A THEORY OF DECOMPOSITION IN SYSTEM ARCHITECTING

Hitoshi Komoto[1] and Tetsuo Tomiyama[2]
(1) National Institute of Advanced Engineering Science and Technology, Japan (2) Delft University of Technology, the Netherlands

ABSTRACT
The divide-and-conquer principle is a technique to obtain solutions for a large-scale problem by dividing it into smaller and manageable subproblems and by integrating these subsolutions. In engineering design, the principle is often used not just as a complexity management method but also as an embodiment method, although its formalization is unclear if not non-existing. This paper attempts to formalize the principle in the context of design of complex multi-disciplinary systems such as mechatronics systems. It proposes a theory of decomposition in conceptual design (system architecting), which extends the decomposition theory in traditional engineering design based on functional decomposition. The theory is applicable to system decomposition processes, in which building blocks necessary for decomposition are not available or must be newly designed during the processes. The theory uses parameter relations governed by physical phenomena realizing functions. A case study of system architecting of a printer is illustrated as a demonstration of the theory.

Keywords: Conceptual Design, System Architecting, Decomposition, Complex Systems, Divide-and-Conquer Principle

1 INTRODUCTION
The divide-and-conquer (D&C) principle is a well-known technique to tackle large-scale problems. It is applied to cut down a big problem into a set of smaller problems of a manageable size that have (known) solutions. Often, the problem can be hierarchically decomposed, resulting into a hierarchy of subproblems. The solution to the original big problem is obtained by adding/integrating subsolutions for those subproblems (Figure 1). In order for this principle to be valid, the principle assumes some properties to the problem [1].

- The hierarchical division should eventually arrive at subproblems that are solvable and have solutions.
- Interactions among the subdivided problems should be minimal and manageable. In other words, a solution for one subproblem should not interfere with other subproblems.
- Solutions for the original should be obtainable by "adding" or "integrating" subsolutions.

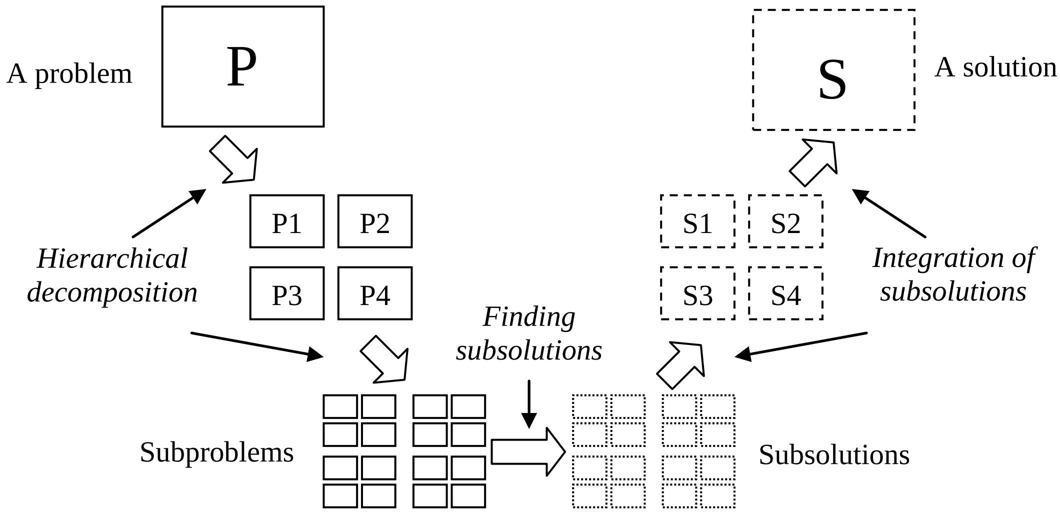

Figure 1. The divide-and-conquer principle

The D&C principle is also used for engineering design [2-5]. For example, the Pahl & Beitz method [3] first builds a hierarchical functional structure and then breaks this function structure into subfunctions that can be embodied with mechanisms and working principles. The final solution is not a simple sum of individual subsolutions that realize subfunctions, though.

Although the D&C principle is perhaps the most powerful method for size and complexity reduction, unfortunately, its procedure is not well-formalized as an algorithm. Because of this, for example, there is no known method to decompose a system uniquely. This means subjective differences among design results even for an identical design task.

The D&C principle becomes more crucial when dealing with complex problems. For instance, the design of modern mechatronics systems, such as hybrid vehicles, medical instruments, and high-end printers, is a complex problem both quantitatively (multi-disciplinarity) and quantitatively (the number of components). Functions of these systems are becoming more and more multi-disciplinary (e.g., mechanical, electrical, electromagnetic, chemical, optical, and thermodynamic) [6]. The number of components is also increasing to exhibit required functions through complex physical phenomena. The general trend of miniaturization is that more components are packed in a smaller space. Although the D&C principle seems to be the sole method that can deal with complex systems, there remains a question if we are still able to use the D&C principle without modification [1].

