

Structuring a Product Design Process for Optimizing the Scale of Derivative Development

Yui Komano¹, Kazuya Oizumi², Fuyuku Katsu³, Yasushi Hattori³, Hiroyasu Miyoshi⁴,
Kazuhiro Aoyama⁵.

¹Department of Systems Innovation, School of Engineering, The University of Tokyo

²School of System Design and Technology, Tokyo Denki University

³Excellent Development System Promotion Department, JATCO

⁴Sysytem Design Research Institute, Inc.

⁵RACE: Research into Artifacts, Center for Engineering, School of Engineering, The University of Tokyo

Abstract: In derivative development, by planning a reasonable design process in advance, the development scale can be suppressed. However, no systematic method to structure a process is established. This research proposes a method to structure a process so that the scale can be optimized. For this purpose, a hypothesis about the rationale of senior designers is made and implemented as a mathematical model on a computer to help designers structure a process. A process is described as sequences between tasks that are defined as functional measures and design parameters to achieve them, which are depicted in domain mapping matrix. Pareto optimal processes in three metrics depicting the rationale are suggested. The designer examines the contents of the processes among different scales and determines an appropriate process and thus the appropriate scale. Case studies on a continuously variable transmission confirm the effectiveness of the proposed method.

Keywords: Design change, derivative development, development scale, design process, house of quality

1 Introduction

Derivative development is a common product development method, in which limited improvements are made to an existing product. As the range of changes from the existing product entails the scale of development, and thus the cost and development time, deciding the optimal range of changes is important. By planning a reasonable design process in advance, the development scale can be suppressed. In practice, the planning of a design process relies on the experience of senior designers and no systematic method is established. As product complexity increases, it becomes difficult to continue this current practice. Therefore, a method for logically structuring a design process is demanded. This research intends to propose a method to structure a design process so that the scale of derivative development can be optimized. For this purpose, a hypothesis about the rationale of senior designers is made and implemented as a mathematical model on a computer to help designers structure a design process.

2 Related works

Several studies have been performed on the management of design processes using matrices. Especially, process grouping and sequencing has been studied in numerous articles. Eppinger et al. (Eppinger et al., 1994) proposed a method to structure a design process based on input-output relationships between tasks. By reordering the tasks according to the design process,

a group of tasks that must be performed simultaneously and a sequence of tasks are calculated. Carrascosa, Eppinger, and Whitney (Carrascosa et al., 1998) extended the method by incorporating the probability of change and its impact on a time basis that enables process lead time simulation. Alison and Papalambros (Alison and Papalambros, 2007) studied partitioning and coordination of a system design with a genetic algorithm. In this method, a system is divided into several subsystems while the amount of coordination between subsystems is minimized.

Engineering changes have been studied as well. Eckert et al. (Eckert et al., 2006) proposed a method by means of design structure matrix (DSM) to predict change propagation on a product. In this method, the likelihood and impact of change between components are respectively shown in DSMs. Change propagation can be predicted by applying calculations of probability and expectancy. To handle change propagation in detail, it is desirable to describe products in multiple domains. Hamraz et al. (Hamraz et al., 2012) proposed change propagation modeling in multiple domains by applying a well-known FBS model (Function-Behavior-Structure) as a product model to multiple-domain matrix (MDM). As a result, it is possible to decompose and express change propagation using functions, behaviors, and structures as media, and improve the explanation. Conversely, it is difficult to describe the model because the number of product components to be described increases and it is necessary to set the occurrence likelihood and the size of the influence between the components.

In these works, DSM that describes products in a single domain is often employed to manage the design process. However, as Hamraz (Hamraz et al., 2012) described, handling multiple domains enriches the information obtained from the analysis. To discuss the development scale, it seems important to distinguish original change and the change choice. While the required changes are stable, design change choices that realize those required changes are optional. Initially, changes are requested for functional product measures. Next, changes are made on physical product features. Therefore, employing the House of Quality (used in quality function deployment (QFD)) seems relevant as it contains both domains and is widely acknowledged.

