V. Srinivasan
PhD Student
e-mail: srinivasan@cpdm.iisc.ernet.in

Amaresh Chakrabarti
Professor
e-mail: ac123@cpdm.iisc.ernet.in

Centre for Product Design and Manufacturing,
Indian Institute of Science,
Bangalore, 560 012, India

An Integrated Model of Designing

Current design models and frameworks describe various overlapping fragments of designing. However, little effort exists in consolidating these fragments into an integrated model. We propose a model of designing that integrates product and process facets of designing by combining activities, outcomes, requirements, and solutions. Validation of the model using video protocols of design sessions demonstrates that all the constructs are used naturally by designers but often not to the expected level, which hinders the variety and resulting novelty of the concepts developed in these sessions. To resolve this, a prescriptive framework for supporting design for variety and novelty is proposed and plans for its implementation are created. [DOI: 10.1115/1.3467011]

1 Introduction

Designing is a process that spans from the identification of a need to the development of solutions to satisfy the need. Designing involves multiple facets: product, process, people, methods, tools, organization, microeconomy, and macroeconomy [1] and is complex due to interactions within and among these facets.

The overall aim of design research is to improve the chances of producing a successful product by making designing more effective and efficient by developing knowledge in various forms [2]. This aim can be realized by (a) formulating and validating an understanding of the current designing in the form of models and theories and (b) developing and validating support in the form of frameworks, guidelines, methods, and tools founded on the current understanding, in order to improve the current designing. Since designing requires knowledge of its facets, design models and frameworks should take into account the knowledge needs of these facets.

The current literature contains various design models and frameworks, a majority of this deal with the process or product facet. However, a review of the models and the frameworks (see Sec. 4) reveals a lack of shared understanding across these models as to how designing takes place. Each model focuses on some (fragment) of these facets and often uses different constructs to represent the same product or process element [3]. Currently, little effort has been taken to consolidate these different fragments into an integrated model of designing. The overall objectives of this paper are the following:

(a) Develop an integrated model of designing that combines significant product and process elements identified from literature.
(b) Evaluate the model by checking whether all its elements are present in current design sessions.
(c) Propose a framework based on the model to be supported on an interactive platform.

2 Motivations

2.1 Conceptual Design. Pahl and Beitz [4] defined conceptual design as a phase in which solution principles are developed. French [5] distinguished conceptual design from the other design phases in terms of its greater fluidity and flexibility, thus, offering maximum scope for striking improvements. Pahl and Beitz [4] argued that a lasting and a successful solution is more likely to spring from a choice of solution principles than from an exaggerated concentration on technical details. Berliner and Brimson [6] argued that on average, about 80% of a product's cost over its life cycle is committed during conceptual design. At this early phase, usually only conceptual sketches and rough schematics with little additional information are available; yet, nearly all the important decisions have to be made during the phase [7].

2.2 Activities. The engineering design is viewed as human problem solving [8,9]. However, designing involves both problem finding and problem solving [10]. An activity in designing is defined here as a deed of problem finding and problem solving and, therefore, is a type of process facet. Activities in designing play a significant role in the success of the end product and successful products can be developed if activities in the early phases of designing can be executed proficiently [11]. It is, therefore, important to identify the significant activities in designing. Several researchers used activities in designing. For instance, Stauffer and Ullman [12] proposed *generate*, *evaluate*, and *decide* and Blessing [8] proposed *generate*, *evaluate*, *and select*, etc.

2.3 Outcomes. During designing, designs evolve through multiple levels of abstraction, each level provides a particular description of the design where descriptions at higher levels of abstraction provide greater flexibility in the interpretation of the design by committing less to its details. An outcome is defined here as a property of a design at an abstraction level and, therefore, is a type of product facet. Therefore, outcomes exist at different levels of abstraction. As designing progresses, the abstraction reduces and the outcomes become more detailed. Nidamarthi and coworkers in [13,14] stressed the importance of capturing outcomes since these influence fulfillment and satisfaction of requirements. Researchers proposed outcomes at various abstraction levels. For example, Blessing [8] proposed *problem*, *requirements*, *functions*, *concept design*, and *detailed design* and Gero [15] proposed *functions, expected behavior, actual behavior, structure*, *design description*, etc.

2.3.1 Physical Laws and Effects. Physical laws and effects are principles of nature that govern change [16]. A physical law, in its widest sense, represents the functional connection between variables, geometric parameters, material constants, and basic constants [17]. Natural laws (comprising physical laws and effects) provide important information for supporting invention and development of artifacts [18]. Zavbi and Duhovnik [17] argued that if operation of existing technical systems can be explained using physical laws, then, these can also be used to design technical systems. They considered physical laws as the basic and richest source for designing. Designing using physical laws prevents a designer's fixation on adaptations of the existing solutions or composition of solutions from the existing components [19] and provides greater ability to innovate [19,20]. A conceptual solution can be described as a causal network of physical effects [21]. However, synthesizing artifacts directly from physical effects is hard

Contributed by the AI/Knowledge Based Systems Committee of ASME for publication in the JOURNAL OF COMPUTING AND INFORMATION SCIENCE IN ENGINEERING. Manuscript received August 17, 2009; final manuscript received June 2, 2010; published online September 13, 2010. Guest Editors: Ashok K. Goel and Andrés Gómez de Silva Garza.

since effects have been created and described by scientists primarily for explanation of phenomena rather than for synthesizing artifacts that embody these phenomena [20].

2.4 Requirements and Solutions. Chakrabarti et al. [22] defined a requirement as a characteristic, which a designer is expected to fulfill through the eventual design. In designing, there can be requirements on product, process, people, tools, organization, and environment (i.e., facets of designing), only requirements related to product are addressed in this paper. Since designs go through several levels of abstraction during designing, each level can have its own requirements. A requirement is defined here as an expression of what a design should have at an abstraction level and is a type of product facet. A design process is initiated with the recognition of a need, leading to the establishment of requirements for the intended product [4]. Therefore, capturing requirements becomes essential and should be a central issue in design research [23]. Several researchers used requirements as an element in designing, e.g., *specifications* [4], *requirements* [13], etc.

A solution is defined here as a means to satisfy requirements and is, therefore, a type of product facet. As pointed before, designs proceed through several levels of abstraction during designing. This indicates that a solution can exist at various abstraction levels. Cooper [11] measured the success of a product in terms of how well it satisfied its requirements and found that profitable products satisfied the end user needs better than its competitors. Nidamarthi et al. [13] considered requirements satisfaction as a success criterion for its necessity in achieving customer needs. Several researchers proposed solutions as an element in designing: Cross [24] used *subsolutions* and *overall solution* and Nidamarthi et al. [13] used *solutions*, etc.

