Proceedings of TMCE 2012, May 7–11, 2012, Karlsruhe, Germany, Edited by I. Horváth, A. Albers, M. Behrendt and Z. Rusák

AN EXTENDED, INTEGRATED MODEL OF DESIGNING

Belgavkar Sudhindra Rao Chita Ranjan
Innovation, Design Study and Sustainability Laboratory
Centre for Product Design and Manufacturing, Indian Institute of Science
ranjan@cpdm.iisc.ernet.in

Srinivasan V
Amaresh Chakrabarti
Innovation, Design Study and Sustainability Laboratory
Centre for Product Design and Manufacturing, Indian Institute of Science
{srinivasan, ac123}@cpdm.iisc.ernet.in

ABSTRACT

The goal of design research is to understand and support designing for product success. Activities, outcomes, requirement-solution and system-environment views play a significant role in product success. Thus, it is important to explicitly incorporate these views in designing. Literature has been reviewed to identify constructs of these views. Earlier, an integrated model of designing was developed by integrating the views of activities, outcomes, and requirement-solution, but the model did not incorporate the system-environment view. We propose a system-environment view in which environment is an explicit and evolvable construct, and combine this with the earlier integrated model of designing to propose an extended, integrated model of designing. Examples are used to illustrate the constructs of the proposed model, and various phenomena related to designing.

KEYWORDS

Model, SAPPhIRE, GEMS, system, environment, element, activity, outcome, requirement, solution

1. INTRODUCTION

Design research aims to improve the chances of producing a successful product by making designing more effective and efficient [1]. To realize this, Blessing and Chakrabarti [1] propose: (a) formulate and validate an understanding of current designing, and (b) develop and validate a support, founded on the understanding, to improve the current designing. In this research, a model of designing is defined as a description of how designing is carried out, to reflect the current understanding.

Designing involves multiple facets – product, process, people, tools, environment, micro- and macro-economy [1]. This research focuses only on product and process facets. A system can be seen as a combination of several subsystems, elements and their relationships. A system interacts with its environment to satisfy its requirements. During designing, one must take into account the effect, of any changes in the system being designed, on the need for a change in the environment with which it interacts, and vice-versa. Therefore, designing must involve considering *both* the system being designed and its environment together, and allowing them to *co-evolve* if necessary. We term this simultaneous and inter-dependent change in both system and its environment as *system-environment co-evolution*. In theories or models of designing, it is important to incorporate *system-environment* view (Sy-En view) - considering relationships, elements, subsystems, system as well as environment, to help explain or design interactions among these and their co-evolution. The four constructs - relationships, element, subsystem and system (but not environment) in contrast - are henceforth referred together as the *system view*.

Activity in designing is defined as a deed of problem finding and solving [2, 3]. Activities play an important role in the success of the end product [4].

Outcome in designing is defined as a property of a design at a particular level of abstraction [2]. Outcomes in designing influence aspects like requirements identification and satisfaction [5, 6].

Requirement is defined as an expression of what a design should have at a level of abstraction. Solution is defined as a means to satisfy requirements. A design process is initiated with the recognition of a need, leading to the establishment of requirements for the intended product [7]. Therefore, capturing requirements is essential and a central issue in design research [8]. Cooper [4] and Nidamarthi *et al.* [6] measured the success of a product in terms of how well it satisfied its requirements.

From literature (Section 3.1) it is found that various models of designing are developed based on one or more of the views of: system-environment, activity, outcome and requirement-solution. Literature also reveals that the four views are not integrated together within a single design model, theory or approach. Therefore, a model of designing that integrates all these views, is novel, and, has the potential to serve as a platform with substantially more explanatory power for supporting complex discourses requiring consideration of multiple views. Examples of such complex discourse include comparing and benchmarking design models, developing a detailed understanding of the different stages of designing, co-evolution, structure sharing, etc.

Hence, the overall objective of this research is to develop a model of designing that addresses these views together. The research approach is as follows:

(a) Design research literature is reviewed to find the extent to which the four views - *system-environment view*, *activity view*, *outcome view*, and *requirement-solution view* - are considered in current work.

(b) Based on this review, the constructs to be used for the four views are identified.

(c) A model of designing based on these constructs is then developed.

2. LITERATURE SURVEY

2.1. Review of Existing Literature

In the following paragraphs, current literature on design theories, models and approaches are reviewed from the aforementioned views: activity, outcome, requirement-solution and system-environment.

Asimow [9] proposed a morphology of design to explain the various activities and outcomes in each of these primary design phases: *feasibility study*, *preliminary design* and *detailed design*. The main activities used are: *analysis*, *synthesis*, *evaluation*, *decision*, *optimization*, *revision*, and *implementation*. The outcomes are *engineering statement of the problem*, *alternative* to *realizable* to *worth-while* to *set of useful solutions*, *tentative selection*, *analytical formulation*, and so on. Asimow used a system view consisting of *system*, *subsystem*, *components* and *parts*. Although Asimow considered interactions between system and environment as important, environment was not used as an evolvable construct.

Gero [10] proposed a descriptive design model based on activities and outcomes, where the purpose of designing is to transform a set of functions {F} into a design description {D} in such a way that the artifact is capable of producing these functions. The activities during the transformation from F to D are: *formulation*, *synthesis*, *analysis*, *evaluation*, *reformulation*, and *production* of a design description. The outcomes during these activities are *functions*, *expected behavior*, *structure*, *derived behavior* and *design description*. The *functions* can be categorized as requirements and *structure*, *expected behavior*, *derived behavior* and *design description* could be categorized under solutions. System-environment view is not considered by Gero.

