Requirement Model for Mechanical, Electrical and Software Integrated Products Using SysML

Tadashi Gotoh[1], Takao Eguchi [1], Tsuyoshi Koga[2], Kazuhiro Aoyama[3]
1 IBM Japan, Ltd., 19-21 Nihonbashi Hakozaki-cho, Chuo-ku, Tokyo, 103-8510, Japan
2 Yamaguchi University, 2-5-1, Tokiwadai, Ube-shi, Yamaguchi, 755-8611, Japan
3 The University of Tokyo, 7-3-1, Hongo, Bunkyo-ku, Tokyo, 113-8656, Japan
(E-mail: tgotoh@jp.ibm.com,eguchies@jp.ibm.com, koga@yamaguchi-u.ac.jp, aoyama@sys.t.u-tokyo.ac.jp)

Abstract: For the development of high-functional and complex mechtornics product, it has been more important to manage the design process with considering multi-domains such as mechanical, electrical and software design. In spite of this, in embedded software development process, the designers cannot grasp the original requirements and intension of previous design and they sometime wasteful design iteration. The authors predict there are some problems for multi-domain integrated environment and suggest the requirement model should be introduced. In addition, the authors show key concepts of requirement model consists of described elements and component levels. They are very effective method for categorization and abstraction. Finally the authors evaluate the ability of description using SysML, with example. In conclusion, proposed requirement model is confirmed that it can surely describe the complex product requirements.
Key words: Requirement engineering; SysML; Model-driven development; Systems engineering

1 Introduction

In recent year, the mechatronics products, i.e., automotive devices, and office equipments and so on, are needed to implement high function feature because of exposures of heavy competition. For the development of such products, it is important for the mechanical, electrical and software designers to communicate the originally product's requirements from conceptual design phase and to detailed design phase. However, at present, the communication among three designer's domains; mechanical electrical and embedded software designs are separated. One reason for this is that drawings and specification documents are not sufficient to describe the original requirements and previous process's design intention. The author predict such insufficient communication leads to unnecessary design iteration and increase the design cost by of the error detection in test phase or the claim from customer in acceptance test phase. In addition to this, for the design change process, the authors predict that the designer cannot identify the scope of change without the traceability of requirement to implementation. From this we can see engineering changes affect unexpected part and it must be the cause of wasteful design iteration too.

In the future, the mechatronics products will become more complex. The authors predict the necessity of integrated multi-domains development environment will be increased. The members of development project should understand easily the original requirements and traceability of requirement-to-design for all area of product.

In this paper, the authors propose the requirement model should be introduced in design process for such integrated design environment[1]. The requirement model provides the environment for the communication of all members and they can understand traceability of requirement-to-specification and "what to make".

2 The Requirements for Requirement Model

There many studies which treat the problems of the product requirements. In mechanical design, VE[2][3] and QFD[4][5] methods are famous for grasp the requirements. In other hand, in software development, UML is the famous modeling language for software based on object-oriented approach[6]. However in the design of multi-domain, there is no unified method established.

In this paper, we will list up the problems and suggest the requirement for requirement model.

2.1 How to describe of requirements and design intension

In mechanical design process, drawings are the specification of form and how to make. In embedded software design process, specification document should specify how to make the program. However drawings and specification documents are focused on described the result of the design, so in many case we cannot find the original requirements and design intention in them. Furthermore, we have the problem that we don't have how to describe them totally in multi-domains.

The authors suggest the requirement model should be able to adopted description method for requirements and design intention.

2.2 Description sufficiency across multi-domains

Each domain's designers write various engineering documents to clarify the design concept. In many case these documents are focused in their own implementation scope. This results in that there is insufficient behavior description of the whole product. Maybe such information is designer's un-written knowledge. However when the complexity of products will be increased, the authors predict this situation leads to miss-understand for requirements and unnecessary design iteration. From this we conclude that we have the problem that we need the method to describe information of whole product in sufficiency.

The authors suggest the requirement model should have structure and categories for ability to describe for whole product.