2.5 Summary and Specific Objectives. From the above, the specific objectives for this paper are formulated as follows:

(a) Develop an integrated model of designing that combines the significant elements of process-facet (activities) and product-facets (outcomes—e.g., physical laws/effects, requirements, and solutions).
(b) Evaluate the model by testing if it can be used to describe design sessions reflecting current practice.
(c) Propose a framework for designing to be supported on an interactive platform, based on the integrated model and observations from the evaluation.

3 Research Approach

The following approach is adopted for the specific objectives of this paper.

(a) Identification of activities, outcomes, requirements, and solutions: Current literature on designing is investigated to identify the various types of activity, outcome, requirement, and solution from which the significant activities, outcomes, requirements, and solutions are determined.
(b) Development of activity, outcome, requirement-solution, and integrated models of designing: The significant activities, outcomes, requirements, and solutions are put together to, respectively, develop models of activity, outcome, and requirement-solution. These individual models are combined to create an integrated model of designing.
(c) Evaluation of activity, outcome, and requirement-solution models of designing: The models of activity, outcome, and requirement solution are evaluated to check if they are inherently used in current designing where the model is not explicitly asked to be followed. Existing video protocols of design sessions from an earlier study [25], which were undertaken before the models are developed, are used for the evaluation. Each design session involved a team T1 or T2 having three designers each (with undergraduate in mechanical or architecture and masters in product design, T1 had an experienced designer—the rest were novice designers), solved a design problem P1 (develop conceptual solutions for an efficient means of keeping the university campus free from dry leaves) or P2 (develop conceptual solutions for a lock that does not require any physical key or numbers to remember), under laboratory settings using a method M1, M2, or M3 (Table 1). Overall, six designing sessions were used (Table 1—the number within the square bracket in each cell denotes the session number) for this evaluation. The designers were instructed to follow a discuss-cum-think aloud protocol, they were provided paper and stationery for documentation. Each design session was assisted by a researcher to clarify any queries the team had while solving the problem. The problems were solved back-to-back and discussions among the teams were not allowed. Even though the same team solved the same problem twice using different methods, this is not an issue for the objective of this research—the focus is on the degree of generality in problem solving across design sessions. Design sessions with problems solved using methods are used because rarely are design problems solved without methods and a general model should provide a basis for designing, both with and without methods. The video sessions are transcribed to include details of the speakers' identity, utterances, hand motions, etc. The transcriptions are coded using the definition of the identified activities, outcomes, requirements, and solutions and analyzed to identify and explain not only the constructs but also the various composite aspects such as synthesis and analysis. A two-way mapping is used for the protocol studies to check if all: (i) instances in the transcription could be represented using the constructs of the model and (ii) constructs of the model have instances in the transcription.

Table 1 Designing sessions

	T1	T2
M1 (functional analysis)	P1 [1]	P2 [2]
M2 (ideal design approach)	P2 [3]	P1 [4]
M3 (innovation situation questionnaire)	P1 [5]	P2 [6]

4 Identification of Activities, Outcomes, Requirements, and Solutions

The current descriptive and prescriptive literature on designing are reviewed to identify the different kinds of activity, outcome, requirement, and solution. The following general observations are made.

(a) The reviewed literature addressed activities, outcomes, requirements, and solutions either individually (e.g., Miller et al. [26] addressed only activities, Bhatta and Goel [27] addressed only outcomes, etc.) or in combinations (e.g., Stauffer and Ullman [12] addressed activities, requirements, and solutions and Pahl and Beitz [4] addressed activities, outcomes, requirements, solutions, etc.). No literature addressed activities, outcomes, requirements, and solutions and also made an explicit distinction between them (e.g., domain theory [28] addressed only outcomes but did not make a distinction between requirements and solutions).
(b) Activities are performed on outcomes, requirements, and solutions. For instance, Stauffer and Ullman [12] performed activities on requirements and solutions and Blessing [8] performed activities on outcomes, etc.

(c) In designing, outcomes evolved as requirements and solutions. For example, in the approach of Chakrabarti et al. [21], *function* and *solution principles* can be considered under requirements and solutions, respectively; in the approach of Pahl and Beitz [4], *specifications* can be considered as requirements and *solution principles*, *function structures, layouts*, etc., as solutions.

(d) Multiple levels of activity and outcome are identified. For instance, Stauffer and Ullman [12] classified activities into two levels: *category* and *operator* where each activity at the category level constitutes some activities at the operator level [12]; *generate, evaluate*, and *decide* are taken as activities at the category level and *select, create, simulate, compare, calculate, accept, reject, suspend, refine*, and *patch* are taken as activities at the operator level. Lossack [29] proposed outcomes at various levels: *requirement, functions, physical principle*, and *embodiments*.

The following are the observations on activities based on the literature survey (see Table 2 for a comprehensive list of activities proposed in literature, secondary level activities are represented within brackets following the primary level activity to which they belong).

(a) Different researchers sometimes used different words to mean the same activity. For instance, to mean *generate*, Stauffer and Ullman [12] used *create* and Blessing [8], Chakrabarti et al. [21], and Cross [30] used *generate*. Nidamarthi [14] used *identify* and *generate* and within *identify* and *generate* used *perceive*, *infer* and *modify*, *create*, and *modify* and *detail*, respectively, and Visser [31] used *develop* (*evocate and elaborate*), etc.

(b) Different researchers sometimes used the same word to mean different activities. For instance, Stauffer and Ullman [12] used *select* as a form of *generate* whereas Nidamarthi [14], used *generate* and *select* as separate entities. Nidamarthi [14] used *modify* as a form of *generate* whereas Stauffer and Ullman [12], used *refine* and *patch* (different forms of modify) as a form of *decide*, Chakrabarti et al. [21] used *modify* as a separate activity.

(c) Not all activities are explicitly mentioned in a model or framework. For example in Ref. [5], at the end of conceptual design, some schemes are *selected*. French [5] mentioned that *evaluation* is present in all the stages but did not mention similar presence of *generation* or *development* of schemes. It can be argued that without their generation or development, these schemes cannot be evaluated.

(d) The presence of feedback arrows could mean *rejection* of current outcomes followed by *generation* of alternative outcomes at the same level of abstraction or *rejection* of current outcomes followed by modification of outcomes at the immediate higher level of abstraction (from which the current outcome was generated). Feedback was used in Refs. [4,5,32].

(e) Points (a) and (b) indicate a lack of shared understanding, as pointed by Chakrabarti et al. [3], who argued that researchers often use different terms to mean the same thing and the same term to mean different things. An approach to alleviate this issue would be to develop a set of generic activities. We propose four generic activities in this paper: *generate*—an activity that brings an outcome into an episode, *evaluate*—an activity that judges the quality, importance, amount or value of an outcome in an episode, *modify*—an activity that changes an outcome in an episode, and *select*—an activity that decides an outcome as acceptable or unacceptable in an episode; an *episode* is defined as an event in designing that involves an exploration of an outcome. Note that *select* takes into account both *acceptance* and *nonacceptance*. Using these generic activities, most of the activities identified in existing literature can be classified, see Table 2. An additional category—others, is included to categorize those activities that cannot be classified using this generic set.

