Designing as Balance-Seeking Instead of Problem-Solving

Filippo A. Salustri
Ryerson University

Damian Rogers
Ryerson University

Nathan L. Eng
University of Cambridge

Please Cite:
Salustri, F. A., Rogers, D., & Eng, N. L. (2009). Designing as balance-seeking instead of problem-solving. *Design Principles and Practices: An International Journal—Annual Review 3*(3), 343-356.
doi:10.18848/1833-1874/CGP/v03i03/57883

Designing as Balance-Seeking Instead of Problem-Solving

F.A. Salustri, D. Rogers, and N.L. Eng

Published in Design Principles and Practices, 3(3):343-356,2009.
http://ijg.cgpublisher.com/product/pub.154/prod.225

Introduction

Designing can have dramatic impact on quality of life, economic well-being, and "progress." The full extent of this is something that practitioners, researchers, and perhaps most especially the public are still comprehending. Designing appears to be an activity that is unique to humans – indeed, it has been suggested by the authors elsewhere [7] that a "designerly organism" has an evolutionary advantage: the first designed artifacts were likely physical tools that helped their makers survive. Today, designing is an ubiquitous activity, and is increasingly acknowledged as essential in business, policy, and other socially constructed areas. Given our ever-increasing understanding of the importance of designing in human existence, it is not unreasonable to want to make sure we are doing it as well as we can.

It is in this spirit that the authors consider the matter of good designing at a philosophical level; specifically, we consider the apparent sympathy between *designing* and *problem-solving*. We will suggest that with respect to a conventional or "lay-person's" sense of problem-solving, designing is *not* a problem-solving activity. We will also argue that thinking of designing as *balancing* could stimulate new perspectives on designing, and lead to new (and hopefully beneficial) methods and processes.

We begin by defining some key terms to our arguments.

Designer is used in this paper as a generic term to denote all active participants in a design process; it is not meant to identify single individuals.

A *model* is any representation of a thing used to understand it from a specific perspective for a particular purpose. Models are by definition imperfect because they must ignore, abstract and structure what is present in the thing being modeled in order to be usable. Designers must therefore maintain awareness of the resultant limitations built into any models used.

We assume all agents exhibit *bounded rationality*, which, per Simon [8], denotes both imperfect knowledge and imperfect reasoning by agents.

A *situation*, per Gero [3], is a collection of active entities (agents, human or otherwise) that influence one another's behaviours in carrying out activities. In designing, we consider the designer, the design, all other agents and entities, as well as information present in the form of experiences, expertise, etc. to all be part of the situation. "Situation" is a term used to denote a real-world thing.

A *context* is a model of a situation, containing as much relevant information as necessary for a specific purpose. While some researchers describe designing as being "situated," we prefer to distinguish between things as they *are* (a situation) and models of those situations. We can only work with models, because we cannot perceive the situations directly. Thus, we prefer to think of designing as *contextual*, to absorb the notion that those context models are not perfect, due to bounded rationality.

A *framework* is an implicit description of a family of processes, intended not to describe the actual processes, but intended to describe the goals that the family of processes can achieve if properly implemented.

A *(set of) requirements* - aka a design brief - is a model of a design problem. As such, requirements do not fully express a design problem, but rather express it as well as can be expected in a given situation, and given bounded rationality.

Designing as Problem-Solving

Design is often thought of as a kind of problem-solving activity, or having characteristics in common with problem-solving. Generally, we think of a design problem as an "undesirable" context, which is corrected by designing something. Since the context (a model of a situation) includes all pertinent agents including designer, we may also think of the context as *changing itself.* There is extensive literature on problem solving in several disciplines and as a framework, some of it at odds with the layperson's conception. For example, in psychology, problem-solving admits there may be many possible solutions to a problem, and in mathematics this is often the norm (e.g. [4]); however, in typical activities such as banking, designing, construction, surgery, etc., problem-solving is taken to imply a single right solution that eliminates the problem permanently. It is certainly true that in many kinds of design practice and research, a "problem" is usually defined in the layperson's sense. This kind of difference between the academic and layperson's definitions of terms is quite common. "Theory" is another term, which in science and mathematics have very specific senses that are significantly different for the common usage. It is therefore not unreasonable to adopt the lay sense of problem solving as the convention.