2.3 Unified method for describe whole design process

In the mechanical design, designers specify the engineering specification which consist of the value of performance, limitation, and constraints, This information is static specification because the values are specified in the certain case, unless which are written or un-written in documentation. In other hand, in software design, designers study the software logic with digital value which is discrete on computer. This information is dynamic specification which is focused system behavior. The authors predict this difference of study manners should be respected and there is the necessity of unification description methods that satisfy the multi-domain's description. From this we conclude that we have the problem we need unified methods for multi-domain design.

The authors suggest the requirement model should be able to include the description method for all of mechanical, electrical and embedded software designs and unify the notation of them.

2.4 The reusability of the requirement definition

The requirements which software designer want to know are need to write by the mechanical designers. However, for the mechanical designers, these requirements are already obtained, unless written or unwritten documentation. If they don't have the opportunities for reuse these requirements, they think it could be extra-burden and they don't think necessity for requirements definition. From this we conclude that we have the problem we should perform reusability of requirements.

The authors suggest the requirement model should be able to easy to reuse for multi-domain designer. For example, the mechanical designer can identify the scope easily in accordance change of requirement.

2.5 The environment for sharing and managing requirement model

In the design process, the designers often use spreadsheet and documentation application, but there is no unified management system for various kind of documentation. In addition to this the designers sometimes fail to update documentation in management system. From this we conclude that we have the problem the difficulty of provide the system to share and maintain the requirement information.

The authors suggest the requirement model should be able to implement in computer application for sharing and managing capabilities. The requirement model should have modeling mechanism and be established by modeling language which satisfy the hereinbefore suggested requirements.

3 The Requirement Model

In this chapter, we will explain summary of the requirement model from two important view points in order to satisfy chapter 2's suggested requirements.

3.1 Described elements

The authors propose the four types of element should be described as the requirements, in reference to object-oriented approach. Four types of element consist of behavior, structure, constraints and required qualities.

3.1.1 Behavior: the description of functionalities

The authors suggest the functionality should be defined to behavior. Behavior will be described by states, transition, event and actions, in accordance to object-oriented approach.

3.1.2 Structure: the description of product configuration information

The authors suggest the product configuration information should be defined the structure which are leveled with appropriate component level. In addition to this, the information flow should be defined between components.

3.1.3 Constraints and engineering specification

The authors suggest the engineering specification and constraints values of them should be defined clearly. Especially, the traceability between mechanical design engineering specification and software constraints should be important for the requirement model.

3.1.4 Required qualities

There are some requirements is described ambiguously, because which are required the qualities for products aspect. But they cannot design the mechanism in their scope. The authors suggest such requirement should be define as required quality which is not defined previous three types of description.

3.1.5 Relationship between them

The four type's elements should have relationship each other. The authors suggest that such relationships should clarify the design dependency.

3.2 Component levels

When the designers develop the simplified product, there are not so many requirements. In this situation the designer can understand the scope of them easily. However for the development of complex products, various specialists are involved and across the many technical organization and join many suppliers. In this condition, there are many requirements and it is difficult to identify the scope by designer only.

The authors propose the component level should be introduced. Component levels are defined appropriate abstraction and granularity. Table 1 is the example of definition of component levels.

Table 1 Example of Component Levels

Component level		Definition	Examples of Cranes
Level 0	Product and environment	Product itself and actors related high functional application	The assembly plant, Roads
Level 1	Functionalities and applications	Fundamental "Function" and High-value "Application"	Crane system, Delivery system, Assemble lines
Level 2	Control system	Control system without inside components	Control system, Hoist system, Trolley system, Bridge system
Level 3	Components of system	Owned and related components	Obstacle sensor, input port, output port PLC Controller

3.2.1 Level 0: The product for development and environments (actors)

This level should be defined the product for development. It is defined appropriate level as highest level. In according to the complexity of products are increasing, we need consider the more related environments.

