Abstract - This paper describes work done at the IARC, in collaboration with an automotive Original Equipment Manufacturer (OEM) and suppliers, in the project Systems Modelling Language (SysML) for Automotive Software Development and Integration. The OEM is interested in how a SysML model could supplement or even replace paper specifications whereas the suppliers are more interested in finding how the resulting SysML model could be used for software development. The project focuses on practical aspects so that deployment of the language and related technology is possible smoothly. The case study involves a Driver Information System for a premium vehicle. Our industry partner supplied the requirements and specification documents. The progress to-date including the challenges faced, results so far and the plan for further work are detailed.

Keywords: Modelling; SysML; automotive; software; integration.

1 Introduction

UML is the de-facto standard for model based specification and development of software across various domains. The standardisation efforts related to UML and the consequent emergence of UML 2.0 has further enhanced its image as a modelling language [1, 2, 3, 4, 5]. In parallel, a new language called Systems Modelling Language (SysML) which is closely-related to UML but better suited to systems engineering is emerging [6, 7]. The use of UML in automotive feature realisation and software development, however, seems to have particular issues which have been documented and commented upon in various sources [3, 6, 7, 8]. Industry experience indicates that there is a level of usage for feature definition using UML which takes place within pockets of the automotive industry. Yet the actual implementation of software for feature realisation is less evident. One of the prominent arguments behind this is the perceived fact that for real-time applications it is better to follow a functional approach, involving industry standard Computer Aided Software Engineering (CASE) tools such as MATLAB/Simulink/Stateflow, rather than an object-oriented approach through UML. However, if UML based specification is transferred to a procedure based software development, as generally happens in industry quite often, there is a high potential for misinterpretation, loss of information and ultimately poor quality software. Therefore, the problem to be investigated here is: - Can the complete development from requirements through to implementation be carried out in a UML-based language, in particular with the advent of enhancements like UML 2.0 and SysML? SysML being geared towards systems and software engineering has a better potential for the automotive domain and hence is worth focussing on.

From the automotive industry perspective, the following issues are very important:

- How should the requirements be modelled using SysML and communicated to the supplier?
- Should the supplier develop the software using the same language? If so, how?
- Does the language have adequate support for the development of real-time embedded automotive software?
- Are there software tools that adequately support the language?
- Are tools inter-operable i.e., are tools capable of exchanging model information?
- Is the use of this modelling language an improvement over the current methods?
- Is this modeling language well-suited for practical applications within the automotive industry?

1 The author is also a faculty in the School of Engineering at The University of Warwick
2 Introduction to SysML

Systems Modelling Language (SysML) is a new modelling language based on the UML. It reuses several diagrams from UML 2.0, improves on some diagrams and introduces some new diagrams. From the 0.9 version of the SysML specification [7], the diagram taxonomy is given below in fig. 1. Some diagrams and changes are mentioned briefly below.

Requirements diagrams have been introduced to model requirements and their dependencies. They allow us to depict how requirements flow down from high-level requirements to those for specific subsystems. They also allow us to depict how aspects of the design meet the requirements. Parametric diagrams give the ability to describe parametric relationships between system properties. Assembly diagrams allow us to model a system as hierarchies of reusable components. Each part within the system can be defined by an assembly with its own parts and ports. In terms of support for modelling behaviour, activity diagram from UML has been enhanced with support for continuous flow. Activities can be disabled during execution and control operators can be used to generate control flows to control other activities. For further details, some references include [6, 7].

3 Methodology

3.1 The case study – selection and description

A suitable case study has been defined with our industrial partner. It is a Driver Information System for a premium vehicle. The specifications and other documents supplied by our industrial partner are being used as a reference.

The main user interface for multiple infotainment applications has to be provided by a touch screen display. There should be two basic forms of it: - For one set of vehicle models, a compact unit accommodating a 7-inch touch screen and with only two push buttons on the lower bezel; and for another vehicle model, a larger unit accommodating the same touch screen, but with ten push buttons arranged to the left and right and two push/turn rotary knobs. The display will be mounted in the mid-fascia area giving the driver and/or front passenger the ability to view information and status from the following options (where fitted):

- Navigation
- Front and rear entertainment
- Trip computer
- 4x4 information
- Phone
- Controller for the entertainment system such as on/off, volume, search etc. (In one vehicle model only)
4 Development of the SysML model

Starting from the OEM’s standard requirements specification document, a SysML model is being developed. Appropriate diagrams such as use case diagrams, assembly diagrams and sequence diagrams are being used to capture requirements. Leading software tools that support SysML are being used for developing the model.

