INTERNATIONAL DESIGN CONFERENCE - DESIGN 2008
Dubrovnik - Croatia, May 19 - 22, 2008.

DESIGN 2008

DESIGN FOR NOVELTY – A FRAMEWORK?

V.Srinivasan and A.Chakrabarti

Keywords: framework, novelty, activity, outcome, requirement, solution

1. Introduction

Designing is complex – it involves interactions among artefact, people, tools, process, organization and environment [Blessing et al., '95]. The aim of design research is to improve designing by understanding and supporting it better. A way to achieve these is by developing a model and framework for designing. A model of designing is taken here as an abstraction of designing that captures artefact, tools, process, organization and environment. A framework for designing is a prescription of how it should be carried out in order to improve its characteristics. Researchers stress the importance of novelty [Chakrabarti, '06; Howard et al., '07], and of considering physical laws and effects on novelty [Zavbi & Duhovnic, '97]. The research outlined here is to develop and empirically evaluate a model of designing that explicitly uses process and artefact elements via Activities, Outcomes and Requirements-Solutions, and based on the empirical findings, propose a framework for designing to encourage novelty.

2. Literature Survey

Relevant findings from literature are as follows: During *conceptual design*, solution concepts are developed to meet requirements of a design task. It is the most creative stage and most changes in a design are best effected and less expensive when worked at this level. A majority of a design's total life-cycle cost is committed during this phase [Chakrabarti et al., '02]. Being an early phase of design, it has many difficulties owing to open-endedness in finalising concept(s). However, relatively less attention has been paid into supporting this stage.

Activities are human problem solving phases in an engineering design process [Simon, '89; Blessing, '94]. *Outcome* of a design refers to the property of an artifact which can be at any level of abstraction, that is used to specify the artifact at that level of abstraction. Certain combination of activities and outcomes in a design are important as they seem to influence various aspects like requirements identification and satisfaction [Cooper, '91; Nidamarthi, '99].

Physical laws and effects are principles of nature that govern a change [Chakrabarti et al., '05]. Many researchers have advocated the importance of designing with laws and effects which can help produce novel and creative products [Chakrabarti et al., '97; Zavbi & Duhovnic, '97; Murakoshi & Taura, '98]. However, synthesizing artifacts directly from physical effects is hard since effects were created mainly for explanation of phenomena rather than for synthesizing artifacts that embody these phenomena and synthesis using them requires more than a straightforward application [Murakoshi & Taura, '98]. This may explain why, notwithstanding the advantages of using effects and laws in design, these have not been adequately represented in the current design models and frameworks.

Requirements express how an artefact should be at any level of abstraction. Many researchers advocate the need to consider requirements as a critical aspect in design, because it initiates a design task and its fulfillment serves as a criterion for design success [Cooper, '91; Pahl & Beitz, '96; Blessing '94]. In an engineering design process, satisfying design requirements is necessary not only to achieve customers' needs, but also to enable development of a design into a product. *Solutions* are means to satisfy requirements and can be at any level of abstraction.

From the above, we argue that there is a need for a model of designing that can accommodate all the above elements and explain typical traits of designing, and based on this model formulate a framework for designing that enables a usage of physical laws and effects as an integral part of designing.

3. Research Approach

The methodology adopted for this research is outlined below:

Step 1 (Development of Model): A model is developed by identifying significant activity, outcome, and requirement-solution-based elements of designing from models in the literature and integrating them.

