Sophistication and Efficiency of Functional Safety Activity by Applying Systems Engineering

N. KANAI

In Functional Safety Activity based on ISO 26262, we have achieved sophistication and efficiency by applying Systems Engineering Method.

(1) Sophistication: Design information, which is becoming more and more complex as the number of functions increases, can be properly managed and visualized to improve explanatory capability.

(2) Efficiency: Document management can be centralized by using SysML tools. As a result, document creation time and verification time can be reduced. (= Reduction of development time)

***Key Words**: ISO 26262, functional safety, Systems Engineering, MBSE, SysML*

1. Introduction

In recent years, the number of functions required of electric power steering (EPS) have increased to include advanced driver-assistance systems (ADAS), autonomous driving (AD) systems, and steer-by-wire (SBW) systems, causing design and verification to become exponentially more complex and difficult than with conventional EPS. Functional safety activities based on ISO26262 require not only the design and verification of safety, but also the design process (design concept validity, traceability of requirements and verification results). Developing new functions such as these requires a higher level of design accountability, as well as efficient development in order to reduce costs while shortening development periods. Meanwhile, there is an increased demand for design processes to be explained when developing somewhat mature design technologies with a track record of past development. Although comprehensive knowledge of both hardware and software is required for functional safety activities, functions carried over from past development often only require the diversion of completed design specifications. This makes it difficult to determine what kind of knowledge and experience (regarding the purposes and applications of functions or signals) are required at the time of development, meaning that activities performed by novices often result in decreased development efficiency due to them being unsure of how to interpret impact analysis when applying to new development.

To facilitate the sharing of such future development requirements and design information, a method called "systems engineering" must be adopted. This method entails functions being designed in line with purposes (requirements), physical configurations being designed to realize those functions, and design information being created by clarifying the relationships between those components[1)].

In this paper, we aimed to achieve "sophistication" by organizing design information in functional safety activities, while achieving "efficiency" by applying MBSE (Model Based Systems Engineering) that uses SysML (Systems Modeling Language) tools to manage this design information.

2. Utilization of Systems Engineering and the "Sophistication" of Functional Safety Activities

This chapter provides a detailed explanation of how systems engineering is used to "sophisticate" functional safety activities via three methods: "construction of functional architecture," "use cases," and "construction of physical architecture."

2. 1 Construction of Functional Architecture

The EPS architecture applied in functional safety activities is a document that is shared both internally and with customers and suppliers, and which attempts to define the physical configuration necessary for realizing functions by including expressions similar to those in design specifications. In order to organize design information using systems engineering, it is constructed as an architecture (EPS functional architecture) that expresses functions by showing only "the chain of behaviors intended by the system," and which qualitatively expresses the purposes that each EPS function is expected to fulfill. **Figure 1** shows an example of EPS functional architecture. This figure shows an image of a torque sensor interface block that converts signals input from the torque sensor into calculated values inside the EPS, and defines it as a function that "quantifies the degree of rotational force."

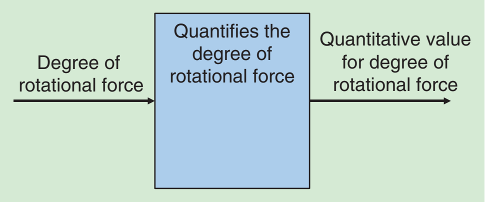
Degree of rotational force
Quantifies the degree of rotational force
Quantitative value for degree of rotational force

Fig. 1 An example of EPS functional architecture

2. 2 Functional Architecture Use Cases

The architecture created using functional expressions was divided into use cases for each EPS operation mode. Because there are cases in which the requirements may be different if the EPS operation mode is different, even if the function is realized using the same physical configuration as a result. **Figure 2** shows EPS operation modes divided into two use cases and defines the functions of each. Those two use cases are: "when assisting driver steering (Steer by Driver Mode)" and "when the vehicle decides to steer the EPS (Steer by Vehicle Mode)," such as that of an ADAS system. While Steer by Driver Mode is defined as "quantifying the degree of rotational force for calculating basic supporting forces (input value for calculating the amount of assistance)," Steer by Vehicle Mode is defined as "quantifying the degree of rotational force for judging driver steering (input value for steering judgment)." This shows that even though they have the same function of "quantifying the degree of rotational force," the application differs depending on the use case.