Visser [11] proposed a descriptive model of designing based on activities and requirement-solution. This model proposed two approaches. One approach is proceeding in iterative cycles: problems are worked out in depth-each, solution to a problem is taken as a problem to be solved until a final solution is reached. The second approach is leading progressively from a global problem specification to a detailed solution: problems are decomposed into sub-problems, which imposes integration of intermediate solutions. This solution has to be detailed and concrete enough to specify its manufacturing details (that is, in the case of artifacts' design). In this model, a problem is decomposed into sub-problems; solutions are found for each and integrated. This suggests the implicit presence of the system view in the design process. The activities involved are *developing a solution proposal*, *evaluating a solution* and *accepting* or *rejecting the solution* based on criteria. The outcomes are *solution proposals, accepted or rejected solutions*.

Blessing [12] developed PROSUS (Process-based Support System) – a prescriptive approach for designing. A *design matrix* is the core of PROSUS; this is a structured set of *issues* and *activities*. The activities are *generate*, *evaluate* and *select*, which are used to solve the *issues*. The *issues* consist of *problems, requirements, function, concept* and *detailed design*. The outcomes are *proposals for each issue*, *arguments* and *decisions* on the *proposals*. Blessing uses a *product hierarchical tree* in the *product model* to relate the *design matrices* of product, assemblies, standard component or components. The relationships between these are stored as a *relationship model*. The design matrices are placed at the nodes of a *product hierarchical tree.* Apart from hierarchical relationships, several other relationships, such as spatial or functional, exist among product, assemblies, standard component or components. The relationship model represents the product elements, their relationships and the type of relationship(s). In this model, system view, outcomes and activities are all linked together. However, the requirement-solution view is not made explicit, although *proposals*, *arguments* and *decisions* at *problem* and *requirement* levels could be interpreted as requirements, and *proposals*, *arguments* and *decisions* at *function*, *concept* and *detailed design* levels could be interpreted as solutions**.**

Hubka and Eder [13] developed the theory of technical systems as an outcomes based model of designing. They use a system-environment view that consists of: *system*, *subsystem*, *elements*, *components*, *environment* and *active environment. Environment* and *active environment* are used in the model of designing but are not evolvable. In this model, outcomes at various levels of abstraction are at different levels of the technical system: *purpose*, *transformation process structure*, *function structure*, *organ structure* and *component structure*. The *purpose* and *transformation process* of an outcome could be interpreted as requirements, while the *function structure*, *organ structure* and *component structure* could be interpreted as solutions. These, however, are not made explicit in the model.

Hubka and Eder [13] subsequently combined the theory of technical systems (TS) with a design process model based on activities and outcomes. The activities are *elaborate and clarify the assigned specification*, *establish function structure–investigate alternatives*, *establish organ structure–investigate alternatives*, *establish component structure–investigate alternatives*, *establish component structure–investigate alternatives*. A model of finer activities that occurs in a design process is also developed. The outcomes of the above activities are *design specification* as a *transformation process, function structures*, *organ structure*, *component structure* **-** *preliminary layout*, *component structure* **-** *dimensional layout*, *component structure* and *detail drawings.* The activities are performed on the outcomes in the theory of TS. While the outcome *design specification* can be interpreted as requirements, and the *function structure*, *organ structure, component structure – preliminary layout*, *component structure – dimensional layout*, *component structure* and *detail drawings* can be interpreted as solutions; this, however, is not explicit in this theory.

Pahl and Beitz [7] proposed a prescriptive design approach. In this model, the different stages are *task clarification*, *conceptual design*, *embodiment and detail design.* In *task clarification*, they prescribed setting up requirements list based on subsystems. In conceptual design, the outcomes are *function structure* (a system of sub-functions), *working principles* (for individual sub-functions), *working structures* (combination of working principles), *principle solution variants* (combination of working structures), and finally *solution principle* (or a concept). In the embodiment design, the outcomes are embodiment determining *main function carriers*, *preliminary layouts* and *form designs*, which are used to develop *detailed layouts*, *form designs* and *definitive layout*. In the detail design, the outcomes are *detail drawings*, *overall layout drawings*, *assembly drawings*, and *parts list*. Pahl and Beitz explicitly use activities, outcomes, and requirement-solution views in their approach. The system view can be seen at several levels within conceptual design: *function structures* at function level, and *working structures* at working principles level. In *function structure*, functions can be seen at system, subsystem and element levels. *Working structures* are developed at subsystem and elements level. At the *embodiment stage*, *layouts* are developed at the system level, and *form designs* at the element level. The *drawings* produced are at different system levels. Both *assembly drawings* and *layout drawings* show system and subsystem level details. *Part drawings* show element level details. Environment though mentioned in the early stages, is not included in any of the views.

Chakrabarti *et al.* [14] proposed a prescriptive approach for conceptual design of micro-sensors based on activities, outcomes, and requirement-solution views. The process of designing involves the activities: *generating a design concept*, *identifying potential behavioral problems* and *trying to fix these problems*. The outcomes proposed are *functions*, *solution principles* and *embodiments*. The activity cycle is performed on *solution principle*, which is one of the outcomes proposed. In general, *function* and *solution principles* could be classified as requirements and solutions respectively, although this distinction is not made in [14]. They, however, represent *function* and *solution principles* respectively using an input-output description and concatenated laws and effects. This suggests that each function is divided into sub-functions for which solutions are combinations of *elementary devices*.

