Proceedings of ICAD2013
The Seventh International Conference on Axiomatic Design
Worcester – June 27-28, 2013
ICAD-2013-15

LESSONS LEARNED FROM TEACHING AXIOMATIC DESIGN IN ENGINEERING DESIGN COURSES

Ang Liu
angliu@usc.edu
Department of Aerospace and Mechanical Engineering
Viterbi School of Engineering
University of Southern California
Los Angeles, CA, USA.

Stephen Lu
sclu@usc.edu
Department of Aerospace and Mechanical Engineering
Viterbi School of Engineering
University of Southern California
Los Angeles, CA, USA.

ABSTRACT

Axiomatic Design is an important design theory that is often taught in many engineering design courses. This paper presents a case study to summarize our various lessons learned of teaching Axiomatic Design in practice. Based on the study of 30 team design projects that were collected from a graduate level engineering design course, we observed some common challenges/difficulties that student designers often encounter when learning and practicing Axiomatic Design Theory. These lessons are organized according to their relevance to several key concepts in Axiomatic Design: domains, hierarchy, the zigzagging process, the design axioms, and constraints. For each practical challenge/difficulty, we prescribe some relevant theoretical foundations and related design methods to facilitate the understanding and practice of the Axiomatic Design.

Keywords: Axiomatic Design, design education.

1 INTRODUCTION

Engineering design courses play the role of equipping student designers with the required knowledge and expertise to solve practical design problems. In the past, the vast majority of design courses primarily focused on the technical design phase by teaching students how to analyse, optimize, and improve a given problem (i.e. an existing product), while those truly important concepts and methods that are critical for the early design stages (e.g., the functional and conceptual design phase) are often either simply ignored or superficially covered. As a consequence, most existing design courses are famous for producing "engineers" who only know how to solve the problem right instead of "designers" who also understand how to frame the right problem. As the importance of design creativity and early stage design decision making draws increasing attentions in both academia and industry, today's design education is experiencing a profound paradigm shift from teaching students specific design techniques and knowledge to teaching them general "design thinking".

Axiomatic Design (AD) has many unique features that make it a perfect candidate to expedite such a paradigm shift. Above all, AD is a domain-independent theory that can be applied in different design fields. Such a universal applicability of AD theory is important to cultivate student's general "design thinking". Furthermore, AD has been extensively studied in the past. There exist many practical applications of AD theory in both academia and industry, which can be incorporated into the teaching as illustrative examples to guide the design practice. Last but not least, relatively speaking, it does not require much sophisticated technical or mathematical knowledge to grasp the essences of AD theory. Therefore, the theory can be taught to different levels of student designers including freshmen in college [Thompson, 2009].

The teaching of AD is not foreign to the design community [Tomiyama *et al.*, 2009]. In general, there exist two common strategies to teach AD theory within different engineering design courses. Some instructors introduce AD only as one of the many design theories and methodologies together with the teaching of other approaches (e.g., systemic design approach [Pahl *et al.*, 2007]). Some others treat AD theory as the main subject of the course and focus on guiding student designers how to employ AD to solve real-world problems via practice oriented assignments such as case studies and design projects. The disadvantage of the former strategy is that there is often not enough time and assignments for the student designers to develop a deep understanding of AD theory. In contrast, the disadvantage of the latter strategy is that designers are often unable to see the whole picture of how AD theory is related to other design approaches. Regardless of the strategies adopted, another common weakness of the current AD teaching is that the theory remains mostly taught as an analysis tool for alternative comparison, evaluation, and selection, while its unique values in supporting design synthesis are far from fully released. This is evident by the fact that majority of current AD teaching primarily highlights the importance of the two design axioms without elaborating on the theoretical rationales and practical meanings of other key concepts of AD theory such as "domains", "hierarchy", "zigzagging", etc.

In the past few years, we have been exploring a new strategy to teach AD theory in a more effective and systemic manner. Specifically, we still treat AD theory as the main subject of the course, but in the meantime we also incorporate some related design methods to complement the teaching of AD. These complementary methods serve to deepen the understanding of certain blurry aspects of AD theory. The complementary methods are not randomly selected, but rather they are chosen to address a common difficulty designers often encounter when learning and practicing AD theory. This paper provides a detailed case study to summarize our various lessons learned in adopting

Copyright © 2013 by ICAD2013

this new strategy to teach AD theory in a graduate level engineering design course.

The rest of the paper is organized as follows. Section 2 introduces the background of the case study in terms of its participants, design problem, and data collection. Section 3 elaborates the various lessons we have learned from this case study that are relevant to multiple key concepts in AD theory: domains, hierarchy, the zigzagging process, the axioms, and constraints. Section 4 ends this paper with conclusions and the limitations of this study.