3 Product and process model

To describe a product, express the development scale, and determine a design process, a product model proposed by Oizumi et al. (Oizumi et al., 2014) is employed. In the proposed method, the product model is perceived as a House of Quality (used in QFD).

3.1 Product model

As shown in Figure 1, the following three element domains represent a product.

- Functional metrics (fm): These are the observable metrics used to evaluate the product functionality. These cannot be directly designed, but are rather realized through the determination of design parameters. As an attribute of a functional metric, importance (imp) is attributed as a numerical value.
- Design parameters (dp): These are elements of a product that illustrate its directly designed constraints.
- Components (comp): These are the physical elements that constitute a product. Every design parameter belongs to one component.

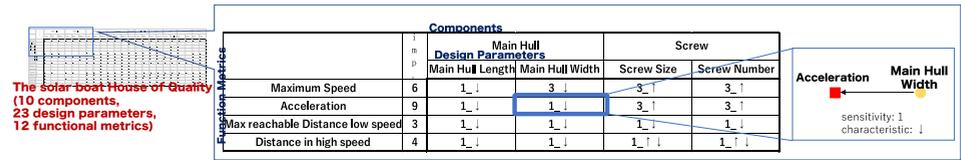


Figure 1. The solar boat House of Quality

Table 1. Possible values attributed to an effect relationship

Sensitivity	(Toward improvement of functional metric) 1: not strong enough 3: strong enough
Characteristics	↑: Larger the better ↓: Smaller the better ↑ ↓: Closer the better ?: Unknown

A change in a design parameter affects functional metrics. To detail the relationship between these effects (hereinafter called “effect relationship”), the following two attributes are given as shown in Table 1.

- Sensitivity: Sensitivity is an extent to which a functional metric responds to a design parameter change.
- Characteristics: Characteristics are the directions to which a design parameter needs to be changed to improve a functional metric.

3.2 Expression of development scale in the product model

Derivative development comprises several functional metrics where change requests are issued. These are called *change-requested functional metrics*. Furthermore, design parameters to be changed during development are called *change design parameters*. The determination of which design parameters to adopt for implementing the requested changes implies the development scale. Thus, in the product model, the development scale is expressed as change design parameters. In addition, functional metrics affected by the alteration of change design parameters while the changes of them are not requested are called affected functional metrics. Thus, the product parts to be changed comprise change-requested functional metrics, change design parameters and affected functional metrics. These are collectively called the *change implementation set* (Figure 2).

3.3 Expression of design process in the product model

It is assumed that structuring the design process is to determine which design parameters should be derived to achieve the required level of each functional metric and order of

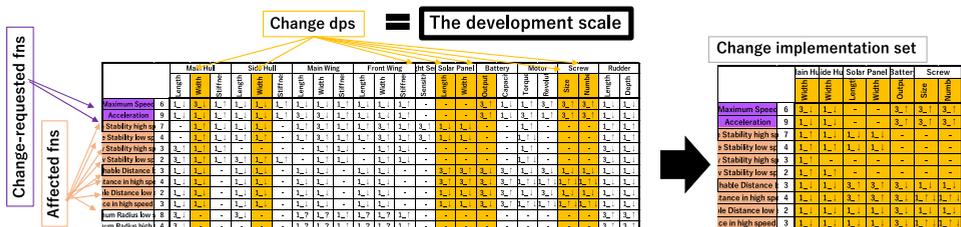


Figure 2. Change implementation set

Function Metrics	Components							
	Side Hull		Screw		Solar Panel		Main Hull	Battery
Design Parameters	Side Hull Width	Screw Number	Solar Panel Width	Solar Panel Length	Main Hull Width	Battery Output	Screw Size	
Role Stability high speed	1_↓	Task3	1_↓	1_↓	1_↑			
Role Stability low speed	1_↓	Task1	1_↓	1_↓	1_↑			
Max reachable Distance low speed	1_↓	1_↓	3_↑	3_↑	1_↓	3_↓	1_↓	
Distance in high speed	1_↓	1_↓	3_↑	3_↑	1_↓	3_↓	1_↓	
Maximum Speed	1_↓	3_↑			3_↓	3_↑	3_↑	
Acceleration	1_↓	3_↑			1_↓	3_↑	3_↑	
Yaw Stability high speed					1_↑			Task2
Yaw Stability low speed	1_↓				1_↑			
Max reachable Distance low speed cloudy	1_↓	1_↓	1_↓	1_↓	1_↓	3_↓	1_↓	
Distance in high speed cloudy	1_↓	1_↓	1_↓	1_↓	1_↓	3_↓	1_↓	