(f) *Explore* (meaning "to search and discover" [33]), used in Ref. [30], can be explained as a sequence of *generate, evaluate, modify*, and *select* and (ii) *communicate* [30,34] is not a part of problem solving but succeeds it. *Guide*, as used in Ref. [35], is not necessarily a characteristic of general problem solving because it may bias the solutions in a specific manner. *Represent* in Ref. [34] may be considered as a part of *generation*. In Ref. [36], (i) *justify* can be seen as a combination of *evaluate* and *select* and (ii) *solve* can be seen as a sequence of *generate, evaluate, modify*, and *select*.

The following outcomes are proposed in current literature: problem, requirement, function, concept design, and detailed design [8], function (as input-output) and solution principle (using physical laws and effects) [21], problem, selected schemes, arrangement drawings, and working drawings [5], process (input-output), function, organ, and part (as sketched layouts, dimensional layouts, detailed part drawings, etc.) [34], transformation (input-output), organ (wirk-elements), and part (as parts and assembly relations) [28], function, expected behavior, structure, actual behavior, and design description [15], structure (as components and relationships), behavior (using state and active functions), and function (as input state to output state) [27], function and attribute [37], needs, desires, requirements, problem, solution, design documentation and rational recovery statements, and design presentation [36], action, state change, phenomenon, effect, input, organs, and parts [16], function and means [38], specification, function structure, principal solution, module structure, preliminary layouts, definitive layouts, and product documents [32], specifications, principle solution, preliminary layout, definitive layout, and product documentation [4], requirement, functions, physical principle, and embodiments [29], need, specification, function structure, principal solution, structure, preliminary form, definite form, and product [7], functions, physical laws, basic schemata, and structure [17,39]. The following are the observations on outcomes:

(a) Outcomes are addressed at various levels of abstraction.

(b) Only the following researchers explicitly use laws and effects while addressing conceptual design: Chakrabarti et al. [16,21], Zavbi and Duhovnik [17], Lossack [29], and Rihtarsic et al. [38].

(c) The SAPPhIRE (State change, Action, Parts, Phenomenon, Input, oRgans, Effect) model [16] is investigated in more detail for reasons that follow. SAPPhIRE (Fig. 1) is a model of outcomes, developed to explain the causality of natural and engineered systems. The constructs of the model have been integrated from theory of technical systems [34], domain theory [28], metamodel [40], and function-behavior-state model [41]. The definition of the constructs is as follows: *phenomenon*—an interaction between a system and its environment, *state change*—a property of a system and its environment involved in an interaction and changes during an interaction, *effect*—a principle of nature that governs an interaction and comprises both physical laws and effects, *action*—an abstract description or high-level interpretation of an interaction, *input*—a physical quantity of the form of material, energy, or information that comes from outside the system boundary and is essential for an interaction, *organs*—the properties and conditions of a system and its environment that are required for an interaction, and *parts*—the physical components and interfaces that constitute the system and its environment. The model is explained as follows:

Table 2 Classification into generic activities

Literature	Generate	Evaluate	Modify	Select	Others
[12]	Generate (select and create)	Evaluate (simulate, compare, and calculate)	Decide (refine and patch)	Decide (accept, reject, and suspend)	
[8]	Generate	Evaluate		Select	
[21]	Generate	Evaluate	Modify		
[14]	Identify (perceive and infer), generate (create and detail), and select (identify)	Analyze (question, verify, weigh, relate, and visualize), evaluate (identify, question, relate, and verify), and select (compare)	Identify (modify) and generate (modify)	Choose (decide) and select (decide)	
[30]	Explore and generate	explore and evaluate	Explore	Explore	Communicate
[26]		Test	Operate	Exit	
[31]	Construct and develop (evocate and elaborate)	Evaluate		Accept and reject	
[35]	Formulate and generate	Evaluate			Guide
[34]	State, search, and prepare	Evaluate, verify, and check		Decide	Communicate and represent
[5]		Evaluate			
[15]	Formulate and synthesize	analyze and evaluate	Reformulate		Document
[36]	Generate, form, synthesize, specify, and solve	Justify, analyze, assess, evaluate, and solve	Revise and solve	Justify and solve	
[32]	Define, determine, search, develop, and prepare	Clarify, analyze, and evaluate			Divide and complete
[4]	Plan and clarify (find, formulate, elaborate, and adapt), develop (establish, search, combine, and firm), define (prepare), and prepare (elaborate and complete)	Plan and clarify (analyze and clarify), develop (identify), evaluate, calculate, define (check), define (check), and prepare (check)	Develop (upgrade and improve), define (refine and improve), define (upgrade and improve), and prepare (upgrade and improve)	Plan and clarify (select) and define (eliminate)	
[29]	Define, find, and describe	Evaluate		Select	
[7]	Generate, synthesize, and shape		Shape	Select	

components and interfaces of a system and its environment (parts) create properties and conditions (organs). When the system and its environment are not in equilibrium, there is transfer of a physical quantity in the form of material, energy, or signal (input) across the system boundary. This quantity with relevant properties and conditions together activate a principle (effect), which is responsible for an interaction (phenomenon) between the system and its environment. This interaction changes some property of the system and its environment (state change), which can be interpreted at a higher level of abstraction (action).

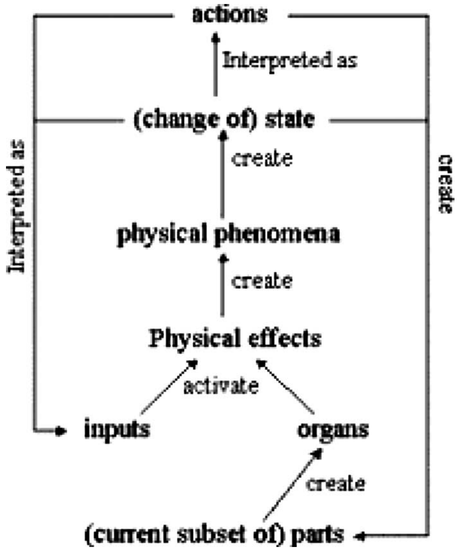

Fig. 1 SAPPhIRE model of causality [16]

The following are the observations on requirements and solutions (see Table 3 for the kinds of requirement and solution proposed in literature).

(a) Researchers used different terms to express requirements, probably to refer to requirements at different abstraction levels. For instance, requirements from customers are expressed as *needs*, which are refined by designers into a *problem* with an *objective function* and *constraints*, and used in designing as a list of *specifications* or *requirements*. In this paper, all forms of requirements are generally taken as *requirements*. Solutions are also addressed at different abstraction levels. See Table 3 for the list of generic requirements and solutions.

(b) Separating requirements and solutions is debatable because solutions at higher abstraction levels become requirements for developing solutions at lower abstraction levels.