2 CASE STUDY

The subjects to study are 30 team design projects that are all collected from a graduate level engineering design course, "Advanced Mechanical Design", which is offered by the Aerospace and Mechanical Engineering Department at the University of Southern California. These course projects are all accomplished by different design teams across 5 semesters during the years 2009-2012. The course participants are all graduate students registered in the University of Southern California, majoring in engineering related fields such as mechanical engineering, aerospace engineering, industrial engineering, etc. At the beginning of every semester, the class is equally divided into 6 design teams, each with 4-6 students. Almost half of the course participants are distance students who have full-time and engineering-related jobs. Therefore, in some sense, this study can be regarded to have included the feedback of both "expert designers" (distance students) and "novice designers" (on-campus students).

This design course consists of three sequential teaching/learning modules: the identification of design targets, the generation and selection of design concepts, and the modification of the chosen concept. It normally takes 3 lectures plus one design review presentation to finish every module. During the review presentations, every project team is allowed 15 minutes to present its design results up to the stage. At the end of the course, every team is also required to compose and submit a provisional patent application report to summarize the innovativeness of its final design results. Within each module, different design approaches are explained to provide designers with the right "tool" to address diverse challenges in different design phases. The theoretical rationale and practical requirements of including every approach and how these chosen approaches contribute to the teaching of AD will be elaborated in section 3.

The design approaches covered in the first module include: Quality Function Deployment [Akao, 2004], the Kano Customer Satisfaction Model [Berger *et al.*, 1993], and the Smart Question Approach [Nadler and William, 2004]. The focus of this module is to teach student designers how to leverage various customer needs in the market segment to frame the unique decision opportunities and determine the real design targets (i.e., functional requirements). The second module consists of two approaches: the Synthesis Reasoning Framework (i.e., SRF) [Lu and Liu, 2011] and Axiomatic Design Theory. Based on our previous work, the SRF can provide some theoretical explanations for certain blurry aspects (e.g., why it is important to distinguish between the different design domains) of AD theory. The focus of this module is to teach student designers how to create multiple new concepts via a systemic synthesis reasoning process guided by the SRF, and then select the best concept via the design axioms prescribed by AD theory. Finally the third module teaches student designers how to improve/modify an existing product (for example, their chosen concept) using Complexity Theory [Suh, 2005] and TRIZ [Altshuller, 1999]. Note that the technical design phase (which further transforms the modified concept into the production process) is not addressed in this course.

The specific problem to address is "to design a computer input artifact that avoids and/or reduces the user's repeated stress injuries (RSI) on the dominant hand". The same assignment has been used in the past four years. It is appropriate for a graduate level engineering design course because it addresses a recently emerging customer need (i.e., to reduce RSI) on a widely seen and commonly used product (i.e., computer input device). On one hand, the product itself is already familiar to the designers. On the other hand, however, depending on the unique choice of target customers, this problem is still open to many creative solutions.

The data are collected from the design documents (i.e., presentation slides and the final provisional patent application report) each team submitted and the video records of their three design review presentations. All verbal materials are properly transcribed. Specifically, there are four types of data that are relevant to the study of AD: the design architecture (i.e., domain and hierarchy), the zigzagging process, the usage of the Independence Axiom (i.e., the design matrix) and the Information Axiom (i.e., the probability density function graph), and the sketches or CAD drawings of the final solution. Here we provide a real project accomplished in this course as the illustrative example (Figures 1-3). At the end of the course, we also conducted an informal survey in order to collect student's feedback towards the various design approaches covered in this course. Specifically, we require every team to reflect on their entire design process and summarize the 5 most important design concepts/principles/axioms/knowledge that they have learned in this course, and the 5 greatest challenges that they have faced when applying these design methods in practice.

3 LESSONS LEARNED

This session summarizes various lessons we have learned from teaching this course. These findings are organized into five subsections with each focusing on an important concept in AD theory including: domains, hierarchy, zigzagging, the axioms, and constraints. In each subsection, first we briefly review the theoretical meaning of the concept, next we present some common mistakes of interpreting the concept based on our observation in practice, and finally we prescribe our method to deepen student's understanding of the concept.