Figure 3. Expression of design process on the House of Quality

determining design parameters. A design process is expressed based on the product model and can be visualized in a House of Quality as shown in Figure 3. The following three element domains represent a design process.

- Design policy (Oizumi et al., 2014): A design policy is given as a set of preferences of effect relationships composed in a design parameter. For example, in Figure 4, “Screw size” is preferred to be increased for the purpose of improving “Maximum Speed” and “Acceleration”, but is not preferred to be decreased for the purpose of improving “Max reachable Distance low speed”, “Distance in high speed”, “Max reachable Distance low speed cloudy” and “Distance in high speed cloudy”. This is the “Screw size” design policy.
- Task: A task indicates that the functional metrics of a task are realized by the design parameters of that task. For example, in Figure 5, in “Task1”, “Screw Number”, “Solar

Function Metrics	Design Parameters	
		Screw Size
	Role Stability high speed	
	Role Stability low speed	
	Max reachable Distance low speed	1_↓
	Distance in high speed	1_↓
	Maximum Speed	3_↑
	Acceleration	3_↑
	Yaw Stability high speed	
	Yaw Stability low speed	
Max reachable Distance low speed cloudy	1_↓	
Distance in high speed cloudy	1_↓	

Preference

- Preferred (Red)
- Controversial (Grey)
- Not preferred (Green)

Figure 4. Expression of design policy

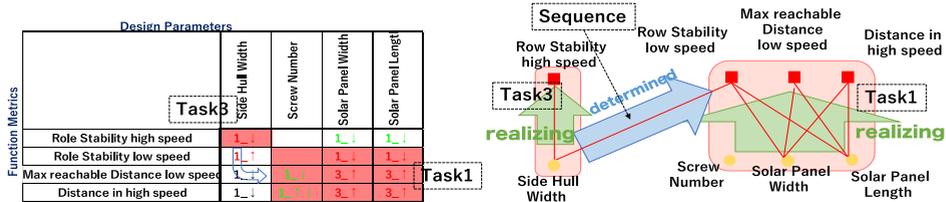


Figure 5. Expression of task and sequence

Function Metrics	Design Parameters			
	Side Hull Width	Screw Number	Solar Panel Width	Solar Panel Length
Role Stability high speed	1_↓		1_↓	1_↓
Role Stability low speed	1_↓	Task3	1_↓	Task1
Max reachable Distance low speed	1_↓	1_↓	3_↑	3_↑
Distance in high speed	1_↓	1_↓	3_↑	3_↑

A preferred effect relationship outside tasks

A preferred effect relationship inside tasks

Figure 6. Meaning of design policy and task

Komano, Yuichi; Oizumi, Kazuya; Katsu, Fuyuku; Hattori, Yasushi; Miyoshi, Hiroyasu; Aoyama, Kazuhiro

Panel Width” and “Solar Panel Length” are determined to realize “Row Stability low speed”, “Max reachable Distance low speed” and “Distance in high speed”.

- Sequence: A sequence gives a preconditioned value for a task defined in a former task, thus an order between tasks. For example, in Figure 5, “Side Hull Width” is determined in “Task3” and is used as is in “Task1” and “Task3” precedes “Task1”.

As explained above, a design policy shows by which design parameters a functional metric should be achieved. In contrast, an effect relationship inside a task means that the design parameter can be determined to achieve the functional metrics of the task. As shown in Figure 6, when a preferred effect relationship is inside a task, a design parameter that should be determined to achieve a functional metric can be determined in that manner, which is favorable. Therefore, a design process would be improved if more preferred effect relationships are included in tasks.

