DEALING WITH COMPLEXITY IN DESIGN: A KNOWLEDGE POINT OF VIEW

T. Tomiyama

Delft University of Technology
Faculty of Mechanical, Maritime and Materials Engineering
e-mail: t.tomiyama@tudelft.nl

Keywords: design knowledge, complexity, axiomatic design, knowledge structuring

***Abstract**: This paper tries to establish a scientific understanding of complexity of multi-disciplinary product development from the viewpoint of knowledge structure. It first discusses why such multi-disciplinary product development is complex and why complex problems are difficult to solve. It then analyzes the source of complexity from the viewpoint of knowledge structure and identifies "complexity by design" and "intrinsic complexity of multi-disciplinarity" when multiple theories are involved during design. Examples illustrate how the idea of knowledge structure based complexity can explain why multi-disciplinary design problems often turn out to be ill-structured.*

1. INTRODUCTION

The complexity of products is increasing due to technological advances that reflect customers' ever increasing requirements and due to the multi-disciplinarity of products (such as mechatronics products). Accordingly, the development team needs to grow and the number of stakeholders also needs to increase. This means that the complexity of product development processes also significantly increased. While these two types of complexity, *viz*., product complexity and process complexity, steadily increase, the pressure on product development processes is also increasing. Beyond traditional targets such as functions, cost, and quality, speed (time-to-market and time-to-delivery), sustainability, and product-service systems are becoming relevant and even crucial for competitive product development.

This paper is an attempt to understand these two types of complexity from the viewpoint of knowledge structure. In particular, we focus on complexity of multi-disciplinary design and propose methods to tackle them.

In Chapter 2, we distinguish the differences between difficult mono-disciplinary design problems and complex multi-disciplinary design problems on which the paper focuses.

Well-known techniques to deal such complexities are Design Structure Matrix (DSM) [1] and Suh's Axiomatic Design (AD) [2]. Suh extended AD to deal with complexity in design recently [3]. Chapter 3 briefly reviews DSM and AD.

While the difficulty of solving difficult mono-disciplinary problems reside mostly within the product itself (such as the complexity of the governing equations and the number of components), the difficulty in solving complex multi-disciplinary design problems results from the involvement of multiple theories. To understand the complexity of multi-disciplinary design problems, Chapter 4 first discusses relationships among different theories. A knowledge theory offers solutions in a single problem discipline (domain). While the difficulty of solving mono-disciplinary problems entirely depends on the complexity of the theory itself, the difficulty of finding solutions for multi-disciplinary problems arises from interferences among the involved theories. Based on these interference mechanisms, the chapter identifies two different types of interactions among theories, i.e., "complexity by design" and "intrinsic complexity of multi-disciplinarity."

Chapter 5 illustrates the differences of "complexity by design" and "intrinsic complexity of multi-disciplinarity" through examples of multi-disciplinary problem. These problems are complex, ill-structured (as opposed to well-structured), and difficult to solve, because the set of theories used to describe the problems can have interactions with each other and because these interactions are sometime unidentified or hidden until they are discovered at a later stage of product development.

Chapter 6 presents some methods to tackle the problems associated with multi-disciplinary design problems. One is to establish a theory that can tackle these multi-disciplinary problems as a whole, which is nothing but a scientific activity. If such a theory is difficult to establish, we may try to "integrate" or "fuse" existing theories to arrive at such a unified theory. To do so, ontology across different theories plays a crucial role. One another way is to develop a "design interference detector" that can be based on qualitative physics.

Chapter 7 concludes the paper.

2. ISSUES OF PRODUCT DEVELOPMENT

2.1. Expanding concepts and boundaries of products

Modern products are becoming extremely complex primarily due to advances of technology that drive toward further miniaturization, high quality, more functionality, and yet cheaper prices. Examples of such multi-disciplinary products are opto-mechatronics products, such as digital cameras, integrating not just mechanical engineering, electronics, and control engineering, but also optics and sophisticated software systems [4].

Multi-disciplinarity comes also from increasing concerns about the global environment, which focus on product life cycle issues. These include product life cycle management, end-of-life treatments, and different forms of added value generation such as service and product-service systems [5].

2.2. Difficult mono-disciplinary problems

There could be two typical examples of difficult mono-disciplinary problems.
For instance, the number of components in a modern product has significantly increased and each of these components has become complex. A simple example is the shape of a component. A geometrically complicated shape has more design parameters to be analyzed and determined than a simple shape. Consequently, the design of such a component can be difficult in spite of being mono-disciplinary, because the number of parameters is big, which often results in computational complexity problems as well.

The other case can be observed, when involved phenomena are governed by "difficult" governing equations. Often these equations do not have analytical solutions, or numerical methods can be prohibitively expensive or even do not exist.