 Copyright © 2013 by ICAD2013

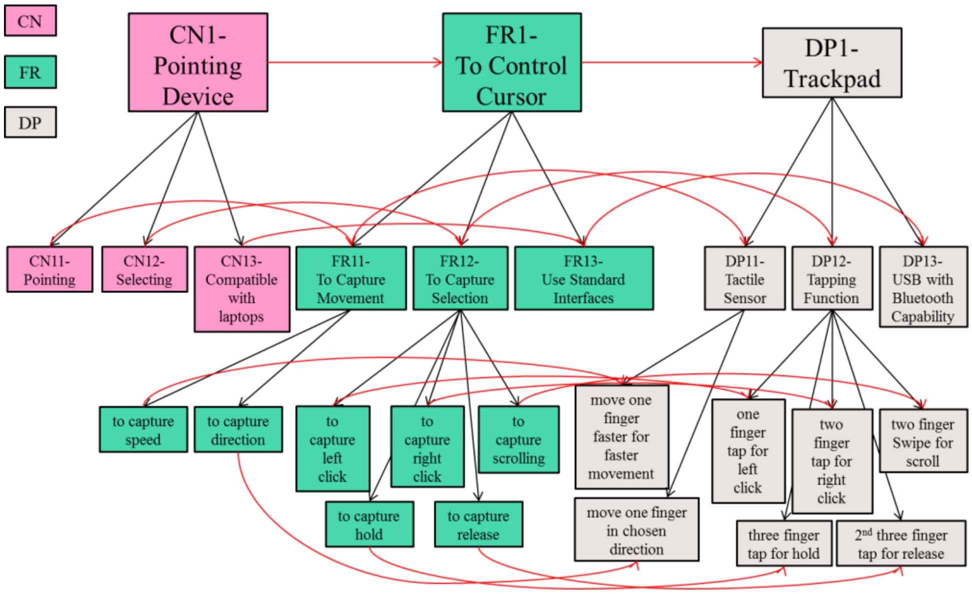

Figure 1. An illustrative example of the "zigzagging" process.

$$
\begin{Bmatrix} FR121 \\ FR122 \\ FR123 \\ FR124 \\ FR125 \end{Bmatrix} = \begin{bmatrix} X & 0 & 0 & 0 & 0 \\ 0 & X & 0 & 0 & 0 \\ 0 & X & X & 0 & 0 \\ 0 & 0 & 0 & X & 0 \\ 0 & 0 & 0 & X & X \end{bmatrix} \bullet \begin{Bmatrix} DP121 \\ DP122 \\ DP123 \\ DP124 \\ DP125 \end{Bmatrix}
$$

FR121: left click	DP121: One finger touch
FR122: right click	DP122: Two finger touch
FR123: scroll	DP123: Two finger swipe
FR124: capture hold	DP124: three finger touch
FR125: capture release	DP125: three finger touch

First 3-finger touch is first required to hold before a second 3-finger touch can be used to release

Two-finger swipe must be a firm action. If action is tentatively, it could be misread as a two-finger touch.

Figure 2. An illustrative example of the "design matrix".

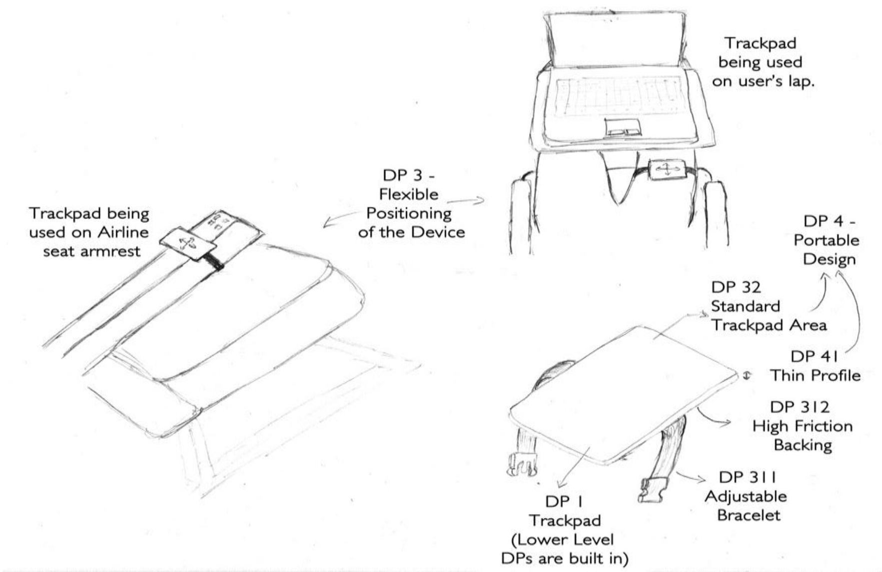

Figure 3. An illustrative example of the final solution sketching.

Copyright © 2013 by ICAD2013

3.1 DESIGN DOMAINS

Domains are a new concept introduced by AD theory to distinguish between different types of activity/decisions in the design process. According to Suh, there exist four types of fundamental design domains: (1) the customer domain, (2) the functional domain, (3) the physical domain, and (4) the process domain. The specific design decisions that are addressed in each domain are customer needs (CN), functional requirements (FR), design parameters (DP), and process variables (PV) respectively. For each pair of adjacent domains, the left domain represents the "what" (or "ends") that designers intend to achieve, whereas the right domain represents the "how" (or "means") that the designers propose to achieve the "what". In the design process, decisions in the "what" domain are constantly transformed into decisions in the "how" domain via a horizontal mapping operation.

