Chapter 8
Managing Design Constraints in Synthesis Reasoning

S. C.-Y. Lu and A. Liu

Abstract Constraints in the context of design synthesis represent various bounds on acceptable technical systems. In early design stages, constraints must be carefully managed in order to keep synthesis on the right track. This paper presents a new constraints management model which is developed based on a domain-independent synthesis reasoning framework. Within the proposed model, design constraints are classified into four types: internal input constraint, external input constraint, internal system constraint, and external system constraint. For each type of design constraints, the model prescribes a respective management strategy.

8.1 Introduction

In general, design synthesis can be seen as a reasoning process from an intangible intent (e.g., what) to some more tangible instantiations (e.g., how) [1]. This process is especially difficult at early design stages (i.e., functional design and conceptual design phases) when both design intent and design constraints are still intangible and subjective. Design intent identifies the goal for synthesis reasoning, whereas design constraint establishes a bound within which synthesis reasoning is carried out. Since the design intent and design constraint play different roles in synthesis reasoning, they should be explicitly distinguished. Otherwise, synthesis could be mistakenly diverged from goal-driven design to constraint-driven design.

S. C.-Y. Lu (✉) · A. Liu
University of Southern California, Los Angeles, CA 90089, USA
e-mail: sclu@usc.edu

A. Liu
e-mail: angliu@usc.edu

A. Chakrabarti (ed.), *CIRP Design 2012*, DOI: 10.1007/978-1-4471-4507-3_8,

Design constraint plays an important role within design synthesis in specific to creative alternative creation. In the alternative creation stage of design synthesis, "novelty" and "appropriateness" of solution alternatives ideated are two important factors that must be considered [2]. The importance of constraints in such creative tasks has already been extensively studied by cognitive psychology [3, 4]. Despite such importance, few efforts have been devoted to studying ways to manage design constraints in the context of synthesis for early-stage creative design. Most existing constraint management models heavily rely on the traditional problem-solving methods [5]; hence they often ignore the impact of constraints on problem-framing which is particularly important in creative design synthesis. Hence, these existing models cannot be directly used to address those early-stage design constraints which are all intangible, subjective and dynamic.

This paper presents a new constraint management model to support design synthesis. Various design constraints are firstly classified into several kinds, then for each kind, the model prescribes a unique management strategy. Constraints, although important, are only partial description of the design synthesis process. The management of constraints can only be effective when incorporated into the whole picture of design synthesis. For this reason, our constraint management model is developed based on an existing domain-independent synthesis reasoning framework.

The rest of the paper is organized as following. Section 8.2 summarizes the distinguishing characteristics of design constraints, and explains the reasons why a new constraint management model is needed. Section 8.3 formally presents the proposed model. Section 8.4 concludes this paper with outlooks for future works.

8.2 Constraints in Design Synthesis

8.2.1 Characteristics of Constraints in Design Synthesis

At early design stages, design constraints are mostly intangible and hard to quantify. Constraint can be seen an element factor that restricts a system from achieving a potentially higher goal [6]. Such definition suggests that the specification and valuation of design constraints rely on two variables: a specific goal and a tangible system. In the context of design, the former refers to the initial design intent (i.e., an input of design), whereas the latter means the final technical system (i.e., an output of design). At early design stages, the initial design intent needs to be further specified/structured and the final technical system is yet "to-be" created via synthesis. As a result, by definition, the design constraints cannot be expressed very explicitly and precisely at early stages. For instance, at the conceptual design phase, designers are unable to precisely estimate the actual product deliver time which is an important constraint within the entire product development cycle. The lack of quantitative information at early stages significantly increases the management difficulty of design constraints that have major impacts on later design decisions and final design outcomes.

All types of design decisions comprise subjective and objective parts. The former is more evident during the early design stages (e.g., functional design and conceptual design phases); whereas the latter is apparent toward the end (e.g., technical design phase) [7]. Design constraints are no exception. In the past, design constraints are often treated as purely objective, whereas their subjectivities are often neglected. This is largely because most of traditional approaches focus on analysis activity that occurs in later stages, as opposed to the synthesis activity that exists in early stages. Similar to those objectively defined constraints (e.g., geometric shape and physical laws) which play important roles in analysis during technical design phase, the subjectively constructed constraints are equivalently important to synthesis during the functional and conceptual design phase. Nevertheless, the impact of latter to design is far from fully explored. Additionally, as today's product development task becomes increasingly complex, diverse expertise from different stakeholders are often required in order to collaboratively carry out design synthesis. When multiple stakeholders each has individual preferences are required to jointly decide the boundary conditions (e.g., the bounds, limits, etc,) of design synthesis, it becomes even more difficult to capture and manage the subjectivity of design constraints.