2.3. Complex multi-disciplinary problems

The multi-disciplinary nature of modern products exhibits another type of difficulty in product development, i.e., complex multi-disciplinarity, which is the focus of this paper. When multiple domains are involved in a design problem, unless there is a uniform theory that can attack the problem as a whole, we will be forced to use a set of theories each of which is valid only in one domain. Although these theories are in principle independent from each other, they can have (sometimes even) intrinsic interactions with each other for a variety of reasons. In Chapter 4, we will analyze these reasons.

In solving such a multi-disciplinary problem, these interactions among theories can cause problems. Of course, when an influence from one theory to another is very small or minor, we can safely neglect them. However, this depends very much on actual design. In other words, interactions that could have been neglected safely (or perhaps even unidentified) in one design case can cause significant problems in other design cases (such as contradicting parameter values).

When these interactions are well-known before the design takes place, we can have counter-measures. However, if unknown or unidentified, at a later stage the designer will be surprised by those hidden or neglected interactions between theories. Once this type of interactions is detected, the designer will be forced to perform unwanted design changes or even to re-design from the beginning. These will delay the project significantly and thus have strong cost implications.

2.4. Complex multi-stakeholder product development

Since the product life cycle issues are not minor any more, product development inevitably involves multiple stakeholders. These stakeholders often come up with design issues that contradict with each other, resulting in complexity product development processes.

As the products as well as product development processes became more and more complex, it was essential to have better collaboration among different product development stakeholders. This is where CSCW (Computer Supported Collaborative Work) comes in [6, 7].

3. COMPLEXITY MANAGEMENT METHODS

There are some methods proposed to tackle the complexity management problems in design. One is Design Structure Method (DSM) [1] and the other is Axiom 1 of Suh's Axiomatic Design (AD) [2].

3.1. Design structure matrix

In DSM, dependencies among elements of a design process are represented and optimized by partitioning and clustering operations of the DSM. A DSM represents N^2 relationships among such N elements as components, stakeholders (or organization involved), design subprocesses (activities), or design parameters [1].

For instance, Table 1 depicts a DSM among five components (processes, parameters, etc.) and should read that, element A determines nothing but is dependent on elements B and C, element B determines elements A and C, and it depends on element C. This DSM obviously has two element clusters, *viz.*, one including A, B, and C, and the other C, D, and E. This implies the whole process can be partitioned into two clusters. A DSM, which is as close as diagonal, represents a simpler design process in which design parameters can be determined sequentially. There are several known algorithms to obtain such a rearranged DSM from an original DSM [1].

Table 1: *A typical DSM*

	A	B	C	D	E
A					
B	x		x		
C	x	x		x	x
D					x
E				x	

3.2. Axiomatic design

With his axiomatic design method, Suh has represented relationships between functional requirements (FRs) and design parameters (DPs) in the framework of design matrix (DM) [2] as follows.

$$\{FR\} = [D]\{DP\} \tag{1}$$

If there are two FRs and two DPs, Equation 1 can have the following three situations in which x signifies some sort of relationships or influences and 0 denotes no relationship. Note that we here consider that upper triangular case is similar to the lower triangular case of Equation 3.

$$\begin{Bmatrix} FR1 \\ FR2 \end{Bmatrix} = \begin{bmatrix} x & x \\ x & x \end{bmatrix} \begin{Bmatrix} DP1 \\ DP2 \end{Bmatrix} \tag{2}$$

$$\begin{Bmatrix} FR1 \\ FR2 \end{Bmatrix} = \begin{bmatrix} x & 0 \\ x & x \end{bmatrix} \begin{Bmatrix} DP1 \\ DP2 \end{Bmatrix} \tag{3}$$

$$\begin{Bmatrix} FR1 \\ FR2 \end{Bmatrix} = \begin{bmatrix} x & 0 \\ 0 & x \end{bmatrix} \begin{Bmatrix} DP1 \\ DP2 \end{Bmatrix} \tag{4}$$

Equation 2 signifies a situation called "coupled design" in which the two functional requirements cannot be independently fulfilled by the two design parameters due to interferences between $DP1$ and $DP2$. Equations 3 and 4 are better situations in which the two functional requirements can be independently fulfilled by the two design parameters. The former is called "decoupled design" in which $DP1$ must be first determined and then $DP2$ can be obtained, while the latter "uncoupled design" in which $DP1$ and $DP2$ can be independently determined. Suh's Axiom 1 (Independence Axiom) states "Maintain independence of functional requirements," (i.e., ideally uncoupled design or at least decouple design) at any time during the functional decomposition process.

Note that for the sake of simplicity, we only discuss cases in which the numbers of *FP* and *DP* are the same and the design matrix is square. If not, it is a redundant or impossible design case.