In the informal survey, one third of the design teams reported that they encountered frequent difficulties in distinguishing between CNs and FRs in practice. The confusion of CNs and FRs can be indirectly reflected by the fact that many designers often use the term "customer requirement" to represent either "customer need" or "functional requirement" by mistake. As a result of such confusion, instead of using the mapping operation to transform CN (as "what") to FR (as "how"), designers often generate CNs and FRs separately and then link them afterwards. For example, we have observed some extreme cases where the designers first proposed multiple need/requirements all at once, then categorized them into the customer or functional domain (as CN or FR) accordingly, finally to identify and establish any appropriate CN-FR mapping relationships. Compared to the distinction of CNs and FRs, it is much easier for the designers to clearly differentiate between FRs and DPs. In the study, there are only a few design teams that regard the FR-DP distinction as one of their major challenges in this course.

According to Suh, customer needs describe "the benefits customers seek" from a product, whereas "functional requirements" prescribe how to provide customers with the desired benefits. By definition, it is clear that the former should be collected from the customers based on their experience and preference, whereas the latter should be determined by the designers based on their design knowledge and expertise. If these two types of decisions are confused, designers can easily lose their autonomy in the design process. Although the customer involvement [Kauliu, 1998] in new product development is drawing increasing attentions in recent years, most of its successful applications are limited to the industrial product development in which it is much easier to identify the lead users [Urban and Hippel, 1988]. But for the vast majority of design tasks (particular consumer product design), it is still critical to explicitly differentiate between CNs and FRs particularly at the early design stages.

Based on our observation, the primary reason of such difficulty is that course participants in the classroom environment often play dual roles of both customers and designers. On one hand, due to the difficulty (with regards to both time and resources) of conducting independent survey of customer needs from the market segment, course participants often imagine themselves as the target customers and brainstorm for CNs based on their own product using experience. This is evident by the fact that 8 out of 30 design teams identified the CAD engineers (which is exactly the professions of many distance students enrolled in this course) as the initial target customers. In addition, our assignment addresses a very specific design problem (i.e., reduce RSI) on a commonly seen and used product (i.e., computer input device). This relatively small design scope also limits the designer's ability to identify many concrete CNs. In practical applications when customer needs are often collected and analysed by other stakeholders (e.g., marketing people) than the designers themselves, it is reasonable to expect that the difficulty of distinguishing CN with FR might be significantly reduced.

Our strategy is to introduce multiple customer need identification methods to complement AD theory. The current methods that we teach include the Quality Function Deployment (QFD) and the Kano Customer Satisfaction model. In the informal survey, almost half of the design teams attributed the Kano Customer Satisfaction model as particularly useful in helping them to predict the future customer "wants" based on the existing customer "needs". In the future, we also intend to tailor the "smart question approach" in the context of AD theory to guide the designers to systemically carry out the functional design phase (i.e., the mapping from CN to FR) based on three fundamental questions "how to describe your CN as initially unique", "what purposeful information is needed to generate the initial FR", and "how to organize the chosen CN and FR using a system viewpoint".

3.2 DESIGN HIERARCHY

Hierarchy, which represents the "design architecture", is another important concept in AD theory. Within every domain, a separate hierarchy must be created to properly organize design entities of the same kind according to their different levels of abstraction. In AD theory, a decomposition operation has been prescribed to guide designers to build design hierarchies in a systemic manner (as opposed to an ad hoc manner). Designers carry out the decomposition process layer by layer until the design becomes fully implementable or until the available design resources (such as schedule or budget) are exhausted. Because participants of this study are all graduate students in engineering majors, both "hierarchy" and "decomposition" are relatively familiar concepts to them. As a result, very few design teams regarded the concept of "hierarchy" as a difficulty of learning AD theory.

In practice, a common mistake regarding "hierarchy" is that designers often incorrectly place decisions of different kinds (i.e., CN, FR, DP, and PV) into the same hierarchy. As a consequence, rather than building four "small" hierarchies that organize different kinds of design decisions separately, designers often end up with a "large" hierarchy that completely mix-up all kinds decisions. This is to say that, diverse design decisions are decomposed into multiple segments of the same hierarchy (Figure 4). Among the 30 design samples, we have observed 5 such mistakes. Based on our observation, there are two main causes. On one hand, as we mentioned in section 3.1, it is by nature difficult for novice designers to clearly distinguish between different kinds of

 Copyright © 2013 by ICAD2013

design decisions (e.g., CN and FR) at the early design stages when everything remains relatively intangible. On the other hand, it is also because most designers lack a deep understanding of the unique two-dimensional structure of AD theory.

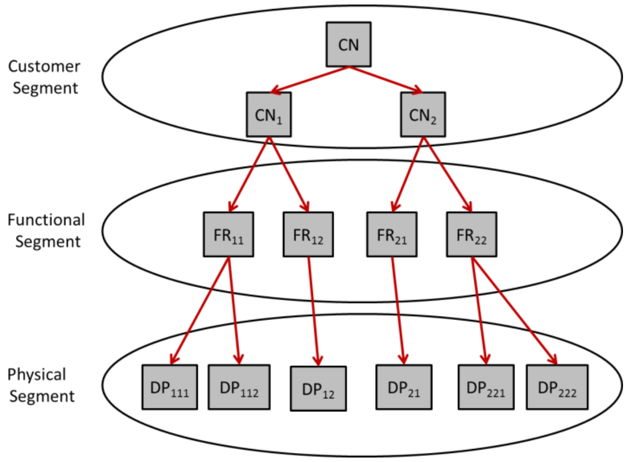

Figure 4. A common mistake in building a hierarchy in Axiomatic Design.