4 Proposed method

Based on the product model, candidates of a design process are explored for a certain development scale. Optimum processes are obtained for several development scales. By examining processes, designers can determine the proper development scale. Here, change implementation sets are prepared by designers. Next, a computer explores the optimum design process structure for each change implementation set. Examination of optimum design processes of the scales and the development scale determination is left to designers.

4.1 Determining the design policy (Oizumi et al., 2014)

First, the design policy is determined for a change implementation set. In practice, the design policy is something determined by a designer. However, the design policy may differ by the development scale. As the proposed method explores processes for several development scales, it is essential to establish a design policy for each scale, which requires significant effort to describe. Therefore, the design policy is calculated in this method.

To determine preferences of effect relationships, risks need to be calculated. Risk means the risk of negative effects on other functional metrics that may be caused by a change in the corresponding design parameter. Initially, each design parameter is judged whether it causes tradeoffs among functional metrics. In the case where a design parameter does not cause any tradeoff, those effect relationships have no risks. If it causes tradeoffs, those effect relationships have risks. To determine the level of risk, dominant effect relationship(s) need to be specified. It is assumed that the dominant factors of determining a design parameter are the sensitivity and importance of a functional metric. Thus, as shown in Figure 7, dominance of effect relationships is calculated as sensitivity multiplied by the functional metric importance. Effect relationship(s) with the highest dominance are specified as dominant relationship(s) and its characteristics become the dominant direction. If the characteristics of an effect relationship comply with the dominant direction of the corresponding design parameter, the effect relationship has low risk, otherwise it has high risk.

By using sensitivity and risk, preference is determined. For the first calculation, preference is given in ordinal scale of zero to two as seen in Table 2. For example, if an effect relationship has low risk and its sensitivity is one, the preference is calculated as one. For the second calculation, first round results are recalculated as seen in Table 3 to ensure that there is at

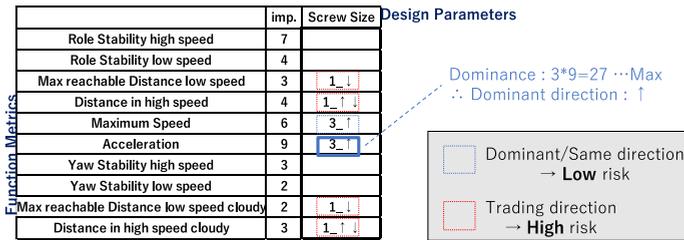


Figure 7. Calculation of risk

Table 2. First round calculation of preference based on risk and sensitivity

	Sensitivity: 3	Sensitivity: 1
No Risk	2	1
Low Risk	2	1
High Risk	0	0

Table 3. Second round preference calculation

For each functional metric		Preference of each effect relationship		
Max	Min	2	1	0
2	0	Preferred	Controversial	Not Preferred
2	1	Preferred	Controversial	N/A
2	2	Preferred	N/A	N/A
1	0	N/A	Preferred	Not Preferred
1	1	N/A	Preferred	N/A
0	0	N/A	N/A	Preferred

least one preferred effect relationship for each functional metric. The left side of Table 3 shows maximum and minimum preference values of effect relationships connected to a functional metric. The right side of Table 3 shows the recalculated preference value. For example, where maximum and minimum values for a functional metric are one and zero, an effect relationship whose first round preference value was one finally has “preferred” as a preference value.

4.2 Exploration of an appropriate design process

Structures of a design process are explored by dividing change implementation sets into tasks and changing their sequences upon calculated design policy. To evaluate design processes, the following three metrics are defined.

- Appropriateness of dp affiliation: When a preferred effect relationship is included in a task, it means that the preferred design parameter determined for the functional metric is appropriately affiliated to the task determining the functional metric. Thus, the more preferred effect relationships that are included in tasks, the more appropriate the process is. Arai (Arai, 2016) used the modularity (Newman, 2004) as a metric of cohesion when clustering effect relationships. Employing this, the modularity of preferred effect relationships is used as a metric.