We were able to clearly delineate the boundaries of EPS by dividing function use cases based on the operation mode, enabling functional requirements to be expressed more clearly (i.e., not only function properties, but also situations in which the function is necessary).

2. 3 Construction of Physical Architecture

Using the same procedure as that used for the functional architecture, we constructed an architecture (EPS physical architecture) that expresses physical attributes by showing only the "system's functional chain," and which quantitatively expresses the design information that EPS each function is expected to provide. **Figure 3** shows an example in which the functions in **Fig. 2** are defined as realization methods. This example shows architectures that define methods of implementing the function of "quantifying the degree of rotational force" in Steer by Driver Mode and Steer by Vehicle Mode, which are divided into use cases. However, both architectures express the fact (design specifications) that the functions are to be realized using the same physical attribute, which is the "T/S signal interface," despite them being functions with different applications.

In this activity, a preexisting EPS was used as a model for redefining the functional architecture of a preexisting physical architecture. However, in creating the functional architecture that served as the purpose of the original systems engineering, we felt it necessary to construct a physical architecture for realizing it.

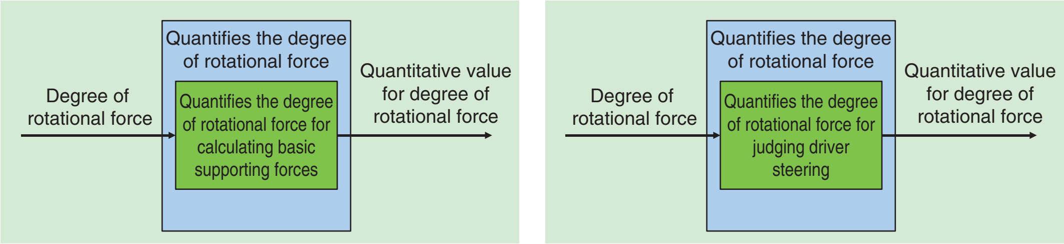
Degree of rotational force
Quantifies the degree of rotational force
Quantifies the degree of rotational force for calculating basic supporting forces
Quantitative value for degree of rotational force
Degree of rotational force
Quantifies the degree of rotational force
Quantifies the degree of rotational force for judging driver steering
Quantitative value for degree of rotational force

Fig. 2 Difference in functional definition depending on Operation mode (Left: Steer by Driver Mode, Right: Steer by Vehicle Mode)

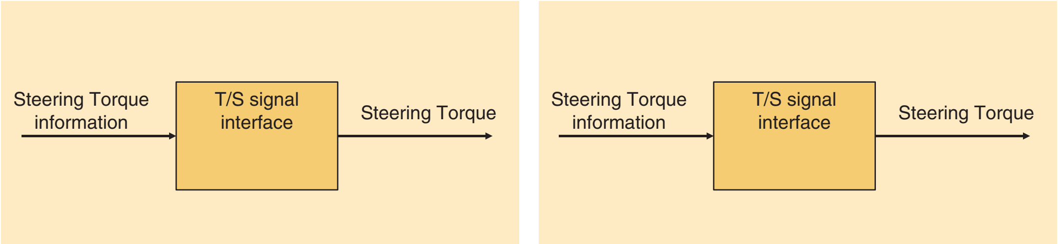

Fig. 3 Physical definition depending on Operation mode (Left: Steer by Driver Mode, Right: Steer by Vehicle Mode)

2. 4 Advantages and Challenges of “Sophistication” Using Systems Engineering

By restructuring the EPS architecture by dividing it into functions and physical attributes, we were able to clarify the required functions (objectives) and the corresponding physical attributes (realization methods). To enable the visualization of design information that should be shared during development in which functions are reused, and by clarifying the functions (purposes) of new development, we feel it is possible to proceed with development while mutually confirming whether the selected physical attributes (realization methods) are appropriate. Even when changes occur, the extraction of information — such as whether the change is due to physical factors (changes to realization methods) or due to functional factors (changes in required functions), as well as the extent of impact (impacted operation mode) — could be comprehensively expressed in the architecture, enabling the creation of design information that prevents omissions during impact analysis. However, although this enabled design information to be organized, it resulted in an increase in the amount of information to be managed. We anticipate that future increases in required functions and operating modes required for EPS will make design information more complex and make verification more difficult. Because it is difficult to link and manage this design information manually, we used the SysML tool to convert it into a single model-based information structure, which we then applied to the development process using MBSE in order to improve “efficiency.”