More recently, Suh introduced complexity as "the measure of uncertainty in achieving the functional requirements of a system within their specified design range" and classified four different types of complexity, *viz.*, time-independent real complexity, time-independent imaginary complexity, time-dependent combinatorial complexity, and time-dependent periodic complexity [3]. Time-independent real complexity is the situation in which FRs are not always satisfied. Time-independent imaginary complexity reveals when there are many FRs and the design is a decoupled design. It is called imaginary, because this corresponds to a situation in which different orders in solving the design matrix result in different difficulties in solving the design.

Suh identified two types of time-dependent complexity, i.e., time-dependent combinatorial complexity and time-dependent periodic complexity. These arise "when the system range moves as a function of time." According to Suh, "time-independent imaginary complexity and time dependent periodic complexity can happen only when many FRs must be satisfied at the same time, whereas time-independent real complexity and time-dependent combinatorial complexity can exist regardless of the number of FRs that must be satisfied at the same time."

He further suggested four strategies to tackle these complexities: (1) minimize the number of FRs, (2) eliminate time-independent real complexity, (3) eliminate time-independent imaginary complexity, and (4) transform a system with time-dependent combinatorial complexity into a system with time-dependent periodic complexity.

This paper deals with time-independent real complexity in Suh's terminology among other things.

3.3. DSM and AD

While DSM (in particular, parameter-based DSM) tries to optimize design processes based on dependencies among design parameters (therefore purely from the viewpoint of design process knowledge), AD aims at "good design" with Axiom 1 (functional dependence) that recommends to re-organize a design problem based on dependencies between function requirements and design parameters from the viewpoint of design object knowledge.

As we can easily imagine, a completely or almost "diagonal" DSM after clustering and partitioning corresponds to an uncoupled or decoupled DM, respectively. Dong and Whitney [8] clarified the mathematical relationships between DSM and AD. They stated that "when AD fails to provide uncoupled or decoupled solutions that are feasible in the business context and design iterations are inevitable, DSM can be used subsequently to apply management leadership on the design process."

3.4. Complexity of design

Both in DSM and AD, it must be noted that when non-diagonal elements are significant or non-zero, design is considered complex. On the contrary, if a design matrix (in case of AD) has only diagonal elements, the design is well-structured and can be decomposed into independent subprocesses. This is obvious when we have two independent two design cases (i.e., Equations 5 and 6), we obtain an uncoupled design as a result of combining these two (Equation 7, which is the same as Equation 4).

$$\{FR1\} = [X]\{DP1\} \tag{5}$$

$$\{FR1\} = [X]\{DP1\} \tag{6}$$

$$\begin{Bmatrix} FR1 \\ FR2 \end{Bmatrix} = \begin{bmatrix} x & 0 \\ 0 & x \end{bmatrix} \begin{Bmatrix} DP1 \\ DP2 \end{Bmatrix} \tag{7}$$

However, if FRs and/or DPs in two design cases have unidentified relationships among them, they will constitute a decoupled or coupled design case.

4. RELATIONSHIPS BETWEEN THEORIES: KNOWLEDGE STRUCTURE

The focus of this paper is the complexity resulting from the involvement of multi-disciplines in product development. Earlier, Meijer, *et al*. [9] pointed that reducing complexity of design processes based on AD's Axiom 1 would succeed when there is intrinsic nature of well-structured design object knowledge, which guarantees the functional independence. This paper is a follow-up of this discussion.

To understand the complexity resulting from the involvement of multiple disciplines, in this section we examine the structure of knowledge represented by relationships among theories. Here, a theory denotes a self-contained axiomatic system consisting of a number of axioms and concepts (or terminology). The relationships among these concepts are defined by axioms and theorems derived from axioms. A theory is a mono-discipline system by definition and can represent only a piece of knowledge about one discipline.

The discussion about difficult mono-disciplinary situations in Section 2.2 refers to the complexity of individual theories. This is caused by, for example, the number of parameters and complicated relationships among them within a domain. In contrast, we discuss here complex multi-disciplinary situations.

Two theories can have different relationships with each other. In the following, we will examine these situations.

4.1. Two theories are independent

Two theories are in principle independent, because if not independent that means the axioms of the two theories are interfering each other. However, this is only syntactically true and two theories can have different types of semantic relationships.

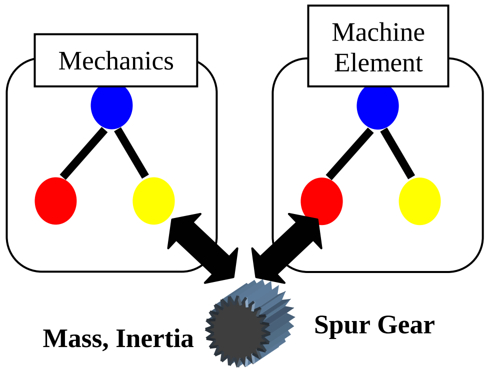

Fig. 1: *Two theories are independent, but share one physical object by designation in different contexts*

The first case is a situation in which two theories can have relationships because of designation (or instantiation) of entity. Let us discuss this in the following example.