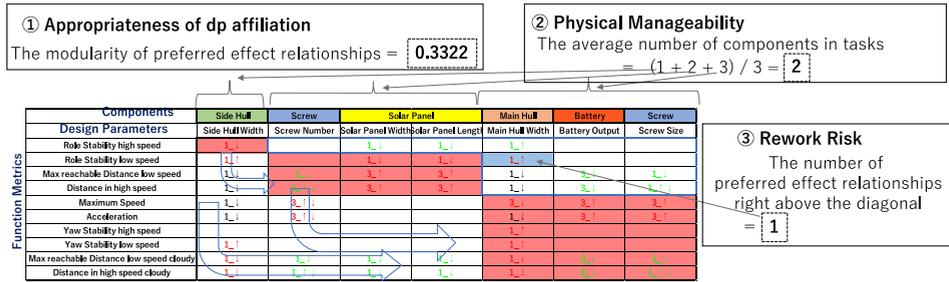


Figure 8. Three viewpoints to evaluate a design process

- Physical Manageability: Design reviews are often conducted by teams that are composed for components. Thus, the process management becomes more difficult when numerous components are involved in a task. Therefore, the average number of components in tasks are used as a metric.”
- Rework Risk: When a preferred effect relationship is outside of tasks, it means that the design parameter that has a major effect cannot be used for achieving the functional metric. Thus, the existence of a preferred effect relationship possibly becomes a cause of reworks. Eppinger (Eppinger et al., 1994) proposed that in DSM relationships right above the diagonal possibly becomes a cause of reworks. Employing this, the number of preferred effect relationships right above the diagonal is used as a metric.

It is assumed that the experienced designers structure a design process to be Pareto optimal for these three metrics. Because the solution space could be so large, a hot start technique is employed. First, the optimal design process for “Appropriateness of dp affiliation”, which seems to prevail among the metrics, is deduced. Then, neighborhoods are explored.

As shown in Figure 9, tasks are determined by maximizing the modularity of preferred effect relationships. First, a binary matrix of a change implementation set is constructed by depicting only preferred effect relationships. The binary matrix is seen as a monopartite graph regardless of the functional metric or design parameter. The Newman method (Newman, 2004) is applied to deduce tasks that maximize the modularity. Then, As shown in Figure 10, sequences between tasks are decided by the number of effect relationships between them. Tasks are ordered so that a large number of the effect relationships would become a prior task to a posterior task. Thus, the rework risk would be minimal under a given set of tasks.

Finally, multi-objective optimization is performed by exploring neighborhoods from the deduced design process. For every step, a new design process is randomly produced from each existing design process. When randomly producing a process, as shown in Figure 11, one functional metric or design parameter is randomly selected and moved to another randomly selected task. In this way, a new task configuration is determined. Then tasks are sorted in the sequence so that the rework risk is minimized. Inferior solutions are culled in a specified number of steps. The exploration is terminated when a determined number of processes are explored. The designer examines the contents of the obtained design processes and determines an appropriate design process.

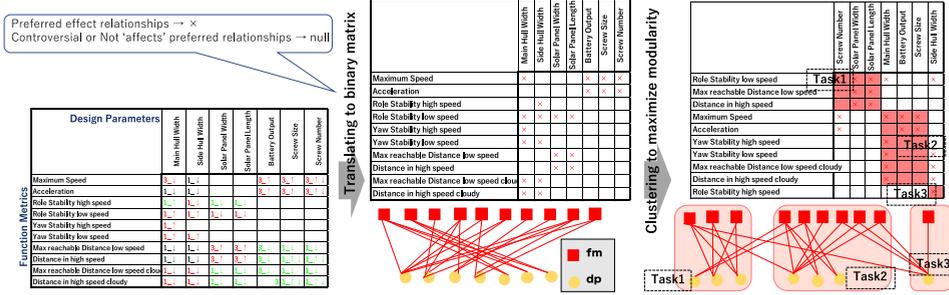


Figure 9. Calculation of task of the best design process about “Appropriateness of dp affiliation”