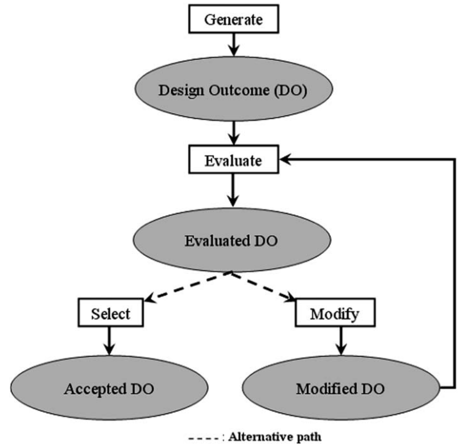

Fig. 2 GEMS activity model

(c) Solutions are generated from requirements and are modified to comply with the requirements and vice-versa. This leads to a continuous updating of the list of requirements and solutions during designing—co-evolution of requirements and solutions. A number of researchers (Table 3) reported the occurrence of co-evolution of requirements and solutions during designing.

5 Development of an Integrated Model of Designing

Table 2 suggests that most of the activities proposed in literature can be mapped into the generic activities. This indicates that the generic activities may be present and sufficient to describe designing. The generic activities together comprise the generate-evaluate-modify-select (GEMS) cycle (Fig. 2) for an episode. *Generation* brings a design outcome into an episode. This is followed by *evaluation* of the outcome, which checks its suitability against some criteria. If the outcome does not match the criteria, it is *modified* and *evaluated* again. A set of iterations ensue between modification and evaluation. If the outcome matches the criteria, it is *selected* as acceptable. If the outcome does not match the criteria even after modification(s) or cannot be modified, the out-

Table 3 Classification into generic requirements and solutions

Literature	Requirements	Solutions	Co-evolving
[12]	Constraint	Proposal	-
[8]	Proposals of problem and requirements	Proposals of function, concept, and detailed design	-
[21]	Function	Solution principle	-
13] and [42]	Requirement	Solution	Yes
[30]	Problem—goals, constraints, and criteria	Concept	-
[31]	Problem	Solution	-
[35]	Problem	Solution	-
[34]	Problem	Solution	-
[5]	Need and problem statement	Schemes, layout drawings, and working drawings	-
[34]	Process	Function, organ, and part	-
[28]	Transformation	Organ and part	-
[37]	Functions	Attributes	-
[36]	Needs, desires, requirements, and problem	Solution and design documentation	-
[38]	Functions	Means	Yes
[24]	Overall problem and subproblems	Subsolutions and overall solution	Yes
[43]	Problem	Solution	Yes
[32]	Specification	Function structure, principal solution, module structure, preliminary layout, definitive layout, and product documents	Yes
[4]	Specifications	Principle solution, preliminary layout, definitive layout, and product documentation	Yes
[29]	Problem	Solution	-
[7]	Need and specification	Function, structure, principle solution, structure, preliminary form, definite form, and product	Yes

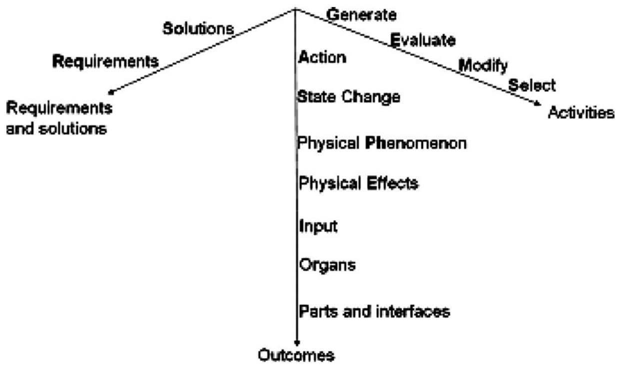

Fig. 3 GEMS of SAPPhIRE as req-sol

come is selected as unacceptable. The model represents a general problem finding and solving cycle, i.e., generate, evaluate, modify, and select requirement or solution. It is expected that an episode starts with a generation and ends with a selection.

The SAPPhIRE model [16] is chosen as the outcome model for the following reasons.

(a) One objective of our work is to understand the relationship between physical laws and effects and designing. The SAPPhIRE model makes explicit use of *effects* in describing the causality of systems, thereby, describing the role of effects in explaining the outcomes of designing.

(b) The model accommodates three major representations of function—action, state change, and input, thereby, providing a rich description of function. Physical phenomenon, physical effect, and organs have never been used together in existing literature. These constructs, together with their links to functionality, provide a richer description of behavior. Organs and parts together describe multiple and integrated representations of structure. Thus, the use of the SAPPhIRE model promises to provide a richer description of function, behavior, and structure.

Table 3 suggests that outcomes from literature can be classified under the generic terms *requirements* and *solutions*. In addition, various researchers have showed the occurrence of co-evolution of requirements and solutions. Hence, a co-evolving model is taken as the requirement-solution model.

The models of activity, outcome, and requirement-solution are combined to create an integrated model of designing—GEMS of SAPPhIRE as req-sol (Fig. 3). Graphically, the space enclosed by the three axes in the figure can be used to describe designing.

6 Evaluation of Activity Model

6.1 Individual Activity Findings. The protocol studies confirmed the presence of the proposed activities—generate, evaluate, modify, and select, in all the six designing sessions studied [1–6]. An example of each activity is given from the transcribed utterances: generate (*D: So, what has to be achieved is that the campus has to be kept free from dry leaves*—designer defines the purpose of the design exercise by generating a requirement), evaluate (*D: Is sweeping okay?*—designer evaluates an idea for clearing dry leaves), modify (*D: Instead of manual sweeping, collection is a better term*—designer generates a solution for clearing dry leaves and then, feels collection is a more general term), and select (*D1: Some secret code is required because each individual will have it differently; D2: Yeah*—the first designer generates a solution to have a safe and private locking system, which is accepted by the second designer). Figure 4 shows the percentage frequency of the individual activities in the six sessions. The following are observed:

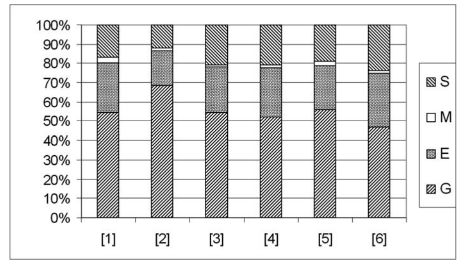

Fig. 4 Percentage frequency distribution of activities

(a) The instances of activities in all the sessions in descending order are generate, evaluate, select, and modify, irrespective of any design method, problem, or team. It could be argued that the design outcomes are generated first but only some of them are evaluated, selected, and modified.

(b) The percentage of *modify* is less than that of *select* in all the cases. This could be for either all the design sessions were of short duration and designers wanted to accept (hence, *select*) rather than modify or the implicit expectation that the designers produce original rather than redesigns.

