new requirements and 6 missing signals for the system.

SysML offers an advantage in modelling by considering different aspects such as specifications and requirements, structure, behaviours, and constraints. Using this method, the system engineer can analyse, verify and validate intended functions in the earlier stage of the product development life cycle. The model can also be reusable for the new functional objective. Also, the activity diagrams, sequence diagrams, and state machine diagrams can be used by the supplier to design software code.

5. References
[1] D. N. Luta, A. Raji and W. Fritz, 2014 Application of Model Based Systems Engineering in the modelling of photovoltaic system, *Energy Technology Conference*, Istanbul
[2] Khouloud Nasraoui, Najeh Lakhoua, Lilia El Amraoui 2017 Analysis and Modeling of a Wind Power System based on SysML, *International Conference on Recent Advances in Electrical Systems*, Tunisia
[3] X. Liu and Y. Cao, Design of UAV Control System Virtual prototype using Rhapsody and Simulink 2010 *International Conference on Computer Design and Applications*, Qinhuangdao, pp. V3-34-V3-38
[4] Mary Bone, Robert Cloutier 2010 The Current State of Model Based Systems Engineering: Results from the OMG™ SysML Request for Information 2009. *8th Conference on Systems Engineering Research*, Hoboken, NJ
[5] Neelakantam, Pavani. 2013 Modeling Flight Software from Architectural Design Patterns. *International Journal of Computer Applications*. 62. 38-43.

Acknowledgments
The authors would like to thank the system engineering team, KPIT Technologies Ltd. for your suggestions, guidance, and inspiration in carrying out this project. This project was financially supported by KPIT Technologies, Pune, India.