To help us evaluate the model, discussions with OEM system engineers and other stakeholders are held on a regular basis. The relevant elements of SysML language for the OEM are being identified by the OEM representatives, IARC and also the suppliers. Similarly, the relevant elements of SysML for the supplier would be identified during the course of the project.

5 The systems modelling experiences

The first step involved in modelling was to capture the use cases as shown in fig. 2.

Figure 2: Use Cases for the Driver Information System application

The diagram shows the various actors who would use the system and the associated use cases. For example, the “Driver” would use the system for viewing navigation information (as depicted by the connection to “View Nav. Info” use case) and so on, and the “Rear Passenger” would use the system for accessing rear entertainment (“Access Rear Ent.” Use case).

The original requirements specification document was not very clear in this regard and certainly not concise and visual. These use cases were refined as in fig. 3.

Figure 3: Refinement of a use case

The above refinement using the “<<include>>” stereotype shows that “View Steering Angle”, “View Gear Position” etc. are all part of the “Access 4X4iInfo” use case. Similarly, the “<<extend>>” stereotype could be used to extend a particular use case with additional features.

Apart from the clarity and conciseness, the advantage of using the use case diagram is that it aids in clearly modelling each of the use cases further. By identifying important uses through notes on the diagram, it can also help, at later stages of the system development lifecycle, in planning the development of functionality in a desired order.

SysML also provides requirements diagrams wherein the textual requirements from the requirements specification documents could be shown and then depict which use cases satisfy those requirements. Additionally, we can link these requirements to diagrams such as sequence diagrams which help us verify those requirements. We have not used these diagrams yet because they have not been provided in the tool apart from the fact that we could add these at a later stage in the modeling process. Even though, the tool does not provide requirements diagrams yet, these can be depicted by using stereotypes.

Also, links between requirements in the requirements management tool and the modeling tool we used were possible. This also helps in having traceability throughout the system development lifecycle.

Assembly diagrams were used to depict the various sub-systems. Fig. 4 shows an assembly diagram for the configuration in one of the vehicle platforms. They show the various parts and ports including some IO Flows. The parts include the various buses, gateways, electronic control units (ECUs) and other devices. In this assembly diagram, the buses have only been named generally as Bus1, Bus2 and so on but it still conveys the system structure. Production of this diagram involved discussion with engineers. In the original documents, information for the electrical platform for both sets of vehicle models was described simultaneously. Some non-standard diagrams
were used in those documents to partially depict the system. In addition, all these vehicles could have many optional components such as TV Tuner and CD player. As a result, the original documents gave a confusing picture of the system structure. The assembly diagrams are a concise and clear way of communicating this information.

![Figure 4: The Assembly Diagram](image)

In the requirements documents, scenarios were described textually. Such scenarios were not only occupying several pages but were also in a non-standardised format. The differences between similar scenarios for 2 different vehicle platforms could not be made out easily. We used sequence diagrams in SysML to model these scenarios. Fig. 5 and fig. 6 show examples for 2 different platforms. It is much easier to see the differences here.

![Figure 5: Sequence diagram showing how OffRoad view is obtained on the display (Platform 1)](image)

Sequence diagrams helped in exploring scenarios such as how steering angle information could be viewed as in fig. 7. However, this diagram does not show how the steering angle information is obtained from other systems in the vehicle. This is shown on another diagram as in fig. 8. Information such as timing can also be shown here. Any additional information helps to get the overall systems perspective which can help engineer the application appropriately. When alternative sequences are shown as above, the feature of the software tool to animate the diagram helps. It shows the messages flowing between objects in sequence and presents options to choose a particular alternative sequence when alternative behaviours are described. Stepping through such sequences is also useful when the sequence diagrams are large involving many actors, interface devices and associated messages shown on them.

### 6 Summary of lessons learnt

Currently, we have not seen much use of SysML within the automotive industry. In some requirements specification documents, we have seen class diagrams being used, though only for specifying some inputs and outputs to those classes. Through the work in our case study, we see that by producing models with use case diagrams and sequence diagrams, we can augment the paper specifications to clearly capture and communicate requirements. Assembly diagrams help in capturing systems or subsystems as collections of reusable parts and connections between them. Through the case-study, we are thus identifying key diagrams that an automotive OEM could use to produce a model of the system and send it across to a supplier. Through our modelling experience, we are also trying to understand the right way of modelling such as which diagrams to use and exactly how much information to convey through individual diagrams and so on. The supplier could then use the model and understand the requirements better and thus produce systems and software that accurately reflects the requirements. In terms of deploying SysML in projects within the automotive industry, we have been talking to various stakeholders within the OEM and suppliers. There is a need for training in SySML and more importantly on concepts of object orientation so that benefits can be obtained.