Step 2 (Empirical Validation of the Model): To check empirical validity of the model, it was evaluated against protocols of design sessions where this model was not asked to be followed. Video recordings of design sessions from an earlier study were transcribed and coded. These sessions involved either team T1 or T2 (Table 1), each of three designers, who solved design problems P1 or P2 (Table 2), using three alternative methods, M1, M2, or M3 (Table 3), as per Table 4 (bold-numbers within square-brackets denote session number) under laboratory setting. The design problems involved developing conceptual solutions, and took 45 minutes each. The designers were instructed to discuss-aloud and provided papers for documentation. Each session was assisted by a researcher, to help clarify any queries. All the problems were solved back-to-back, and discussion across the teams was not allowed. Even though the same team solved the same problem twice using different methods, this is was not an issue for the objective of this research as we are interested only in the degree of generality in the pattern of problem solving across design sessions. We used protocols involving methods because rarely are design problems solved without methods, and a general framework should provide a basis for designing, with and without methods. The protocols were analyzed to identify and explain activities, outcomes, and requirements-solutions, as well as various typical composite aspects such as synthesis, analysis etc. A two-way mapping was used for the protocol studies to check if all: (i) instances in the protocol could be represented using the model elements, and (ii) elements in the model have instances in the protocol.

Step 3 (Proposal of a Framework): Based on the inadequacies in current designing from the empirical findings, a framework with activity, outcome and requirement-solution elements is proposed.

4. Results & Discussion

4.1. Development of Model

Development of Activity Model: There are a number of models from literature that are primarily activity-based. Based on protocol studies of five individual designers Stauffer & Ullman [1991] proposed a Generate-Evaluate-Decide model of a mechanical engineering design process. Ten different activities were grouped under three categories - generate, evaluate, and decide to describe the process. The protocol data was classified into the above activities, portions of which were selected at random, and analyzed to recognize patterns of design activity. Most of the patterns are described by four types of sequence: generate and test, generate and improve, means end analysis and deductive thinking. These patterns constitute *generate*, *evaluate* and *decide* activities in general. However, these studies focused on only portions of designing. Blessing [1994] used descriptive studies to propose a component-based prescriptive model-PROSUS (Process-based support system)-which consists of (a) Activities (*Generate, Evaluate, and Select*) and (b) Issues (problem statement, requirements, function structure, concept design, detail design, manufacturing etc.). Generate, evaluate, and select were proposed at each of the issue-levels. Based on protocol studies from a requirements-identification and satisfaction perspective, Nidamarthi [1999] identified two levels-problem understanding and problem solving. Each was characterised by primary-level activities - *identify*, *analyse*, and *choose*, and, *generate*, *evaluate* and *select*, in the respective levels. Each primary-activity was characterised by several secondary-level activities. In general, *identify*, *analyse*, and *choose* seems similar to *generate*, *evaluate* and *select*. Chakrabarti et al. [1997] proposed a model for the design of micro-sensors where design concepts are generated, behavioral problems with the concepts are identified and problems

resolved. This cycle seems analogous to *generating* concepts, *evaluating* these to identify behavioral problems, and *modifying* concepts to solve the problems. However, the model was not supported by any empirical observation.
Based on the above review, *generate, evaluate, modify and select* are generally involved in a design process. A **GEMS** (***G**enerate-**E**valuate-**M**odify-**S**elect*) model is taken as a model of activities.

Table 1: Designers' Background

Teams	Designers	Education		Nature
		Bachelors	Masters	
T1	D11	Mechanical	PDM	Novice
	D12	Mechanical	PDM	Novice
	D13	Mechanical	PDM	Experienced
T2	D21	Mechanical	PDM	Novice
	D22	Mechanical	PDM	Novice
	D23	Architecture	PDM	Novice

PDM: Product Design and Manufacturing

Table 2: Problem Goal

Problem	Goal
P1	To develop a conceptual solution for an efficient means of keeping the university campus free of dry leaves
P2	To develop a conceptual solution for a locking system that does not require any physical key or numbers to remember.