3.2.2 Level 1: The product's main- functionalities and across-applications

This level should be defined mainly functionalities of the product. For the complexed product, it needs other functions which are across the functionalities. The authors define this functions is application. and it is important for high-functional features.

3.2.3 Level 2 and Level 3: System and its components

These two levels should be defined control system and its logical components. Level 2 should be defined the control system for implementation of functionalities and application of Level 1. Logical component will be defined with "logical method". For examples, in crane system we can define control system as Level 2 and define the sensors, input port, PLC controller, output port and actuator in accordance of adoption PLC control.

3.3 Satisfaction the requirements of requirement model

These for types of description and component level will satisfy the requirements in chapter2. There are the reasons are follows;

- All original requirements will deploy to four types of element and develop further level elements and we can have the traceability with relationship. So requirements and intention will be shown in requirement model (Chapter 2.1)
- We can describe the whole and higher-abstract level behavior in Level 0 or Level 1. We can describe the requirement for input to detailed design in Level 2 and Level 3. From these we conclude that requirement model satisfy the whole product behavior and detail behavior description in same requirement model (Chapter 2.2)
- The static specification will be described constraints and engineering specification. The

dynamic specification will be described behavior. The relationship can be established between them. From these we conclude that requirement satisfy the design manner in multi-domain (Chapter 2.3)

- The relationships are established between any types of described elements. If one element change the content, we can find dependency with relationship information. From these we can conclude that requirement model have capability to show the scope of change. (Chapter 2.4)
- The requirement of Chapter2.5 is implementation of computer application. This requirement will be satisfied by modeling language and tools implementation and explain the following Chapter 5)

4 Modeling Process for the Requirement Model

In the chapter 3, the authors show the two key concepts, four type's described elements and component levels, for requirement model. In this chapter, the authors define the modeling process for the requirement model in accordance two key concepts. The process consists of three procedures. These procedure are shown Figure 1.

4-1 Define of product's component levels

We will define the component levels for the product. It is important that we need define the understandable level for the designers with considering product features, sizes, and development formation and so on.

4-2 Deploy the requirements

We will deploy the requirement to the requirement model. For examples, the functional requirements are described by behavior. The specification and performance targets are described by engineering specifications.

4-3 Breakdown the elements

We will breakdown the element to the next component level. For examples, structure and behavior are decomposed each other to further level[7]. Constraints will be bound to engineering specifications and the attribute which will be defined further components. The required qualities will be allocate to other described elements; structure, behavior, constraints and engineering specification.

5 Modeling Case Using SysML

SysML is proposed as the modeling language by OMG, which is based on object-oriented