257
Presses Menu HardKey
seq
seq
seq

Driver
Front Passenger
Menu Button
4X4iInfo Button
Touch Screen Display
Driver InfoSys

Presses Menu HardKey
Menu HardKey Pressed

4X4iInfo Button Pressed
Update Display to Menu

Update Display to last OffRoad

Presses Menu
HardKey

Presses Menu HardKey
Menu HardKey Pressed

4X4iInfo Button Pressed
Update Display to Menu

Update Display to last OffRoad

Figure 6: Sequence diagram showing how OffRoad view is obtained on the display (Platform 2)

Driver
Front Occupant
4X4iInfo Button
Touch Screen Display
Steering Wheel
Driver InfoSys

Presses 4X4iInfo HardKey
4X4iInfo HardKey Pressed

If first time selection show Compass View else seq
end alt
Driver steers
Steer Left
Update Angle

Information on which parts of the Driver InfoSys are involved is shown on a separate diagram.

Figure 7: Sequence Diagram to show the sequence involved for displaying steering angle

steering angle to Bus 1
Gateway reads information
Converted message gatewayed
Gateway reads information
Converted message gatewayed
steering angle info. displayed

steering angle to Bus 1
Gateway reads information
Converted message gatewayed
Gateway reads information
Converted message gatewayed
steering angle info. displayed

Figure 8: Sequence showing how the Driver Information System obtains the steering angle information
7  Further Investigation

The work done so far certainly promises a lot of improvement over current practice. However, there are more questions to answer before SysML becomes a preferred route to systems and software development within the automotive industry. One important question is whether OO software is more reliable? How easy is it to verify the software? For the popular MATLAB/Simulink/Stateflow route of developing software, there is the possibility to verify the model against requirements through formal verification tools. Another question is whether it is possible to generate and use C code easily from SysML, if that is desired for efficiency and other practical reasons. If the OEM and suppliers use different tools, is it still possible to ensure reliability when SysML models are exchanged? Are there languages other than SysML which are better? Ultimately, the chosen language has to fit in to the system development lifecycle supporting verification and validation and easy integration into all the lifecycle phases.

As the systems are increasingly being developed in a supplier-OEM collaboration, changing the methods of capturing requirements impact both the OEM and the supplier. Hence, the companies have to work together and change to new methods. Because this requires time and effort, they usually wait for technology to be in ready-to-use form with little or no problems but by doing so, opportunities are missed in terms of the benefits that can already be obtained. Hence, there is a lot of sense in trying to keep pace with new technologies and try using them even if it is gradually.

Many automotive systems also involve some legacy. Systems developed for particular vehicle platforms have to be maintained for several years even if no new vehicles using that platform are being produced. Hence, the industry would have to deal with such issues while taking on new technologies.

8  Conclusions

From our current experience, we feel that using a SysML model to capture and communicate system requirements is worth the effort. As we have seen through some of our modelling experiences described in the paper and through our discussions with our partners, we believe that using a SysML model can significantly improve requirements capture and augment current paper specifications. It offers the potential to model system of systems and enable suppliers to develop reliable software more easily. A visual modelling approach such as this one can help engineers coming from different countries with different backgrounds, both technical and cultural. A SysML tool could also help different engineers to simultaneously work on different aspects of the model after which some consistency checks could be made. However, SysML as a language, is still being updated and standardized and hence there is an urgent need for some stability in the specification. For example, there is a difference even in the nomenclature of diagrams such as between version 0.9 of the SysML specification and the latest version 1.0 alpha [7]. Having stability in the specification would enable tool vendors to release tools which do not change often and they can focus on making the tools much more user-friendly. Within the automotive industry, systems usually evolve and not really start from scratch, and hence, SysML models for existing systems would have to be created. To do this would require engineers to re-visit how requirements were modeled in the past. To really benefit from all the upcoming technology, the automotive industry has to be open enough to upgrade employee skills and invest in new techniques and technology.

References