Table 3: Design Method

Method	Description
M1	Functional Analysis
M2	Ideal Design Approach
M3	Innovation Situation Questionnaire

Table 4: Protocol Design Sessions

	T1	T2
M1	P1 **[1]**	P2 **[2]**
M2	P2 **[3]**	P1 **[4]**
M3	P1 **[5]**	P2 **[6]**

Development of Outcome Model: Notwithstanding many existing models that use *outcomes* in various forms and levels of abstraction, a more desirable model should explicitly use *physical laws and effects* for designing, which is one way of achieving the primary aim of this paper - addressing novelty. The **SAPPhIRE** model of causality [Chakrabarti et al., '05] (Figure 1) is one such model. Apart from *Laws* and ***E**ffects*, the model uses ***A**ction*, ***S**tate change*, ***P**arts*, ***Ph**enomenon*, ***I**nput*, and *o**R**gan*- to provide a richer description of an artifact. However, the model was developed originally to explain the causality of natural and engineered systems, and has not been tested for its ability to act as a model of designing. Considering design to be a transformation from a function-level (F) to a strutucture-level (S) through a behaviour-level (B), we find that each of the constructs of the SAPPhIRE fall into one of these categories i.e. Function: Action, State Change, Input, Behavior: Physical Phenomenon, Physical Effect, Organ, and Structure: Part (Figure 2). Hence, the *SAPPhIRE* model is taken as a model of outcomes.

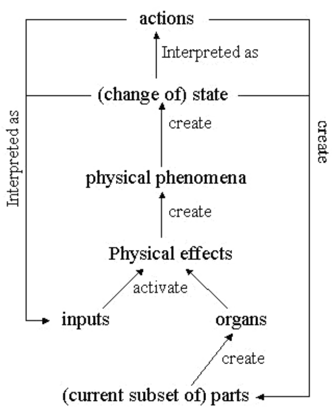

Figure 1: SAPPhIRE model of causality

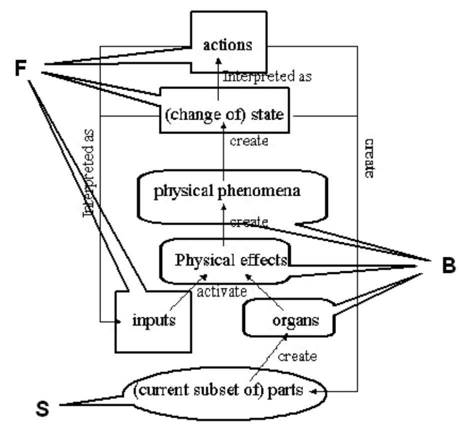

Figure 2: SAPPhIRE model with FBS-attributes

Development of Requirement-Solution Model: Pahl & Beitz [1996] propose a prescriptive model of design in which task clarification involves collecting information to prepare a list of *requirements* (specifications). *Solutions* are framed at different levels of abstraction based on the requirements; for instance, function structures and solution principles at the conceptual-level, preliminary layouts and form designs at the embodiment-level etc. The requirements are updated based on the solutions framed at different abstraction levels. A model of creative design, proposed by Maher et al. [1996], is based on a co-evolution of *requirement* (problem) and *solution* space, which happens throughout the design process along with an exchange of information between the two spaces. Nidamarthi et al. [1997] observed a co-evolution of *requirements* and *solutions* i.e., requirements influenced solutions in their generation, evaluation and selection and vice-versa. They identified two kinds of requirements: solution-specific and solution-neutral. They found instances of designers synthesizing solutions to satisfy given requirements and solution-specific requirements to realize the solution towards achieving the final design. Nidamarthi [1999] proposed a model of designing which divides it into two phases: problem understanding and problem solving. During problem understanding, *requirements* are identified, analysed and chosen and during problem solving, *solutions* are generated, evaluated, and selected. The above findings indicate there might be a *requirement-solution co-evolution* model of designing.

4.2. Empirical Validation

The following report significant findings from the protocol studies conducted in this work:

Individual Activity Findings: The protocol studies confirmed the presence of activities: generation, evaluation, selection, and modification in the design sessions. Table 5 shows the way the activities have been identified in the protocol with an example each. Table 6 shows the percentage frequency distribution of the individual activities in the six sessions, and helps derive the following findings:

- The instances of activities in all cases in descending order are: *generation, evaluation, selection and modification*, irrespective of any design method, problem or team. It would be logical to argue that the design outcomes are generated first but only some of them are evaluated.
- The percentage of *selection* and *modification* approximately sum together to equal the percentage of evaluation. Therefore, one could infer that evaluated outcomes are either selected or modified.
- The percentage of *modification* was observed to be less than *selection* in all the cases. This could be for two reasons: first, all the design sessions were timed for 30 minutes only and designers wanted to accept (hence, select) rather than modify, and second, design exercise required the designers to come up with an original design rather than a redesign.
- If *selection* was identified, then *rejection* would also be a part of the activity. However, we could not find instances of explicit *rejection* probably because of short time durations that designers could not afford to reject outcomes and maybe preferred to modify them.