In order to efficiently and effectively develop complex multi-disciplinary systems including mechatronics systems, the V model of product development (Figure 2) is recommended [7,8]. Within the V model, system architecting is a process for conceptual design of system architecture, in which system architects perform the following tasks:

- Identification of requirements to be translated to system descriptions.
- Hierarchical decomposition of the system to subsystems and eventually to components. The D&C principle is applied here.
- Definition of behaviors and structure of these subsystems and their interfaces.

At the succeeding stages in product development, these decomposed subsystems are designed, integrated, validated, and verified with reference to their definition specified in system architecting [9].

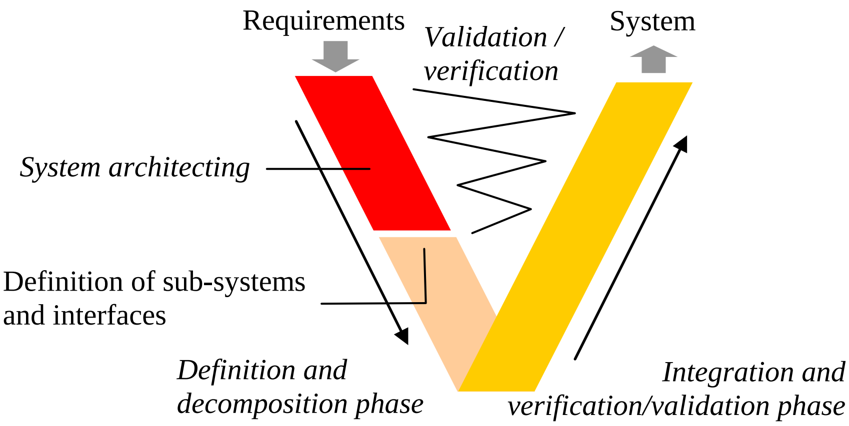

Figure 2. The V-Model of product development

In this paper, we focus on the decomposition process during system architecting. During hierarchical decomposition based on the D&C principle, a system is decomposed into smaller subsystems using building blocks that can construct the subsystems. At a different hierarchy level, a different set of building blocks is used. As the decomposition proceeds, the description level of building blocks becomes more concrete and eventually arrives at, e.g., machine elements, established components and mechanisms in mechanical design. In control design, these fundamental building blocks are block diagrams that represent sensors, actuators, and controllers, while in software design, they can be subroutines and functions that implement fundamental algorithms. In order for designers to be able to construct embodiment with building blocks from functional decomposition of requirements, building blocks must be indexed by functions and connected to behaviors and structure. The past literature show that fundamental building blocks can be defined in various ontologies, including function elements [2], working principles [3], physical features [4], solution patterns [10], and contact and channel model elements [11]. This variety of fundamental building blocks results in different representations of subsystems and their interfaces included in developed design concepts.

When this decomposition method is applied to complex systems architecting, the following insufficiencies can be identified.

- *Systematic design of new building blocks*: Traditionally, building blocks (and their functional descriptions) used for the design of a specific system were developed and refined empirically. When no building blocks satisfy a certain functional requirement, new building blocks needs to be developed. However, no rational and systematic rules have been established so far for the design of new building blocks.
- *System decomposition without building blocks*: During the hierarchical decomposition, it is not always possible to find established building blocks. In such a case, system decomposition should be performed without knowing the detailed description of subsystems included in building blocks. This is true for system architecting of complex systems, because the hierarchy can be very deep and it is often difficult to prepare building blocks at every level of hierarchy.

The objective of this paper is to establish a theory of system decomposition based on the D&C principle in system architecting of complex systems that can deal with the above-mentioned situations. In the D&C principle, functions are hierarchically decomposed by searching for corresponding building blocks from the library using function as an index (Figure 3 (a)). Subsystems and their interfaces are defined by combinations of building blocks. In contrast, the proposed method does not necessarily assume the availability of building blocks. Instead, it assumes that subsystems and their interfaces are gradually created and refined through hierarchical functional decomposition in terms of the relations among system parameters. It is done by adding physical phenomena corresponding to functions and finding possible clustering patterns of the parameters derived from physical phenomena (Figure 3 (b)). Applicability of these different approaches to system decomposition in system architecting depends on the maturity of available design knowledge, which will be discussed in Section 2 in the paper.