3. Achieving “Efficiency” Using MBSE

This chapter explains how we achieved “efficiency” by using SysML tools to convert the organized design information into an MBSE format within the tool as a single information structure.

3. 1 Constructing an Information Structure Inside the SysML tool

Design information for each function and physical attribute was used to create an information structure within the SysML tool.

①The context of the target system (all environments and systems surrounding the target system) was defined as a layer structure.

②A system diagram was created for each architecture (design information for each block and signal were defined as parameters).

By doing so, we were able to express ① the EPS’s relationship with the outside and ② its internal structure as design information that expresses functions and physical attributes.

③Relationships between functional and physical expressions were allocated.

By linking information that expresses functions and physical attributes, we were able to clarify relationships and construct a single information structure within the tool. As a result, we were able to understand which functions are comprised of which physical attributes (realization methods) within the tool, meaning that even if changes occur to either the functions or physical attributes, impacted areas can be managed without omission. **Figure 4** shows an example information structure constructed in the SysML tool.

3. 2 Visualizing Design Information and Extracting Output Information

In order to enable application to the actual development process, we used tool functions to construct a mechanism capable of outputting the information structure in the format of formal documents. In the conventional process, documents were manually created and modified in their own format. However, extracting from the aggregated information structure not only reduces the amount of time necessary to create documents, but also enables missing document modifications and inconsistent design information to be confirmed using the tool. Moreover, constructing a mechanism has enabled the information required at each stage of the development process to be visualized and output in any format, such as by extracting only impact analysis targets, which has contributed in facilitating the process of explaining design information during design reviews, etc. **Figure 5** shows changes to activity processes when the SysML tool was applied.