The main conclusion from this section is that designing can be represented using four activities – generate, evaluate, modify and select.

Activity Patterns Findings: Table 7 shows the percentage frequency distribution of the activity-patterns observed for six sessions, and helps derive the following observations:

- The prominent patterns of activities in descending order are: *G, GES, GE* and *GEM*, across all design methods, problems and teams.
- It seems logical to conclude that activity patterns should culminate in selection. However, we observe many activity patterns to end in either evaluation or modification. This could be because these outcomes are either not considered, or implicitly evaluated and selected. Some patterns have generation only and the associated outcome is not considered later.
- Certain patterns have multiple evaluations, selections and modifications. It could be because the designers are working in a team and each member could have his/her own point of view leading to different criteria for evaluation. It could also point to the iterative nature of design.
- Activity patterns generally follow a sequence of generation and evaluation, followed by modification or selection. A modified outcome is evaluated until it gets selected or not considered further (Fig. 3).

Individual Outcome Findings: Table 8 shows instances of the outcomes from the protocol. Table 9 shows the percentage frequency distribution of the outcomes for the cases, from which we infer that:

- A high incidence of *action-level* description is noticed in all the cases since these were derived or taken directly from the problems given to designers.

- A low percentage of *state change-level* description was noted, possibly since it is another way of expressing action, and instances of it could have been included under action.
- A high-percentage of *part-level* descriptions was identified, probably because designers in general possess good part-knowledge.
- The percentage of *effect*- and *organ-level* descriptions was low. Reasons could be (i) effects and laws are not a part of natural (without model) way of designing, (ii) designers lacked effects/laws knowledge, (iii) designers did not know how to use them, (iv) problems did not require use of effects, or (v) methods did not specify use of effects. Similar results were also reported in a different study in [Sarkar & Chakrabarti, '07].

Table 5: Instances of Activity

Activity	Code	Protocol Instance
Generate	G	D1: So, what has to be achieved is that the campus has to kept free from dry leaves *(Generation: Campus to be kept free from dry leaves)* *[Episode: Designer defines the purpose of design exercise by generating a requirement]*
Evaluate	E	D1: Second is sweeping D1: Is sweeping okay? *(Evaluation: Checking the worth of sweeping)* *[Episode: Designer generates an idea for clearing-off dry leaves and estimates its worthiness]*
Modify	M	D1: Instead of manual sweeping, collections is a better term *(Modification: Change from 'manual sweeping' to 'collection')* *[Episode: designer generates a solution for clearing dry leaves (manual sweeping) and then feels collection maybe a more general term]*
Select	S	*D1:* Some secret code is required because each individual will have it differently D2: Yeah *(Selection: D2 accepting the solution proposed by D1)* *[Episode: First designer generates a solution to have a safe, private locking system which is accepted by the second designer]*

Table 6: % of activity in each session

Activity	[1]	[2]	[3]	[4]	[5]	[6]
G	47	56	46	46	55	49
E	29	26	28	29	23	28
M	6	4	3	4	3	2
S	18	14	23	21	19	21

Table 7: Percentage of activity-patterns in each session

Activity Pattern	[1]	[2]	[3]	[4]	[5]	[6]
G	53	58	52	47	61	50
GE	11	13	6	7	3	10
GES	19	17	32	37	31	33
GEM	6	7	1	3	4	2
GESE	1	1	1	2	0	0
GEMES	1	1	2	1	0	1
GESES	2	2	4	4	0	3
GESEM	1	0	0	0	0	1
GESESE	0	1	0	0	0	0
GEMEMES	0	0	1	0	0	0
GESESES	0	0	1	0	0	0
GESEMEM	0	0	0	0	0	1
GESEMES	6	0	0	0	0	0
GESEMESESESESEM	0	0	0	0	0	1