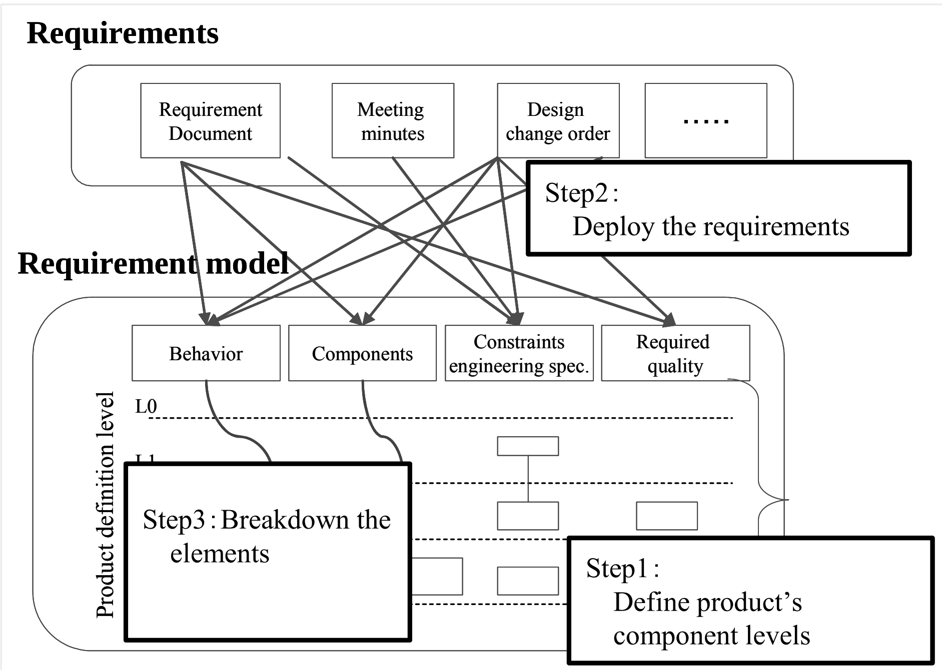


Figure 1 The Procedures of Modeling for Requirement Model

approach[8]. SysML can describe the software and hardware's function, requirements, constraints and so on[9]. This modeling method implemented by some tools and

In this chapter, we have example as the design of Intelligent ceiling crane, the authors explain how the designers can model using SysML in order to evaluate the requirement model capability.

In this example, the designers will design the crane to using the assembly plant. This crane will have the intelligent functionality and will be able to avoid the obstacles automatically. In addition , the crane will control to minimize the load vibration. The purpose of the crane development is improvement performance of production. The crane system must recognize the obstacles, which are moving big parts in delivery system and the workers besides assembly lines and so on, and transport the products and interim assembled materials with good efficiency.

The modeling procedures are described the following orders.

- Define the component levels
- Decompose the structure and behavior
- Define the constraints bound to target specification and other attributes.
- Develop the required qualities

Figure 2 shows applied SysML diagrams for described elements.

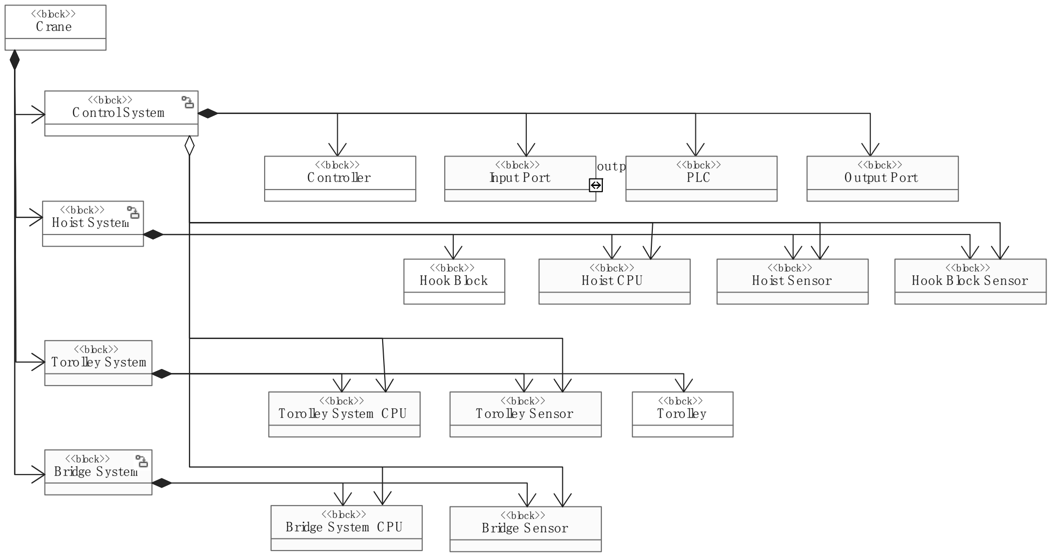


Figure 2 The Procedures of Modeling for Requirement Model

5.1 Define the component level

The designers should define the appropriate component levels based on the product features, the target of development and the development formation with suppliers and so on. In this example, the designers define the levels based on the target of development especially. The designer define the assembly plant as level 0 and the crane ,delivery system and assemble lines as level 1. The scope of the development, intelligent control system is defined as level 2. Figure 2 shows the definition and deployment the component in the component levels with SysML block definition diagram.

