A Theoretical Comparison of a Framework for Designing with Existing Models and Approaches

Srinivasan V* and Amaresh Chakrabarti **

Innovation, Design Study and Sustainability Laboratory, Centre for Product Design and Manufacturing, Indian Institute of Science, Bangalore, India
** srinivasan@cpdm.iisc.ernet.in ** ac123@cpdm.iisc.ernet.in*

Abstract: The objective of this paper is to theoretically compare a prescriptive framework for designing (GEMS of SAPPhIRE as req-sol) with the existing design models and approaches to assess the similarities and differences. The framework integrates activities, outcomes, requirements and solutions; addresses conceptual and early embodiment stages; and is intended as a support for creating novel designs. The comparison criteria are: nature of work (descriptive or prescriptive); kinds of activities, outcomes, requirements and solutions; stages of designing supported and purpose of the work. Results of comparison reveal that the framework maps well with the existing work; however, no single existing work includes activities, outcomes, requirements and solutions, and supports novelty in conceptual stage.

Keywords: framework, GEMS of SAPPhIRE, activity, outcome, requirement, solution.

1. Introduction

Designing involves interactions within and among its elements – product, process, people, tools, organisation and environment in which designing takes place [1]. The current literature contains several designing models and approaches, which respectively describe and prescribe designing using one or more of the above elements. A prescriptive framework for designing – GEMS of SAPPhIRE as req-sol – was developed in [2] as a support for developing novel designs and integrates elements of product and process. The objective of this paper is to theoretically compare the framework with the existing literature to analyse its similarities and differences.

2. GEMS of SAPPhIRE as req-sol – A framework for designing

A prescriptive framework for designing was proposed in [2] as a support for developing novel designs. The framework integrates activities, outcomes, requirements and solutions. An activity is defined as a deed of problem-solving; an outcome is defined as a property of a design at an abstraction level; a requirement is defined as what a design should have at an abstraction level; and a solution is defined as a means at an abstraction level to satisfy requirements. The following specific elements were identified in [3] for the framework: **G**enerate, **E**valuate, **M**odify and **S**elect (**GEMS**) (Table 1) as activities; **S**tate change, **A**ction, **P**arts, **Ph**enomenon, **I**nput, o**R**gan and **E**ffect (**SAPPhIRE**) (Table 2) as outcomes; and co-evolving **req**uirements and **sol**utions. An idea is defined here as a solution that satisfies only a sub-function. A concept is defined here as a solution that satisfies an overall function i.e., all sub-functions. A concept is thus a combination of ideas of all sub-functions. For example, a rover has to rove (overall function) and has sub-functions: mobility, stability and steering; solutions of mobility comprise ideas for mobility; concepts for roving will combine ideas of mobility, stability and steering.

The framework is divided into two stages: requirement synthesis and solution synthesis (Fig. 1) (Notation: G: Generate, E: Evaluate, M: Modify, S: Select; a: action, s: state change, i: input, ph: phenomenon, e: effect, i: input, r: organ, p: part; re: requirement, so: solution; Y: Yes, N: No; x[y(z)]: An activity 'x' of requirement or solution 'y' at an outcome level 'z'. For e.g. G[re(a)] means Generation of requirement at the action-level, S[so(p)] means Selection of solution at the part-level, etc.; E[y(z)⇔y'(z')]: An evaluation of solution y which is at an outcome level z against requirement or solution y' which is at an outcome level z'. For example, E[so(p)⇔re(p)] means Evaluation of solution at part-level against requirements at part-level, E[so(p)⇔so(r)] means Evaluation of solution at part-level against solution at organ-level). In *requirement synthesis*, requirements at different abstraction levels including SAPPhIRE are generated, evaluated, modified and selected. In the requirements synthesis stage, 'others' category contains requirements that pertain to: (a) ideas at more detailed levels of abstraction, and (b) concepts. *Solution synthesis* is sub-divided into: *idea synthesis* and *concept synthesis*. In *idea synthesis*, ideas at different abstraction levels of SAPPhIRE are generated, evaluated, modified and selected, one by one, in the order of decreasing abstraction level – action, state change, phenomenon, effect, input and organ, and part. In *concept synthesis*, concepts are generated (from selected ideas), evaluated, modified and selected. The activities and outcomes respectively comprise the process and product elements; product elements evolve during designing as requirements or solutions.