The system view here [14] is at the function, input-output and solution principle levels.

Nidamarthi [5] proposed a descriptive model of designing based on activities and requirement-solution-associated information. The model consists of two stages: *problem understanding*, and *problem solving*. Under these stages, the activities are sub-divided into primary and secondary levels. In *problem understanding*, the activities are *identify* (*perceive, infer, modify*), *analyze* (*question, relate, weigh, verify, visualize*), and *choose* (*decide*). In *problem solving*, the activities are *generate* (*create, modify, detail*), *evaluate* (*identify characteristic, question, relate, verify*), and *select* (*identify things to do, compare, decide*). Nidamarthi *et al.* [6] were the first to empirically validate the existence of co-evolution of requirements and solutions in designing.

French [15] proposed a descriptive model of designing based on activities and outcomes. According to French, designing has following stages: *analysis of problem*, *conceptual design*, *embodiment of schemes*, and *detailing*. French argued that the *evaluation* activity is present in each stage. The outcomes are *statement of the problem*, *selected schemes*, *general arrangement drawings*, and *working drawings*. The *statement of the problem* can be classified as requirements and the rest as solutions. The *selected schemes* are solutions at the system level. *General arrangement drawings* are solutions at system and subsystem levels. *Working drawings* or *part drawings* are solutions at the element level. The system view is not made explicit.

Cross [16, pp. 3-11] developed a descriptive model of designing based on activity, and requirement-solution views. Activities are *exploration of problem-space*, *generation of a concept*, *evaluation of the concept against goals*, *constraints and criteria of the design problem*, and *communication of the evaluated design to manufacture*. The outcomes are problems which contain *goals*, *constraints* and *criteria*, which we categorize as requirements; *solution space*, *concept sketch*, *drawing* or other *models of design* for visualization, evaluated and improved design, and *final drawing* or *description of the artifact*, are categorized as solutions. Problems and solutions are at various levels of abstraction. Problems and solutions are developed together. For properly defining design problems, Cross suggests the use of problem structures (i.e. to subdivide a problem into sub-problems and link these) and decision trees. The problem structure and decision trees are system views at earlier stages of design. The solutions developed start as solution space at the system level, then to system and subsystem level (i.e. as concept sketch, drawings, evaluated drawings) to elements level (i.e. in final production documents). From these, it can be seen that the system view is present in Cross' model, but has not been made explicit.

VDI 2221 (explained in [16, pp. 38 - 41]) is a prescriptive, systematic approach for designing technical systems and products. It combines activities, outcomes and requirement-solution views. System view is explicitly used at the second stage of the VDI 2221, where functions and *function structure* are determined. At the third stage, *solution principles* are developed for sub-functions, and combined to form a *principal solution*. At the fourth stage, the *principal solution* is divided into realizable modules and its *module structure* is developed. At the fifth stage, key modules are developed into *preliminary layouts*, which are further developed in the sixth stage into a *definitive structure*. In the seventh stage, final product documents are produced. The outcomes are at different levels of abstraction: *specification, function structure, principal solution, module structure, preliminary layouts, definitive layout, and product documents*. While the specification can be interpreted as requirements, the remaining outcomes can be interpreted as solutions. This view is not explicit in this model. VDI 2221 follows a general systematic procedure of breaking the problem into sub-problems, finding suitable sub-solutions, and combining these into an overall solution. From this general systematic procedure used, the system view (which does not include environment as an evolving element) and outcomes are linked with requirement-solution. However, not all the relationships among the system view, outcomes view and requirement-solution view are made explicit.

Cross [16, pp. 42] developed an integrative model of designing – which is a symmetrical problem-solution model in which an overall problem is divided into sub-problems, sub-solutions are found, and from these an overall solution is developed. Within this model, seven stages are proposed: *clarifying objectives*, *establishing function structures, setting requirements*, *determining characteristics*, *generating alternatives, evaluating alternatives*, and *improving details*. While the activities view and requirement-solution view are explicitly present, the system view (problem to sub-problem, and sub-solution to solution distinctions) is implicit. However while objectives, function structures, requirements, characteristics, alternative solutions, and detail can be interpreted as outcomes, this is not made explicit.

In the Domain theory [17], a mechanical artifact to be designed can be seen in three domains – *transformation, organ* and *part*. This theory has evolved from theory of technical systems [18]. In the Domain theory, there are three principal synthesis steps: two types of steps in each domain, *detailing* and *concretization* and a *synthesis step* jumping from one domain to another. The outcomes are the *characteristics* in the step of *detailing* and *the values to these characteristics* in the step of *concretization*. These outcomes are present in each domain. The outcomes of transformation domain can be classified under requirements, while the outcomes of organ and part domain can be classified as solutions.

Lossack [19] developed a prescriptive design approach based on activities and outcomes. The activities are *define problem, find a solution, describe solution, evaluate solution, select solution*; among these *problem* at *requirement level* could be categorized as requirements, while *solutions* at *function, physical principle, embodiment level* could be categorized as solutions. However, the requirement-solution view is not made explicit. The outcomes are *requirement, function, physical principle* and *embodiment*. At *function level*, a *task structure* is used to present the relationships among tasks which build a hierarchy. At *physical principle level*, a physical solution is described covering all the information of the physical inter-relationships of an artifact. Both these levels suggest the presence of the system view, by the use of *task structure*, *physical principle structure* and *geometry structure*. However, the system view is not explicit in the other two abstraction levels: requirement level and embodiment level. Environment is not included in these views.