From these procedure we conclude that the clarified part by using model are ;

- The scope which the designer should study and develop
- The related components, such as delivery system and assembling lines
- The component levels which are shared by all project members

5.2 Decompose the structure and behavior

The designers can decompose the element with the structure and behavior. In this example, the procedures are explained from addition the intelligent function to control system (Level 2) to update the structure of components (Level 3) by separated six steps. Figure 3 is shown the six steps using SysML.

5.2.1 Step1: Define the component

The designer will define the component as the scope of development. In this example, control system (Level 2) is defined as the mainly scope of development. The bridge system, trolley system and hoist system are defined as originally existing systems. This step is represented by block definition diagram

Table 2 Applied SysML Diagrams for Described Elements

Type	Description	Diagram
Behavior	State and transition	State Machine Diagram
	Actions which perform the function	Activity Diagram
Components	Structure of whole product	Block Definition Diagram
	Interface between components	Internal Block Diagram
Constraints	Engineering specs	Requirement Diagram
	Constraints	Requirement Diagram
	Constraints relation	Parametric Diagram
Required quality	Quality requirement which cannot be deployed other elements	Requirement Diagram
Relationship	Relation between elements	Various relationships
	The result of design process	Allocation

5.2.2 Step2: Define the state and transition

The designer will define the state and transition of the control system. In this example, the designers will add the step mode, which is one of intelligent mode, in addition to manual mode. The designer will define the sub states of step mode. This step is represented by state machine diagram

5.2.3 Step3: Define the actions

The designer will define the actions in sub-states. In this example, the designers will develop “In execution” state’s actions. This step is represented by activity diagram.

5.2.4 Step4: Define the components

The designer will define logical components (Level 3) of the control system. In this example, the designer will adopt the PLC logic controller as control system. So, obstacle sensor, input port, output port and controller are added as logical components for control system in this step.

5.2.5 Step5: Allocate the action to the components

The designer will allocate the actions to the logical components which are defined in step4. The designer will study the allocation with some viewpoints. Figure 2 shows the swim lanes in activity diagram to define the allocations.

5.2.6 Step6: Clarify the data among the components

The designer will examines the data flow using object node in the activity diagram.
The added object flows mean the information among the logical components. Step6 in figure 2 is shown the object nodes in activity diagram and the data flow will be described in internal block diagram.

From these six steps we conclude that the clarified parts by using model are;

- The structure and components (Level 3)
- The behavior of the control system (Level 2)
- Actions and the allocation to the component (Level 3)
- The data flow among these components (Level 3)

5.3 Define the constraints bound to target specification and other attributes

The designer can specify the constraints. In this example, the consideration of constraints is from the target specification to the attribute of components by separated four steps. Figure 3 is shown SysML diagrams in these steps.