Table 1: Definition of activities

Activity	Definition
Generate	An activity that brings an outcome into an episode[1]
Evaluate	An activity that judges the quality, importance or value of an outcome in an episode[1]
Modify	An activity that changes an outcome in an episode[1]
Select	An activity that chooses an outcome as acceptable or unacceptable in an episode[1]

Table 2: Definition of outcomes [4]

Outcome	Definition
Phenomenon	An interaction between a system and its environment.
Effect	A principle of nature that underlies/governs an interaction.
Input	A physical variable that crosses the system boundary, and is essential for an interaction between a system and its environment.
State	A property at an instant of time of a system (and environment), that is involved in an interaction.
Action	An abstract description or high-level interpretation of an interaction between a system and its environment
o**R**gan	A set of properties and conditions of a system and its environment required for an interaction between them.
Part	A set of physical components and interfaces that constitute the system and its environment.

3. Research Methodology

The research methodology has the following steps:

(1) Review existing literature: The existing literature is reviewed to identify the elements of process and product for the comparison.

(2) Compare the framework with the existing literature: The framework is compared with each of the existing literature using the following criteria, to identify similarities and differences: (a) nature of the

[1] An episode is defined as a situation within which something happens and that can help explain it

existing literature – descriptive or prescriptive, (b) stage(s) of designing addressed, (c) the kinds of activities, outcomes, requirements and solutions used, and (d) the specific purpose of the existing work.