Design constraints are dynamic at early design stages. This is particularly evident for the type of subjective constraints that are resulted from designer's previous decisions. As a particular design decision is changed, so do the associated constraints. Even for some external constraints (e.g., budget, schedule, etc.) which are imposed to designers by outside parties, their specific value of limitation may remain negotiable until the technical systems are finalized. In addition, as new objectives, components, information, and knowledge are increasingly added to the technical system via synthesis, new constraints will constantly arise and grow.

Constraints are intangible, subjective and dynamic at early design stages. Hence, they are often confused with the functional requirements (i.e., FRs). FRs are the real targets of design, whereas design constraints are only the bounds to acceptable solutions which must satisfy the desired FRs. Unlike FRs which should be stated and maintained independent of each other, design constraints do not have to satisfy such an independence axiom. In addition, it is often unnecessary to specify the tolerance for the constraints, whereas the FRs normally have a design range associated with them [8]. In terms of the mutual relationship between FR and design constraints, it becomes more efficient to select FRs when the design synthesis is appropriately constrained [8]. In any case, a true creative design synthesis should be more target-driven than constraint-driven.

8.2.2 Need for a New Constraint Management Model to Support Design Synthesis

It is important to note that design synthesis, in both sprite and process, is not equivalent to constraint satisfaction. There have been many early studies to formulate design task as a constraint satisfaction problem (CSP), and then to adopt the existing constraint-based systems (CBS) to manage design constraints [5]. Despite few successful implementations on simple design task, it has been proven to be very difficult to directly apply CBS to support design synthesis. This is largely because the natures of constraints (i.e., intangible, subjective, and dynamic) at early design stages are very different from the implementation prerequisites (i.e., tangible, objective, and static) of typical CBS. Constraint satisfaction is a part of design synthesis; however, design synthesis cannot be simplified as a pure constraint satisfaction process.

Recently there have been many attempts to apply preference aggregation principles from social choice to support design decisions [9]. Specifically, multiple individual preferences are properly combined in order to rank-order alternatives from most to least desirable. Such thinking, although useful in certain type of design decisions, cannot be directly adopted to manage constraints for design synthesis. Unlike design objectives which can be compared by their relative importance and design solutions that can be measured by their absolute performance, all types of design constraints are equally important in terms of preventing non-functional or unacceptable solutions. Hence, there is no strong need for an absolute ranking of all constraints. Nevertheless, the diverse preferences towards constraints are still important in design; they should be comprehensively collected and systemically combined to ensure that the "fence" of design space is seamless and flawless. Alternatively, we can say that the preference towards certain design constraint should be treated as a sort of "veto power" from individual designers to the final group design decision. As a matter of fact, the existence of constraints in design is one major difference between engineering design and social choice [9].

There are also many efforts devoted to develop certain constraint-based automatic reasoning and logic programming to support design decisions [10, 11]. These methods, although are effective in some specific domains (e.g., geometry design and digital circuit verification), do not meet the general applicability requirement of the new constraint management model for design synthesis. The "subjectivity" of design constraints must be carefully controlled in order to maintain such domain-independent requirement. Specifically, the new model must provide the clear definition, criteria, and classification of design constraints in order to objectively sort different kinds of constraints and distinguish the design target from design constraints. As well, for each type of constraints, the new model must prescribe an appropriate strategy to address it objectively.

8.3 A Constraint Management Model for Design Synthesis

8.3.1 Classification of Design Constraints

Various constraints must be considered in design synthesis. In general, these constraints can be classified into two types: "input constraints" which apply to the overall design task, and "system constraints" that apply to specific design decisions [8]. The input constraints are closely associated with the given design task (i.e., the assignment); hence they are designer-independent and must be satisfied by all proposed solutions. Whereas the system constraints are resulted from designers' previous decisions, therefore they are always designer-dependent and specific to certain design decisions. For examples, in industries, corporate strategy, market competition, government regulation, budget, and schedule, which are imposed to designers associated with the initial design task, can all be categorized as input constraints. Behaviors of certain device, capability of particular manufacturing machines and some domain-dependent physical laws, which relate to the realization of specific design objective, should be classified as system constraints. In short, the major difference between input constraint and system constraint lies in the original source. If the constraint comes from a general design assignment, it is defined as an input constraint; whereas if the constraint results from designers' specific decisions, it is defined as a system constraint.