The distinction between "domains" and "hierarchy" is an important feature that distinguishes AD theory from other design theories and methodologies, because this leads to the new perspective of treating design as a two-dimensional thinking instead of the traditional one-dimensional thinking. Specifically, the mapping between adjacent domains (from "what" to "how") forces the designer to reason along the horizontal direction seeking for a particular "how" to realize a general "what". In contrast, the decomposition between adjacent layers within the same hierarchy guides the designer to reason along the vertical direction looking for multiple particular "sub-whats" to detail the general "what". This two-dimensional "domain-hierarchy" structure is in sharp contrast with other approaches, such as the Analytical Hierarchical Process (AHP) [Saaty, 1990], which focuses on decomposing a complex decision problem into multiple relatively simpler sub-problems, so that each of the sub-problems can be analysed separately. In AHP, any relevant aspects of the highest problem can be organized into the same hierarchy via decompositions. When comparing the AHP theory with AD theory, it is clear that the former follows a typical one-dimensional thinking via only the vertical decomposition, whereas the latter follows a two-dimensional thinking via both the horizontal mapping and vertical decomposition. Although such a two-dimensional (i.e., domain and hierarchy) structure is complicated to build, it is very important for early stage design. On one hand, it guides the designers to perceive the subtle distinctions between similar decisions and hence place them into different domains. On the other hand, by doing so, it becomes much easier to trace back different previous decisions and make design modifications accordingly. The latter is particularly meaningful for the design of complex systems in which decisions in different domains are often made by separate stakeholders (e.g., marketing people, product designer, manufacturing engineers, etc.).

In our previous work, we proposed to use the "analytic-synthetic distinction" to guide the designer to strictly follow the two-dimensional thinking prescribed by AD theory [Lu and Liu, 2011]. The "analytic-synthetic distinction" is a fundamental distinction in philosophy to differentiate two types of propositions namely the "analytic proposition" and the "synthetic proposition" [Kant, 1781]. By dictionary definition, proposition means the activity of proposing something new to be considered and accepted. Any proposition must contain two components: a subject and a predicate. The former is the input of a proposition, whereas the latter is the output of the proposition. In some sense, design can be regarded as a "proposition making" process in which designers propose some new systems (i.e., predicate) to accomplish certain intended goals (i.e., subject). Analytic proposition is a type of proposition whose predicate is contained within its precedent subject, whereas synthetic proposition is a kind of proposition whose predicate is not contained within its precedent subject. We argue that the horizontal mapping across adjacent domains should be made via making synthetic propositions, while the vertical decomposition across adjacent layers within the same domain should be made via making analytic propositions [Lu and Liu, 2011].

3.3 ZIGZAGGING PROCESS

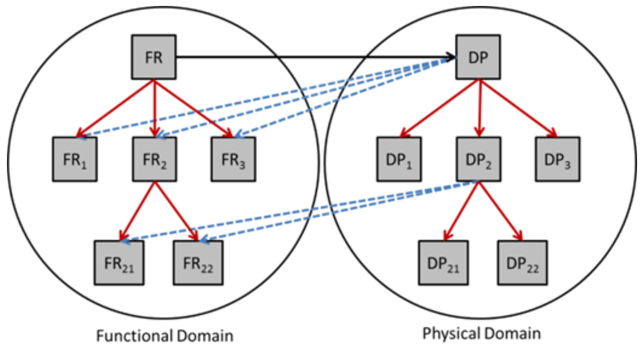

Figure 5. Decomposed-of vs. constrained-by relationships in the "zigzagging" process.

Based on the two-dimensional (i.e., domain and hierarchy) structure, AD theory prescribes a unique "zigzagging" process to develop hierarchies by alternating between adjacent domains. Specifically, when decomposing a general FR into multiple sub-FRs (e.g., FR_1, FR_2, and FR_3), the generation of a sub-FR must consider its superior FR-DP pair (see Figure 5). In other words, to arrive at a sub-DP (say DP_1) from a general FR, designers must go through a three step zigzagging process: (1) a "zig" from FR to DP, (2) a "zag" from parent FR-DP pair to FR_1, (3) another "zig" from FR_1 to DP_1.