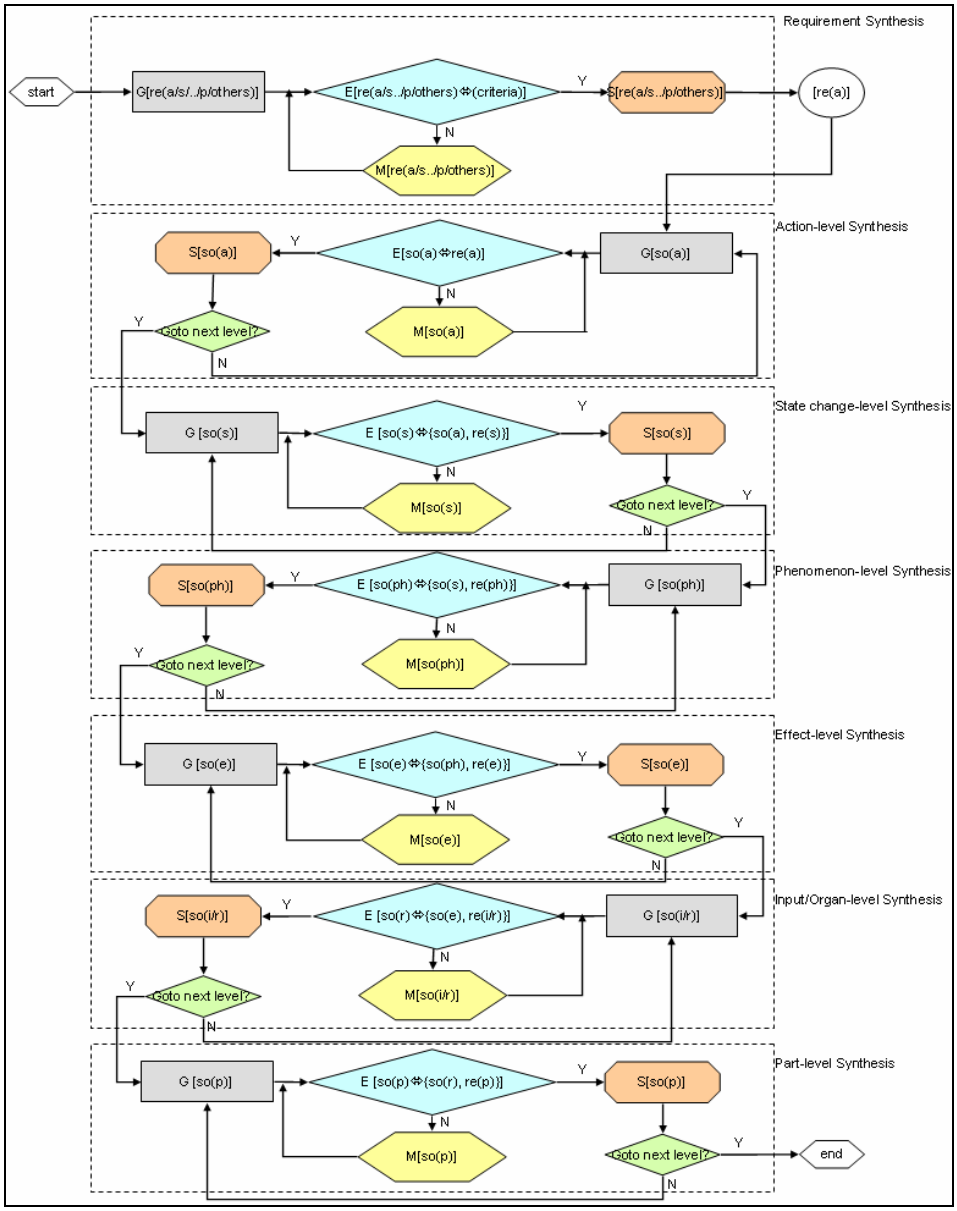

Figure 1: Pictorial representation of requirement and solution (idea) synthesis

4. Results

This section reports findings from the review and comparison of the framework with the existing work.

Based on protocol studies of individual designers, Stauffer and Ullman [5] developed a descriptive model of the mechanical engineering design process. They identified two levels of activities: *categories* (generate; evaluate; decide) and operators (select and create; simulate, compare and calculate; accept, reject, suspend, refine and patch). Stauffer and Ullman categorised the protocol data using the categories, selected portions of the classified data at random and analyzed these portions to recognize patterns of categories. Most of these could be described by the sequences: generate and test, generate and improve, means end analysis and deductive thinking. These sequences in general, constitute generate-evaluate-decide. The sequences were applied on constraints and proposals. The model in [5] addresses the conceptual stage of designing, to provide a better understanding to the

process of mechanical engineering design and use this understanding to develop computer-aided tools. In general, create; simulate, compare and calculate; refine and patch; and, accept and reject of [5] respectively resemble generate; evaluate modify and select of [2]. Overall, generate, evaluate and decide of [5] resemble generate, evaluate; modify; and select of [2]. Constraints and proposals of [5] resemble requirements and solutions of [2].

Blessing [6] proposed a prescriptive process-based approach for designing. The design matrix is the core of this approach and consists of activities and outcomes. Activities – generate, evaluate and select – are applied to outcomes at different abstraction levels (referred in [6] as proposals) – problem, requirements, function, concept and detailed design. A design matrix relates to a product or to one of its assemblies or components; several matrices are used in any one process. The approach addresses designing from problem identification to detail design. This approach is used to develop a computer-based design support tool to aid the improvement of the design process in terms of its effectiveness and efficiency. Generate, evaluate and select of [6] respectively resemble generate, evaluate and select of [2]. In general, problem and requirements, and function, concept and detail design can be respectively classified as requirements and solutions.

Chakrabarti et al. [7] proposed a prescriptive approach for designing micro sensors based on activities and outcomes. The process of designing sensors involves: generating a design concept, identifying potential behavioral problems with the concept and trying to fix these problems. This is analogous to generating a design concept, evaluating the concept (to identify problems) and modifying the concept (to fix the problems). The outcomes exist at two abstraction levels: function, represented using input-output description of physical quantities and solution principle, represented using alternative chains of laws and effects. This approach is valid for the conceptual stage of designing. Generate, evaluate and modify of [7] respectively resemble generate, evaluate and modify of [2]. The input (of the function) and solution principle of [7] can be classified respectively as input and effects of [2].