Figure 1 illustrates such a situation. The theory of mechanics and the theory about functions of machine elements are independent. Newtonian mechanics forms a theory that begins with the law of universal gravity and three laws of motion. Knowledge of a class of mechanical component (such as spur gear) also forms a theory. It may contain such statements about its functions and attributes as "a spur gear pair transforms mechanical rotational power", "the moderation ratio is determined by the ratio between the numbers of gear teeth of the pair", etc.

During a design process of gear pairs, the designer has to consider the inertia of a particular gear pair instance, because he/she designated (or instantiated) a gear pair as a means to transform mechanical rotational power. In other words, at the moment of the designation of a particular spur gear these two theories share a common concept. This creates a connection between these two theories and adds new complexity in the whole knowledge system because of the instantiation of this concept. We call this complexity "complexity by design," because it happened due to the choice of a particular instance. If we chose a hydraulic solution to transform energy, different theories would be incorporated.

4.2. Two theories share a common concept

Although independent, two theories can contain the same concept as is the case of Figure 2. These theories are closely related knowledge systems intrinsically. An example is motor design in which dynamics and electromagnetics can be related with force as a sole interface parameter. This is the case often observed in multi-disciplinary product development. This situation is knowledge integration [4, 10, 11]. The complexity found for this case results from the fact that two theories share common concepts (rather than instances) and can be

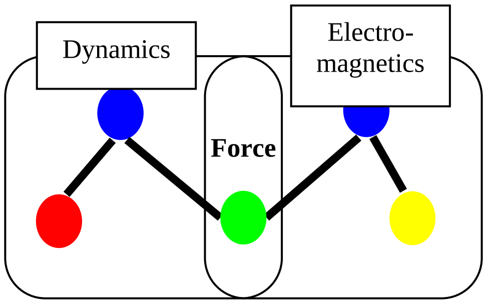

Fig. 2: *Two theories share a physical concept: Knowledge Integration*

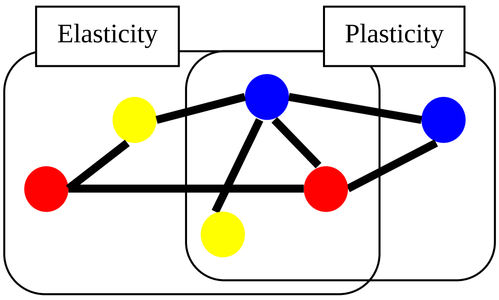

Fig. 3: *A set of concepts applied to different situations resulting in different (independent) theories*

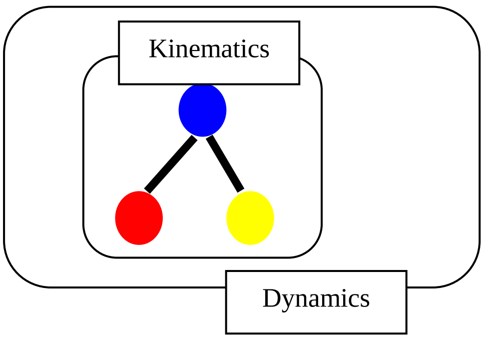

Fig. 4: *One theory subsumes another theory*

called “intrinsic complexity of multi-disciplinarity.”

Figures 3 and 4 are special cases of Figure 2. In Figure 3 elasticity theory and plasticity theory share some portion, but they differ in their applicable ranges. Elasticity theory is valid under a condition that the deformation is relatively small and there is a linear relationship between load and elongation, whereas plasticity theory is applicable when such a linear relationship does not exist.

Figure 4 depicts a situation in which one theory subsumes another. Most of the concepts of kinematics are part of dynamics.

4. 3. Knowledge fusion

Knowledge fusion creates an inter-disciplinary new knowledge system that can be operated as a whole over fused domains (Figure 5). While knowledge

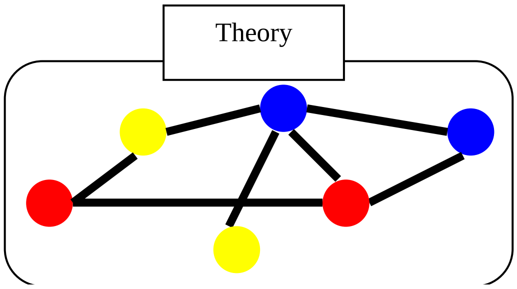

Fig. 5: *An inter-disciplinary fused theory: Knowledge fusion*

integration is still a collection of independent knowledge systems with clearly defined interfaces that describe common concepts among those integrated knowledge systems, knowledge fusion is an ideal situation in which these systems have been totally fused to create a new knowledge system.