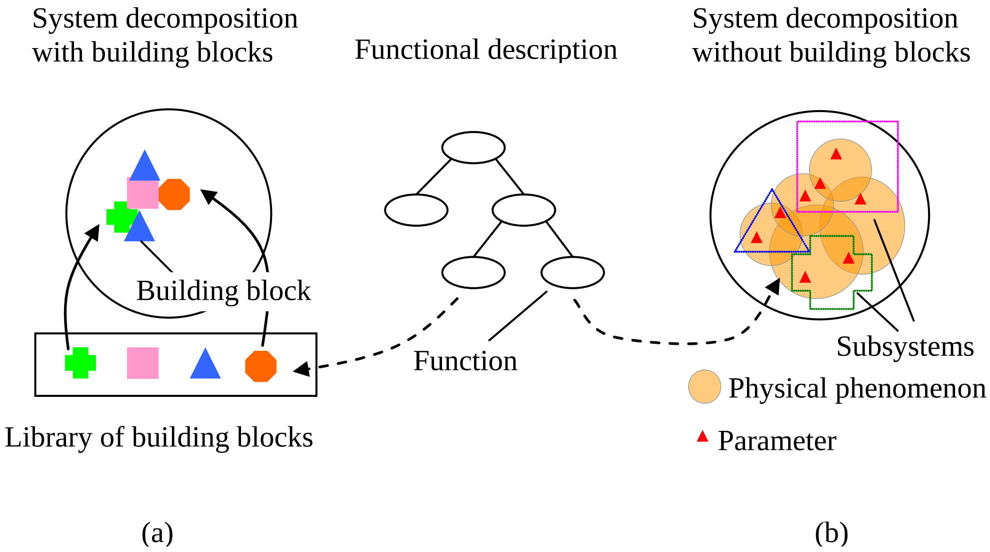

Figure 3. System decomposition with building blocks (a) and without them (b)

The paper is structured as follows. Section 2 classifies system decomposition processes in terms of the maturity of design knowledge in system architecting, which shows the need of the decomposition theory proposed in the paper. Section 3 describes the proposed decomposition theory. Section 4 illustrates a case study of system decomposition process in a printer design case. Section 5 briefly describes the state of art of methods to support system decomposition in engineering design in order to clarify the difference between them and the proposed theory. Section 6 summarizes and concludes the paper.

2 CLASSIFICATION OF SYSTEM DECOMPOSITION PROCESSES

System decomposition processes in system architecting are basically conceptual design of complex systems. Here, the D&C principle plays the central role but actual decomposition takes place in different ways which are summarized in Table 1. This paper mainly focuses on functional decomposition of multi-disciplinary systems.

Table 1. Various Decomposition

Technical decomposition	Spatial, Functional, Behavioral, Process wise, Disciplinary
Organizational decomposition	Departmental, Life Cycle wise

The simplest type of decomposition is "cutting a pie into small pieces" in which the D&C principle just divides a problem spatially into equal pieces. In engineering design methods (e.g., [3]), the D&C principle not only decomposes a problem into smaller subproblems but also finds embodiment (solutions) for these subproblems. A solution to a subproblem is chosen from "building blocks" and a solution to the original problem is obtained by integrating such building blocks. A subproblem can further be decomposed into sub-subproblems. Therefore, the system decomposition within engineering design consists of the following two steps:

- Hierarchical decomposition of the system description in the functional domain (i.e., functional decomposition).
- Search for solutions from building blocks in the physical domain using functions in the functional decomposition as indices.

Thus, the design knowledge in the system decomposition process is divided into both functional and physical domains, i.e., functional decomposition in the functional domain and a set of solutions in the physical domain. The physical domain is viewed from behavioral aspects (e.g., physical phenomena and processes), from a spatial aspect, and from any other relevant aspects.
The system decomposition process can be classified into three phases with respect to the necessity of functional decomposition and the availability of building blocks (Figure 4).

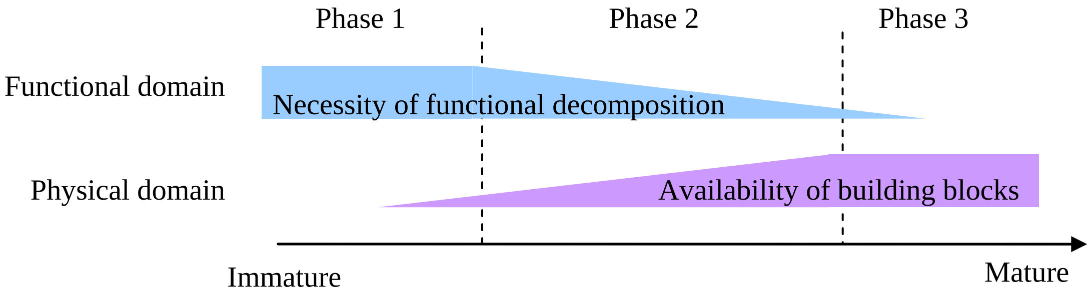

Figure 4. Classification of system decomposition processes

2.1 System decomposition without building blocks (Phase 1)

Hierarchical decomposition of the system descriptions eventually arrives at the physical phenomenon level. Since at this level there is no convenient building blocks, system architects need to refine the functional description to search for primitive elements, such as mass, spring, and dumper in mechanical design, that can describe physical phenomena. These primitive elements are different from building blocks in that they are not initially recognized as solutions to functions, but they are organized so that they become solutions to specific functions. The system description in the physical domain consists of a large number of these primitive elements. Design synthesis by means of combination of these primitive elements is thus exhaustive but is hardly tractable. Since the synthesis process is independent of building blocks or existing functions, system architects can end up with system architecture with unique subsystems and their interfaces. System architects may follow this type of system decomposition, when they deal with completely new customer requirements and/or they have little knowledge about the physical realization of these requirements.