Based on protocol studies of designing of both individual and team designers working in laboratory and industrial settings, Nidamarthi [8] proposed a descriptive model of designing that combines activities, requirements and solutions. In this model, designing is divided into two phases: problem understanding (PU) and problem solving (PS). In PU, designers identify, analyse and choose design problems. In PS, designers generate, evaluate and select solutions. Two levels of activities – primary and secondary – are identified; secondary-level activities are categorized into a primary activity. Nidamarthi interprets modification (secondary activity) as a form of identification in PU and generation in PS phase. Nidamarthi also showed that requirements and solutions co-evolve, and mutually influence one another. Identify and generate; analyse and evaluate; and choose and select of [8] respectively resemble generate-cum-modify; evaluate; and select of [2]. PU and PS phases in general respectively resemble requirement and solution synthesis stage in [2].

Cross [9] proposed a descriptive model based on activities. The four staged model can be described as: exploration of a problem-space; generation of a concept to the problem; evaluation of the concept against the goals, constraints and criteria of the design problem; and communication of the evaluated design to manufacture. This can be seen as a combination of activities (exploration, generation, evaluation, communication), requirements (problem) and solutions (concept). A feedback loop from evaluation to generation signifies that not all evaluated designs are communicated but at times a more satisfactory design needs to be generated. This approach spans from exploration of problem space to description for manufacturing. Exploration in [9] is equivalent to generation, evaluation, modification and selection in [2]. Generation and evaluation in [9]

respectively resembles generation and evaluation in [2]. Problem and solution in [9] can be generally categorised respectively under requirements and solutions.

Miller et al. [10] proposed a descriptive model of a fundamental sequence of thinking processes based on activities. The **T**est-**O**perate-**T**est-**E**xit (TOTE) model has four steps: testing of initial state, operation of change, testing of resulting state, exiting if result of testing is satisfactory, otherwise the operation is adopted and repeated. In complex thinking processes, the TOTE sequences are linked in a chain or several modification processes are executed before a testing process takes place. Since this is an approach of thought process, it should be applicable to all stages of designing. Testing and adapting in [10] are respectively equivalent to evaluating and modifying in [2]. Exiting in [10] means acceptance leading to selection or non-acceptance leading to non-selection, and therefore, resembles selection in [2].

Based on an observational study of an experienced designer working in aerospace industry, Visser [11] developed a descriptive model of problem solving, based on activities, requirements and solutions. At a high level of abstraction, designing may be described as: (a) proceeding in iterative cycles, and (b) leading progressively from a problem specification to a detailed solution. In each cycle a solution to a problem is developed, evaluated, accepted or rejected. The two forms of solution development are: (a) solution evocation – designer recalls a solution from memory that matches a problem/solution association and (b) solution elaboration – when an evoked solution fails, a designer creates new solutions without recalling from memory. Development (evocation and elaboration), evaluation and acceptance/ rejection of [11] respectively resemble generation, evaluation and selection of [2]. Problem specification and solution of [11] can generally be categorised into requirements and solutions.

Hubka and Eder [12] proposed a hierarchy of activities in designing at different levels (design stages, design operations, basic operations, elementary activities and elementary operations). Two kinds of relationships exist between the activities at different levels: (a) hierarchical – every activity at a level contains activities at the next lower level and so forth and (b) block dependency – some activities are repeatedly applied in a certain sequence. Their views are based on their experience and not on direct observations of designing. In this paper, only the activities at the level of basic operations – state the problem; search for solutions; evaluate, decide; communicate solution; prepare information; verify, check; and represent information – are considered because they constitute the problem-solving process. State, search and prepare, and evaluate, verify and check of [12] can be represented respectively using generate and evaluate of [2] . Problem and solution of [12] can be generally classified respectively under requirement and solution.