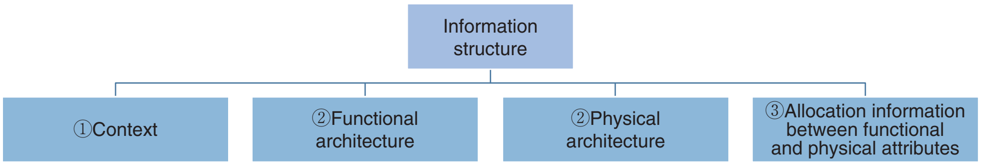

Fig. 4 Example information structure in the SysML tool

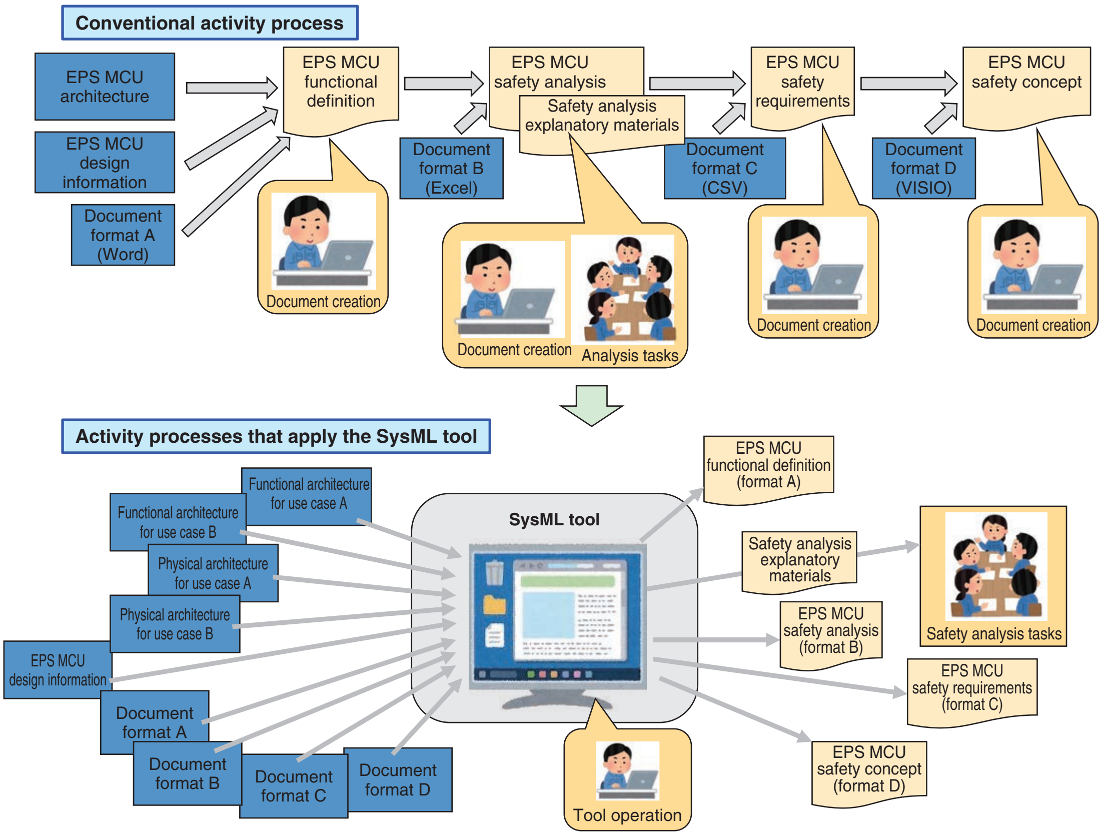
Conventional activity process
EPS MCU architecture
EPS MCU design information
Document format A (Word)
EPS MCU functional definition
Document creation
EPS MCU safety analysis
Safety analysis explanatory materials
Document format B (Excel)
Document creation
Analysis tasks
EPS MCU safety requirements
Document format C (CSV)
Document creation
EPS MCU safety concept
Document format D (VISIO)
Document creation
Activity processes that apply the SysML tool
Functional architecture for use case A
Functional architecture for use case B
Physical architecture for use case A
Physical architecture for use case B
EPS MCU design information
Document format A
Document format B
Document format C
Document format D
SysML tool
Tool operation
EPS MCU functional definition (format A)
Safety analysis explanatory materials
Safety analysis tasks
EPS MCU safety analysis (format B)
EPS MCU safety requirements (format C)
EPS MCU safety concept (format D)

Fig. 5 Changes in the activity process when the SysML tool is applied

4. Conclusion

Using systems engineering, we defined the functions required for the EPS based on their behavior and performed “sophistication” by linking them with the design information. Furthermore, because utilization of the SysML tool resulted in an increase in information management and utilization tasks (impacted area extraction, creation of explanatory materials and documentation), we had the machine perform information visualization tasks. Doing so enabled us to focus on thinking tasks, hence achieving “efficiency.”

During these activities, “functional definitions” during functional safety activities were described as an example and the effectiveness of systems engineering was shown. This concept can also be applied to post-processing activities such as “safety analysis,” “safety requirements,” and “safety concepts.” For example, it is possible to manage information by creating and linking design information within the tool to perform impact analysis from the viewpoint of “higher-order → lower-order” or “lower-order → higher-order” when changes occur. Furthermore, this concept can be applied to functional safety activities, as well as general design activities.

However, the SysML tool is not just for simply outputting documentation. Because the usefulness of the SysML tool in design work (in facilitating the explanation of design information) will only become apparent when design information is properly organized, its true utilization requires the acquisition of systems engineering knowledge and an increased level of awareness at the organizational level regarding the operation of SysML tools, etc.

References

1) S. Friedenthal, A. Moore, R. Steiner: A Practical Guide to SysML Third edition, The MK/OMG Press (2014).

N. KANAI*

* *Engineering Planning Dept., Automotive Business Unit*