INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 05 MELBOURNE, AUGUST 15 – 18, 2005

A BEHAVIOURAL MODEL FOR REPRESENTING BIOLOGICAL AND ARTIFICIAL SYSTEMS FOR INSPIRING NOVEL DESIGNS

Amaresh Chakrabarti, Prabir Sarkar, B. Leelavathamma, B.S. Nataraju

Abstract

Inspiration is useful in problem solving. Our aim is to develop an integrated approach for systematic idea generation in product design using inspiration from natural as well as artificial systems. A new, generic model has been developed for representing causality of how functionality in natural and artificial systems is achieved; this is implemented in a piece of software for automated, analogical search of relevant ideas from large databases of natural and artificial systems, annotated with the constructs of the model, to solve a given problem. Two search strategies, consisting of three different search types have been formulated to facilitate designers with the necessary flexibility to search the databases, as per requirements of a given problem. In house design experiments, conducted for evaluating effectiveness of the support, showed a clear indication that the software enhances designers' ability to generate a large number of solutions, even after their own ideas are exhausted.

Keywords: Bio-mimicry, design problem solving, idea generation, inspiration, behavioural language, design by analogy

1 Introduction

According to Larry et al. [1], inspiration is a factor that is brought on by accidental stimuli. Knowledge and experience are often available but one needs some new elements that trigger a new combination. Inspiration can be received from people that we come in contact with, by our exposure to new ideas that are parallel to our own, and by contact with ideas that are in contrast to our proposed solution of a problem, or by some adjoining thought. Inspiration is useful for exploration and discovery of new solution spaces [2]. Several pieces of literature indicate that presence of a stimulus can lead to more ideas being generated during problem solving [3]. Various researchers have validated this claim by conducting empirical studies and by developing stimulus rich creativity techniques- the results are optimistic [4, 5].

Natural world, evolving through ages, has produced a large variety of systems for variety of purposes; artifacts and systems developed my humans add to this variety. Both these systems are rich in knowledge that could be useful for solving various design problems. One of our aims is to tap and channel this knowledge to present day designers in order to help them solve new design problems.

Literature provides several accounts of research [6,7] with the goal of learning from nature with product development in mind. However, unlike in the artificial domain, where existing systems are routinely used for inspiration (such as in compendia, case based reasoning systems etc), those from the natural domain are rarely used in a systematic way for this purpose.

Analogy has long been regarded as a powerful means for inspiring novel idea generation, as seen in several systems based on analogy [8] and creativity methods developed with the specific aim of fostering analogical reasoning [9]. The objective of this work is to develop a computational tool for supporting designers to generate novel solutions to product design problems by providing naturally or artificially-inspired analogical ideas. The current focus of application is on developing novel mechanisms.

2 Objectives

The particular intention is to use the corpus of the diverse motions that natural and artificial systems exhibit as a potent source of inspiration for solving product design problems, especially in inspiring creativity and innovation of novel products. The emphasis is not on mimicking natural phenomena, but rather getting inspired from primarily the behavioural aspects of natural phenomena. The task was to first develop two databases: a database of natural systems (e.g., insects, plants etc.) exhibiting diverse movements, and a database of simple to relatively complex artificial mechanical systems that are capable of providing various behaviours (e.g. vacuum cleaners, clutches etc.). The second task was to analyse these natural and artificial systems, and develop a language for describing their motion behaviours. A behavioural language has a set of constructs that is used to represent the functioning of an artificial or a natural system. The third task was to develop a piece of software with appropriate interface and reasoning procedures to aid in the following process. Designers, with a problem to solve, would explore motions of various types (from the natural or artificial systems in the databases), and use the behavioural representation developed to describe the problem in terms of the constructs of the representation – the software would then search the databases for entries that could be used to solve the problem. There are three major steps involved:

- Create databases of natural and artificial systems (discussed in section 3).
- Develop a common, behavioural language for representing these systems and their functionality (discussed in section 4).
- Develop procedures for interactive, analogical generation of alternative ideas for solving a given design problem (discussed in section 5).