French [13] developed a descriptive model of designing based on activities, outcomes, requirements and solutions. Based on an initial statement of a need, a problem statement is formulated after an analysis of the problem. A problem statement has three parts – the statement of design problem, limitations placed upon solutions, and criteria of excellence. In conceptual design, schemes are produced for the problem statement. In the embodiment of schemes, selected schemes are developed into a set of general arrangement drawings. In detailing, the general drawings are further detailed to working drawings. According to French, activities like evaluation are present in all the stages. He also indicates the iterative nature of designing with feedback loops from embodiment stage to conceptual design to problem analysis, indicating that modification and rejection maybe present. This approach spans designing from problem analysis to detail design without elaborating on any particular stage. In general, statement of the problem, and selected schemes, general arrangement drawings and

working drawings of [13] could be respectively classified under requirements and solutions.

Hubka's theory of technical systems [12] is a prescriptive, outcome-based approach to describe a technical system through the following abstraction levels: process, function, organ and constructional structure. The purpose of a technical system (TS) is described in terms of the output effects delivered to accomplish a technical process, which involves a transformation of an operand from its initial to final state. In the process structure, all the internal transformations of inputs to the TS (desired and undesired) into desired outputs (output effects) and secondary outputs are described. An effect is an output of a chain of transformations, where each transformation is based on the laws of physics, chemistry, biology, etc. In the function structure, all functions (defined as the property of TS to fulfil its purpose) – transformation and purpose – that are needed in the TS to create the effects that achieve the desired transformation of the operand are described, including the internal processes of the TS. In the organ structure, organs (defined as function carriers) are described. Each organ realises one or more of the functions or property of the system, through some physical effect. In the constructional structure, the components and their relationships that realise the organs are described. The constructional structures exist at various levels of detail – sketch layouts, dimensional layouts, detail part drawings, etc. Several components along with their relationships help realise an organ and several organs together help realise a function. The technical process; transformation of operands from input to output state; functions; and components of [12] respectively resemble action; state change; physical phenomena and parts of [2].

Andreasen's domain theory [14] is an outcome-based prescriptive approach that describes the mechanical artefact to be designed using three abstraction levels, termed as domains: transformation, organ and part. In the transformation domain, the purpose of an artefact along with its user is described as a means to support a transformation process. The interaction between the product and its user produces effects necessary for the transformation of an operand from its initial to end state. A series of operations (horizontal chain of causality) where each operation alters one or more of the characteristics of the operand, transforms the operand from the initial to the end state. In the organ domain, organs (active elements that create the required effects in the mechanical product) are described. The mode of action of an organ is based on a physical effect. An organ consists of wirk elements, which is classified into wirk surface, volume and field. In the part domain, the wirk elements are allocated among the parts in such a way that the parts can be produced and assembled, according to the requirements of the wirk elements. Each organ is controlled by several wirk elements and each part consists of several organs. This approach supports designing spanning from task identification to detail design. The transformation process, organs and parts of [14] respectively resemble state change, organs and parts of [2].

Gero [15] developed Function-Behaviour-Structure (FBS), a descriptive model of designing based on activities and outcomes. The model is based on three properties of an artefact: Function (F), defined as what the artefact is for, Behaviour (B), defined as what the artefact does and Structure (S), defined as what the artefact consists of. The transformation from function to design description is through a series of following eight steps: formulate a set of expected behaviour (B_e) from function (F); synthesise a structure (S) from the set of B_e; analyse the S to derive actual behaviour (B_a); evaluate the B_a and B_e to check if they match, document the design description (D) from the S; reformulate the S, if the behaviours do not match in analysis; reformulate the B_e, if the behaviours do not match in analysis; and reformulate F, if the behaviours do not match in analysis. This model spans designing from need identification to design description. Action, state change and input; physical phenomena, physical effects and organs; organs and parts of [2] are different forms of respectively representing function; behaviour

and stru cture of [15]. Formulate, syn thesise and document; evalua te and a nalyse; reform ulate of [15] respectively resemble generate, evaluate and modify of [2].