Meanwhile, constraints can be either internal or external of the technical system being designed [6]. Design begins with an initial intent (i.e., goal), and this intent grows to become a sophisticated technical system by purposefully synthesizing relevant resources, information and constraints. During such synthesis process, the gradual evolution of the technical system is constrained by both internal and external forces. The internal constraint is a part of the technical system; hence, it limits the evolvement of the technical system only from inside. The internal constraint is evident when design targets demand more than the current technical system can deliver. For instance, if the chosen machine cannot successfully produce the required component, then it becomes the internal constraint of the manufacturing process. In contrast with internal constraints, external constraints are not part of the technical system; as a result, it bounds the expansion (rather than evolution) of technical system from the outside. The external constraint appears when the technical system tries to function more than it currently capable of or jump out of the scope of a given design task.

Meanwhile, constraints for design synthesis can come from both social and brute realities. Social reality knowledge is those stakeholder-dependent agreements resulted from social interactions; whereas the brute reality knowledge is those stakeholder-independent natural laws derived from domain physics. For example, the constraints from social reality can include the preferred strategies and business objectives of the company in terms of the product outcomes as well as the existing market competitions identified through benchmarking. The constraints derived from brute reality (which must be treated as non-negotiable "hard"

Table 8.1 Classification of constraints in design synthesis

Type of constraint	Input constraint		System constraints	
	Internal	External	Internal	External
Level of abstraction	General	Specific	Mostly general	Specific
Level of flexibility	Mostly Rigid	Rigid	Flexible	Rigid
Level of variability	Dynamic	Static	Dynamic	Static
Lever of subjectivity	Objective	Objective	Subjective	Mostly Objective
Practical example	Initial CN	Competitions	Behaviors of DP	Physical laws

constraints in synthesis) include, for examples, available production facilities, budget limits, and particular physical knowledge of the application domains, such as physics, materials, etc.

Based on the above discussions, design constraints can be classified into four types: internal input constraint, external input constraint, internal system constraint, and external system constraints. Table 8.1 summarizes their different characteristics. Specifically, internal input constraint defines the constraint which must be part of the technical system but are not chosen by designers themselves. External input constraint represents the constraint that is not contained in the technical system but is part of the design task description or particular assignment. Internal system constraint refers to the constraint which is chosen by the designers to be part of the technical system. External system constraint describes the constraint that is resulted from designer's previous decisions but not part of the final technical system. As Table 8.1 indicates, among all types of constraints, internal system constraints are most difficult to manage due to their special characteristics at early stages of design.

To create a new technical system, the design synthesis process starts with the functional design phase when the designers rationally choose a set of functional requirements (FR) to fully satisfy the given customer needs (CN). The internal input constraints are determined by the initial CN, whereas the external input constraints (e.g., budget and schedule) are imposed to the designers as part of the given design task. At this stage, due to the specific choice of FR, certain external system constraints (e.g., market competition, corporate strategy, government regulations, etc.) are incorporated in the decision making process (Fig. 8.1). Next, the designers proceed to the conceptual design phase to select certain design parameters (i.e., DPs) that can satisfy FRs. At this stage, many internal system constraints will emerge to limit the realization and further decomposition of FRs. For instance, if the behaviors of a particular DP cannot successfully satisfy the respective FR, this DP will become the internal constraint of the evolving technical system. Similarly, according to the Axiomatic Design Theory, the DPs at higher level of the DP hierarchy may become constraints of FR at the next level of the FR hierarchy [7]. The last stage is the technical design phase, when the process variables (PVs), which can satisfy the DPs, must be determined to manufacture the technical system. In this stage, more external system constraints must be

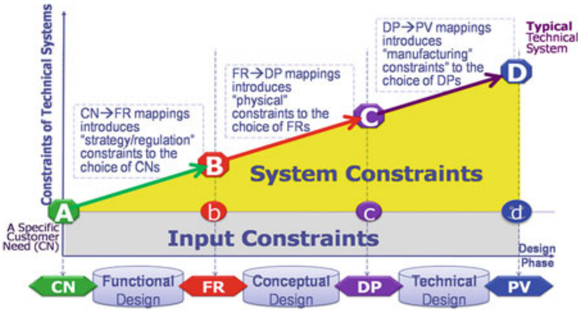
Constraints of Technical Systems
DP→PV mappings introduces "manufacturing" constraints" to the choice of DPs
Typical Technical System
FR→DP mappings introduces "physical" constraints to the choice of FRs
CN→FR mappings introduces "strategy/regulation" constraints to the choice of CNs
D
C
B
A
System Constraints
b
c
d
Input Constraints
A Specific Customer Need (CN)
Design Phase
CN
Functional Design
FR
Conceptual Design
DP
Technical Design
PV

Fig. 8.1 Input/system constraints in different design phases

considered (Fig. 8.1). For instance, certain physical laws and available manufacturing capability, which are external of the technical system, will become constraints of production of the chosen DP.