2.2 System decomposition with building blocks (Phase 2)

As system architects learn and design a number of system architecture, they find some patterns in the design information, abstract such patterns as building blocks, and associate them with specific functions. Although functional decomposition is necessary to find building blocks through functions, the functional description is simpler than that used in the system decomposition described in Section 2.1. It is because they can find building blocks associated with the functional decomposition at higher hierarchical levels. Nevertheless, new building blocks have to be designed using building blocks at a lower level, when no building blocks satisfy certain functions. The past literature has focused on this phase of system decomposition more than that in Section 2.1 and proposed various units of building blocks, including function elements [2], working principles [3], physical features [4], solution patterns [10], and contact and channel model elements [11]. These diverse units result in different representations of subsystems and their interfaces included in developed design concepts. It is a future research topic to study differences of these units in the decomposition phase (e.g., in terms of the applicability to system decomposition at different levels of system abstraction).

2.3 System decomposition without explicit use of functional knowledge (Phase 3)

With sufficient refinement of functional description and corresponding building blocks, new designs are functionally described as a combination of existing functions, and physically realized by combinations of existing building blocks corresponding to these functions. When the role of functional description to find new functions becomes less important due to the maturity of design knowledge, the functional description tends to become implicit knowledge of system architects, and they directly explore combinations of building blocks without explicit functional description. In such a case, resulting system architecture is limited to the combination of existing building blocks. Thus, the origin of the complexity of system decomposition in this phase is combinatorial complexity rather than the complexity in finding appropriate functional description.

3. A THEORY OF SYSTEM DECOMPOSITION PROCESS

This section proposes a theory of system decomposition as a design process model in system architecting of complex systems. The theory is aimed to understand the system decomposition classified in Section 2.1 (Phase 1), in which building blocks are not available. The theory supports system decomposition in both functional and physical domains. This means that physical decomposition is available in the middle of functional decomposition, while the traditional system decomposition model in engineering design (e.g., [3]) assumes that physical decomposition becomes available after completing functional decomposition. In short, the proposed theory can explain how subsystems and interfaces are defined and updated in the physical domain through functional decomposition without explicit use of building blocks.

3.1 Elements of the system decomposition model

The proposed theory is basically explained with formal operations on the system description made of the following elements. Figure 5 illustrates relations among these elements.

- *Function model*: A hierarchical description of a system in the functional domain. A function model is developed based on the D&C principle. A function model is used to derive physical phenomena described below. For instance a function *to transfer heat* can be realized by a physical phenomenon *heat transfer*. Functions in the function model at arbitral hierarchical levels can be connected with physical phenomena. By doing so, the description of a system in the physical domain is available in the middle of functional decomposition.
- *Physical phenomena*: Physical phenomena define processes that take place in the system and relations among parameters. For instance, *convective heat transfer* (physical phenomena) can be described by a hear flow from *source* to *object* through *fluid*, and the quantity of heat flow depends on the temperature of *object* and *source*, and the thermal conductivity of *fluid* (see Figure 6). Here, *source*, *object*, and *fluid* are entities defined in the physical phenomenon, but independent of the parameter network described below. Such knowledge about causality and parameter relations can be generic or specific to designed systems and extracted from engineering textbooks and design documentations.
- *Parameter network*: A parameter network is the description of a system including parameters and their relations. These parameters can be defined in both functional and physical domain (such as

function requirements and design parameters [12] or properties and characteristics [10], respectively). The parameter network is updated by adding physical phenomena to it. Figure 5 illustrates addition of physical phenomena to parameter network and resulting update of parameter network. The parameter network defines system decomposition in the physical domain in terms of the relations among entities defined below.

- *Entities*: An entity is characterized by a set of parameters in the parameter network of the system. Relations between two entities are defined by relations among the parameters of these entities. Entities and their relations are regarded as subsystems and their interfaces. The parameter network in the middle of Figure 5, for instance, represents two entities, whose interface is defined by two pairs of parameter relations. The definition of entities and their relations are also updated by adding physical phenomena to the parameter network. New entities are defined when there are parameters in the parameter network, which are not used to characterize other entities.