Bhatta and Goel [20] develop a normative theory of conceptual device design called model-based analogy (MBA). MBA takes a design problem as *functional requirements* and *structural constraints* on the desired design, and gives the outcomes in the form of a *structure* that realizes the specified functions and structural constraints. The design problem in the form of *functional requirements* and *structural constraints* could be categorized as requirements, and *structures obtained* can be categorized as solutions. In addition, MBA uses a Structure-Behavior-Function (SBF) model that explains how a structure realizes a desired function. The structure of a device in SBF models is represented hierarchically in terms of its constituent structural elements and relations among them such as *part-of, includes,* and *parallely-connected.* This shows the implicit presence of the system view at structural level in the model. However, environment is not included in the view.

Campbell and Rai [21] develop a prescriptive approach for computational design synthesis. At the beginning, a design *problem* is *formulated* by the customers, which includes *constraints* and constructive *objective functions* or *design goals*. The activities are representation, generation, evaluation and guidance which are performed only on solutions. The outcomes of these activities are *candidates*, *generated solutions*, *evaluated solutions*, *objective values*, and new and better solutions. While these outcomes fall under solutions, the problem descriptions, constraints and objectives could be categorized as requirements.

According to Ullman [22], six basic actions take place in design problem-solving: *establish*, *plan*, *understand*, *generate*, *evaluate*, *decide* and *communicate*. Ullman uses the concept of product life-cycle management that integrates six major types of information: one of these is systems engineering, considered as '*support for technical development of the function of the product*'. He uses the following decomposition for mechanical devices: system, subsystem, assembly, and component. He considered it important to use function-behavior-performance. He also discussed that based on the need, solutions can be one or many. Activities, outcomes, requirements, solutions and the system view are all explicitly used in his model of designing, but environment is not considered.

Ulrich and Eppinger [23] developed a generic product development process with the following phases: *planning*, *concept development*, *system level design*, *detail design*, *testing and refinement*, and *production ramp-up*. Each phase consists of activities and outcomes at various abstraction levels. They particularly discuss the *concept development* phase in greater detail. The *needs* are developed and *hierarchy of the needs* is organized. Problem is decomposed into sub-problems, and sub-solutions are developed and systematically combined to form various concepts. This leads to *concept selection* and *testing*. The next phase is system level design (or *product architecture*) in which the product is divided into *function* and *physical elements*. *Functional elements* of a product consist of *individual operations* and *transformations*. *Physical elements* of a product are *parts*, *components* and *sub-assemblies*. In this model, activities, outcomes, requirements, solutions and system view are explicitly present.

Chakrabarti *et al.* [24] developed a model of causality to describe how engineered and biological systems work, so as to use these systems to trigger ideation of new solution spaces during product development. The model consists of seven elementary constructs **S**tate change, **A**ction, **P**art, **Ph**enomenon, **I**nput, o**R**gan, and **E**ffect (SAPPhIRE). Chakrabarti et al. claim that the model through its constructs provides a rich description of function, behavior and structure of a system.

Howard *et al*. [25] combined a *modular system hierarchy* and *design activities process*. They proposed that a design process should refer to the types of activities carried out when given a design task, which should be independent of the systems level at which the task is set. They argued that the stages of *analysis of the task*, *conceptual design*, *embodiment design*, and *detailed design* form the *design activities process*. The activities, however, are not discussed in any further detail. The outcome of the *design activities process* from a design module at one system level feeds down to another system level module, each of which should go through every design activity, and so on.

Srinivasan and Chakrabarti [2] developed an integrated model of designing (GEMS of SAPPhIRE as req-sol) that combines Generate-Evaluate-Modify-Select (GEMS) activities model, the SAPPhIRE outcome model and the co-evolving requirement-solutions (req-sol) model. This model is validated by comparing it against existing protocol studies of design sessions where the model was not explicitly asked to be followed. It is found that the constructs of the model can be used to describe activities, outcomes, requirements and solutions in designing.

Environment based Design (EBD) methodology by Zeng [26] has three main activities: environment analysis, conflict identification, and solution generation. A major operation in EBD is *structure operation*, which is a means to represent a hierarchical system. According to Zeng, “in the design process, any previously generated design concept can be treated as an environment component for the succeeding design, as a result, a new state of design can be defined as the structure of the old environment (Ei) and the newly generated design concept (Si), which is a partial design solution”. Zeng [26] calls this change in state of environment as “evolution of environment”, where the new environment consists of the earlier environment plus the new design. However, this work does not propose system-environment co-evolution. This is because Zeng considers system and environment to be mutually exclusive; therefore, the new environment created by adding a change in the system amounts to changing either the system or its environment, but *not both*. The views of activity, requirement-solution and system-environment are explicit in the model. However, the outcome view is implicit and overlaps with the requirement-solution view.

2.2. Findings from Literature Survey

Table 1 summarises the views of activity, outcome, requirement-solution and system-environment in the current design theories, models and approaches. The following are the findings from the literature survey:

1) Blessing [11], VDI 2221 [16], Pahl and Beitz [7], Ullman [22] and Ulrich and Eppinger [23] explicitly address the views of activity, outcome, requirement-solution and system view (without environment), but not the system-environment view.