However, the fit between designing and problem-solving, in the lay sense, is not very good. Some efforts to address this mismatch have been pursued in design (e.g. [6]) but have not really been accepted in practice.

The authors have identified four key characteristics (so far) of problem-solving frameworks that highlight the misfit with designing; we describe them here.

Solutions end problems. While it may be that a problem sometimes vanishes upon solution, we cannot predict the nature of the new context because of bounded rationality. The context resulting from implementation of a solution is also imperfect, and is only one possible outcome of the designer's activity. Therefore, we can expect new problems to arise because of the solution. Indeed, it is not so much that the problem *ends* upon solution; but rather that it just *changes*. We might consider, then, as a general constraint on designing in the problem-solving framework, that developing a solution must minimize the detrimental impacts of future problems that arise from the implementation of the designed solution. However, it is difficult, if not impossible, to know what those future problems are.

Problems are static once specified. This is a common misconception, based on the common practice of "freezing" requirements, which model a design problem, before intensive designing starts. If they were not frozen, they could easily change faster than the designers could keep up. Thus, in current practice and for practical reasons, requirements are typically frozen to give the designers a fixed target, as it were. However, freezing the requirements does not freeze the problem on which they are based. Indeed, a significant amount of project management in many large-scale design tasks (e.g. buildings, urban planning, commercial or military aircraft) arises directly from managing changes to requirements after they are supposed to be frozen, but that have become "stale" due to unforeseen external forces. Furthermore, requirements are only imperfect models of design problems, which further increase the likelihood that there will be errors that must be corrected after the requirements are frozen. If the requirements remain frozen, the designers risk developing a wrong solution for what the problem will have become. This dilemma has been recognized, at least implicitly, by the drive toward shorter lead-times; the shorter the lead-time, the less likely the requirements are to have become stale. However, as the rate of technological, social, political, and economic change continues to increase, it will become ever harder to develop designs fast enough. The fundamental matter here is not that design problems change, but that we treat them as if they were fixed.

Design problems are solved by choosing the "best" alternative solution. The term "best" is a relative term, but is usually treated as an absolute. That is, in problem-solving, there is a tacit assumption that it is possible to unequivocally identify the most appropriate solution. In fact, "best" should be defined with respect to: the design alternatives known to the agents (a function of bounded rationality); the accuracy and completeness of the requirements (which should be allowed to change over time); the accuracy and completeness of the expected future context when the design solution will be implemented; and other factors. However, problem-solving as it is commonly practiced ignores effects one might anticipate in the future, usually because they cannot be quantified to the same degree as the requirements (which are essentially historical in nature). Without a willingness to consider the future effects of design solutions, and to build requirements that account for those effects to the best of our ability, the consequences of our design interventions cannot be controlled or mitigated at all. Yet, the conventional problem-solving framework provides no mechanisms for this.

Problems are typically solved algorithmically or heuristically. We often name as "problems" situations for which a solution can be worked out algorithmically or heuristically. For instance, finding the roots of a binomial equation is a typical "problem" that is "solved" by students. The prevalence of this view of problem-solving implies that design problem-solving is similar, but this is not true. Designs emerge through human interaction, over time, and via reflection about the context. Preliminary design implementations and prototypes are very often used to elicit information from users that further clarify the specifics of why the "as-is" situation is undesirable. This reinforces the idea of coevolution of problem and solution [6], where the act of solving a design problem illuminates the problem itself. The authors have argued elsewhere [7] that designing is an appropriate approach precisely where algorithmic and heuristic methods fail - i.e. with so-called wicked problems. In any case, design appears to be non-algorithmic and non-heuristic, which places it at odds with the conventional notion of problem-solving.

We propose that these four characteristics of problem-solving make it an inappropriate way to think about designing. In the next section, we will explore another way to think about design and suggest it has particular merits over the problem-solving framework.

Designing as Balancing

The authors believe that continued use of the problem-solving framework for designing will not be particularly fruitful, given the mismatches between problem-solving and design that we outlined above. Over the last few years, we have developed an alternative view that we think could be more beneficial: that designing is balancing a situation.