Outcome-Pattern Findings: Table 10 shows line-diagrams depicting patterns of outcomes observed from the protocol studies. The diagrams in the table show the relationship between the elements of the SAPPHIRE model as observed. The findings from the table are as follows:

(a) In all cases the designers started solving from an action-level description. The diagrams feature descriptions of higher-level abstractions to lower-level abstractions i.e., starting from action-level descriptions and ending up with part-level descriptions passing through one or more of the intermediate-levels of abstraction like phenomenon, state change etc. In most cases there is a direct jump from action- or phenomenon-level to part-level description. The transition from a higher-level abstraction to a lower-level abstraction confirms the *synthetic* nature of design.

(b) Contrary to (a), action-level and phenomena-level descriptions were also derived, respectively, from phenomena-level and part-level descriptions. This again seems to confirm the strong part knowledge of the designers as they seem to know the working and the function of the part. This transition from a lower-level abstraction to a higher-level abstraction confirms the *analytical* nature of design.

(c) In all the cases, design sessions culminated with a part-level description but not detailed to the extent of manufacturing. This is expected for a conceptual design problem.

Requirement-Solution Findings: Table 11 gives an instance of requirement and solution from the protocol. Requirements came from three sources - method, designer, and problem (Table 12). Table 13 shows the percentage distribution of frequency of the types of requirements, from which we infer that:

- In most cases, a high percentage of requirements from method was observed. It probably points out that designers are driven by the instructions in the method.
- Only a small percentage of requirements came from the problem in most cases. It could have been that the information in the problem was not enough for the designers, which had motivated them to look into the method and their own knowledge for guidance.
- A large percentage of requirements came from the designers in most of the cases.

Table 14 shows the percentage frequency distribution of requirements and solutions. Four kinds of relationships between requirements and solutions were observed: (a) requirement-requirement: represents requirements clarification, (b) requirement-solution: represent cases where a solution is derived to satisfy a requirement, (c) solution-requirement: depicts a situation where a requirement was derived from a solution i.e., a previously derived solution is used as a requirement in the current situation, and (d) solution-solution: resembles a solution clarification.

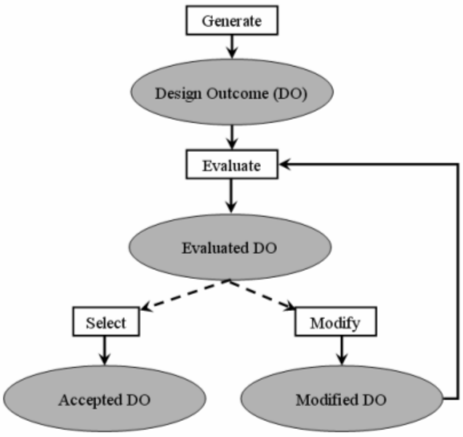

Figure 3: Activity model of design

Table 9: Percentage of outcomes in each session

Outcome	[1]	[2]	[3]	[4]	[5]	[6]
s	0	10	0	0	1	0
a(i)	33	21	46	44	17	16
p	14	21	37	36	40	50
ph	45	37	15	20	37	28
r	2	9	2	0	5	5
e	6	2	0	0	0	1