(c) Even though *select* accounts for both acceptance and rejection, explicit reject could not be observed in the sessions. This could be due to either short duration or designer preference to modify rather than reject the outcomes that did not comply with the evaluation criteria.

(d) The main conclusion from this section is that all the utterances in the design sessions can be represented using the four generic activities—generate, evaluate, modify, and select.

6.2 Activity Pattern Findings. Table 4 shows the frequency of the activity patterns observed in the six sessions, each activity pattern referring to the activities in an episode. The following observations are made.

(a) The prominent activity patterns in descending order are G, GES, and GE across all design methods, problems, and teams.

(b) It is logical to argue that all activity patterns are expected to culminate in selection. However, many observed activity patterns seem to end in either evaluation or modification. This could be because these outcomes after generation are either not considered or implicitly evaluated and selected. Some patterns have generation only, possibly due to the outcome being implicitly evaluated and selected.

(c) Certain patterns have multiple evaluation, selection, and modification, possibly since each designer in a team had a different point of view, leading to different evaluation criteria. It could also indicate the iterative nature of designing.

(d) Activity patterns generally follow a sequence of generation and evaluation, leading to modification or selection. A modified outcome is further evaluated until it gets selected or not considered further.

(e) We conclude that the GEMS model (Fig. 2) is sufficient to describe the activity patterns.

7 Evaluation of the Outcome Model

7.1 Individual Outcome Findings. This study confirmed the presence of all the outcomes in all the designing sessions. An example of each outcome is given using a transcribed utterance from the design sessions: action (*D: So, what has to be achieved is that the campus has to be kept free from dry leaves*, the "action" is to "keep campus free from dry leaves"), state change (*D: Sys-*

Table 4 Frequency of activity patterns in designing sessions

Activity patterns	[1]	[2]	[3]	[4]	[5]	[6]
G	166	144	97	74	181	98
GE	29	11	5	8	4	10
GEM	11	3	0	1	10	3
GES	49	19	28	34	88	73
GESE	1	1	0	0	0	0
GEMES	2	0	0	1	1	1
GESES	11	4	4	5	2	3
GESEM	0	1	1	0	0	0
GESESE	0	0	0	0	0	1
GEMESES	1	0	1	0	1	0
GESESES	2	1	4	1	0	3
GESEMES	1	0	0	0	0	0
GESESEM	0	0	0	1	1	1
GESEMEM	0	0	0	0	0	1
GESESESE	0	0	1	0	0	0
GESEMESESES	0	0	0	0	0	1

tem's primary useful function is that it should lock when it is required to lock and open when it is required to open, the state change is from "lock to unlock and unlock to lock"), phenomenon (*D1: So, the functions that the system will take care of are cleaning, loading, transportation, unloading, and disposal of dry leaves*, the phenomena are cleaning, loading, transportation, unloading, and disposal), effect (*D1: Because of the force of gravity—gravitational force (D2 writes on the paper)*, the effect is Newton's law of gravitational force), organs (*D: So input is gravity, self-weight, and weak link* and the organs are self-weight and weak link), and parts (*D: Transportation can be done by carrying the bins and baskets manually, small trucks, or tractors* and the parts are bins, baskets, trucks, and tractors). While coding, actions were also interpreted as descriptions of input-output and so, inputs are counted under actions. Figure 5 shows the frequency of the outcomes for the six cases. The following inferences are drawn.

(a) High frequency of *action* level descriptions is noticed in all the sessions. It is probably because designers possess good knowledge of action.

(b) Fewer *state change* level descriptions are observed, probably because state change is another way of interpreting action and some instances of state change could have been included under action.

(c) High frequency of *part* level descriptions is noticed. It is probably because designers, in general, possess good knowledge of parts.

(d) The frequency of *phenomenon* level, *effect* level, and *organ* level descriptions are lower than that of action level descriptions. This could be because (i) designers do not know how to use these constructs while designing, (ii) designers lack knowledge of these constructs, or (iii) the methods used do not specify the use of the constructs.

(e) Similar results were also reported in a different study by Sarkar and Chakrabarti [44] on search spaces explored by novice and experienced designers while designing problems. Chakrabarti and Taura [45] discussed possible difficulties in analyzing and synthesizing designs using *effects*.

7.2 Outcome Pattern Findings. Figure 6 shows line diagrams depicting the patterns of outcomes as observed. The findings are the following.

(a) In all the sessions, the designers start solving from an action level description. The diagrams feature descriptions from higher level abstractions to lower level abstractions, e.g., starting from action level descriptions and ending with part level descriptions, passing through one or more intermediate levels of abstraction such as state change, phenomenon, etc. The transition from a higher level abstraction to a lower level abstraction confirms the synthetic nature of design.

(b) Action level and phenomenon level descriptions also emerge from, respectively, phenomenon level and part level descriptions. This seems to confirm again the strong part-knowledge of the designers—as they seem to know how to derive the behavior and function of the parts. This transition from a lower level abstraction to a higher level abstraction confirms the analytical nature of design.

(c) In most cases, there is a direct jump from action level or phenomenon level to part level description during synthe-

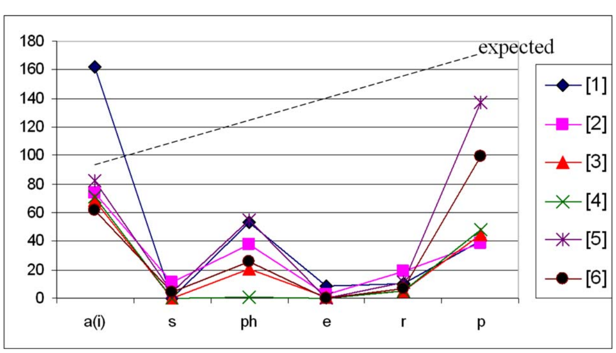

Fig. 5 Frequency of outcomes in designing sessions

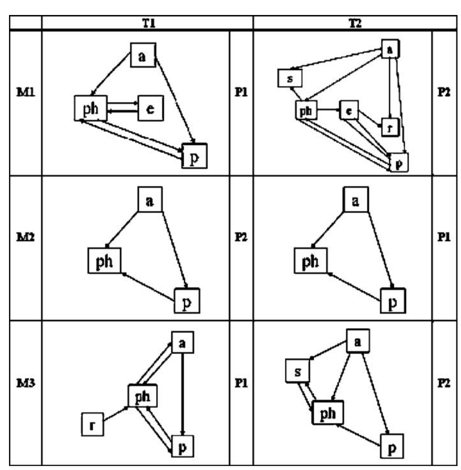

Fig. 6 Patterns of outcomes

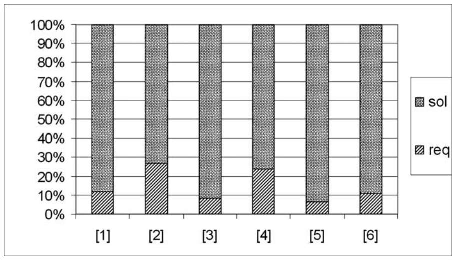

Fig. 7 Percentage frequency distribution of req and sol

sis and from part level to phenomena level description in analysis. In other words, designers do not always explore, or to the same extent, the outcomes at all levels of abstraction.