The authors regard a "problem" as a mismatch between one's goals or desires and "how things are." This perspective is based largely on Alexander's work (*misfits* in [1], and *conflicting forces* in [2]). We might argue that problems are just models of situations containing these kinds of mismatches, but one might interpret this as meaning we intend that situated mismatches is a more fundamental perspective. We do not intend this; rather, we suggest that thinking in terms of situated mismatches is just a more designerly way of thinking (as opposed to artistic or scientific thinking).

The current state ("how things are") can be described in terms of a set of (not necessarily physical) forces that result in a condition of some stability. Consider ground-source heating (GSH) as an example. GSH is generally recognized as a sustainable way to heat a home with low operating costs, but GSH is more expensive to install than other less sustainable technologies. Therefore, GSH is not very popular. The forces here are capital cost and pollution on the one hand, and operating cost and sustainability on the other. These forces balance at a point where relatively few people use GSH. As these forces change, the balance point will change, and more (or fewer) people will use GSH.

Designing, then, changes a situation through the introduction of a new product, process, or other artifact, such that the forces change, bringing the balance point closer to one consistent with our goals and desires.

We can visualize this as in Figure 1, which shows "before and after" renderings of a context. The central node is the feature, product, or entity of consideration (such as GSH in the example above). The white nodes represent sources of mismatch – what Alexander would call "misfit variables" – that exert forces denoted by arrows on the central node. The forces caused by the mismatch variables balance each other in a particular way. The left diagram might model the current state of GSH, and the skew in the diagram might indicate GSH is not popular because of its large capital cost. The right diagram shows a different balance point, one in which GSH might be more widely adopted. To get to the new balance point, some design intervention was undertaken. Note that the intervention need not have been directly on GSH technology, but may have been a redesign of regulations, or of other aspects of the context. We note that balance is never perfect because we are all subject to bounded rationality, and the models we use are always imperfect.

It is also important to recognize that the authors are considering the impact of the overall design activity - of design itself - and not the impact of individual artifacts on a situation here.

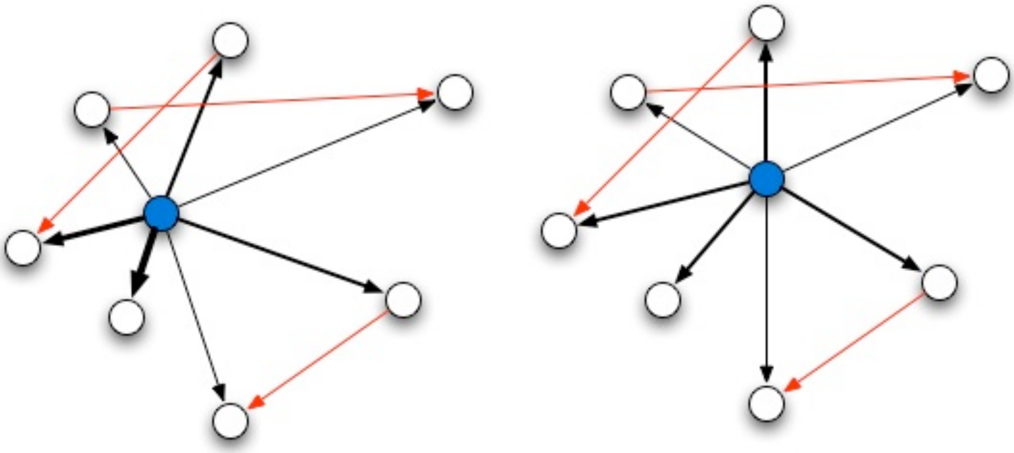

Figure 1: (left) an unbalanced situation; (right) a more balanced situation.

Finally, we note that assuming some (possibly qualitative) metric can be associated with the mismatch variables (to determine the strength of the forces), it is possible to use simple graph theoretic algorithms to explore how balance points are reached. This is a matter of on-going research for the authors.