Table 8: Instance of outcomes

Outcome	Code	Instance
Action *(interpreted as input-output description)*	a (i)	D1:So, what has to be achieved is that the campus has to be kept free from dry leaves *(Action: Interpreted as dry leaves → no dry leaves)* *[Episode: Designer states the requirement to be fulfilled i.e., keep the campus free from dry leaves and at a higher level of abstraction, closer to the problem]*
State change	s	D1:System's primary useful function is that it should lock when it is required to lock and open when it is required to open *(State change: unlock → lock or lock →unlock)* *[Episode: Designer stating the purpose of the lock, which has to be designed and is closer to a solution]*
Part	p	D1: transportation can be done by carrying the bins and baskets manually, small trucks or tractors *(Part: Bins, baskets, small trucks, tractors)* *[Episode: Designer generates ideas for transporting dry leaves from one place to another]*
Phenomenon	ph	D1: So, the functions that the system will take care of are: cleaning, loading, transportation, unloading and disposal of dry leaves. *(Ph: cleaning, loading, transportation, unloading, and disposal)* *[Episode: Designer generates the process of keeping the campus free from dry leaves]*
Organ	r	D1: So input is gravity, self-weight, weak link *(Organ: Weak link)* *[Episode: designers reasoning the factors responsible for the fall of a leaf]*
Effect	e	D1: Because, of the force of gravity-gravitational force (D2 writes on the paper) *(Effect: Newton's law of gravitational force)* *[Episode: Designer explains the cause of fall of a leaf]*

Table 10: Outcome patterns in each session

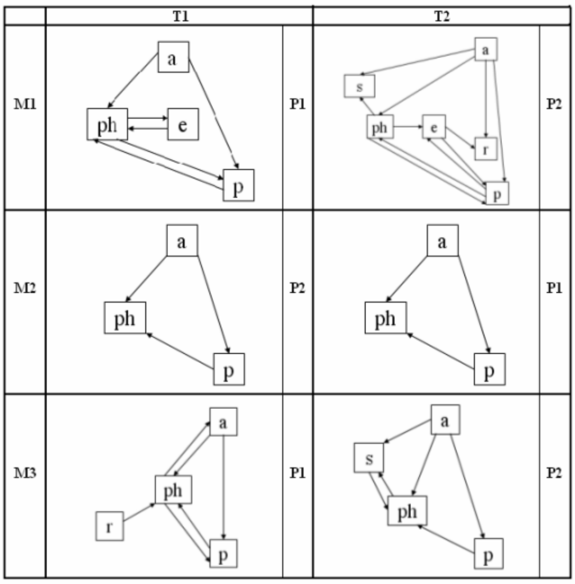

Table 11: Instance of requirement and solution

Content	Code	Instance
requirement	re	D: So, our main focus will be here (points at the paper and shades a part in the sketch), wherever people want to access, those places need to be kept clean. *[Designer creates a requirement by identifying the areas to be cleaned]*
solution	so	D': This is a net-arrangement (pointing at the sketch) and this is a vacuum cleaner (pointing at the sketch) *[Designer generates a solution for collecting dry leaves and storing in a net-arrangement]*

Table 12: Instance of requirement from method/designer/problem

Source	Code	Instance
Method	re(met)	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) *[Designer reads out the requirement specified by method (ISQ) from the instruction-sheet i.e., to generate evaluation criteria-technological, economy etc.]*
Designer	re(des)	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, but, however, it can be battery-powered]*
Problem	re(pro)	D': So, what has to be achieved is that the campus has to kept free from dry leaves. *[Designer spells out the objective of the design exercise i.e. to keep university campus free from dry leaves taken directly from the problem given]*

Combined - Activity and Outcome Findings: Tables 15-16 report the percentage frequency of combined activity and outcome findings from the six sessions and helps derive the following observations:

- Action, part and phenomenon had several instances of generation, evaluation, modification and selection. These also had activity-patterns involving multiple evaluation, modification and selection.
- On the contrary, organ and effect had very few instances of individual activities. No modification was observed for effects. These outcomes did not have activity patterns involving multiple evaluation, selection and modification. These findings could again point to the difficulty that designers face when working with effects and organs.
- Even state change, did not have any modifications and activity patterns involving multiple evaluation, selection and modification. It could be because some state change-level descriptions could have been included under action because of little differences between the two outcomes.