5.3.1 Step1: Specify the target specification

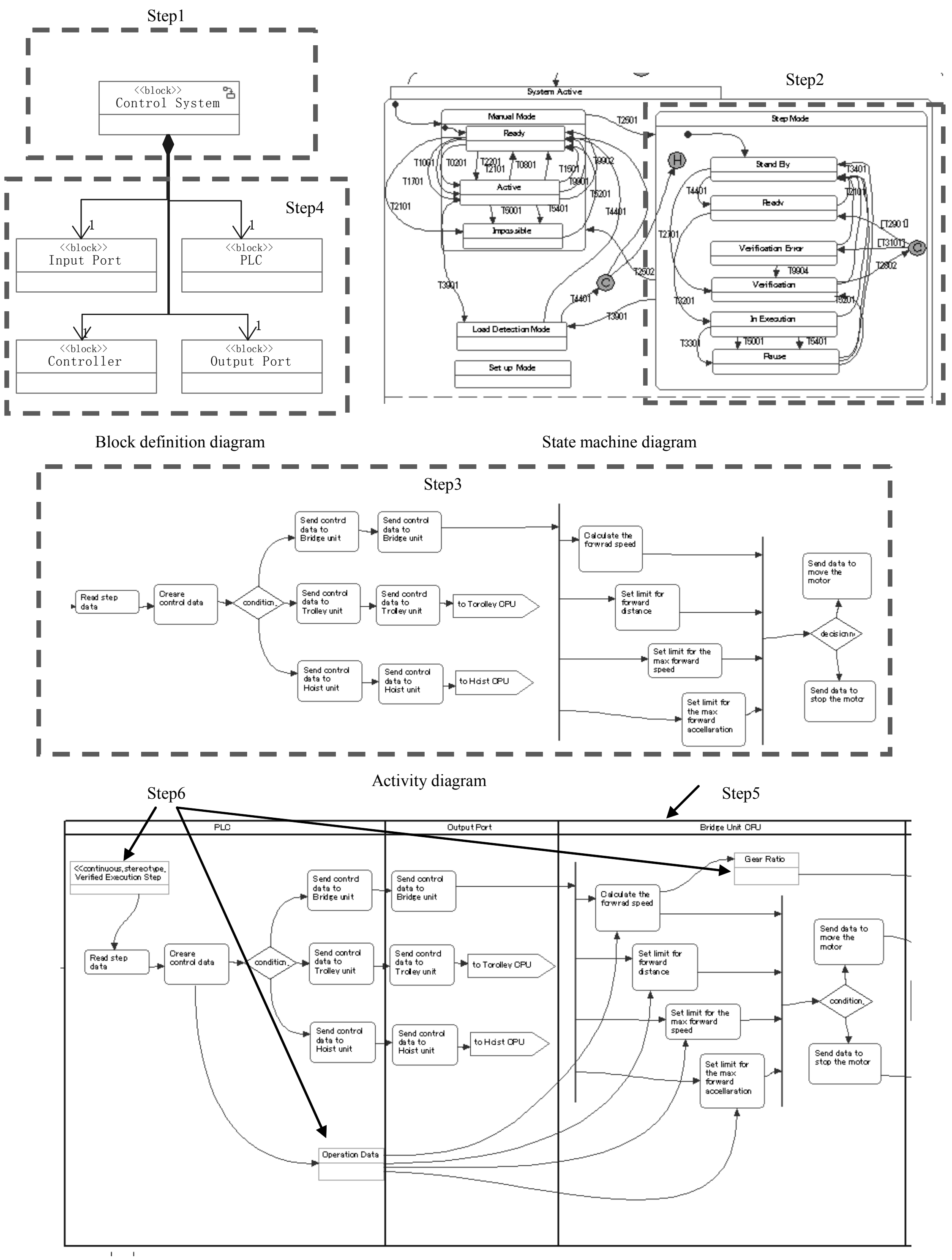
The designer will specify the target performance specification of the system. In this example, the designer will specify the productivity performance as engineering specification. Figure3 is shown by the engineering specification in requirement diagram.

5.3.2 Step2: Define the constraints between Level 0 and Level 1

The designer can define the constraints bound to the specification and attribute of Level 1 component. In this examples, the designer can define constraints of the target productivity performance bound to Operating time (Level 0) and tact time for the crane which is attribute of ceiling crane system (Level 1). Figure3 shows the parametric diagram.

5.3.3 Step3: Define constraint between Level 1 and Level 2

In this example, the designer can define tact time’s constraints bound to the average time of conveyance and the mean time of avoidance obstacles. The average time of conveyance is attribute of ceiling crane system (Level 1) and the mean time of avoidance obstacles is attribute of control system (Level 2). Figure3 shows the parametric diagram.



Block definition diagram　　State machine diagram

Activity diagram

Activity diagram with swim lane and object node

Figure 3　Applied SysML Diagram for Decompose the Structure and Behavior

5.3.4 Step4 Define constraints between Level 2 and Level 3

In this example, the designer will define the mean time of avoidance's constraints bound to the average width of parts and the sensor detection angles. The sensor detection angle is the attribute of sensor (Level 3 component)

From these four steps we conclude that the clarified parts by using model are;

- The constraints bound attributes in each levels
- The traceability from Level 0 to Level 3

5.4 Develop the required qualities

The required quality is one of the described elements, which is vague requirement for the quality. The designer can develop them in accordance with decomposition of structure and behavior.