(d) In all the sessions, the designs culminate in a part level description but are not detailed to the extent of manufacturing. This is expected for a conceptual solution.

8 Evaluation of Requirement-Solution Model

Requirements and solutions were found in all the designing sessions. The findings are as follows.

(a) The requirements come from three sources—method, designer, and problem. Figure 7 shows the distribution of the percentage frequency of requirements and solutions. An example of each type of requirement and solution is given using transcribed utterances from the design sessions: req(method) (*D: Next step is criteria for selecting solution concept—desired technological characteristics, desired economic characteristics, desired timetable, expected degree of novelty, and other criteria (reads from the instruction sheet)*—the designer reads out the requirement specified by method (ISQ) from the instruction sheet i.e., to generate evaluation criteria-technological, economy, etc.), req(problem) (*D: So, what has to be achieved is that the campus has to be kept free from dry leaves*—designer describes the objective of the design exercise, i.e., to keep university campus free from dry leaves, taken directly from the problem given), req(designer) (*D: So, it (lock) should not be electricity dependent or it can have its own battery, why not?*—designer creates a requirement that the locking system should not be electricity dependent, however, it can be battery-powered), and sol (*D: This is a net-arrangement (pointing at the sketch) and this is a vacuum cleaner (pointing at the sketch)*—the designer generates a solution—a vacuum cleaner and net—for collecting and storing dry leaves, respectively).

(b) Four kinds of relationships between requirements and solutions are identified. *Requirement-requirement* represents requirement clarification (i.e., evaluation, modification, and selection of requirements), *requirement-solution* represents development of solutions to satisfy a requirement (i.e., generation of solutions), *solution-requirement* depicts development of requirements from solutions (i.e., generate requirements from solutions), and *solution-solution* represents solution clarification (i.e., evaluation, modification, and selection of solutions). An example of each relationship is given using transcribed utterances from the design sessions: req-req (*D1: Now there is a constraint, no physical key, no code, in the sense, alpha-numeric. D1: I have a doubt, is alpha (numeric) also included in this?*—designer D1 develops a requirement for a conceptual solution of a locking system and evaluates it to check if "alpha-numeric" is also included in it), req-sol (*D: If humans are there, then, supervising can be done by cameras like its done here (points at the camera)*—designer D proposes to use cameras as a solution for the requirement of supervision), sol-req (*D: unloading and disposal, these are the areas to be satisfied by the system*—designer D identifies "unloading" and "disposal" as requirements, which were previously developed as solutions for the cleaning of dry leaves), and sol-sol (*D1: Can it be a keyboard password? D2: Yes, it can be a keyboard password*—designer D1 clarifies whether keyboard password can be used as a solution for the locking system and is accepted by designer D2). The *req-sol* and *sol-req* relationships indicate the co-evolution.

(c) We conclude that designing can be described using a requirement-solution co-evolution model.

Table 5 Frequency of activities and outcomes

	G	E	M	S
a(i)	520	245	23	184
s	19	6	1	5
ph	194	91	7	72
e	13	1	0	1
r	57	25	1	20
p	408	124	10	101

9 Evaluation of Combined Model

9.1 Combined—Activity and Outcome Findings. Table 5 reports the combined frequency of activities and outcomes in the six sessions. The following observations are made.

(a) Action level and part level descriptions have many instances of generation, evaluation, selection, and modification. This seems to further confirm that designers had good knowledge of these constructs.

(b) Phenomenon level, organ level, and effect level descriptions have far fewer instances of individual activities. No modification is observed for effects since modify is synonymous to replace and effects cannot be altered. These numbers verify the difficulty that designers face when working with these constructs.

(c) State change level descriptions also have fewer instances of individual activities. It could be because some state change level descriptions have been included under action due to the little difference in representing these two outcomes.

9.2 Combined—Requirement-Solution and Activity Findings. The percentage distribution of activities for requirements and solutions (Fig. 8) show similar patterns in all design sessions, less modification of requirements than of solutions. Similar general trend is seen in Fig. 4.

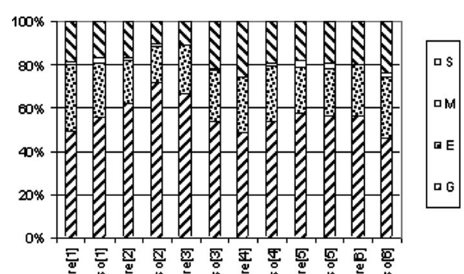

Fig. 8 Percentage distribution of activities in req and sol

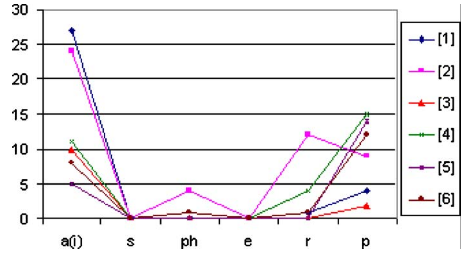

Fig. 9 Frequency of outcomes of req

9.3 Combined—Requirement-Solution and Outcome Findings. For all sessions, Figs. 9 and 10, respectively, show the frequency of outcomes, of requirements, and of solutions. Except for one session where a higher number of organ level description is found, similar patterns as in Fig. 5 are observed.

10 A Framework for Design for Novelty and Its Planned Implementation

In this section, we describe how the proposed model of designing has been helpful in identifying an issue with current designing that has significant influence on variety and novelty of concepts. Based on this and the knowledge of the integrated model, a framework for supporting variety and novelty is developed and a plan for its implementation proposed.

Novelty means being new and original and something not formerly known [46] and unusualness or unexpectedness [47]. Novelty is considered a measure of creativity of engineering products [47,48]. Researchers highlighted the positive effects of novelty on the success of a product [49–51]. Variety signifies how different concepts are from one another [52]. Creativity is a thought process that generates ideas; however, only a small number of these actually end up as innovation [53]. This stresses the importance of producing a large variety of alternative solutions during designing. Srinivasan and Chakrabarti [52] verified empirically that novelty of a concept space is influenced by its variety while both depend on the number of outcomes explored.

During designing, a single-to-many mapping from a higher to a lower level of abstraction is expected, i.e., an action can be satisfied by multiple alternative state changes, a state change by multiple alternative phenomena, and so forth (shown by a dashed-line in Fig. 5), resulting in development of a variety of alternative concepts. Observations in this paper indicate that designers do not uniformly explore all the constructs (shown by the low numbers of phenomenon, effect, and organ level descriptions in Fig. 5), resulting in fewer concepts than possible, thereby, inhibiting variety and novelty.