Bhatta and G oel [1 6] p roposed a prescriptive outcome-based a pproach - S tructure-Behaviour-Function (SBF) language. This approach provides conceptual primitives for representing and organising knowledge of structures, behaviours and functions of a device. The stru cture of a d evice is seen as a co nstitution of co mponents an d substances, and is represen ted h ierarchically in terms o f its co nstituent struc ture elements and relations among them. The behaviour is seen as a se ries of state c hanges from an i nput t o an output state. The function i s a behavioural abstraction and is represented as a schema that specifies the behavioural state the function takes as input an d ou tput. Th e tran sition fro m o ne state to anoth er is don e using 'allow functions'. Th is app roach addresses co nceptual design. T he f unction, behaviour, structure a nd 'allow f unctions' of [16] respectively resemble action, state change, parts and organs of [2].

Reich [17] reviewed Yoshikawa's General Design Theory (GDT), which is a prescriptive mathematical approach based on knowledge of outcomes. A ccording to Reich, GDT has a flavour of descriptive theory of d esign but provides a prescription for development o f CAD system s. Th e th eory atte mpts to ex plain d esigning using set theory. I n a st ate of i deal k nowledge, des igning i s a di rect mapping f rom t he k nowledge i n f unction t o t he attribute space, under the assumptions that both the knowledge spaces have a topological structure and can be managed with infinite resources. Under the ideal state, a successful termination of design is guaranteed. In a state of real knowledge, the function a nd attribute knowledge s paces do not have a topological structure, cannot be manipulated by infinite resources and hence, successful design is no t guaranteed. GDT prescribes that the real knowledge sho uld be sup plemented with additional d esign strateg ies an d assu mptions, to arriv e at a d esign solution. This theory has been used to provide guidelines for building CAD systems. The function and attribute of [17] is equivalent respectively to action and parts of [2].

Tomiyama e t al. [18] pr oposed a pr escriptive ap proach – Metam odel, w hich is b ased on GDT. In th e real knowledge, design is a stepwise evolutionary transformation process. After candidate solutions are generated for given specifications, they will be improved towards a final solution that satisfies the specifications with the use of Metamodel. A model here is an abstraction of the actual en tity that serves as a focus for a particular design stage. The metamodel mechanism integrates several models by integrating the background theories of the models. The following steps are followed: a current metamodel is observed, solutions are thought and obtained, solutions are evaluated using the models, a decision on which solution to adopt is taken and the next metamodel is made, if there is no solution that satisfies the requirements, then the current or older metamodel is revised. Metamodel serves as a central modelling basis of CAD and also serves as a mechanism to model physical phenomena. The mechanism also serves as a workspace for designers by providing them with various work models in the design process. Obtain, evaluate, re vise and decide of [18] are si milar resp ectively to generate, ev aluate, mo dify and select of [2]; physical phenomena of [18] resembles physical phenomena of [2].

Smithers [19] pr oposed $K_L D_O^E$, a d escriptive, g eneral, know ledge-level th eory of d esigning b ased on knowledge of activities (form, revise, justify, synthesise, specify, solve, analyse, assess and ev aluate), outcomes (requirements, pr oblem, sol ution, documentation st atements and presentation state ments), requ irements an d solutions. The theory is represented in the form of knowledge process diagram, to show the different knowledge types ge nerated a nd used during designing. T he t heory id entifies knowledge asso ciated with: requ irements; problem; sol ution; s olution a nalysis, asses sment an d e valuation; desi gn d ocumentation a nd rational rec overy;

and design presentation and, claims these to be necessary and sufficient in designing. These knowledge kinds are tagged with one of the knowledge types: domain, task, inference, process and justification. The theory being generic does not discuss about the content of the knowledge types because this would be equivalent to making claims about particular kinds of designing in particular domains and contexts. The theory under different conditions is used to account for, explain and define routine, innovative and original designing, leading to a theoretical definition for creative designing. The theory supports designing spanning from need identification to design documentation. Form, synthesise and specify; analyse, assess and evaluate; revise of [19] respectively resemble generate; evaluate; and modify of [2]. Justify and solve of [19] may be seen as a combination of evaluate and select; generate, evaluate, modify and select of [2], respectively. In general, requirement and problem statements; solution and documentation statements of [19] can be respectively categorised under requirements and solutions.