The required quality should be developed to the other describe element; structure, behavior, constraints and engineering specification in appropriate component level. In this example, the minimization of the vibration of loads is required quality. So we set separated two steps to develop it using SysML. Figure 4 shows the diagram using SysML.

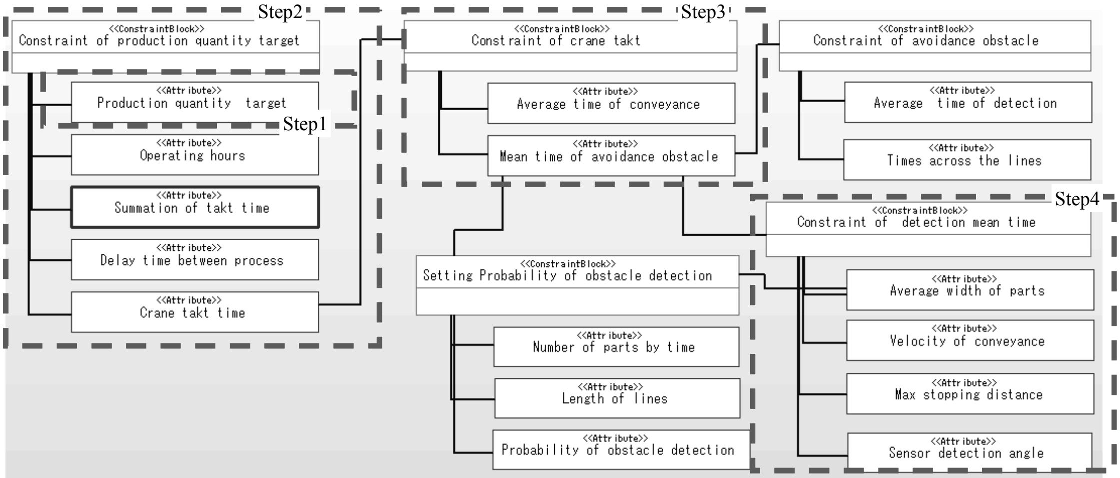
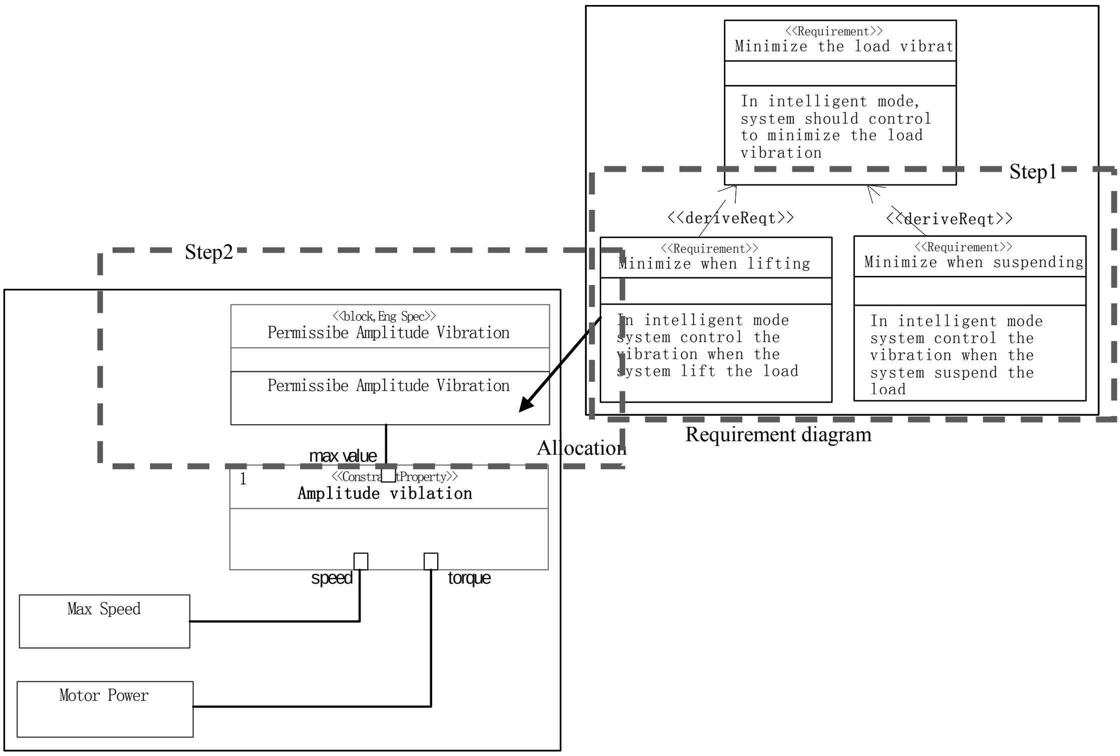