2) Constructs of the system view have been variously included in current theories, models and approaches. For example, Asimow [9] uses the system view to include system, subsystem, components and parts, Ullman [22] uses this view consisting of system–subsystem–assembly–component, and so on. However, not all the theories, models and approaches include this view. Even those that do recognize the system view, not all consistently address the view at all levels of abstraction.

3) Current models of designing focus on developing *only* the system and not its environment. Hence, it is understandable why environment has never been considered an *explicit, evolvable construct* in design process models. This absence of representation of environment as an explicit, evolvable construct has probably prevented detailed explanations and explorations of system-environment co-evolution, an important phenomenon in designing.. This is in spite of environment and its interaction with system being considered as critical by many researchers [9, 26].

4) Co-evolution of system and environment during the design process has not been represented in the reviewed theories, models and approaches. Zeng [26] proposes evolution of either system or environment, but not their co-evolution.

5) Current design theories, models and approaches link together activity, outcome, requirement-solution and system-environment views in fragmented ways. They do not consider all these views together. Even those theories, models and approaches that consider this, do not represent environment as a construct. Only few consider all these four views in designing (Table 1). While some take into account some of these views, none takes all the views into account together. However, the constructs of these views are not integrated consistently across all the stages.

Table 1: Classification of literature in terms of views of activity, outcome, requirement-solution and sy-en

Existing Work	**Activity**	**Outcome**	**Requirement**	**Solution**	**Sy-En view**
Asimow [9]	✓	✓			
Gero [10]	✓	✓			
Visser [11]	✓		✓	✓	
Blessing [12]	✓	✓	✓	✓	
Pahl and Beitz [7]	✓	✓	✓	✓	
Hubka and Eder [13, Figure 7–3]		✓			✓
Hubka and Eder [13, Figures 7-12, 7-13]	✓	✓			✓
Chakrabarti et al.[14]	✓	✓			
French [15]	✓	✓			
Nidamarthi [5]	✓		✓	✓	
Cross [16, pp. 3-11]	✓		✓	✓	
VDI2221 [16, pp. 38-41]	✓	✓	✓	✓	
Cross [16, p. 42]	✓		✓	✓	
Hansen and Andreasen [17]	✓	✓			✓
Lossack [19]	✓	✓			
Bhatta and Goel [20]		✓			
Campbell and Rai [21]	✓		✓	✓	
Ullman [22]	✓	✓	✓	✓	
Ulrich and Eppinger [23]	✓	✓	✓	✓	
Howard et al. [24]	✓				
Srinivasan and Chakrabarti [2]	✓	✓	✓	✓	
Zeng [26]	✓		✓	✓	✓

3. IDENTIFICATION OF CONSTRUCTS OF VIEWS OF ACTIVITY, OUTCOME, REQUIREMENT-SOLUTION

The model of activity was developed based on an extensive literature survey and empirical validation [2]. The different constructs used are *generate*, *evaluate*, *modify* and *select*. The following are the definitions of these activities. **Generate** is an activity that brings an outcome into an episode. **Evaluate** is an activity that judges the quality, importance, amount or value of an outcome in an episode. **Modify** is an activity that changes an outcome in an episode. **Select** is 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.

The model of outcomes developed by Chakrabarti *et al*. [24] was used and empirically validated in [2]. The definition of the different constructs is given in [2] as follows. **Phenomenon** is an interaction between an entity and its surroundings. **State change** is a property of the entity and its surroundings involved in the interaction. **Effect** is a principle of nature that governs the interaction. **Action** is an abstract description or high-level interpretation of the interaction. **Input** is a physical quantity in the form of material, energy, or information that comes from outside the boundary of the entity and is essential for the interaction. **Organ** is a set of properties and conditions of an entity and its surroundings that is required for the interaction. **Part** is a set of physical components and interfaces that constitute the entity and its surroundings. **Entity** is defined as a subset of the universe that is under consideration, and is characterized by its boundary; **surroundings** is defined as all the subsets of the universe except for the entity; interaction is communication between a entity and its surroundings to reach equilibrium.

The co-evolving model of requirement-solution was developed and empirically validated in [2]. The definition of the constructs is as follows. **Requirement** is defined as an expression of what a design should have at a level of abstraction. **Solution** is defined as a means to satisfy requirements.

Srinivasan and Chakrabarti [2] developed an integrated model of designing, GEMS of SAPPhIRE as req-sol. The integrated model is developed by combining the individual models of activity, outcome and requirement-solution. According to this model, GEMS activities are performed on the SAPPhIRE outcomes, which evolve as requirements or solutions.