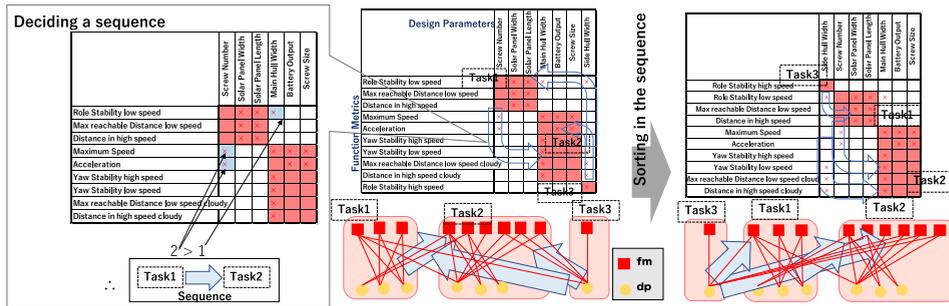


Figure 10. Calculation of sequence of the best design process about “Appropriateness of dp affiliation”

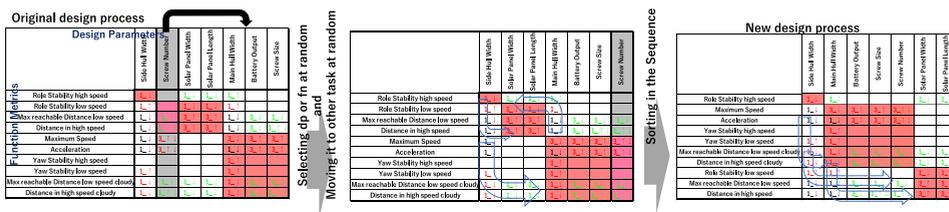


Figure 11. Producing new design process at random

4.3 Examining design processes to determine development scale

An appropriate design process is structured for each development scale that is determined as above. Finally, the designer determines the appropriate development scale by examining the contents of processes among different development scales.

5 Results and discussion

The effectiveness of the proposed method was verified on the Continuously Variable Transmission (CVT) in the following two test cases. The House of Quality of the CVT comprises 102 functional metrics, 12 components, and 162 design parameters.

5.1 Validity of the design process derived by the proposed method

In this test case, for nine functional metrics that have been required to change, 49 design parameters (entailing development scale) were selected, which concomitantly decide

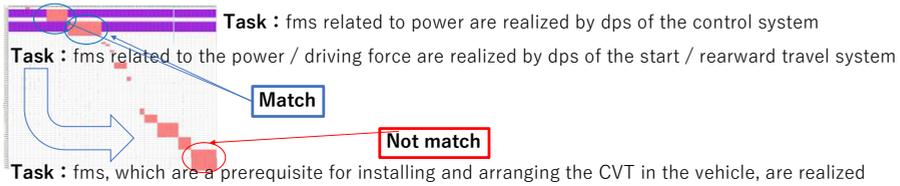


Figure 12. Design process derived by the proposed method

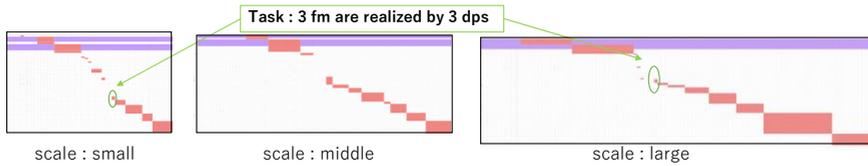


Figure 13. Design processes for three scales

affected functional metrics. These functional metrics and design parameters compose a change implementation set. Next, design processes were explored, and from the obtained quasi-optimum processes, the designer determines the appropriate design process by examining details of the processes. As there were a large number of candidate processes, the designer arbitrarily selected the processes to be examined. As the process structure of the Pareto solutions are quite similar to each other, even the arbitrary selection potentially avoids bias. However, there is room for improvement.