However, knowledge fusion is not automatically possible and is yet a research issue. Mechatronics is now considered to form an integrated knowledge system, but still we can see distinctions among mechanical technology, control technology, electronics, sensor technology, and software technology. Although details of knowledge fusion are yet research issues, we may point out that knowledge structuring efforts considered useful [9].

4.4. Summary

In summary, we identified two reasons for two theories to arrive at complex situations. One is “complexity by design” in which independent theories become relevant due to the instantiation of a physical object common to these theories. The other is “intrinsic complexity of multi-disciplinarity” in which two theories get integrated through one common concept as an interface between them.

It is important to notice that “intrinsic complexity of multi-disciplinarity” happens regardless of the choice of design object, whereas “complexity by design” happens only when designed in that way. In other words, “complexity by design” may have a way to avoid; in contrast, “intrinsic complexity of multi-disciplinarity” cannot be avoided because physics dictates in that way.

5. DESIGN IN A MULTI-DISCIPLINARY DOMAIN

5.1. Well-structured problems

It is often said that design involves ill-structured problems. This is obvious in multi-disciplinary design cases [11]. To understand ill-structured nature of design in the context of knowledge structure, we first try to understand well-structured problems that can be solved by the "Divide and conquer" strategy as follows (Figure 6).

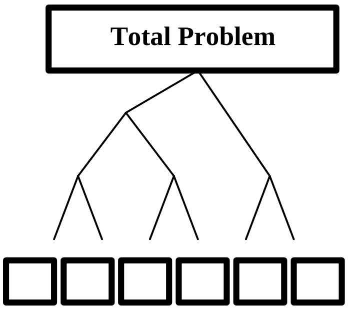

Fig. 6: *Divide-and-conquer approach to a well-structured problem*

In case of a well-structured problem, the problem can be divided into a number of subproblems, aspect-wise, component-wise, or process-wise. Each of these subproblems has a clear system boundary, governing principles, and conditions (such as requirements). These subproblems have minimum interactions with each other, so that ideally they can easily be separated from each other and solved independently. This situation corresponds to uncoupled design in AD, whereas decoupled design in AD can be still well-structured but difficult (in the sense of difficulty but sequentially solvable) because of interactions among parameters.

A subproblem can be solved, sometimes by analytical or numerical computations, or sometime by database lookup. Even if solution methods are not known, at least we know how to generate solution candidates in the given situation (problem space) and to test those candidates against requirements (generate and test method).

Finally, a solution for the original problem can be synthesized by combining solutions for subproblems.

Often, solving such a well-structured problem boils down to a search problem within a problem space. In design context, when we know building blocks within the problem space, design boils down to select such building blocks and to combine them. The system boundary is clear and solutions will be found in this problem space.

An example of well-structure problems could be design of products with modular architecture, such as desktop PCs, in which design is a combination of modules [12]. However, even for such a problem, there is no guarantee that the problem can easily be solved, for example, due to computational complexity (e.g., combinatorial explosion).

5.2. Winch design

Let us consider a design example of a winch. If the length of wire is not too long, usually the design criterion is the strength of the wire (Figure 7). Therefore, there is only one *FR*1 (to support the load *L*). Regarding DPs, tensile strength of the wire, σ, must be larger than $4L/\pi D^2$ (where *D* is the diameter of the wire). This boils down to two DPs, *DP*1 (material, *M*) and *DP*2 (diameter *D*). As seen in Equation 8, this is redundant design. (In practice, this does not cause a problem, because the wire material is usually steel.)

$$\{L\} = [x \quad x]\begin{Bmatrix} M \\ D \end{Bmatrix} \tag{8}$$

This is correct when the wire is vertically hung (Figure 7) or horizontally supported with frictionless wheels (Figure 8). If there is any friction (due to weight or curvature of the wire), the tension increases because of frictions.