Hall [27] in his systems engineering process used a system-environment view which consists of *system*, *environment*, *subsystems* and *objects*. In [27], Hall considered initial and final environment in the systems engineering process, but environment was not considered in the processes in between. Deng *et al.* [28] considered e*nvironment* as an explicit element in their *Function-Behavior-Working Environment-Structure* model of designing. INCOSE [29] for the systems engineering process used a system view with the following constructs: *system*, *element* or *segment*, *subsystem*, *assembly*, *subassembly*, *components* and *parts*. Hubka and Eder [13] used a system-environment view that consists of: *system*, *subsystem*, *elements*, *components*, *environment* and *active environment.*

The different constructs of the system-environment view proposed by Ranjan *et al*. [30] are relationships, elements, subsystem, system and environment. In this, both system and environment are evolvable constructs in the process of designing. The above constructs are defined as follows. A **system** is the overall product being designed, at any level of abstraction. A **subsystem** is a subset of a system that can be further divided. An **element** is a subset of a system or a subsystem, which cannot be further divided. An **environment** refers to all subsets of the universe apart from the system. The **relationships** are how system, environment, subsystems, and elements are linked with one another. Elements (and subsystems) combine together to comprise subsystems. All subsystems and elements combine together to comprise the system. This system-environment view is adapted as it uses environment as an explicit and evolvable construct. These constructs are illustrated with an example of a ballpoint pen. At the level of abstraction of *part*, the ballpoint pen, a system, consists of a refill, body and a cap. The refill, a subsystem, can be further sub-divided into elements, nib, ink and ink-reservoir. The body consists of elements, upper-body and lower-body. The environment for the ballpoint pen consists of papers on which it has to write and an agent that uses it to write. This system hierarchy also exists at other levels of abstraction. For example, at the *action* level, the action of the system (ballpoint pen) is ‘to write on paper’ which can be sub-divided into the following subsystems at action-level: store ink, supply ink, etc.

4. PROPOSED MODEL

Based on the constructs identified from the previous section, we propose an extended, integrated model of designing by combining the constructs of the views of activity, outcome, requirement-solution and system-environment. The model is represented in Figure 1, in a 4-dimensional space with the four views being the axes. Designing can be described using these views, as a series of steps (i.e. as a point in this 4D space), where each step is represented as a combination of one construct from each axis.

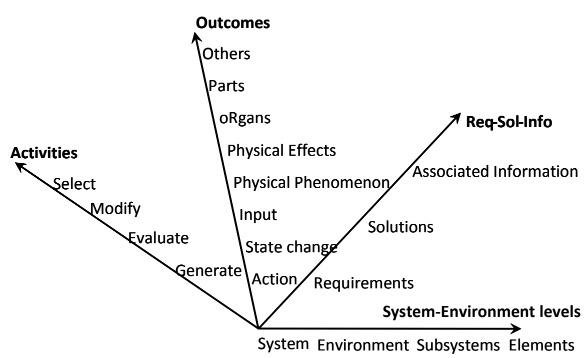

Figure 1 Extended GEMS of SAPPhIRE as req-sol

The model is explained as follows: The SAPPhIRE outcomes are generated, evaluated, modified and selected (select includes both acceptance and rejection). These outcomes can be requirements, solutions or associated information. In a design process, the outcomes evolve as requirements, solutions and associated information. These outcomes can be at any level of system-environment view, i.e., these outcomes can be at system, subsystem, element, environment and relationship level. The outcomes evolve at different levels of system and environment view. Thus, according to this model *the GEMS activities are performed on SAPPhIRE outcomes, which evolve as requirements or solutions of relationship, element, subsystem, system or environment.*

4.1. Preliminary Validation of the System-Environment View

As a preliminary validation of the proposed model, Ranjan *et al.* [30, 31] used protocol studies of a series of design sessions, to carry out an empirical investigation of the presence of the system-environment view in the design sessions. This involved checking whether or not all the constructs in the proposed system-environment view are present in the design sessions. The investigation confirmed that the constructs are present in the design processes.

4.2. Example of the proposed model - Ballpoint Pen

The example below (Tables 2-7) is an analysis of a ball-point pen [32] using the proposed model of designing. Hence, the activities involved in obtaining these outcomes are not explained in the example.

Table 2 Part level outcomes

Parts	Requirements	Solutions
System	Writing device	Ball-Point Pen
Subsystem	Container and supplier of ink within writing device Mechanism for writing Facility for comfortably holding the pen	Refill Nib Body or cover
Elements	Avoid bleeding or blotting on the paper or agent Material to create mark on paper	Cap; upper cover; lower cover; ink; ink-reservoir; Roller-ball; Socket;
Environment	Surface for writing Movement of writing device Quick drying of ink	Paper Agent Atmosphere

Table 3 Input/oRgan level outcomes

Input/oRgan	Requirements	Solutions
System	Ink should dry quickly. Writing should not get erased for a period. To make sure there is minimum or no bleeding on paper and agent.	Movement of pen; Smooth motion of pen on the paper; Gravitation pull of the earth for the ink flow; Suitable atmosphere for evaporation of ink; Proper dimensions of the narrow gap between roller ball and socket
Subsystem	Temperature Ingredients of the ink, viscosity of the ink, heat transfer co-efficient of the parts and so on.	Empty space between refill and cover acting as an insulation Material of the cover; Adhesion coefficient between ink and cover;
Elements	Gravity Mass of the ink	Movement of pen leading to rotation of roller-ball The roller-ball and holding tube should form a spherical joint creating a roller mechanism; The dimensions should be such that there is space for ink to flow.
Environment	Gravity	Gravity Right atmosphere for evaporation of ink

Table 4 Effect level outcomes

Effects	Requirements	Solutions
System	Amount of deposition depends on applied force and friction between writing device and writing surface	Newtonian laws of motion → Gravity flow → Adhesion →Evaporation and Diffusion
Subsystem	Temperature dependence of liquid viscosity	Fluid takes the shape of the container or Adhesion between ink and inner wall of the cover or Insulate against temperature
Elements		Newtonian laws of motion Gravity flow Adhesion A mount of Deposition Evaporation Diffusion of ink [Fick's laws of diffusion]
Environment		Gravity