8.3.2 Strategies of Resolving Design Constraints

For each type of design constraint, we prescribe a unique strategy to address it in design synthesis. This is necessary, because different kinds of constraints appear in different design phases and play diverse roles in synthesis. Hence, they cannot be resolved using one universal strategy. For the internal input constraints (e.g., customer demands), since they are derived from the initial design intent (i.e., to satisfy CNs), they cannot be simply removed or ignored. Because such constraints are often started as intangible and lack of details, the best strategy is to define more specifications to avoid any potential violation in the upcoming decisions. For the external input constraints (e.g., budget and schedule), because they are mostly imposed on designers by outside parties (e.g., management), the best strategy is to carry out collaborative engineering negotiation [12].

For the internal system constraints (e.g., behaviours of certain device), since they are resulted from designers' own decisions, they are subject to change. Note that even if certain internal system constraints can be removed by changing the previous decision, new internal system constraints will always arise due to the same decision change. Because the internal system constraint is mostly subjective, flexible and dynamic, whether to remove it or comply with it often relies on designers' preference aggregation result. For the external system constrains (e.g., physical laws and manufacturing capacity), since they are introduced to designers due to certain system requirements (e.g., security and quality), they should be treated as hard constraints and cannot by violated in any circumstance. Therefore, the best strategy is to add extra buffers (e.g., safety factors) to the technical system to ensure that the external system constraint is never starved [6]. Table 8.2 summarizes the specific strategy for each type of constraints.

Table 8.2 Strategies to resolve different types of constraints

Type of constraint		Type of strategy
Input constraint	Internal of system	Define more specifications to avoid violation
	External of system	Engineering negotiation
System constraint	Internal of system	Preference aggregation
	External of system	Add extra buffers to avoid violation

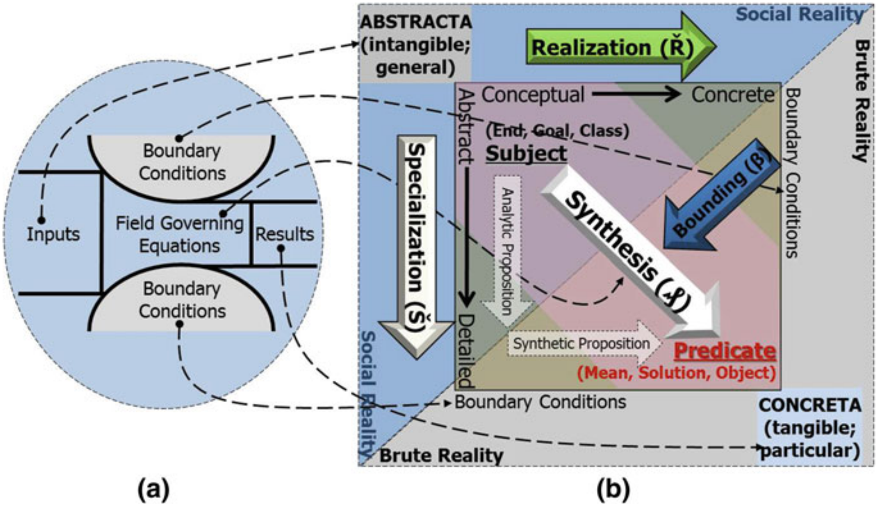

Fig. 8.2 Design synthesis as a progressive constraining process. **a** An initial-boundary-valued problem. **b** A structured synthesis reasoning framework

8.3.3 Design Constraints in Synthesis Reasoning

When viewed as a generic reasoning activity, synthesis is intrinsically an ill-posted (or under-constrained) problem. This suggests that synthesis reasoning can only produce a specific result under a proper set of constraints. Conceptually speaking, synthesis reasoning can be modelled as solving an initial-and-boundary-valued problem [1]. Figure 8.2 shows how an initial-and-boundary-valued system model (Fig. 8.2a) is conceptually mapped to an existing synthesis reasoning framework (Fig. 8.2b). The squared region in Fig. 8.2b represents a bounded (or constrained) "synthesis reasoning field" within which synthesis reasoning constantly transforms relatively conceptual/abstract design intent to more concrete/detailed design instantiation by following certain governing equations.