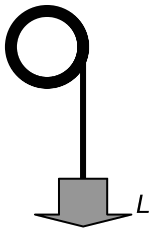

Fig. 7: *Vertical winch*

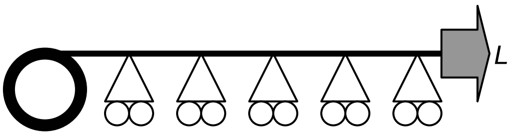

Fig. 8: *Horizontal winch*

However, if the wire is very long, we need to take the weight of the wire into account. The maximum length is determined by specific density and tensile strength. If we need to design a wire above this limitation, we need to change material first. This means a new *FP*2 must be introduced which is to reach certain distance *H*. *FP*2 is determined by DP1, material. There will be two DPs needed: one is *DP*1 (material) that determines both tensile strength (σ), and, specific gravity (ρ), and the other is *DP*2 (diameter, *D*). As depicted in Equation 9, this is decoupled design. In case the wire is used in liquid, however, we take further buoyant force into account, which may virtually reduce the load but still Equation 9 holds for this case.

$$\begin{Bmatrix} L \\ H \end{Bmatrix} = \begin{bmatrix} x & x \\ x & 0 \end{bmatrix}\begin{Bmatrix} M \\ D \end{Bmatrix} \tag{9}$$

The winch has a drum on which the wire is wound up. Due to the tension, the wound up wire receives vertical force. If the wire is not strong enough, the beginning of the wire which is wound up around the drum surface may collapse (especially in case of wires made of compound material). This enforces another design consideration, which involves tension (T, determined by L and wire weight which is released from the drum), compression strength of the wire (σ_c, again determined by material M), and drum geometry (G). This results in Equation 10 that is (still) decoupled design and can be solved "sequentially" more or less.

$$\begin{Bmatrix} L \\ H \\ C \end{Bmatrix} = \begin{bmatrix} x & x & 0 \\ x & 0 & 0 \\ x & x & x \end{bmatrix} \begin{Bmatrix} M \\ D \\ G \end{Bmatrix} \tag{10}$$

This example shows a typical case of "complexity by design," although it is in a mono-discipline of strength of materials, because the instantiation of a wire in a particular physical environment forced us to introduce new parameters, which increased "mono-disciplinary difficulty" of the design.

5.3. Ill-structured problems

Ill-structured problems stand in sharp contrast with well-structured problems. First of all, the system boundary cannot be clearly identified. Secondly, it is almost impossible to identify subproblems that can be independently solved, because a variety of aspects interrelate to each other. Coupled design cases of AD (Equation 2) correspond to ill-structured problems and their complexity results from mono-disciplinary difficulty of and/or multi-disciplinarity.

As opposed to *modular architecture*, for example, products such as cars have highly interrelated and tuned structure called *integral architecture* [12]. Equation 2 illustrates such interrelationships among design parameters. Due to coupling between $FR1$ and $FR2$ through $DP1$ and $DP2$, there is no way but to solve this equation as a whole. There are two major reasons for these interrelations among aspects.

If there is no theory that can comprehensively explain the given situation, it is necessary to employ a set of existing theories to tackle the problem but this will cause "complexity by design" and/or "intrinsic complexity of multi-disciplinarity" as discussed in Section 4.

When explicit relationships among those theories are known before solving the problem, we might be able to forecast what can happen if our knowledge is complete. This is the case of intrinsic complexity of multi-disciplinarity. However, this is not the case for product development in general. Even though we know explicit relationships among involved theories before solving the problem, in complex design problems often the system boundary is not clearly described and the division of the problem can be inadequate.

When no explicit relationship among those theories is known before solving the problem, at a later stage we may find unidentified physical connections within the problem that interfere with each other due to "complexity by design" introduced by instantiating an entity at a later stage. These implicit, hidden interferences among theories can surprise designers in the middle of product development.

The discussions above identified two major reasons for ill-structured problems in product development. The first is the lack of sufficiently systematized design knowledge that can tackle complex multi-disciplinary problems as a whole (such as the case in Figure 5). The second is that the introduction of a set of independent domain theories to solve the problem will inevitably cause "complexity by design" and/or "intrinsic complexity of multi-disciplinarity".

The second case can be explained by the following example (Figure 9). Consider a design of a simple mechatronics mechanism (e.g., micro-cantilever) that can be simplified as a system consisting of a single-supported beam (cantilever) and lumped mass. This mechanism has an initial function requirement to oscillate the mass with specified amplitude (O). There are two design parameters, i.e., design of the beam (mass and stiffness of the beam determined by its geometry, B) and design of the oscillation system including the control (C). At this stage we obtain Equation 8 that implies redundant design.

$$\{O\} = \begin{bmatrix} x & x \end{bmatrix} \begin{Bmatrix} B \\ C \end{Bmatrix} \tag{8}$$

Suppose that due to control requirements, we need to reduce the mass for wider bandwidth and that the beam is made of silicon that is fragile. If we do this (i.e., lighter design), generally the strength also reduces unless the geometrical shapes are modified. This suggests a weaker design of the cantilever and fatigue caused by vibration becomes an important design criterion. Therefore, we can identify another function requirement to sustain for a given time span (T). However, obviously T is determined by the beam design and external excitation determined by C. This results in a coupled design case depicted by Equation 9.

$$\begin{Bmatrix} O \\ T \end{Bmatrix} = \begin{bmatrix} x & x \\ x & x \end{bmatrix} \begin{Bmatrix} B \\ C \end{Bmatrix} \tag{9}$$

This coupled design happened partly because there is no unifying theory that can deal with mechatronics machine design comprehensively and partly because of physical interactions among design parameters (see Figure 2). We used control theory, dynamics knowledge, mechanics knowledge, and knowledge on the strength of materials. Although these theories are independent in principle, they share certain physical concepts (such as deformation, stiffness, force, strength, etc.). Due to these conceptual level interactions among different theories (i.e., intrinsic complexity of multi-disciplinarity), the design becomes intrinsically coupled.