Hubka [20] proposed a prescriptive law that combines requirements and solutions. It states that, in the hierarchy of functions which contribute to the realisation of an artefact's overall purpose function, there exist causal relations, determined by the means that realise the functions. Transformation between these takes place using relationships that link functions to means, where each choice of a means leads to uncovering further functions and then to further means and so forth, developing into a function-means tree. In general, functions, means and their co-evolution in [20] respectively resemble requirements, solutions and their co-evolution in [2].

Maher et al. [21] developed a descriptive model of creative design based on the co-evolution of problem and solution space during which there is constant exchange of information between them. In general, problem, solution and their co-evolution in [21] respectively resemble requirements, solutions and their co-evolution in [2].

Verein Deutscher Ingenieure (VDI) [22] developed an outcome-based prescriptive approach to the design of technical systems and products based on activities, outcomes, requirements and solutions. The approach divides designing into seven stages: clarify and define task, determine functions and their structures, search for solution principles and their combinations, divide into realisable modules, develop layouts of key modules, complete overall layout and prepare production and operating instructions; it has the following outcomes: specification, function structure, principal solution, module structure, preliminary layouts, definitive layout and product documents. Specifications from the first stage are constantly reviewed and used in all the stages; this resembles the co-evolution between requirements and solutions in the framework. The guideline prescribes that several solution alternatives should be analysed and evaluated at each stage. Iterations between the stages, results in back and forth transitions between stages; suggests that modification and rejection maybe present. This approach spans from task clarification to manufacturing. Determine, search, develop, complete and prepare; clarify, analyse and evaluate of [22] respectively resemble generate and evaluate of [2]. Specifications; function structures, solution principles, module structures, layouts and product documents in [22] can be respectively categorised as requirements and solutions.

Pahl and Beitz [23] proposed a prescriptive approach that combines activities, outcomes, requirements and solutions. The approach divides designing into stages – planning and task clarification, conceptual design, embodiment design and detail design, which have as outcomes – specifications, concept, preliminary layout, definitive layout and documentation. Outcomes are requirements (specifications) or solutions (concept, preliminary layout, definitive layout, documentation). Planning and task clarification of this approach resembles requirement synthesis stage in the framework. Each of the stages has different activities – analyse, find, select,

clarify, elaborate, identify, establish, search, combine, evaluate, eliminate, check, prepare, upgrade and improve. Upgradation and improvement of design specification, after each stage of designing, indicates the co-evolution between requirements and solutions. This approach spans designing from task clarification to product documentation. Find, elaborate, establish, search and prepare; analyse, clarify, check and evaluate; upgrade and improve; select and eliminate of [23] respectively resemble generate, evaluate, modify and select of [2].

5. Discussion

The following are some salient points of discussion based on the above comparison:

a) Existing work use different activities for different stages of designing. Most of these activities can be categorised into one of the generic activities of the framework. The sequence of activities in the framework matches well with the existing work. It can be concluded that the framework's activities map well with the activities of the existing work.

b) Existing work use different kinds of outcomes in different stages of designing. A subset of these outcomes with similar names matches well with the framework's outcomes. Another subset with a different terminology also resembles the framework's outcomes. Much of the existing work does not make explicit use of physical laws and effects. Overall, most of the framework's outcomes map well with the outcomes of the existing work.

c) Existing work address requirements and solutions at different abstraction levels and their co-evolution. Solutions of the framework match well with most of the solutions (at conceptual and early embodiment stages) of the existing work. It can be concluded that the requirements and solutions of the framework and their co-evolution map well with the existing work.

d) Overall, the framework maps well with the existing work. However, the uniqueness of the framework can be argued through the following points:

   i. It uses the same set of activities with a similar sequence at all the abstraction levels.
   ii. The combination of outcomes provides a rich description of functionality (action, state change and input), behaviour (physical phenomenon, physical law/effect and organ) and structure (organ and parts).
   iii. The explicit use of physical laws and effects.
   iv. It supports novelty at the earlier stages of designing.
   v. Even though different existing work have similarities with the framework, no single work supports novelty in conceptual design through a combination of activities, outcomes, requirements and solutions (especially physical laws and effects) used by the framework.