Similarities with “Control Theory”

If the notion of *designing as balancing* (DaB) holds, then we can consider designing as a phenomenon that responds to variable inputs. The ensuing response (a new design) will alter the situation in a measurable way. One can use such measurements to set an “error” between how things are and how we wish they were (i.e. an imbalance). This error motivates further design to improve the balance. The result is a closed-loop process, in which designers respond to the effects of their own work with further design. A good analogy might be to riding a bicycle: the rider is constantly adjusting his centre of gravity and the position of the handlebars to maintain balance while wind, bumps, and other disturbances seek to unbalance him. To the authors, this is suggests an analogy to a well-known area of study: control theory. We note that “control theory” could also be called “adaptation theory” in that the “control elements” (see diagram below) adapt the behaviour of a system to account for uncontrollable "disturbances."

A typical visualization of a controlled system with feedback, and an example of that kind of system, is given in Figure 2. The "plant" is an entity the output of which can vary from some ideal as a result of external disturbances. The plant's output is analyzed and compared to the ideal (or desired) output. The "control element" then performs some activity that alters the plant's behaviour to offset the disturbances and render an output closer to that desired. In the example, we show how a domestic furnace, thermostat, and thermometer can keep a home at the same temperature regardless of external thermal disturbances. In the example, we have colour-coded the blocks to facilitate seeing how we transformed this basic diagram to show how DaB works.

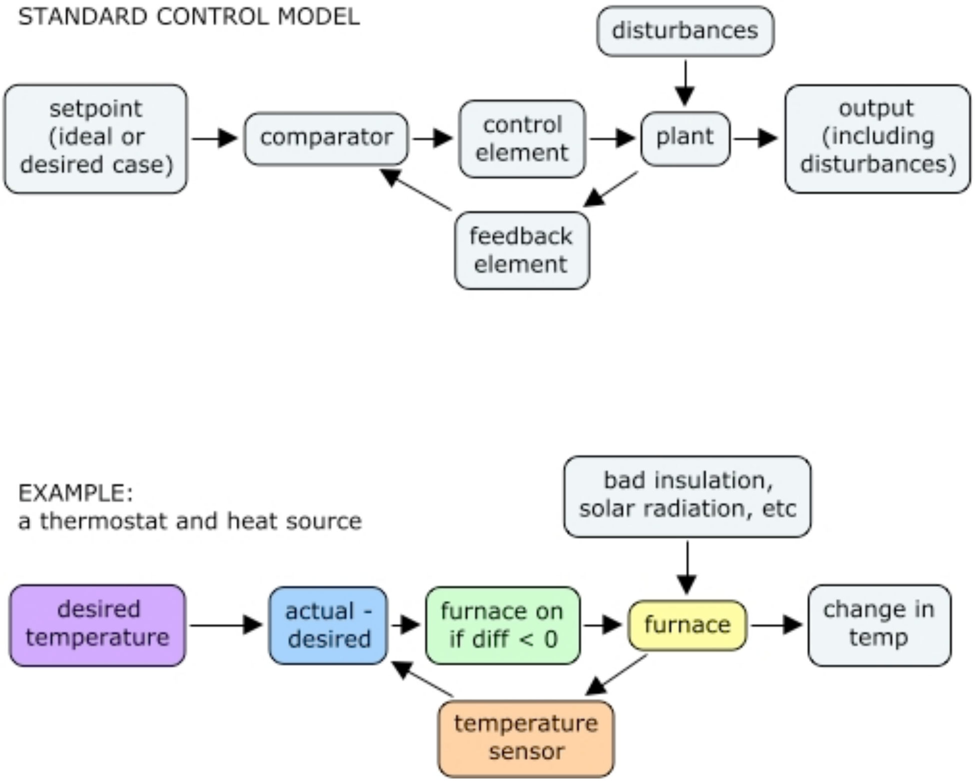

Figure 2: A typical block diagram for a control system with a feedback loop, and example.

The authors have developed two interpretations of the control block diagram with respect to DaB. The first is based on the premise that if designing is balancing, then the designer is the balancer. That is, the design takes the place of the "control elements" (Figure 2). This is shown in Figure 3, which is topologically equivalent to the standard control diagram. The colours in Figure 3 correspond to those used in Figure 2, to help the reader see the correspondence of the two diagrams. There are two substantive changes in Figure 3:

1. One situation changes over time because of actions taken by designers. To show this, we have drawn a single all-encompassing situation node.