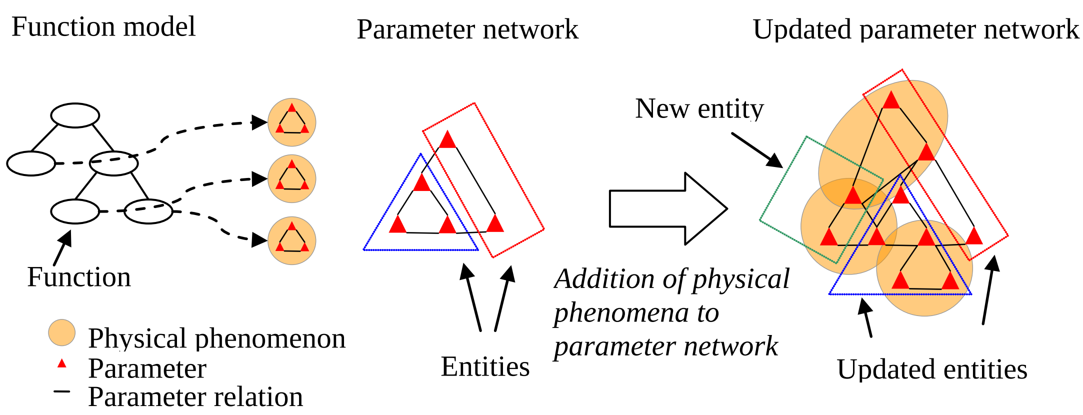

Figure 5. Elements of the system description in the proposed decomposition process model

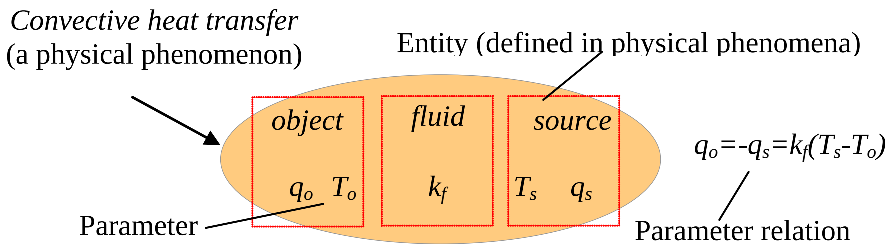

Figure 6. An example of physical phenomenon

3.2. The proposed system decomposition procedure

The proposed theory assumes that system decomposition is a process to define subsystems and their interfaces based on iterative of four steps: Selection of parameters, search and selection of physical phenomena, generation and selection of physical decomposition candidates, and update of parameter network (Figure 7).

1. *Selection of parameters*: Parameters in the parameter network are selected. At the beginning of the decomposition process, initial parameters are introduced based on the requirements in the functional domain (e.g., quality and productivity). Functional decomposition is a means to search for these initial parameters. In the middle of iteration of the decomposition process, parameters are derived from existing physical phenomena in the system, whose value should be changed.
2. *Search and selection of physical phenomena*: Physical phenomena that will influence the selected parameters are searched out of the knowledge of system architects, which can be stored in the respective knowledge base. After that, at least one physical phenomenon is selected with respect to each selected parameter in order to realize the desired value changes.
3. *Generation and selection of physical decomposition candidates*: There are multiple patterns in adding parameters and their relations included in the selected physical phenomena to the

parameter network. These patterns are indeed clustering patterns of new and existing parameters with respect to entities. New entities can be added in this step. Different clustering results in a different physical decomposition of the system. The number of feasible system decompositions (i.e., physical decomposition candidates) can be decreased by adding constraints in the search process. For instance, separation of entities in the definition of physical phenomena (e.g., in Figure 6, *object*, *source*, and *fluid* should be separated entities) is used to limit the number of decomposition candidates [13].

4. *Update of parameter network*: After selecting one of the physical decomposition candidates, the parameter network of the system is updated. The process terminates when the updated parameter network is validated by system architects. Otherwise, the process returns to the first step for the selection of a new set of parameters in the updated parameter network.

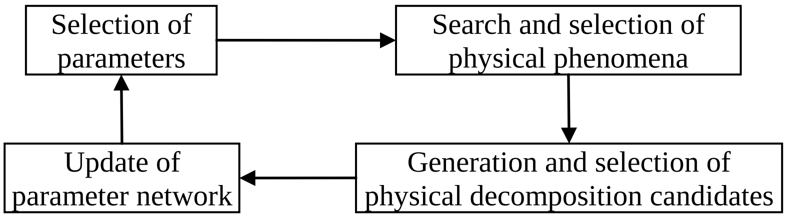

Figure 7. The proposed system decomposition procedure

4 A CASE STUDY OF SYSTEM DECOMPOSITION PROCESS

A case study of multi-disciplinary system decomposition following the proposed theory was conducted in system architecting of a printer, which is a typical complex, multi-disciplinary system [13]. The case study was conducted on a computational tool for system architecting [14]. The tool supports system decomposition processes following the proposed model by automating the search of physical phenomena using the knowledge base and the systematic generation of physical decomposition candidates. The tool was developed as an extension of the Knowledge Intensive Engineering Framework (KIEF) [15], which supports suggestions of physical phenomena based on parameter relations with Qualitative Process Abduction System (QPAS) [16]. The implementation and the detail of system have been described in [13].