Table 13: % distribution of re for each session

	[1]	[2]	[3]	[4]	[5]	[6]
re(met)	64.71	11.76	46.48	24.24	67.92	61.90
re(des)	20.59	73.53	12.68	72.73	30.19	36.51
re(pro)	14.71	14.71	40.85	3.03	1.89	1.59

Table 14: % distribution of re & so for each session

	[1]	[2]	[3]	[4]	[5]	[6]
re	14.85	28.10	31.84	31.43	18.93	28.44
so	85.15	71.90	68.16	68.57	81.07	71.56

Table 15: % distribution of outcomes for each activity

	a(i)	p	ph	r	e	s
G	26	33	33	4	3	15
E	25	35	36	2	1	3
M	25	27	42	6	0	0
S	25	37	34	2	1	3

Table 16: % distribution of outcomes for each activity-pattern

	a(i)	p	ph	r	e	s
G	26	31	33	5	2	2
GE	30	29	39	3	0	0
GES	27	39	30	2	2	1
GEM	34	23	31	11	0	0
GESE	0	50	33	0	17	0
GEMES	14	71	14	0	0	0
GESES	31	38	31	0	0	0
GESEM	20	40	40	0	0	0
GESESE	0	50	50	0	0	0
GEMEMES	100	0	0	0	0	0
GESESES	100	0	0	0	0	0
GESEMEM	0	0	100	0	0	0
GESEMES	0	0	100	0	0	0
GESEMESESESESEM	0	100	0	0	0	0

Combined – Requirement-Solution and Activity Findings: Tables 17 & 18 show the percentage frequency distribution of activities, for requirements and solutions, respectively, for all sessions. The findings and discussion are quite similar to that seen in *Individual Activity Findings*.

Combined – Requirement-Solution and Outcome Findings: Tables 19 & 20 show the percentage frequency distribution of outcomes, for requirements and solutions, respectively, for all sessions. The findings and discussion are quite similar to that seen in *Individual Outcome Findings*.

4.3. Proposal of Framework

The empirical results reveal that designers are not equally proficient with the activities at all the levels of SAPPhIRE for both requirements and solutions. Very few effect- and organ-level descriptions are found when a higher number is expected. While a single-many mapping is expected as one moves from higher to lower levels of abstraction, this is considerably lower at the effect- and organ-level descriptions. Since novelty is critical in design, the use of more laws and effects in designing should be encouraged. Hence, we propose a framework: GEMS of SAPPhIRE as requirements-solutions, by suggesting GEMS to be carried out to a substantial degree at all the levels of SAPPhIRE, for both requirements and solutions.

5. Summary & Future Work

This section summarises important findings from this paper and draws future plans for this research:

- A model that integrates the activity-, outcome-, and requirement-solution based elements has been developed and validated to prove designing can be represented with the above elements.
- There exists a need to support designers with knowledge of physical laws and effects to encourage designing novel artefacts. However, the uniqueness of the framework as a support for 'Design for Novelty' still needs to be evaluated i.e, to check if use of laws and effects promotes novelty.
- The framework is currently limited to supporting only conceptual and early-embodiment design and needs to be extended to support the other phases of design, which is planned for future.

Table 17: % distribution of activity for re in each session

	[1]	[2]	[3]	[4]	[5]	[6]
G	57.63	53.97	53.52	44.59	54.95	62.00
E	25.42	25.40	23.94	28.38	26.37	22.00
M	3.39	4.76	2.82	2.70	2.20	0.00
S	13.56	15.87	19.72	24.32	16.48	16.00

Table 18: % distribution of activity for so in each session

	[1]	[2]	[3]	[4]	[5]	[6]
G	56.20	53.37	54.17	51.06	55.77	48.45
E	24.21	28.22	25.00	26.24	23.59	28.57
M	4.90	3.68	3.33	2.84	2.70	2.80
S	14.70	14.72	17.50	19.86	17.94	20.19

Table 19: % distribution of outcomes for re in each session

	[1]	[2]	[3]	[4]	[5]	[6]
s	0.00	2.94	0.00	3.57	0.00	0.00
a(i)	64.29	38.24	29.17	28.57	50.00	33.33
p	21.43	20.59	58.33	53.57	42.86	56.67
ph	14.29	14.71	12.50	10.71	7.14	10.00
r	0.00	23.53	0.00	3.57	0.00	0.00
e	0.00	0.00	0.00	0.00	0.00	0.00