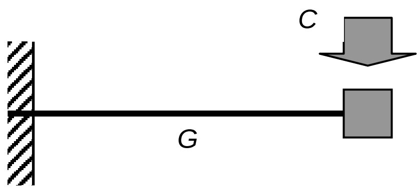

Fig. 9: *Micro-cantilever*

However, at some point during the product development, the designer may even find out that the temperature *t* actually influences on the stiffness and fatigue strength of the material (because for instance the device is installed in a sealed package and the heat conduction is not considered). This will result in Equation 10 with a non-square DM. This is a typical example of “complexity by design,” because the designer designed the cantilever inescapable from heat influences. While this “complexity by design” does not change the process complexity of the problem, it simply complicates the design process in that it will narrow the choice of the material (which is time-independent real complexity according to Suh [3]).

$$\begin{Bmatrix} O \\ T \end{Bmatrix} = \begin{bmatrix} x & x & x \\ x & x & x \end{bmatrix} \begin{Bmatrix} B \\ C \\ t \end{Bmatrix} \qquad (10)$$

5.5. Summary

Table 2 summarizes the discussions so far and shows the relationships among:

- Coupled, decoupled, uncoupled (AD).
- Multi-disciplinary design, mono-disciplinary design.
- Well-structured problems, ill-structured problems.

6. TECHNOLOGICAL WAYS TO DEAL WITH ILL-STRUCTURED PROBLEMS

In the previous section, we identified how complexity of ill-structured problems in product development increases. One is the lack of sufficiently systematized design knowledge which can tackle multi-disciplinary problems as a whole. The other is the complexity of the problem introduced by a set of domain theories that may cause “complexity by design” and/or “intrinsic complexity of multi-disciplinarity”.

Then, how can we solve these problems? An obvious, straightforward approach is to establish a unified theory to tackle a multi-disciplinary design problem such as mechatronics design. To do so, first we need to build ontological knowledge that describes relationships among various theories. This is the approach of knowledge integration and fusion through ontological integration [10, 13].

Such ontological descriptions about a theory will help to identify conditions in which the theory is valid and interfaces among theories.

To avoid finding unexpected relationships among theories, resulting from “complexity by design” or “intrinsic complexity of multi-disciplinarity”, it might be useful to develop a “design interference detector” or a qualitative physics based reasoning system that envisions possible interactions among employed theories. An early attempt can be found in [14, 15]. The basic idea is to first build a knowledge base that contains as many physical phenomena as possible and then to build a reasoning system to envision possible phenomena that can happen to design objects. The system is able to detect “hidden” interactions among theories especially for situations depicted in Figures 1 and 2.

Table 2: *Summary*

Disciplines	Complexity	How complexity is introduced	AD classification	Structuredness
			Uncoupled design	Well-structured
Mono-disciplinary	Computational complexity		Uncoupled design Decoupled design	Well-structured
			Coupled Design	Ill-structured
	Complex relationships among parameters	Complexity by design	Decoupled design	Well-structured
			Coupled design	Ill-structured
Multi-disciplinary		Complexity by design	Coupled design	Ill-structured
		Intrinsic complexity of multi-disciplinarity	Coupled design	Ill-structured

CONCLUSIONS

This paper is our attempt to establish a scientific understanding of complexity, which is a first step toward complexity management in design. Without such an understanding, it might not be possible to reduce the complexity or to keep it under control. Although there are such pioneering research results of DSM [1] and AD [2], complexity in design has not been dealt with in the context of knowledge structure.

This paper discussed complexities of multi-disciplinary product development from the viewpoint of knowledge structure. We first outlined the directions of the next generation product development and concluded such multi-disciplinarity is unavoidable. Second, we analyzed the difference between difficult mono-disciplinary problems and complex multi-disciplinary problems. Third, we reviewed two methods to deal with such complexities, *viz*., DSM and AD.

In Section 4, we analyzed the source of complexity in multi-disciplinary problems from the viewpoint of knowledge structure and identified “complexity by design” and “intrinsic complexity of multi-disciplinarity” when multiple theories are involved during design. We explained these two types of complexity result from indeed knowledge structure, namely, the relationship among multiple theories. One important conclusion here is that two independent theories can have relationship either by having a common instance physical object or by having an interface concept among them. These become sources of unidentified relationships that cause troubles in design.