We observed that designers often make mistakes with regards to the (2) "zag" step. Specifically, when a superior FR is decomposed into multiple sub-FRs, the resulting sub-FRs often directly become "part-of" their superior DP, which are certain behaviors of the chosen device. This is to say that, the "constrained-by" relationship between the superior DP and sub-FRs (the blue dash arrow in Figure 5) is mistakenly replaced by the "decomposed-of" relationship (the red solid

Copyright © 2013 by ICAD2013

arrow in Figure 5). Below are some examples of such incorrect understanding of the (2) "zag" step.

a) FR: convert user's natural motion to game navigation
DP: motion sensing system
Sub-FR: sense rotational motion
b) FR: to avoid losing and easy to switch when users type
DP: a device that can be worn
Sub-FR: to wear on the head
c) FR: to keep user alert
DP: a device that doesn't cause fatigue
Sub-FRs: to avoid users becoming fatigued for 4 hours

In all three examples, the sub-FR can be regarded as "part-of" the superior DP instead of the superior FR. The superior DP is the physical solution (i.e., how) that realizes the superior FR (i.e., what). "Part-of" the superior DP should be its more detailed components (i.e., sub-DPs) with certain behaviors. If the diagonal "constrained-by" relationship is confused with the vertical "decomposed-of" relationship, it is likely that the sub-FRs become indifferent with derived behaviors of the superior DP. As indicated by previous research, the confusion of function and behavior will greatly hinder the designer's creativity particular at early design stages. Last but not least, if the generation of a sub-FR only considers the impacts of the superior DP (while ignoring the impact of the superior FR), the two-dimensional "zigzagging" process and structure of AD theory will also be reduced to the one-dimensional structure of the Function-Means Tree [Bracewell, 2002] (see Figure 6).

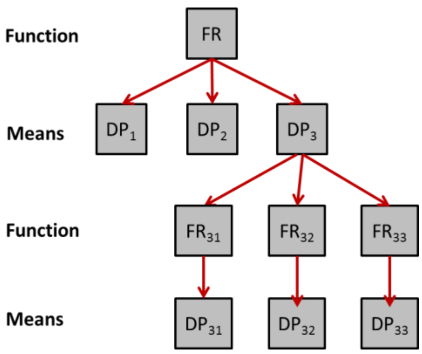

Figure 6. One dimensional Function-Means Tree.

To deepen the understanding of the zigzagging process, our approach is to introduce an extra "bounding" operation (in addition to the existing "mapping" and "decomposition" operations in AD theory) to represent the reasoning forces coming from the superior DP to the sub-FRs. By doing so, the generation of sub-FRs must consider both the superior FR via a decomposition operation and the superior DP via a bounding operation. The former creates a "part-of" relationship between FR and sub-FRs, whereas the latter establishes a "constrained-by" relationship between superior DP and sub-FRs. In addition, we also detail the "zigzagging" process into a 3-phase and 9-step synthesis reasoning process, (Figure 7) [Lu and Liu, 2011].

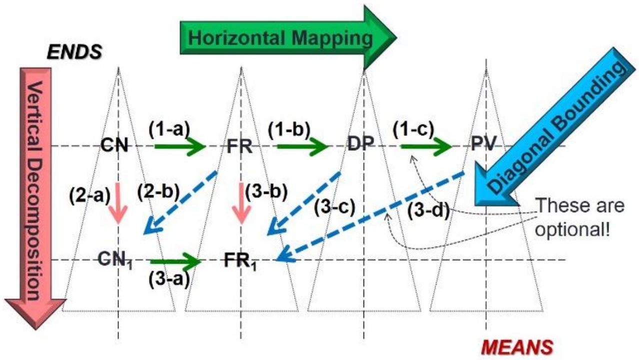

Figure 7. A "zigzagging" synthesis reasoning process [Lu and Liu, 2011].

3.4 Design Axioms

The two design axioms (i.e., the Independence Axiom and the Information Axiom) are the most essential (as well as famous) concepts in AD theory. At every decision point of the zigzagging process, the Independence Axiom is used to maintain the functional independence of the design and to characterize multiple design alternatives (or options) into three categories: uncoupled, decoupled, and coupled; and then the Information Axiom is employed to compare those alternatives (that comply with the Independence Axiom) in order to select the best alternative that has the minimum information content (or the maximum probability of success).

Half of the design teams attributed the "functional thinking" behind the Independence Axiom as the most important lessons they learned in this course. Based on our assessment of their design results, the vast majority of teams are able to correctly employ the Independence Axiom to structure their functional and physical hierarchies towards the uncoupled or at least decoupled designs. Among the three categories of designs (uncoupled design, coupled design, and decoupled design), most in-class questions from student designers go to the decoupled design such as "how to eliminate the unwanted sequence?" As expected, the majority of teams consider the design matrix as particularly useful in clarifying the FR-DP interactions. Furthermore, designers often confuse the "functional coupling" (i.e., FR-FR coupling or FR-DP couple) with the "physical coupling" (i.e., DP-DP coupling). According to the designers, it is most difficult to identify the FR-FR coupling in practice, because it requires much deeper abstract thinking.