The system decomposition in the case study was based on the realization of three major function requirements of a printer in Figure 8. These requirements were used to define the initial parameters of the printer, whose values are changed by physical phenomena to meet the requirements. For instance, the productivity depends on the velocity of sheets. The initial parameter network was manually developed considering such parameter relations (Figure 9). It consists of entities containing parameters and physical phenomena containing the indices of parameter relations, which are separately shown in the bottom frame. The tool currently supports arithmetic equations and qualitative proportional and differential relations as types of parameter relations.

After the initialization, three parameters corresponding to the function requirements in Figure 8 were used to instantiate the physical phenomena in Figure 8, which results in an extension of the initial parameter network. Here, the background knowledge of designers gave freedom in selecting physical phenomena among those suggested by the tool (by giving the parameters as triggers). For instance, other thermal phenomena can be selected in order to change the temperature of paper. After selecting four physical phenomena, the designer selected one of the physical decompositions suggested by the tool, which were derived from possible unification of the knowledge about parameter relations defined in these physical phenomena. In this example, 151 candidates have been systematically generated. Figure 10 shows one of the suggested physical decompositions. In Figure 10, parameters corresponding to the function requirements and the instantiated physical phenomena are labeled following Figure 8. In this step, four system elements were introduced and their parameters related with the instantiated physical phenomena have been identified. One of them (new entity A in Figure 10) plays a role of both the sheet transfer belt and the conductive heater and contains both kinetic and thermal parameters. Printers including such a subsystem are available in current market. The detail of the printer should be further designed considering relations among these parameters.

By conducting such case studies multiple times, the authors expect that designers can find typical physical decompositions (i.e., partial parameter network), which are frequently selected among systematically generated physical decompositions. Such typical physical decompositions are regarded as new building blocks in the succeeding product development. The proposed decomposition method can be useful to find new building blocks based on the knowledge about physical phenomena (how parameters are related) rather than the knowledge about entities (which parameters are contained).