Table 5 Phenomenon level outcomes

Phenomenon	Requirements	Solutions
System	Deposition of ink on paper	Ink flows from ink-reservoir to nib → Filling of the ink in the narrow gap of the nib → Roller ball rotates → Paper and ink get in contact with each other → ink gets deposited on the paper → ink evaporates and diffuse to get the writing on paper
Subsystem	Bleeding of ink	Containment of leaked ink
Elements		Ink flows from ink-reservoir to nib Filling of the ink in the narrow gap of the nib Roller ball rotates Paper and ink get in contact with each other ink gets deposited on the paper ink evaporates and diffuse to get the writing on paper
Environment	Deposition of ink on paper	

Table 6 State change level outcomes

State Change	Requirements	Solutions
System		Ink in the pen → Ink on the paper and the writing job is done
Subsystem	Bleeding to no-bleeding or Bleeding to less-bleeding	Leakage to no leakage; or no containment to containment of leakage
Elements	No mark on paper to mark on paper	Ink in ink reservoir to Ink on roller-ball Ink on roller-ball to Ink on paper
Environment		No ink to ink

Table 7 Action level outcomes

Action	Requirements	Solutions
System	Write on a paper	Make a readily distinguishable mark on the paper
Subsystem	Contain and supply ink to write Eliminate or minimise the bleeding of ink on agent	Avoid leakage or contain leakage.
Elements	Write on paper	Allow ink from ink-reservoir to paper
Environment	Writing	Allow ink to make a readily distinguishable mark

This example (Tables 2-7) illustrates how a part level outcome, the *body*, shares different actions such as *facility for comfortably holding the pen* and *avoid bleeding or blotting on the paper or agent*, but at different levels of the system-environment view. This is an example of structure sharing – sharing of the same structure for carrying out multiple functions [33, 34, 35].

An example of the system-environment co-evolution during designing can be seen in the development of writing devices with ink inside, from the likes of pen and inkpot. Design effort initially focused on either the pen as the system to be designed with the inkpot as part of the ‘given’ environment, or the inkpot as the system with the pen as part of the environment. In either case, design of one led to redesign of the other. Subsequently, the two were considered together, leading to design of integrated pen-inkpot systems, such as a fountain pen or a ballpoint pen.

5. SUMMARY

From literature, it is found that various design theories, models and approaches include constructs of the system view, but not consistently at all levels of abstraction. Environment is rarely considered as an explicit, evolvable construct in these design theories, models and approaches. The absence of environment as an explicit construct may have prevented explanations and explorations of system-environment co-evolution, an important phenomenon in designing. The views of activities, outcomes, requirement-solution and system-environment are found to be important in designing. Some of the current design theories, models and approaches link together these views, albeit in a fragmented manner. Not all theories, models and approaches consider all the views together, and those that do, do not represent environment. Also, the constructs of these views are not integrated consistently across all design stages.

Srinivasan and Chakrabarti [2] used a comprehensive review of literature to identify the constructs of the views of activities, outcomes and requirement-solution. Using these constructs they developed and validated GEMS of SAPPhIRE as req-sol, an integrated model of designing that combines activities view, outcomes view and requirement-solution view. However, this model did not represent the system-environment view. Based on this model and the constructs of the system-environment view identified in this paper, we propose here an extended, integrated model of designing. This model integrates the system-environment view, activity view, outcome view and requirement-solution view. The originality of the paper lies in this proposal of the model that integrates the four views of designing, so as to help explain various complex phenomena such as system-environment co-evolution, or structure sharing. An explanation of the proposed model is given using the example of a *ballpoint pen* (Tables 2-7): from the example, it can be noted that the system-environment view exists at every level of outcome abstraction. An example is also provided to explain structure sharing using the model. An example explaining system-environment co-evolution is also given. These indicate the potential of the model to explain various complex phenomena related to designing.

6. CONCLUSIONS

In this paper, an extended, integrated model of designing is proposed, which incorporates the system-environment view into the erstwhile GEMS of SAPPhIRE as req-sol model of designing. The proposed model asserts that design progresses as follows: GEMS activities are performed on SAPPhIRE outcomes, which evolve as requirements or solutions of relationship, element, subsystem, system or environment. The model has the promise to explain various complex phenomena such as system-environment co-evolution, other interactions within the system, and requirement-solution co-evolution.

7. FUTURE WORK

A more extensive, empirical validation of the proposed model is required. Based on the proposed model, a framework is planned to be developed for supporting creativity in design.

REFERENCES

[1] Blessing, L., and Chakrabarti, A., (2009), DRM, A Design Research Methodology, Springer, London, UK.

[2] Srinivasan, V., and Chakrabarti, A., (2010), "An Integrated Model of Designing", in Special Issue on Knowledge Based Design, ASME Journal of Computing and Information Science in Engineering (JCISE), Vol. 10, Issue 3, ed. by Ashok K. Goel and Andrés Gómez de Silva Garza, 031013.

[3] Stauffer, L., and Ullman, D., (1991), "Fundamental Processes of Mechanical Designers Based on Empirical Data", Journal of Engineering Design, Vol. 2, Issue 2, pp. 113-125.