In this conceptual framework for synthesis reasoning, constraints are framed as the boundary conditions that take advantage of various bounding information to limit the creation and consideration of possible options. In Fig. 8.2b, the four sides of the squared area describe the internal and external input constraints, the two

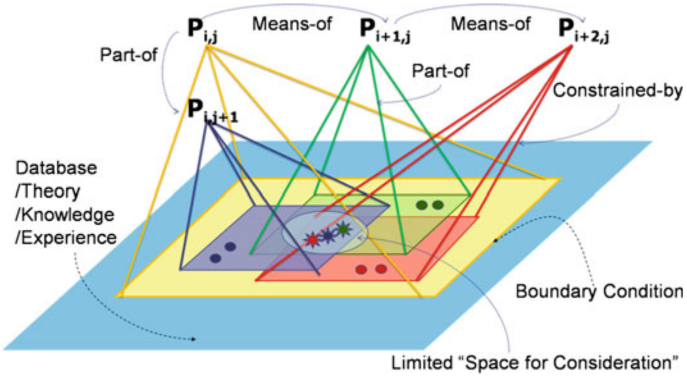

Fig. 8.3 Design synthesis as a progressive constraining process

edges of the diagonal band represent the external system constraints, and the Bounding (*β*) operation indicates the reasoning "forces" from constantly emerging internal system constraints. Note that in synthesis reasoning, the Bounding (*β*) operation establishes a "constrained-by" dependency relationship.

Using such a framework, design synthesis can be regarded as a progressive refinement process. Figure 8.3 below provides an alternative perspective to look at various design constraints in design synthesis. Generally speaking, synthesis is, given an abstract $P_{i,\ j}$, to arrive at a concrete $P_{i\ +\ 1,\ j\ +\ 1}$ that "is-a" tangible "thing". Rather than randomly mapping from $P_{i,\ j}$ to $P_{i\ +\ 1,\ j\ +\ 1}$, design synthesis must follow a systemic manner to firstly establish a "space for consideration", then to ideate possible solutions within this limited space. In this process, $P_{i,\ j}$ as the given intent, serves to define the input constraints; whereas $P_{i\ +\ m,\ j\ +\ n}$ (i.e., the representation of previous decisions made) functions to progressively construct the system constraints via "constrained-by" dependency relationship. Note that, the proposed synthesis reasoning process is based on the Axiomatic Design, and its application scope is limited to the creative engineering design at early stages.

8.4 Conclusion

In design synthesis, to impose certain design constraints to a technical system is an important decision which will affect further evolvement and final outcome of the technical system. Such decision cannot be made randomly, but rather it must be consistent with the initial design goal and all previously decisions. This task is particularly challenging at early design stages where constraints are still intangible, subjective and dynamic. This paper presents the initial development of a new constraint management model for synthesis reasoning. Within the proposed model, design constraints are firstly classified into four types: internal input constraint, external input constraint, internal system constraint, and external system constraint. For each type, the model prescribes a unique management strategy. Finally, we further incorporate the above studies into a domain-independent synthesis reasoning framework in order to systemically build boundary conditions (i.e., constraints) for design synthesis. Future works of this research include developing a tracking method to quantitatively trace the dynamic changes of design constraints and an adductive reasoning based diagnosis method to identify the early violation of constraints. A series of design experiments will also be conducted to illustrate the performance of this new model.

References

1. Lu SC-Y, Liu A (2011) A Synthesis Decision Framework for Early-stage Innovative Design. Proceeding of 2011 CIRP Design
2. Moreau CP, Dahl DW (2005) Designing the solution: the impact of constraints on consumers' creativity. J Consum Res 32(1):13–22
3. Costello F, Mark K (2000) Efficient creativity: constraint-guided conceptual combination. Cogn Sci 24(2):299–349
4. Stokes PD (2001) Variability, constraints, and creativity: shedding light on claude Monet. Am Psychol 56(4):355–359
5. Jeon H (2000) Dynamic constraint management in collective design. Pro Quest Dissertations and Theses
6. Goldratt E (1999) Theory of constraints. North River Press, Croton-on-Hudson
7. Lu SC-Y, Liu A (2011) Subjectivity and objectivity in design decisions. CIRP 7 Ann Manuf Technol 60(1):161–164
8. Suh NP (1990) The principles of design. Oxford University Press, New York, p 401
9. Scott MJ, Antonsson EK (1999) Arrow's theorem and engineering design. Decis Making Res Eng Des 11(4):218–228
10. Shpitalni M, Lipson H (1997) Automatic reasoning for design under geometrical constraints. CIRP Ann: Manuf Technol 46(1):85–88
11. Jaffar J, Maher MJ (1994) Constraint logic programming: a survey. J Logic Program 19(20):503–581
12. Lottaz C, Smith IFC, Robert NY, Faltings BV (2000) Constraint-based support for negotiation in collaborative design. Artif Intell Eng 14:261–280