2. The pink box is new: it includes models of pertinence to the control theory analogy being made. That is, everything but the designer and the designer's artifacts (the product) exist only in the designer's mind.

While this representation of DaB is quite simple, it does appear quite reasonable so far.

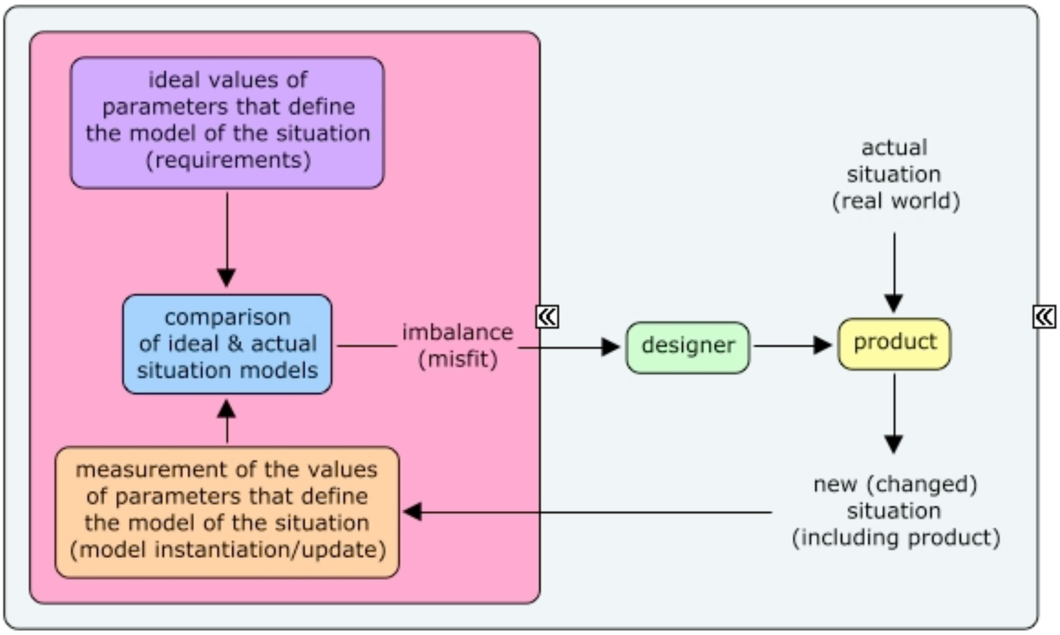
ideal values of parameters that define the model of the situation (requirements)
comparison of ideal & actual situation models
imbalance (misfit)
measurement of the values of parameters that define the model of the situation (model instantiation/update)
designer
product
actual situation (real world)
new (changed) situation (including product)

Figure 3: An interpretation of DaB with the designer as the "control element."

A second interpretation of DaB with respect to control theory is possible. In this interpretation, the designer assumes the role of the "comparator" in the standard block diagram. While the graphical elements have been rearranged here, the topology of this interpretation is consistent with the standard control diagram in Figure 1. Here, the designer embodies the process of specifying precisely the nature of the mismatches between the current state and the required (or desired) state. The control elements now become the sum total of all possible information that can be brought to bear, including the expertise, preference, and judgement of the designer as well as the natural laws. They are the control elements in that, due to bounded rationality, they limit the scope of possible designs. This interpretation too does not present any obvious flaws, insofar as the authors have ascertained so far.