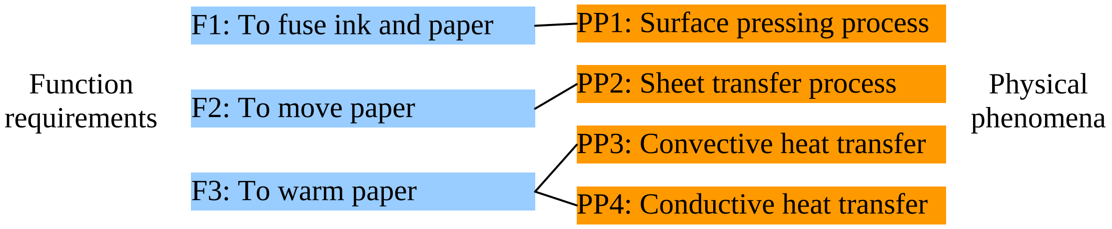

Figure 8. Functions and corresponding physical phenomena

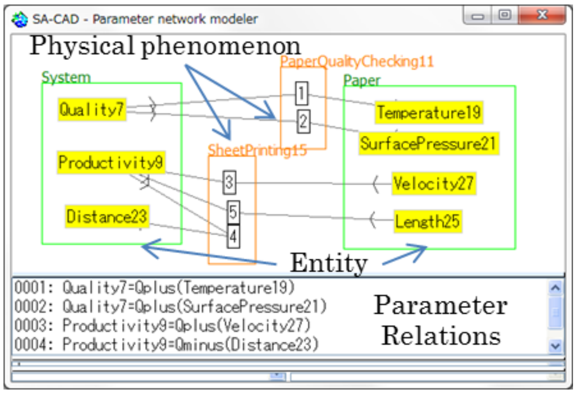

Figure 9. Initial parameter network before function realization

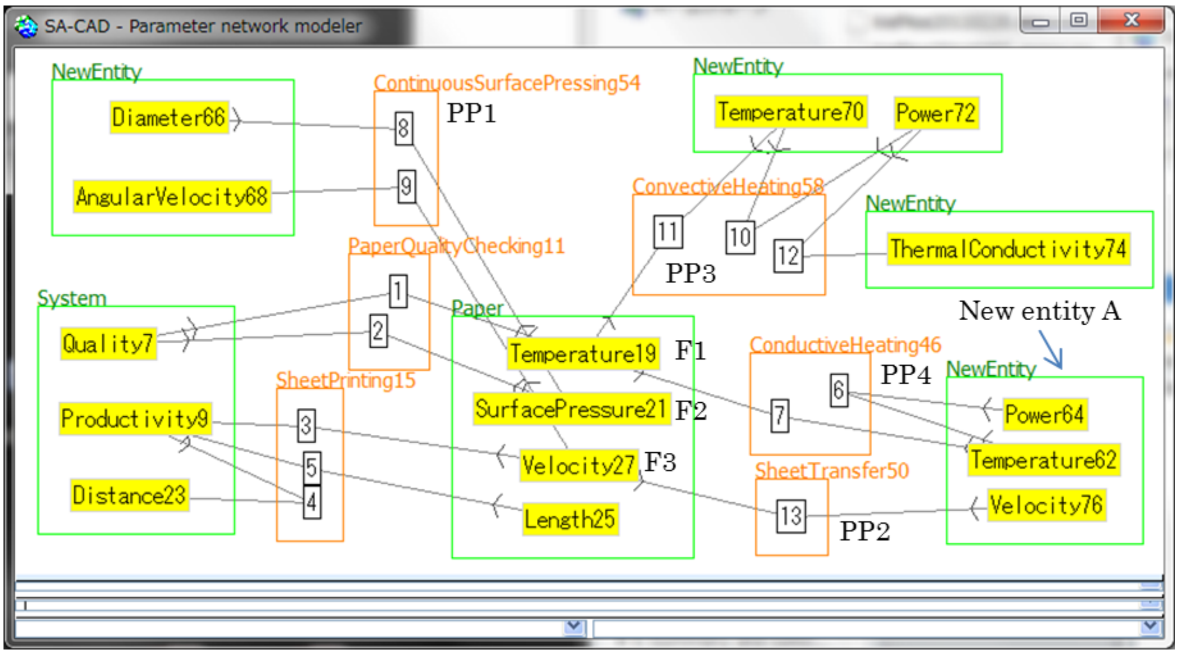

Figure 10. A physical decomposition after function realization

5 RELATED WORK

In the Axiomatic Design [12, 17], a system is described with four different aspects; customer attributes (CA), function requirements (FR), design parameters (DP), process variables (PV). The system descriptions from these four aspects are hierarchically decomposed in a zigzagging manner. However, concrete decomposition operations on the system descriptions have not been explained (e.g., introduction of new function requirements and design parameters).

Systematic engineering design approach [2, 3] employs physical phenomena as the most fundamental building blocks, which are followed by the proposed decomposition method. The proposed decomposition method further considers parameters and their relations included in each physical phenomenon in order to suggest alternative system decompositions in terms of parameters.

The SPALTEN method deals with decomposition of design problems into smaller problems in the early stage of product development [5]. It classifies the types of problem solving styles encountered in problem decomposition processes. Although the focus of the method is more on the organizational decomposition of design problems than decomposition of system descriptions such as functional model and building blocks, the method can be combined with the contact and channel model (C&CM) [11], which is the representation of building blocks developed by the same research group.

Modularization methods have been used to define physical decomposition of a system by clustering components of the system [18, 19], which leads to such concepts as product architecture [20]. These methods assume that system components are well-defined with respect to the objectives of modularization. Although these methods do not assume that necessity of functional description corresponding to components and computed modules, some studies use the functional description of system for modularization (e.g., [19]).

After identifying the parameters of a product though the proposed decomposition method, reconfiguration of relations among the parameters is crucial (i.e., synthesis). Such reconfiguration is performed considering various design criteria (e.g., functional redundancy [4]) and supported by the other design methods reviewed in this section. For instance, the Axiomatic Design [12, 17] is used to analyze relations between function requirements and design parameters.

6 SUMMARY AND CONCLUSIONS

This study has proposed a theory for decomposition in system architecting of complex systems. The theory has extended the decomposition theory based on functional decomposition with the D&C principle followed by combination of established building blocks, which has been traditionally accepted at the conceptual design stage in engineering design. The proposed theory has explained how functional decomposition proceeds with the definition of subsystems and their interfaces in terms of parameter relations. The theory has been implemented with a computational tool for system architecting and a system decomposition case of a printer has been described for the demonstration. The proposed decomposition method can be useful to find new building blocks based on the knowledge about physical phenomena (how parameters are related) rather than the knowledge about entities (which parameters are contained).

We conclude that the proposed theory is crucial in developing design methods, computational tools, and knowledge base to support system decomposition in design of complex systems, in which their behaviors are multi-disciplinary and new designs of building blocks are still required. Future work includes the development of new building blocks based on the proposed method, and the study of complexity in hierarchical system design based on the theory with quantitative evaluation measures.

ACKNOWLEDGEMENTS

This work has been carried out as part of the Octopus project with Océ-Technologies B.V. under the responsibility of the Embedded Systems Institute in Eindhoven, The Netherlands, partially supported by the Netherlands Ministry of Economic Affairs, Agriculture and Innovation under the Bsik program.