To address this issue, a framework is proposed, which prescribes GEMS to be carried out for all the SAPPhIRE levels, for both requirements and solutions [54]. The framework is divided into requirements exploration stage (RES) and solutions exploration stage (SES). In RES, requirements at all the levels of SAPPhIRE are generated, evaluated, modified, and selected. In SES, solutions at all the levels of SAPPhIRE are generated, evaluated,

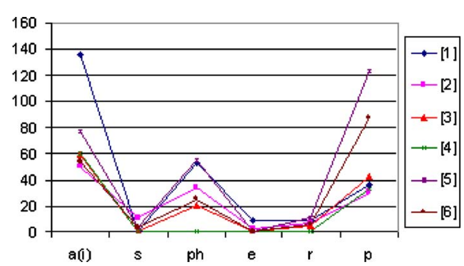

Fig. 10 Frequency of outcomes of sol

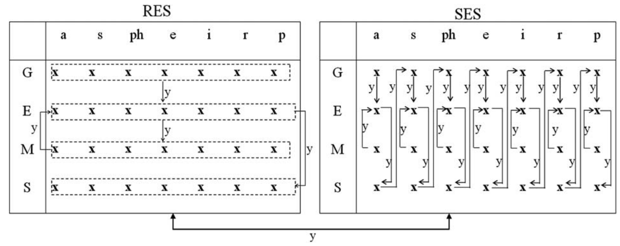

Fig. 11 An interactive platform of the framework (notation: x-assist and y-guide)

modified, and selected in the order of decreasing level of abstraction. The framework should encourage better exploration of all the levels of outcome, thereby, enhancing variety and novelty. The framework is planned to be implemented on an interactive platform (see Fig. 11 where each cell shows an activity-outcome combination, the first and second matrices represent RES and SES, respectively) to support two kinds of tasks, computer *assists* designer and computer *guides* designer. Different media for assistance are planned: IDEA-INSPIRE, an interactive tool to provide analogical requirements and solutions as stimuli from a database of natural and engineered systems [16], a notebook-interface to enable storage and retrieval of data, e.g., for comparison and editing, and a modeling-interface for simulation and evaluation. Generation of requirements and solutions at all levels of SAPPhIRE will be assisted by IDEA-INSPIRE and the notebook-interface, evaluation of requirements and solutions by notebook-interfaces and modeling-interfaces, and modification and selection of requirements and solutions by the notebook-interface. Guidance is given under discretion of the designer after an activity is completed or when switching between RES and SES.

11 Summary

This paper can be summarized as follows.

(a) An integrated model of designing based on process elements and product elements has been developed by combining activities, outcomes, requirements, and solutions.
(b) Empirical validation of the model confirmed that all the proposed activities, outcomes, requirements, and solutions are present in natural design processes. However, it has also been found that an adequate number of phenomena, effects, and organs were not explored, resulting in lower variety and novelty.
(c) In order to alleviate this shortcoming, a framework has been proposed. A plan has also been created for implementing the framework on an interactive platform.

References

[1] Blessing, L., and Chakrabarti, A., 2009, *DRM, A Design Research Methodology*, Springer, London, UK.
[2] Blessing, L., and Chakrabarti, A., 2002, "DRM: A Design Research Methodology," Proceedings of the International Conference on the Science of Design—The Scientific Challenge for the 21st Century, INSA, France.
[3] Chakrabarti, A., Murdoch, T., and Wallace, K., 1995, "Towards a Glossary of Engineering Design Terms," Proceedings of the International Conference on Engineering Design (ICED95), Prague, Czech Republic.
[4] Pahl, G., and Beitz, W., 1996, *Engineering Design: A Systematic Approach*, Springer-Verlag, UK.
[5] French, M., 1999, *Conceptual Design for Engineers*, 3rd ed., Springer-Verlag, London, UK.
[6] Berliner, C., and Brimson, J., 1988, *Cost Management for Today's Advanced Manufacturing: The CAM-I Conceptual Design*, Harvard Business School Press, Boston.
[7] Zeiler, W., Savanovic, P., and Quanjel, E., 2007, "Design Decision Support for the Conceptual Phase of the Design Process," Proceedings of the International Association of Societies of Design Research (IASDR07), Hong Kong.
[8] Blessing, L., 1994, "A Process-Based Approach to Computer-Supported Engineering Design," PhD thesis, University of Twente, The Netherlands.
[9] Simon, H., 1996, *Sciences of the Artificial*, 3rd ed., MIT, USA.