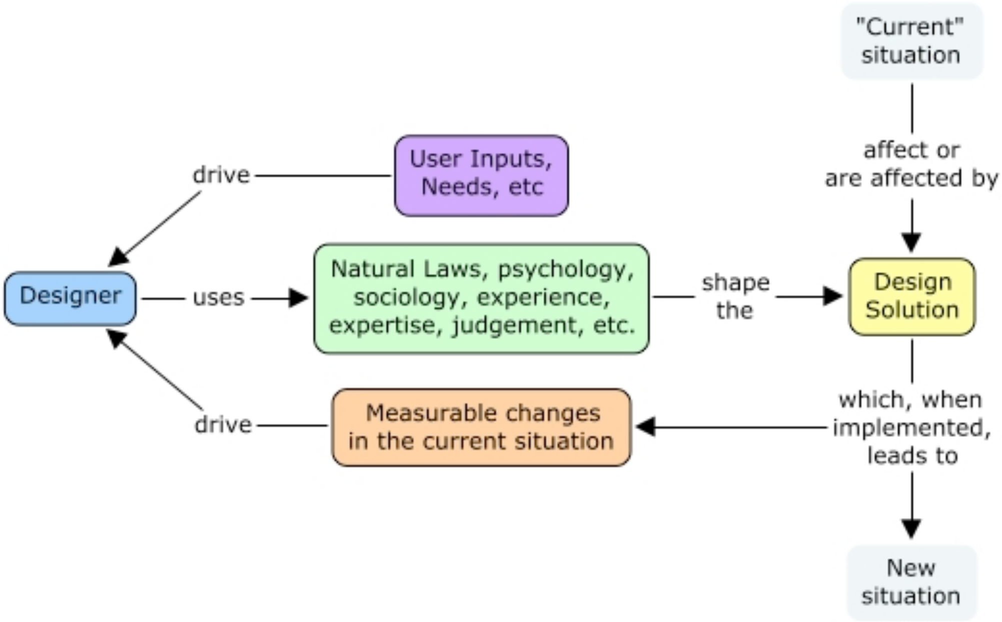
"Current" situation
affect or are affected by
User Inputs, Needs, etc
drive
Designer
uses
Natural Laws, psychology, sociology, experience, expertise, judgement, etc.
shape the
Design Solution
which, when implemented, leads to
Measurable changes in the current situation
drive
New situation

Figure 4: An interpretation of DaB with the designer in the role of "comparator."

In the case of either interpretation, the authors are attempting to capture the macroscopic interaction of designing with the rest of "reality." In both interpretations, we are trying to show a dynamic and continuous interaction between design practise and everything impacted by designing. The interactions are intended to keep the current situation as well-balanced as possible.

Control theory is, of course, far more than just the basic feedback loop we have used here. Our next research question is: assuming the analogy between DaB and control theory as exemplified in Figures 3 and 4 actually holds, what other aspects of control theory can we pass through the analogy to derive new hypotheses, methods, and tools for designing? We have yet to begin addressing this matter.

The authors note that while control theory is commonly associated with artificial systems and the influence of humanity on nature, the control "circuits" like the feedback loops used in Figures 3 and 4 are good models for various naturally occurring phenomena. Many physical, ecological, and biological systems can be modelled - at least partially - with feedback-based representations. As such, we suggest that DaB as depicted here is consistent with our understanding of natural systems. This presents a possibility of connecting DaB to such areas as "natural design" and sustainable design practise. This too will be investigated by the authors.

It may seem that DaB cannot account for so-called "disruptive" design, but this is not necesssarily true. "Disruptive" – indeed, even "balance" – are relative terms. They are defined with respect to some context (a model of real life situations). Thus, the same phenomenon may be disruptive from one perspective, yet balancing from another. The introduction of products like the Apple iPhone or the first military tank has been described as disruptive. However, it is not because they imbalanced a situation, but because of the sudden overall shifts in balance – of market on the one hand and of military/political power on the other.

The iPhone shifted the balance of market significantly in Apple's favour, a situation which seems clearly to have been Apple's intent. Similarly, the tank moved the balance of military power towards a "better" situation for the country that introduced it. Thus, disruption, in the view of the authors, seems to be more a description of the speed with which balance changes rather than whether balance is changed.

Influences of Designing as Balancing

To more fully consider what DaB might mean in practise, the authors consider in this section some of the implications of DaB with respect to some aspects of design activities.

Context and requirements

Ideally, a fully implemented DaB system would result in nearly instantaneous (with respect to the time scale of the process) transmission of contextual changes to the design, and update of appropriate documentation, throughout the design process. This also opens the possibility of allowing clients and users to engage more fully in the design process – a-la participatory design – and interact more with the designers.