REFERENCES

[1] Chmarra M.K., Alvarez Cabrera A.A., van Beek T., D’Amelio V., Erden M.S. and Tomiyama T., Revisiting the Divide and Conquer Strategy to Deal with Complexity in Product Design. 2008 IEEE/ASME International Conference on Mechatronic and Embedded Systems and Application

[2] Rodenaker W.G., *Methodisches Konstruieren*. 4th edn. 1991 (Springer, Berlin).

[3] Pahl G., Beitz W., Feldhusen J. and Grote K.H., *Engineering Design-A Systematic Approach*. Wallace. K. and Blessing, L. (Trans. and Eds.), 3rd edn., 2007, (Springer, Berlin).

[4] Umeda Y., Ishii M., Yoshioka M. and Tomiyama T., Supporting conceptual design based on the function-behavior-state modeler. Artificial Intelligence for Engineering Design, Analysis and Manufacturing. 1996, 10(4), 275-288.

[5] Albers A., Burkardt N., Meboldt M. and Saak M., SPALTEN problem solving methodology in the product development. *In International Conference on Engineering Design*, ICED05, 2005, Melbourne, Australia.

[6] Tomiyama T., D'Amelio V., Urbanic J. and ElMaraghy W. Complexity of Multi-Disciplinary Design. Annals of the CIRP, 2007, 56(1), 185-188.
[7] Forsberg K., Mooz H. The Relationship of Systems Engineering to the Project Cycle. Engineering Management Journal, 1992, 4(3), 36-43.
[8] VDI 2206, *Design methodology for mechatronic systems*. 2004.
[9] Wynn D. and Clarkson J. Model of Designing. In: Clarkson J. and Eckert C. (Eds.) *Design process improvement - A review of current practice*. 2005. Springer, London.
[10] Weber C., Looking at 'DFX' and 'Product Maturity' from the Perspective of a New Approach to Modelling Product and Product Development Processes, In: F.-L. Krause, Editor, *The Future of Product Development*, 2007, pp. 85–104 (Springer, Berlin).
[11] Albers A. and Alink T., Support of Design Engineering Activity for a Systematic Improvement of Products. In: F.-L. Krause, Editor, *The Future of Product Development*, 2007, pp.105–114 (Springer, Berlin).
[12] Suh N.P., Complexity in Engineering. Annals of the CIRP, 2005, 54(2), 46-63.
[13] Komoto H. and Tomiyama T., Multi-disciplinary system decomposition of complex mechatronics systems. Annals of the CIRP, 2011, 60(1) (in press).
[14] Komoto H. and Tomiyama T., A system architecting tool for mechatronic systems design. Annals of the CIRP, 2010, 59(1), 171-174.
[15] Yoshioka M., Umeda Y., Takeda H., Shimomura Y., Nomaguchi Y. and Tomiyama T., Physical concept ontology for the knowledge intensive engineering framework, Advanced Engineering Informatics, 2004, 18, 95-113.
[16] Ishii M., Tomiyama T., Yoshikawa H., A Synthetic Reasoning Method for Conceptual Design. Proceedings of the IFIP TC5/WG5.3 Conference on Towards World Class Manufacturing 1993, 1993, pp. 57-70.
[17] Lu S.C.Y. and Suh N.P., Complexity in design of technical systems. Annals of the CIRP, 2009, 58(2), 157-160.
[18] Erixon G., von Yxkull A. and Arnstrom A., Modularity – the Basis for Product and Factory Reengineering. Annals of the CIRP, 1996, 45(1), 1-6.
[19] van Beek T., Erden M.S. and Tomiyama T., Modular design of mechatronic systems with function modeling. Mechatronics, 2010, 20, 850-863.
[20] Ulrich. K., 1995, The role of product architecture in the manufacturing firm, Research Policy, 24/3:419-440.

Contact:
H. Komoto
National Institute of Advanced Industrial Science and Technology (AIST)
Advanced Manufacturing Research Institute, System Functional Design Group
Namiki 1-2-1
305-8564, Tsukuba, Ibaraki
Japan
Phone: +81 (0) 29-861-4139
Fax: +81 (0) 29-861-7201
E-mail Address: h.komoto@aist.go.jp

H. Komoto is a research scientist at Advanced Manufacturing Research Institute, National Institute of Advanced Industrial Science and Technology, Tsukuba, Japan. He received his PhD in 2009 in the field of computer-aided product-service systems design, at Delft University of Technology Delft, the Netherlands. His research interests include research and development of intelligent CAD systems.

T. Tomiyama is a professor at Intelligent Mechanical Systems Group, Department of BioMechanical Engineering, Faculty of Mechanical, Maritime and Materials Engineering, at Delft University of Technology, Delft, the Netherlands. His research areas include design theory, design methodology, knowledge intensive engineering, service engineering, life cycle engineering, and maintenance.