3 Creation of databases

Two databases (one with natural systems, the other with artificial systems, as entries) have been developed, the details of which are given in Table 1. The information collected contains the following:

1. Details about the function, behaviour and structure of these systems in text format.
2. Diagrammatic, pictorial, video or animation data about the systems and their motions.
3. A summery of the analysed motion.
4. Action, input, physical phenomenon, organ, physical effects, parts and change of state of each motion described.

An example of an entry is given in Figure 1.

Table 1: Details of the two databases developed

DATABASE 1: Natural/Biological systems (contains about 210 entries)		
Arranged according to		*Examples*
	Sub division	
Motion types	Aerial	Bat, Bee, Butterfly
	Aquatic	Dolphin, Catfish, Crocodile
	Climbing	Bushy baby, Caterpillar, Climbing plant
	Feeding	Barnacles, sawfish
	Special	Blue bells, Tumble weeds
	Terrestrial	Camel walking, Fern, Fig
	Underground	Clam mole rat
Organisms	-	Antelope, Baboon, Caterpillar
Search by index	-	Antelope walking, Bird wing, Blue whale swimming
DATABASE 2: Artificial systems (contains about 280 entries)		
Arranged according to		*Examples*
	Sub division	
Basic components		Cam, Follower, Gear, Pulley
Kinematics pairs	Lower pairs	Revolute, Prismatic
	Higher pairs	Cam, Follower, Helical gear
	Wrapping pairs	Belt, Rope
Actuators	-	Electromagnets, Hydraulic pumps
Devices	-	Transport mechanism, Cooling mechanism
Search by index	-	4- bar slider, adjustable cone drive