This would fundamentally change how requirements and design briefs are defined and managed. In the conventional problem-solving (PS) framework, requirements are typically given as expectations of and on the artifact. However, in DaB, requirements focus on context and include three parts:

- the identified "forces" acting in the context,
- measures of the perceived imbalances in the forces, and
- a rationale of why the imbalance is undesirable, which implicitly gives guidance on possible avenues to improve the balance.

In other words, with respect to requirements, problem-solving focusses on defining expectations for the future, while DaB focusses on determining the inadequacies of the present.

If requirements were allowed to change throughout the design process, there would also be a significant impact on the legal status of those requirements. In many industries, requirements or design briefs carry certain contractual obligations: accepting a set of requirements may be (and usually is, in fields like engineering design) construed as a contract to "solve a problem." A detailed study of the legal implications of DaB, with respect to requirements specifications, is beyond the scope of this paper, but it is certainly a significant issue to be treated before DaB can ever gain widespread acceptance.

Design parameters

The authors use the term design parameter to denote a named value that characterizes some aspect of a design. They describe a design that relates back, usually quite directly, to particular requirements.

In PS, design parameters are assumed to be fixed by the time the design is finalized, because they strictly characterize the design as an isolated entity. They are typically a single value possibly with a tolerance (a range of acceptable but sub-optimal values) such that any value outside the tolerance range is entirely unacceptable. Tolerances are typically used only in designs that will have to be directly manufactured (e.g. engineering and related high-technology fields).

In DaB, design parameters must more readily accommodate the possibility of changing requirements. This can be done quite easily by using intervals rather than single values. An overview of intervals is available in [5]. For our purposes, it suffices to think of an interval as a value-range, typically marked by its limits, that brackets all reasonable values without prejudice to one particular value or another. Intervals have arithmetic, algebraic, and calculus operations in direct analogy to the more conventional single-valued mathematics, and can be used in both quantitative and qualitative situations.

How DaB treats design parameters will alter how change management happens. In typical practise, change management can account for a large portion of design cost, because of its administrative overhead. There is vigorous ongoing research to look for ways to eliminate the need for changes, especially in the technical design disciplines. Unfortunately, this is, in the authors' opinion, exactly the wrong thing to do. Changes will happen; we believe trying to control or stop them is doomed to failure. A better course of action is to develop streamlined and highly flexible methods of managing change. For example, the opensource software community has developed a variety of lightweight change management processes, some of which could be applied to non-software situations. This is a separate research question that will be covered in a future publication.

Designing affordances into products is potentially different in DaB than in PS. In PS, affordances designed into products are meant to address the problem directly. In DaB, a given design is only part of an ongoing and evolving process that designers must include in their thinking. In DaB, then, one must consider how any affordances designed into a product will affect future situations. One might consider making more adaptable or flexible the affordances of products being designed currently. The benefits of doing so include product longevity, greater product use, and possibly broader or larger market; but there are also risks including product misuse, greater chance of product failure, and increased liability.

Designing affordances relates to the use of intervals for design parameters. Smaller intervals generally mean more limited and targetted affordances. Smaller intervals lead to products that are generally easier to build, but will be more specific in function, affordance, and, therefore, usefulness and scope. Intervals can therefore be used as a tool to help designers trade off all these issues.

Concept evaluation changes

Any decision-making method can be used to evaluate design concepts. Interval-based design parameters can be used in all decision-making methods. So, concept evaluation can very easily incorporate design parameters as described above. However, the result of a concept evaluation exercise using intervals will be an interval itself. This means that, firstly, more concepts might be identified as preferred (if concepts' intervals fall partly above a preference threshold) and, secondly, the intervals of design concepts being compared may overlap. This will require changes to the way we select the "best" concept in a given comparison, but it will also afford opportunities. For instance, overlaps between concept intervals suggest that those concepts can be grouped and that new concepts could be derived that maximizes interval coverage; such maximizations could indicate broader usefulness of products.

Similar changes can be expected in other methods of design analysis, comparison, and evaluation, such as design critiques. All this leads us to believe that opportunities for new design methods exist in a DaB framework, compared to PS frameworks.