Figure 4 Applied SysML Diagrams for Define Constraints



Parametric diagram

Figure 5 Applied SysML Diagrams for Develop Required Quality

5.4.1 Step1: Derive the required qualities

The designer can derive the required requirements from original requirement. In this example, the designer will derive two requirements; the vibration minimization when lifting and suspending. Step1 in This step represents the requirement blocks of them.

5.4.2 Step2: Allocate the required quality to engineering specification

The designer can allocate the requirement to the other described elements. In this example, the designer will study the vibration limitation based on safety clearance of ceiling layout and define the engineering specification as this limitation. This specification will be allocated from the vibration minimization when lifting. This step represents for allocation between them.

From these steps we conclude that the clarified parts by using model are;

- The derivation of the required quality
- The intension of development the required quality
- The traceability from vague requirement to clear requirement

5 Conclusions

In this paper, the authors predict that insufficient information of original requirement and previous design process's intension lead to unnecessary design iteration. So, the authors propose requirement model should be introduced. In order to clarify the requirement of requirement model, the authors categorize and arrange the problems for the integrated development environment for multi-domains. In addition, the authors suggest the requirement model should adopt two key concepts; described elements and component levels. Finally the authors evaluate the ability of description using example, intelligent ceiling crane design. From these we can conclude that requirement model could be effective for the complex product design process and many part of design scope will be clarified by modeling with SysML.

References

[1] Gotoh T., Eguchi T.,Koga T., Aoyama K. Modeling for Product Requirements Based on Logical Structure of Product[J]. Journal of Japan Society of Design Engineering, 76-771(2010-11) (In Japanese）

[2] R. Anderson, R. Sturges. A Case Study in Extended Value Engineering[C]. Proceedings of the ASME IDETC/CIE, 2002

[3] Del L. Younker: Value Engineering: Analysis and Methodology, Marcel Dekker Inc., ISBN-13: 978-0824706968, 2003

[4]Shun Takai and Kosuke Ishii. Integrating Target Costing Into Perception-Based Concept Evaluation of Complex and Large-Scale Systems Using Simultaneously Decomposed QFD[J]. Journal of Mechanical Design, 128, 6 (2006): 1186-1195

[5] Yoji Akao. QFD: Quality Function Deployment, Productivity Press, ISBN-13: 978-1563273131, 2004

[6] Jacobson Ivar, Grady Booch, and James Rumbaugh. The Unified Software Development Process, Addison Wesley Longman. ISBN 0-201-57169-2, 1998

[7] Koga, T. and Aoyama, K. Product Behavior and Topological Structure Design System by Step-By-Step Decomposition[C]. Proceedings of the ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, DETC2004/DTM-57513 (CD-ROM), Salt Lake City, Utah September, 2004

[8] SysML specification OMG: http://www.omgsysml.org/

[9] INCOSE SysML Tutorial: http://www.omgsysml.org/