By comparing the design process derived by the proposed method with the actual designer's cognition, the validity of a design process derived by the proposed method was verified. As shown in Figure 12, it was confirmed that several structures were consistent with the designer's cognition while some parts are left inconsistent. Thus, the design processes derived by the proposed method have a certain validity. For example, the process parts consistent with the designer were that the functional metrics related to power are realized by the control system design parameters. The functional metrics related to the driving force are realized by the design parameters of the forward/reverse switching mechanism, and these tasks are performed upstream of the design process.

The part inconsistent with the designer was that functional metrics, which are a prerequisite for installing the CVT in the vehicle, are realized downstream. From the designer's cognition, these functional metrics are designed mostly upstream. In practice, these functional metrics, which are a prerequisite for the installation, are handled differently from other functional metrics. In contrast, they are included in functional metrics without distinguishing them by the property. Therefore, it is considered that this problem can be solved by treating a prerequisite as a different element type and tailoring the proposed method to handle the difference.

5.2 Verification of ability to support optimization of development scale

In this test case, for the nine change-requested functional metrics, three different change implementation sets that respectively comprise 49 (small-scale), 79 (middle-scale) and 120 (large-scale) change design parameters were compared.

By comparing the quasi-optimum design processes among development scales, the ability of the proposed method to support the scale optimization was verified. As shown in Figure 13, the contents of the design processes were different among development scales. For example, in the “small-scale” and “large-scale”, there was a task that three functional metrics trading each other are realized by one design parameter, which is unreasonable. Therefore, it was considered that “medium-scale” is the most appropriate development scale among the three tested scales. As demonstrated above, the development scale affects the design process feasibility. By examining details of the obtained design processes for given development scales, a designer could determine the most reasonable process, thus determining the proper development scale as well.

6 Conclusion

Through the verification above, it was confirmed that an appropriate design process could be derived by optimization in three metrics; whether design parameters are properly affiliated to functional requirements; each task does not handle too many components; and risk of rework is minimized. By incorporating a design policy, which may be affected by the scale of development, it was possible to compute the proper affiliation changes of design parameters. It results in different process structures among development scales. As the process structures are given as a combination of functional measures and design parameters realizing them, it is possible for designers to examine the feasibility of the processes. Therefore, the proposed method can support the development scale optimization.

References

- Alison, J. T. and Papalambros, P. Y., 2007. Optimal Partitioning and Coordination Decisions in System Design Using an Evolutionary Algorithm. Paper presented at 7th World Congress on Structural Multidisciplinary Optimization, Seoul, May 21-25.
- Arai, K., 2016. Method for Appropriate Description of Product Information in Design Planning Stage. The University of Tokyo.
- Carrascosa, M., Eppinger, S. D., and Whitney, D. E., 1998. Using the Design Structure Matrix to Estimate Product Development Time. Paper presented at ASME 1998 Design Engineering Technical Conferences, Atlanta, September 13-16.
- Eckert, C. M., Keller, R., Earl, C., and Clarkson, P. J., 2006. Supporting change processes in design: Complexity, prediction and reliability. *Reliability Engineering and System Safety* 91, 1521-1534.
- Eppinger, S. D., Whitney, D. E., Smith, R. P., and Gebala, D. A., 1994. A Model-Based Method for Organizing Tasks in Product Development. *Research in Engineering Design* 6 (1), 1-13.
- Hamraz, B., Caldwell, N. H. M., and Clarkson, P. J., 2012. A Multidomain Engineering Change Propagation Model to Support Uncertainty Reduction and Risk Management in Design. *Journal of Mechanical Design*, 134, (10).
- Newman, M. E. J., 2004. Fast algorithm for detecting community structure in networks. *Physical review E* 69.
- Oizumi, K. and Aoyama, K., 2014. Improvement Design Planning Method That Identifies Change Parts and Their Directions. *Proceeding of the ASME 2014 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*.
- Contact:** K.Aoyama, School of Engineering, The University of Tokyo, RACE: Research into Artifacts, Center for Engineering, 7-3-1, Hongo, Bunkyo-ku, Japan, +81-3-5841-6504, +81-3-5841-0652, aoyama@race.t.u-tokyo.ac.jp.