Need for more careful analysis of possible futures

Underlying many of the possible changes resulting from the adoption of DaB is the notion of being more mindful of the future impact of designs. There are various tools (e.g. in engineering design, Failure Mode and Effects Analysis) that can be adapted to work in diverse design settings. These tools help designers predict what can go wrong with a product, so that those problems can be “designed out” of the product before it goes to market. To extend such methods beyond the technical design fields will require using methods of futurology and future studies, and include the construction of scenarios of what might happen once a product is introduced.

This relates to the description of future situations that include the artifact/product being designed. Given such descriptions of future situations, one can study the balance of those situations and then consider how to change the current design to accommodate perceived shortcomings in those future situations.

Prototype/test activities change

Since DaB focusses on addressing the imbalance of a situation – which is a direct result of forces external to the control of the designer – prototyping and testing of early concepts with the involvement of users or clients becomes even more important. Each external participant in such activities will bring a particular and possibly novel point of view to the activity, any of which may reveal inadequacies in the design.

Prototyping and testing is also important to evaluate performance. Given the potential for increased adaptability of designs developed under DaB frameworks, this is an important issue. Increased adaptability usually means increased efficacy, but increased efficacy also usually means decreased efficiency. A suitable balance between efficacy and efficiency is very important for a successful design, and prototyping and testing is a fundamental way to gather the necessary information for such assessments.

Conclusions

In this paper, the authors have proposed that designing might be more fruitfully thought of as a process of balancing rather than one of problem-solving. Within the length limits of this publication, we have tried to give a high-level overview of our framework, but at the expense of much detail. We have explained how features of problem-solving as understood by a general audience conflicts with features of designing, and how designing as balancing (DaB) addresses those conflicts. We characterize DaB as a framework that is adaptive, responsive, systems-oriented, futurological, and based on the use of intervals rather than single-point values.

DaB is a framework that would, if instantiated in design practise, include less administrative overhead for tracking and recording information, lead to faster response and shorter lead times by design teams, and enable users to participate more fully in the design activity. DaB should lead to more sustainable practises that share many features with processes that occur in nature.

We have not proposed, yet, any particular methods by which these DaB can be instantiated in practise; instead we have provided a general description of the "appearance" – the look and feel, as it were – of what design practise might be, having adopted DaB. Our description in this article could be considered a requirement set itself - a design brief for DaB practise - and for which an appropriate solution must now itself be designed. This is ongoing research and we expect to continue this project for several years.

Whether DaB would ever be adopted by practitioners is also not clear; a framework shift such as that proposed here can be hard to adopt due to corporate culture, resistance to change, and perceived risk. However, until such time as there is actual evidence that DaB would not be taken up in practise, we will continue to pursue it.

Acknowledgements

The authors acknowledge the support of the Natural Sciences and Engineering Research Council of Canada, and the feedback from participants at the 2009 Design Principles and Practises Conference.

References

1. Alexander, Christopher. *Notes on the synthesis of form*. Boston: Harvard University Press, 1964.
2. Alexander, Christopher, Sara Ishikawa, and Murray Silverstein. *A Pattern Language: Towns, Buildings, Construction*. USA: Oxford Univ Press, 1977.
3. Gero, J.S. and R. Sosa. "Creative Design Situations." In Proc. CAADRIA2002, ed. A. Eshaq, C. Khong, M. Neo, and S. Ahmad. New York: Prentice Hall, 2002.
4. Funke, J. "Experimental research on complex problem solving." In *Complex problem solving: The European Perspective*, ed. P.A. Frensch and J. Funke. Hillsdale, NJ: Lawrence Erlbaum Associates, 1995.
5. Habib, Walid, and Allen C. Ward. "Proving the Labeled Interval Calculus for Inferences on Catalogs." In *Design Theory and Methodology*. ed. James R. Rinderle and David G. Ullman. New York: ASME, 1990.
6. Poon, J. and M.L. Maher. "Co-evolution and emergence in design." *Artificial Intelligence in Engineering* 11 (1997): 319-327.
7. Salustri, Filippo A., and Nathan L. Eng. "Design As...: Thinking of What Design Might Be." *Design Principles and Practises* 1 (2007): 19-28.
8. Simon, Herbert A. "Models of Bounded Rationality, Volume 1, Economic Analysis and Public Policy." Cambridge: MIT Press, 1984.