With respect to the Information Axiom, due to the lack of sufficient knowledge of probability theory, it is common to see that designers often make mistakes in drawing the probability density function (i.e., PDF) curves (Figure 8). When multiple "system PDF" curves appear in the same chart, these curves (although with different shapes) must occupy the same amount of areas which represent the "probability". This is to say that, the shape of curves can be either tall but narrow (to represent the small standard deviation) or short but wide (to indicate the large standard deviation), but never both tall and wide (or both short and narrow). This finding suggests that, even for the graduate level engineering design course, it is still necessary to provide some basic statistics knowledge as backgrounds of the Information Axiom. Furthermore, as a

 Copyright © 2013 by ICAD2013

supplement of the Information Axiom, we also teach a set of domain-independent measures that are developed based on relevant studies of abductive reasoning to describe the quality of a concept. These measures include clarity (M_1), feasibility (M_2), testability (M_3), simplicity (M_4), and analogy (M_5).

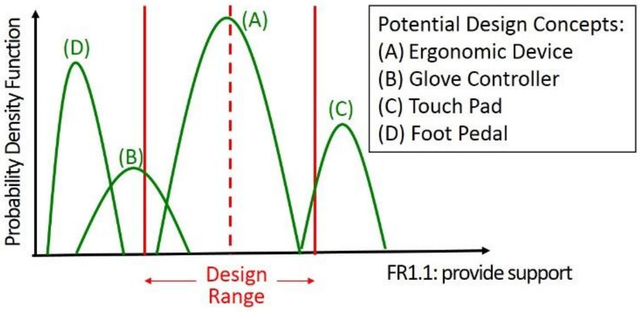

Figure 8. A common mistake of drawing probability density function curves.

After the introduction of AD theory, we also teach students how to use Suh's "complexity theory" [Suh, 2005] as a way to identify different kinds of complexities within existing systems. The intention is to deepen the understanding of the two axioms in a backward manner to highlight the importance of "functional independence" and "physical certainty" in reducing the future complexity. There are 4 design teams that regard "this new way of articulating complexity" as one of their most important lessons.

3.5 DESIGN CONSTRAINTS

In AD theory, constraints are defined as the "bounds of acceptable solutions". At early design stages, constraints are often confused with functional requirements. In this study, for example, one third design teams mistakenly attributed certain constraints as functional requirements. FRs are the real targets (objectives) of design, whereas design constraints are only the limitations of acceptable solutions which are proposed to satisfy the intended FRs. Unlike FRs which must maintain independent of each other, design constraints do not have to comply with the independence requirement [Suh, 1990]. Furthermore, FRs normally have a design range associated with them, while constraints only have a boundary value [Suh, 1990]. With regards to the mutual relationship between FRs and constraints, according to Suh, it becomes more efficient to select FRs when design is appropriately constrained [Suh, 1990]. In any case, FRs must be clearly distinguished from constraints. Otherwise, the design can easily diverge from the right course of objective-driven to constraint-driven.

In practice, a common mistake is that designers often regard weight/size as a FR (e.g., "be portable", "be light", "be small", "easy to carry", etc.) in the functional domain. Note that the weight/size of an integrated technical system is determined by multiple individual DPs. Suppose it is treated as a FR (instead of a constraint), this FR will unavoidably be affected by different DPs at the same time. According to AD theory, this violates the Independence Axiom and thus leads to a coupled design. Another common mistake is that designers often wrongly regard "low cost" as one of its FR to satisfy instead of a constraint to comply with. Similar to the case of weight/size, if cost is seen as a FR, it will also results in coupled designs. This is because that the overall cost of a technical system is also simultaneously determined by multiple DPs (instead of a single DP).

To facilitate the distinction between constraints and functional requirements, we have developed a new constraint management method for AD theory. From the theoretical perspective, we conceptually modeled constraints as the initial/boundary conditions of synthesis reasoning. For the practical perspective, we prescribe a more detailed classification of constraints. Specifically, in addition to Suh's existing classification of constraints as "input constraints" which apply to the overall design task and "system constraints" that apply to specific design decisions [Suh, 1990], we further categorize constraints into "internal constraints" and "external constraints". The internal constraint is a part of the technical system; hence, it limits the further evolvement of the system only from inside. The internal constraint is evident when design decisions demand more than the existing system can deliver. In contrast, external constraints are not part of the technical system; as a result, they bound the further expansion (rather than evolution) of the system from the outside. The external constraint appears when the system attempts to function more than it is currently capable of.

In our new classification [Lu and Liu, 2012], constraints are classified into four more specific categories: internal input constraint, external input constraint, internal system constraint, and external system constraints. Internal input constraints define the constraints which must be part of the technical system but are not chosen by designers themselves. External input constraints represent the constraints that are not contained in the technical system but are part of the design task description or problem statement. Internal system constraints refer to the constraints which are chosen by the designers to be part of the technical system. External system constraints describe the constraints that result from the designer's previous decisions but are not part of the final technical system.

4 CONCLUSION

Axiomatic design is an important design approach that is covered in many existing design courses. How to effectively teach the Axiomatic Design to student designers in the classroom has long been a struggling question for many instructors. In our perspective, the key to success lies in providing related theoretical foundations and relevant design methods to complement the teaching of Axiomatic Design. Based on the study of 30 team design projects that were collected from a graduate level engineering design course, we summarized some common challenges/difficulties that student designers often encounter when learning and practicing the Axiomatic Design in the classroom.