6. Summary

A prescriptive framework for designing is compared with the existing descriptive and prescriptive literature, to assess its similarities and differences. Results of the comparison reveal that the framework maps well with the existing work. However, no single existing work addresses activities, outcomes, requirements and solutions, with emphasis on conceptual design and supports novelty. This contributes to the uniqueness of the framework.

References

1. Blessing, L., Chakrabarti, A. and Wallace, K. (1995) A design research methodology. In Proceedings of ICED95, pp. 50-55.

2. Srinivasan, V. and Chakrabarti, A. (2009) Designing Novel Artefacts: A novel systematic framework. In Research into Design – Supporting multiple facets of product development, pp. 67-75.
3. Srinivasan, V. and Chakrabarti, A. (2008) Design for novelty – A framework?. In *Proceedings of DESIGN 2008*, pp. 237-244.
4. Srinivasan, V. and Chakrabarti, A. (2009) SAPPhIRE – An approach to analysis and synthesis. ICED09. (Accepted)
5. Stauffer, L.A. and Ullman, D.G. (1991) Fundamental processes of mechanical designers based on empirical data, Journal of Engg. Des., Vol. 2, No. 2, pp. 113–125.
6. Blessing, L. (1994) A Process-based approach to computer-supported engineering design. PhD thesis, University of Twente, Netherlands.
7. Chakrabarti, A., Johnson, A. and Kiriyama, T. (1997) An approach to automated synthesis of solution principles for microsensor designs. In *Proceedings of ICED97*, pp. 125-128.
8. Nidamarthi, S. (1999) Understanding and Supporting Requirement Satisfaction in the Design Process, PhD thesis, University of Cambridge, UK.
9. Cross, N. (2000) Engineering design methods – Strategies for product design, John Wiley & Sons, pp. 3-11.
10. Miller, G. A., Galanter, E. and Pribram, K. (1960) *Plans and the Structure of Behaviour*, Holt, Rinehardt and Winston, New York.
11. Visser, W. (1990) Evocation and Elaboration of Solutions: Different Types of Problem Solving Actions. In *Proceedings of COGNITIVA 90*, pp. 689 – 696.
12. Hubka, V. and Eder, W. E. (2002) Theory of technical Systems and engineering design synthesis. In *Engineering Design Synthesis - Understanding, Approaches and Tools*, Springer-Verlag, pp. 49-65.
13. French, M. (1999) Conceptual design for engineers, Third Edition, Springer-Verlag.
14. Hansen, C. T. and Andreasen, M. M. (2002) Two approaches to synthesis based on the domain theory, In *Engineering Design Synthesis - Understanding, Approaches and Tools*, Springer-Verlag, pp. 93-108.
15. Gero, J.S. (1990). Design prototypes: A knowledge representation schema for design. *AI magazine*, Vol. 11, No. 4, pp. 26-36.
16. Bhatta, S. and Goel, A. (2002) Design patterns and creative design, In *Engineering Design Synthesis - Understanding, Approaches and Tools*, Springer-Verlag, pp. 271-284.
17. Reich, Y. (1995) A critical review of general design theory. Research in Engg. Des., Vol. 7, No. 1, pp. 1-18.
18. Tomiyama, T., Kiriyama, T., Takeda, H. and Xue, D. (1989) Metamodel: A Key to Intelligent CAD Systems, *Research in Engg. Des.*, Vol. 1, No. 1, pp 19-34.
19. Smithers, T. (1998). Towards a knowledge level theory of design process, In AI in Design '98, pp. 3-21.
20. Hubka, V. (1967) Fundamental algorithm for the solution of design problems. In *International Scientific Colloquium of the Technical University Ilmenau*, pp. 69-74.
21. Maher, M L, Poon, J and Boulanger, S. (1996) Formalising design exploration as co-evolution: A combined gene approach. In *Advances in formal design methods for CAD*, Chapman and Hall, UK.
22. Cross, N. (2000) Engineering design methods – Strategies for product design, John Wiley & sons, pp. 38-40.
23. Pahl, G. and Beitz, W. (1996) Engineering Design: A systematic approach. Springer-Verlag, UK.