Table 20: % distribution of outcomes for so in each session

	[1]	[2]	[3]	[4]	[5]	[6]
s	0.00	9.76	0.00	0.00	1.75	3.18
a(i)	13.64	2.44	41.56	47.22	9.21	13.38
p	19.19	23.17	29.87	31.94	42.98	49.68
ph	59.60	54.88	25.97	19.44	42.11	29.30
r	3.03	6.10	2.60	1.39	3.95	4.46
e	4.55	3.66	0.00	0.00	0.00	0.00

References

Blessing, L. "A Process-Based Approach to Computer-Supported Engineering Design", PhD Thesis, University of Twente, The Netherlands, 1994.

Blessing, L., Chakrabarti, A., Wallace, K., "A Design Research Methodology", ICED95, Prague, 1995.

Chakrabarti, A., Johnson, A., Kiriyama, T., "An Approach to Automated Synthesis of Solution Principles for Microsensor Designs", ICED97, Tampere,1997.

Chakrabarti, A., Langdon, P., Liu, Y., Bligh, T., "An Approach to Compositional Synthesis of Mechanical Design Concepts Using Computers", Engineering Design Synthesis-Understanding, Approaches and Tools, A. Chakrabarti (ed), Springer-Verlag London, 2002, pp. 179-197.

Chakrabarti, A., Sarkar, P., Leelavathamma, B., Nataraju, B.S., "A Functional Representation for Aiding Biomimetic and Artificial Inspiration of New Ideas", AI EDAM, Vol. 19, 2005, pp. 113-132.

Chakrabarti, A., "Defining and Supporting Design Creativity", Design 2006, Dubrovnik, 2006.

Cooper, R.G., "New Products: What Distinguishes the Winners?", Research Technology Management, November-December, 1991, pp. 27-31.

Howard, T., Culley, S., Dekoninck, E., "Creativity in the Engineering Design Process", ICED, Paris, 2007.

Maher, M.L., Poon, J., Boulanger, S., "Formalising Design Exploration as Co-evolution: A Combined Gene Approach', in JS Gero and F Sudweeks (eds) Advances in formal design methods for CAD, Chapman and Hall, London, 1996.

Murakoshi, S., Taura, T., "Research on the Systematization of Natural Laws for Design Support", 3rd IFIP Workshop on Knowledge Intensive CAD, Tokyo, 1998.

Nidamarthi, S., "Understanding and Supporting Requirement Satisfaction in the Design Process", PhD Thesis, Department of Engineering, University of Cambridge,, 1999.

Pahl, G., Beitz, W., "Engineering Design: A SystematicApproach", Spriger-Verlag, London,UK, 1996.

Nidamarthi, S, Chakrabarti, A and Bligh, T, "The Significance of Co-Evolving Requirements and Solutions in the Design process", ICED, Tampere, 1997.

Sarkar, P., Chakrabarti, A., "Understanding Search in Design" ICED07, Paris, 2007.

Simon, H., "The Sciences of the Artificial", MIT Press, Massachusetts, USA, 1989.

Stauffer, L., Ullman, D., "Fundamental Processes of Mechanical Designers Based on Empirical Data", Journal of Engineering Design, Vol. 2, No. 2, 1991, pp. 113 – 125.

Zavbi, R., and Duhovnik, J., "Prescriptive Model with Explicit Use of Physical Laws", ICED, Tampere, 1997.

Dr. Amaresh Chakrabarti
Associate Professor
Indian Institute of Science, Centre for Product Design and Manufacturing
IDeaS Lab, CPDM, IISc, Bangalore, India
Telephone: +91-80-2293 3136/2293 2922
Telefax: +91-80-2360 1975
E-mail: ac123@cpdm.iisc.ernet.in
URL: http://cpdm.iisc.ernet.in/people/ac/ac.htm