[4] Cooper, R., (1991), "New Products: What Distinguishes the Winners?", Research and Technology Management, Vol. 33, Issue 6, pp.27-31.

[5] Nidamarthi, S., (1999), Understanding and Supporting Requirement Satisfaction in the Design Process, PhD thesis, University of Cambridge, UK,.

[6] 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.

[7] Pahl, G. and Beitz, W., (1996), Engineering design: A systematic approach, Spriger-Verlag, London, UK.
[8] 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.
[9] Asimow, (1962), Introduction to design, Prentice-Hall Inc, Englewood Cliffs (NJ).
[10] Gero, J.S., (1990), "Design prototypes: A knowledge representation schema for design", AI magazine, Vol. 11, No. 4, pp. 26-36.
[11] Visser, W., (1991), "Evocation and elaboration of solutions: Different types of problem solving actions"; Proceedings of 3rd COGNITIVA symposium on at the crossroads of artificial intelligence, cognitive science, and neuroscience, Madrid, Spain, pp. 161-168.
[12] Blessing, L., (1994), A process-based approach to computer-supported engineering design, PhD thesis, University of Twente, Netherlands.
[13] Hubka, V. and Eder, W., (1996), Design Science: Introduction to the Needs, Scope and Organization of Engineering Design Knowledge, Springer-Verlag London Limited, London, UK.
[14] Chakrabarti, A., Johnson, A. and Kiriyama, T., (1997), "An approach to automated synthesis of solution principles for microsensor designs"; Proceedings of International Conference on Engineering Design (ICED97), Helsinki, Finland, pp.125–128.
[15] French, M., (1999), Conceptual design for engineers, Springer–Verlag, Berlin, Germany.
[16] Cross, N., (2000), Engineering design methods– Strategies for product design, John Wiley and Sons, England.
[17] Hansen, C. and Andreasen, M., (2002), "Two approaches to synthesis based on the domain theory", in Engineering design synthesis – understanding, approaches and tools, ed. by A. Chakrabarti, Springer-Verlag, London, pp. 93 -108.
[18] Hubka, V. and Eder, W., (1988), Theory of technical systems, A Total Concept theory for Engineering Design, Springer-Verlag.
[19] Lossack, R., (2002), "Design processes and the context for the support of design synthesis", in Engineering design synthesis–Understanding, approaches and tools, ed. by A. Chakrabarti, Springer-Verlag, London, pp. 213-227.
[20] Bhatta, S. and Goel, A., (2002), "Design patterns and creative design", in Engineering design synthesis – understanding, approaches and tools, ed. by A. Chakrabarti, Springer-Verlag, London, pp.271-284.
[21] Campbell, M. I. and Rai, R., (2003), "A generalization of computational synthesis methods in engineering design"; Proceedings of AAAI Spring Symposium Series, Calif., USA.
[22] Ullman D.G., (2003), The Mechanical Design Process, 4th Edition, McGraw Hill.
[23] Ulrich K. T., Eppinger S.D., (2003), Product Design and Development, 3rd Edition, Tata McGraw Hill.
[24] Chakrabarti, A., Sarkar, P., Leelavathamma, B. and Nataraju, B., (2005), "A functional representation for aiding biomimetic and artificial inspiration of new ideas", Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AI EDAM), Vol. 19, Issue 2, pp. 113–132.
[25] Howard, T., Culley, S., Dekoninck, E., (2009), "The integration of systems levels and design activities to position creativity support tools"; Research into Design: Supporting Multiple Facets of Product Development, ICoRD'09,Bangalore, India, pp.43-50.
[26] Zeng, Y., (2011), "Environment-Based Design (EBD) Methodology", ASME 2011 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, Washington DC, USA, DETC 2011-48263.
[27] Hall A.D., (1962), A methodology for systems engineering, 6th ed. Princeton (NJ), Van Nostrand.
[28] Deng Y.M., Tor S. B. and Britton G. A., (2000), "Abstracting and Exploring Functional Design Information for Conceptual Mechanical Product Design", Engineering with Computers 16, Springer-Verlag London Limited, London, UK, pp 36–52.
[29] INCOSE, (2004), Systems Engineering Handbook - a "What To" Guide for all se Practitioners, INCOSE-TP-2003-016-02, Version 2a, Released by - Technical Board, International Council on Systems Engineering (INCOSE).
[30] Ranjan B. S. C., Srinivasan V., Chakrabarti A., (2012), "System-Environment View in Designing"; CIRP Design 2012, Bangalore, India (accepted).
[31] Ranjan B. S. C., Srinivasan V., Chakrabarti A., (2011), "Investigation of the System-Environment View in Designing"; PLMSS11, Coimbatore, India.
[32] http://en.wikipedia.org/wiki/Ballpoint_pen.
[33] Chakrabarti, A., (2004), "A new Approach to Structure Sharing", ASME Journal for Information in Science and Engineering (JCISE), Vol. 1,Issue 1, pp. 1-78.
[34] Kumari, M.C., and Chakrabarti, A. "A Method for Structure Sharing in Logo Design"; Research into Design: Supporting Sustainable Product Development, ICoRD'11, Bangalore, India, pp.834-842.
[35] Chakrabarti, A., and Singh, V., (2007), "A Method for Structure Sharing to Enhance Resource-Effectiveness", Journal of Engineering Design, Vol. 18, Issue 1, pp. 73-9.