In Section 5, these complexities are illustrated with examples. These examples illustrated how the idea of knowledge structure based complexity can explain why multi-disciplinary design problems often turn out to be ill-structured. We also clarified relationships among various concepts we introduced in this paper.

To deal with multi-disciplinary designs, Section 6 identified three approaches. One is to establish a unifying theory for relevant areas (knowledge integration and knowledge fusion). The second is to build ontological descriptions about relationships among those relevant areas. The third is to develop a qualitative physics based design interference detector and to envision unidentified relationships among theories before the design takes place.

ACKNOWLEDGEMENT

The author gratefully acknowledges the support of the Dutch “Innovation Oriented Research Program Integrated Product Creation and Realization” (IOP-IPCR) of the Dutch Ministry of Economic Affairs. This research was partly conducted as a part of the Smart Synthesis Tools Project within this IOP-IPCR program. The author would like to thank members of the project for their valuable comments through discussions; in particular, special thanks go to Valentina D’Amelio (Delft University of Technology).

References

[1] Browning, T.R., *Applying the Design Structure Matrix to System Decomposition and Integration Problems: A Review and New Directions*, IEEE Trans. Eng. Management, 48/3, 2001, pp. 292-306.

[2] Suh, N.P., *The Principles of Design*, Oxford University Press, Oxford, New York. 1990.

[3] Suh, N.P., *Complexity in Engineering*, Annals of CIRP, 54/2, 2005, pp. 581-598.

[4] Tomiyama, T., Meijer, B.R., *Directions of Next Generation Product Development,* in ElMaraghy, H.A., ElMaraghy, W.H. eds., Advances in Design, Springer, London, ISBN 1-84628-004-4, 2005, pp. 27-35.

[5] Tomiyama, T., *Knowledge Intensive Engineering towards Sustainable Products with High Knowledge and Service Contents*, in Horváth, I., Medland, A.J., Vergeest, J.S.M. eds., TMCE 2000, Third International Symposium on Tools and Methods of Competitive Engineering, Delft University Press, Delft, the Netherlands, 2000, pp. 55-67.

[6] *Proceedings CSCWD (Computer Supported Collaborative Work in Design) Workshops*.

[7] CIRP Working Group on Engineering as Collaborative Negotiation, http://wisdom.usc.edu/ecn/

[8] Dong, Q., Whitney, D.E., *Designing a Requirement Driven Product Development Process*, in Proc. ASME 2001 DETC, Paper No. DETC2001/DTM-2007 (CD-ROM), 2001.

[9] Meijer, B.R. Tomiyama, T., van der Holst, B.H.A., van der Werff, K., *Knowledge Structuring for Function Design,* Annals of the CIRP, 52/1, 2003, pp. 89-92.

[10] Tomiyama, T., Takeda, H., Yoshioka, M., Shimomura, Y., *Abduction for Creative Design*, Proc. of 2003 ASME DETC, Paper No. DETC2003/DTM-48650, ASME, New York, 2003.

[11] Tomiyama, T., *Collaborative Product Development in Ill-Structured Problem Domains*, in Shen, W.M., Lin, Z.K., Barthès, J.P., Luo, J.Z., Deng, J.M., Li, X.P., Yong, J.M., Hao, Q., eds., Proc. 2006 10th Int. Conf. on Comp. Supported Cooperative Work in Design (CSCWD 2006), IEEE Press, ISBN 1-4244-0164-X, Vol. 1, 2006, pp. 15-20.

[12] Clark, K.B., Fujimoto, T., *Product Development Performance,* Harvard Business School Press, Boston, 1991.

[13] Yoshioka, M., Umeda, Y., Takeda, H., Shimomura, Y., Nomaguchi, Y., Tomiyama, T., *Physical Concept Ontology for the Knowledge Intensive Engineering Framework,* Advanced Engineering Informatics, 18/2, 2004, pp. 69-127.

[14] Kiriyama, T., Tomiyama, T., Yoshikawa, H., *A Model Integration Framework for Cooperative Design,"* in Sriram, D., Logcher, R., Fukuda, S., eds., Computer-Aided Cooperative Product Development, Lecture Notes in Computer Science 492, Springer-Verlag, Berlin, 1991, pp. 126-139.

[15] Kiriyama, T., Tomiyama, T., Yoshikawa, H., *Qualitative Reasoning in Conceptual Design with Physical Features,* in Faltings, B., Struss, P., eds., Recent Advances in Qualitative Physics, The MIT Press, Cambridge, MA, USA, 1992, pp. 375-386.