Copyright © 2013 by ICAD2013

Table 1. Summary of lessons learned.

AD Concept	Theoretical Importance	Common Mistake in Practice	Teaching
Domains	Distinguish between different kinds of design decisions	Confusion of CNs with FRs	QFD, Kano customer satisfaction model, and "smart question" approach
Hierarchy	Structure the same kind of design decisions	Mix "what" and "how" in one hierarchy	Analytic-synthetic distinction
Zigzagging	Build hierarchies by alternating between adjacent domains	Confusion with the "function-means" tree structure	A Synthesis Reasoning Process
Independence Axiom	Maintain functional independence	Confusion of functional coupling with physical coupling	Complexity Theory
Information Axiom	Reduce physical uncertainty	Probability density function curves	Statistics knowledge
Constraints	Bounds on acceptable solutions	Confusion of constraints with FRs	A new constraint management method

These lessons are organized according to their relevance to different key concepts of the Axiomatic Design: domains (section 3.1), hierarchy (section 3.2), the zigzagging process (section 3.3.), the axioms (section 3.4), and constraints (section 3.5). For each challenge/difficulty, we prescribe certain theoretical foundations and design methods (as a supplement of the Axiomatic Design) to facilitate the understanding and practice of the concepts (Table 1). Future work includes a rigorous and relevant assessment of designer's understanding towards the Axiomatic Design due to the introduction of these complementary methods.

There are several limitations that should be considered when interpreting the results of this case study. Above all, the informal survey was conducted in the team level instead of the individual level. Therefore, the results may not reflect individual designers' interpretations of Axiomatic Design. Furthermore, this course, which is offered on the Distance Education Network platform at the University of Southern California, consists of both distance students and on-campus students. The former can be regarded as close to the "expert designers", whereas the latter should be treated as the "novice designers". Although it is ignored in this study, it is reasonable to hypothesize that these two kinds of student designers may have very different understandings of Axiomatic Design. Last but not least, although the basic structure of the course remains the same, there were new content, methods, and examples added to each module of the course every time it was offered. On one hand, this is how we constantly adjust the "course design" based on emerging "student needs". But on the other hand, this also means that it might not be completely rigorous to combine all student feedback towards the course in the same study, because strictly speaking they are not learning exactly the same content.

5 REFERENCES

[1] Akao, Yoji. *Quality function deployment: integrating customer requirements into product design*. Productivity Press, 2004.

[2] Altshuller, Genrich. *The innovation algorithm: TRIZ, systematic innovation and technical creativity*. Technical Innovation Ctr, 1999.

[3] Berger, Charles, et al. "Kano's methods for understanding customer-defined quality." *Center for Quality Management Journal* 2.4 (1993): 3-36.

[4] Bracewell, R. H. "Synthesis based on function-means trees: schemebuilder." (2002): 199-212.

[5] Kant, I., *Critique of Pure Reason*, Wood Cambridge University Press, 1781. ISBN: 0-521-65729-6.

[6] Kaulio, Matti A. "Customer, consumer and user involvement in product development: A framework and a review of selected methods." *Total Quality Management* 9.1 (1998): 141-149.

[7] Lu, S. C.-Y., and A. Liu. "A Synthesis Decision Framework for Early-Stage Innovative Design." *Interdisciplinary Design: Proceedings of the 21st CIRP Design Conference*. Mary Kathryn Thompson, 2011.

[8] Lu, S. C.-Y., and A. Liu. "Managing Design Constraints in Synthesis Reasoning." *CIRP Design 2012*. Springer London, 2013. 73-82.

[9] Nadler, G., and William, C. "*Smart questions: Learn to ask the right questions for powerful results*". Jossey-Bass, 2004.

[10] Pahl, G and Beitz, W. "*Engineering design - a systematic approach*", 2nd ed. Springer, London, 2006

[11] Saaty, T. L., "How to make a decision: The analytic hierarchy process", *European Journal of Operations Research*, 48, 9–26, 1990.

[12] Suh N.P., *The Principles of Design*, New York: Oxford University Press, 1990. ISBN 0-19-504345-6

[13] Suh, Nam P. *Complexity: theory and applications*. Oxford University Press, USA, 2005.

[14] Thompson, Mary Kathryn. "Teaching axiomatic design in the freshman year: a case study at KAIST." *Proceedings of the 5th International Conference on Axiomatic Design*. 2009.

[15] Tomiyama, T., et al. "Design methodologies: Industrial and educational applications." *CIRP Annals-Manufacturing Technology* 58.2 (2009): 543-565.

[16] Urban, Glen L., and Eric Von Hippel. "Lead user analyses for the development of new industrial products." *Management science* 34.5 (1988): 569-582.

 Copyright © 2013 by ICAD2013