[10] Smithers, T., 1992, “Design as Exploration: Puzzle-Making and Puzzle-Solving,” *Proceedings of the Workshop on Search-Based and Exploration-Based Models of the Design Process, Associated With AI in Design*, J. Gero, ed., Kluwer, Dordrecht, pp. 1–22.
[11] Cooper, R., 1991, “New Products: What Distinguishes the Winners?,” Res. Technol. Manag., **33**(6), pp. 27–31.
[12] Stauffer, L., and Ullman, D., 1991, “Fundamental Processes of Mechanical Designers Based on Empirical Data,” J. Eng. Design, **2**(2), pp. 113–125.
[13] Nidamarthi, S., Chakrabarti, A., and Bligh, T., 1997, “The Significance of Co-Evolving Requirements and Solutions in the Design Process,” Proceedings of the International Conference in Engineering Design (ICED97), Tampere, Finland, pp. 227–230.
[14] Nidamarthi, S., 1999, “Understanding and Supporting Requirement Satisfaction in the Design Process,” PhD thesis, University of Cambridge, UK.
[15] Gero, J. S., 1990, “Design Prototypes: A Knowledge Representation Schema for Design,” AI Mag., **11**(4), pp. 26–36.
[16] Chakrabarti, A., Sarkar, P., Leelavathamma, B., and Nataraju, B. S., 2005, “A Functional Representation for Aiding Biomimetic and Artificial Inspiration of New Ideas,” AI EDAM, **19**(2), pp. 113–132.
[17] Zavbi, R., and Duhovnik, J., 2000, “Conceptual Design of Technical Systems Using Functions and Physical Laws,” AI EDAM, **14**(1), pp. 69–83.
[18] Koyama, T., Taura, T., and Kawaguchi, T., 1996, “Research on Natural Law Database,” Proceedings of the Joint Conference on Knowledge Based Software Engineering, Bulgaria.
[19] Zavbi, R., and Duhovnik, J., 2001, “Analysis of Conceptual Design Chains for the Unknown Input/Known Output Pattern,” Proceedings of the International Conference on Engineering Design (ICED01), UK.
[20] Murakoshi, S., and Taura, T., 1998, “Research on the Systematization of Natural Laws for Design Support,” Proceedings of the IFIP Workshop on Knowledge Intensive CAD, Japan, pp. 141–160.
[21] Chakrabarti, A., Johnson, A., and Kiriyama, T., 1997, “An Approach to Automated Synthesis of Solution Principles for Microsensor Designs,” Proceedings of the International Conference on Engineering Design (ICED97), Finland.
[22] Chakrabarti, A., Morgenstern, S., and Knaab, H., 2004, “Identification and Application of Requirements and Their Impact on the Design Process: A Protocol Study,” Res. Eng. Des., **15**(1), pp. 22–39.
[23] Chakrabarti, A., 1994, “Requirements Identification: A Central Issue in Design Research,” Proceedings of the International East-West Conference on IT in Design, Russia, pp. 108–117.
[24] Cross, N., 2000, *Engineering Design Methods—Strategies for Product Design*, Wiley, UK, p. 42.
[25] Chakrabarti, A., 2003, “Towards a Measure for Assessing Creative Influences of a Creativity Technique,” Proceedings of the International Conference on Engineering Design (ICED03), Sweden.
[26] Miller, G., Galanter, E., and Pribram, K., 1960, *Plans and the Structure of Behaviour*, Holt, Rinehart and Winston, USA.
[27] Bhatta, S., and Goel, A., 2002, *Design Patterns and Creative Design, Engineering Design Synthesis—Understanding, Approaches and Tools*, A. Chakrabarti, ed., Springer-Verlag, UK, pp. 271–284.
[28] Hansen, C., and Andreasen, M., 2002, *Two Approaches to Synthesis Based on the Domain Theory, Engineering Design Synthesis—Understanding, Approaches and Tools*, A. Chakrabarti, ed., Springer-Verlag, UK, pp. 93–108.
[29] Lossack, R., 2002, *Design Processes and the Context for the Support of Design Synthesis, Engineering Design Synthesis—Understanding, Approaches and Tools*, A. Chakrabarti, ed., Springer-Verlag, UK, pp. 213–227.
[30] Cross, N., 2000, *Engineering Design Methods—Strategies for Product Design*, Wiley, UK, pp. 3–11.
[31] Visser, W., 1991, “Evocation and Elaboration of Solutions: Different Types of Problem Solving Actions,” Proceedings of the Third COGNITIVA Symposium on at the Crossroads of Artificial Intelligence, Cognitive Science, and Neuroscience, Madrid, Spain, pp. 161–168.
[32] Cross, N., 2000, *Engineering Design Methods—Strategies for Product Design*, Wiley, UK, pp. 38–41.
[33] Cambridge University Press, 2009, Cambridge Dictionaries Online, New York, http://www.dictionary.cambridge.org/
[34] Hubka, V., and Eder, W., 2002, *Theory of Technical Systems and Engineering Design Synthesis, Engineering Design Synthesis—Understanding, Approaches and Tools*, A. Chakrabarti, ed., Springer-Verlag, UK, pp. 49–65.
[35] Campbell, M., and Rai, R., 2003, “A Generalization of Computational Synthesis Methods in Engineering Design,” AAAI Spring Symposium Series.
[36] Smithers, T., 1998, “Towards a Knowledge Level Theory of Design Process,” *AI in Design*, J. Gero and F. Sudweeks, eds., Kluwer, Dordrecht, pp. 3–21.
[37] Reich, Y., 1995, “A Critical Review of General Design Theory,” Res. Eng. Des., **7**(1), pp. 1–18.
[38] Hubka, V., 1967, “Fundamental Algorithm for the Solution of Design Problems,” Proceedings of the International Scientific Colloquium of the Technical University Ilmenau, Germany, pp. 69–74.
[39] Rihtaršic, J., Žavbi, R., and Duhovnik, J., 2008, “Physical Nature of Technical Systems,” Proceedings of the International Design Conference (DESIGN2008), Croatia, pp. 53–60.
[40] Tomiyama, T., Kiriyama, T., Takeda, H., and Xue, D., 1989, “Metamodel: A key to Intelligent CAD Systems,” Res. Eng. Des., **1**(1), pp. 19–34.
[41] Umeda, Y., Ishii, M., Yoshioka, M., Shimomura, Y., and Tomiyama, T., 1996, “Supporting Conceptual Design Based on the Function-Behaviour-State Modeler,” AI EDAM, **10**(4), pp. 275–288.
[42] Nidamarthi, S., Chakrabarti, A., and Bligh, T., 2001, “Improving Requirement Satisfaction Ability of the Designer,” Proceedings of the International Conference on Engineering Design (ICED01), Glasgow, UK, pp. 237–244.
[43] Maher, M., Poon, J., and Boulanger, S., 1996, *Formalising Design Exploration as Co-Evolution: A Combined Gene Approach, Advances in Formal Design Methods for CAD*, J. Gero and F. Sudweeks, eds., Chapman and Hall, UK.
[44] Sarkar, P., and Chakrabarti, A., 2007, “Understanding Search in Design,” Proceedings of the International Conference on Engineering Design (ICED07), France.
[45] Chakrabarti, A., and Taura, T., 2006, “Computational Support of Synthesis and Analysis of Behaviour of Artefacts Using Physical Effects: Some Challenges,” Proceedings of the International Design Conference (DESIGN2006), Croatia.
[46] Sternberg, R., and Lubart, T., 1999, *The Concept of Creativity: Prospects and Paradigms, Handbook of Creativity*, R. Sternberg, ed., Cambridge University Press, UK.
[47] Shah, J., Smith, S., and Vargas-Hernandez, N., 2003, “Metrics for Measuring Ideation Effectiveness,” Des. Stud., **24**(2), pp. 111–134.
[48] Lopez-Mesa, B., and Vidal, R., 2006, “Novelty Metrics in Engineering Design Experiments,” Proceedings of the International Design Conference (DESIGN2006), Croatia.
[49] Westwood, A., and Sekine, Y., 1988, “Fostering Creativity and Innovation in an Industrial R and D Laboratory,” Res. Technol. Manag., **31**(4), pp. 16–20.
[50] Ottosson, S., 1995, “Boosting Creativity in Technical Development,” *Proceedings of the Workshop in Engineering Design and Creativity*, W. Eder, ed., Pilsen, Czech Republic, pp. 35–39.
[51] Zimmerman, L., and Hart, G., 1998, *Value Engineering—A Practical Approach for Owners, Designers and Contractors*, Van Nostrand Reinhold, New York.
[52] Srinivasan, V., and Chakrabarti, A., 2010, “Novelty-Outcome Relationships in Engineering Design,” AI EDAM, **24**(2), pp. 161–178.
[53] Cavallucci, D., 2002, *TRIZ, the Altshullerian Approach to Solving Innovation Problems, Engineering Design Synthesis—Understanding, Approaches and Tools*, A. Chakrabarti, ed., Springer-Verlag, London, UK, pp. 131–149.
[54] Srinivasan, V., and Chakrabarti, A., 2009, *Designing Novel Artifacts: A Novel Systematic Framework, Research Into Design: Supporting Multiple Facets of Product Development*, A. Chakrabarti, ed., Research Publishing Services, Singapore.