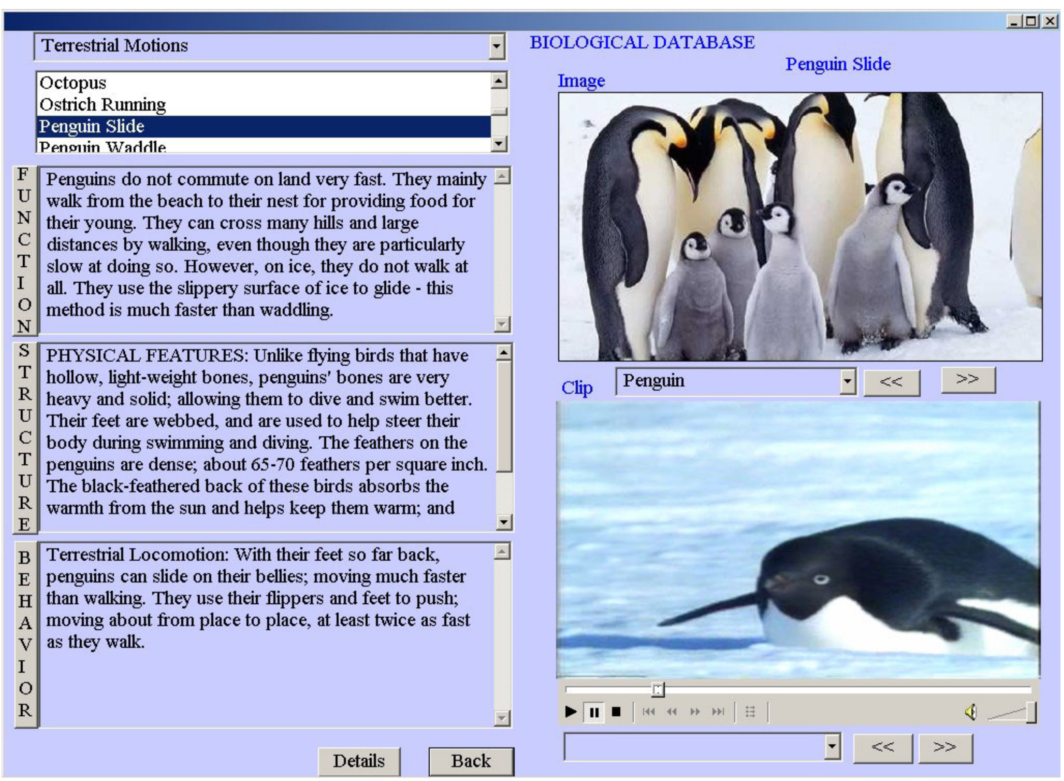

Figure 1 Screen capture of an entry from the database of natural systems

4 Describing natural and artificial systems and their functionality

The main challenge for the behavioural description has been that it must allow the function, behaviour and structure of a system to be linked to each other in a way common for both natural and artificial systems, and allow describing these at various levels of abstraction. At the centre of this work is the development of a uniform functional/behavioural representation for these systems. In literature, several models of functional reasoning are proposed, including a number of function-behaviour-structure models (FBS or SBF models) ([8, 10-15]. There exist many views of function, behaviour and structure [16-21].

We view function as the intended effect of a system [22], and behaviour as the link between function and structure defined at a given level; therefore, what is behaviour is specific to the levels at which the function and structure of a system are defined. We believe that structure - in a richer behavioural representation - must also have the flexibility of being represented using multiple views.

From the existence and usefulness of multiple views of function, structure and behaviour, we feel that these views are but specific and limited aspects of a richer underlying causal description of behaviour. What is problematic in many FBS models is the implicit equation of this rich causal description of the behaviour of a system with a fixed, predefined division of this knowledge into function, behaviour and structure. With the exception of Andreasen's work in which behaviour is viewed at multiple levels (although not integrated into a common causal description), others seem to implicitly assume this overall causal description to be synonymous to a specific FBS model. We see this causal description as separate from an FBS model, and view a given FBS model as a way of viewing particular portions of this description at particular levels of abstraction.

The three pieces of work that influence our work most are the 'Theory of Technical Systems' of Hubka [20], 'Domain Theory' of Andreasen [21], the 'FBS' work Umeda et al. [12], and the 'Metamodel' work of Yoshioka & Tomiyama [23]. Hubka used a four level representation for a product: a process level, a function level, an organ level and a part level. Andreasen modified this into a three level representation in his domain theory: transformation, organ and structure. Umeda et al. developed a function-behaviour-state model in which they use function as the top level, and define behaviour using what they call physical phenomena (e.g, electricity flowing) and structural entities and relationships (e.g., conductors connected via wires). While many of the building blocks used by these authors are essential for developing a rich, causal description of the functioning of an artifact in a given environment, these are not sufficient for producing such a description.

For instance, not all existing views of function, structure and behaviour are represented in any of the above models. In particular, the distinction between the various semantically distinct representations of function (e.g., I/O and state change) remains unclear in the transformation level representation. Structural description is somewhat limited in Tomiyama's work, although it is richer in those of Hubka and Andreasen. Behavioural representation is minimal in Andreasen and Hubka, while it is richer in the work of Yoshioka & Tomiyama and Umeda et al. Organs are rather narrowly defined in Andreasen's work, not taking into account the characteristics of the context or environment in their definition, while they are not used in the FBS model of Umeda et al. What is also missing in these descriptions is the explicit use of physical effects and the link to state changes effected by virtue of the effects. As a result, these representations individually are less than adequate for providing a rich, behaviour level explanation of an artifact or a natural system's functioning.

The concept of organ was first used by Hubka [20], and there are several other views of organs having different perspectives, e.g., Yoshioka & Tomiyama [23]; Chakrabarti & Regno [24]. We define organs as structural prerequisites for physical effects created by assembling the structural characteristics of and relationships between the system and its context/environment in order to activate an effect or effects, so as to make a change of state (which may be no change of state after all) possible. To summarise, many FBS models define *a priori* what the functional level of device is etc. This is arbitrary and counter-intuitive. We need to dissociate the underlying causal description of the functioning of a system from the subjective and partial FBS views on this description.

We therefore need a richer, encompassing causal description of the functioning of a system. This should allow multiple FBS views to be taken on the description, in terms of multiple levels of granularity of structure and behaviour and multiple aspects of the causality as intent or function. This should also allow multiple, existing views of function and behaviour to be embedded and discernible within the description. Structure should be represented at different levels and should include properties of the artifact or system as well as relevant properties of its context or environment.

Our experience shows that what is considered input for one system is a state change for another. In other words, input and state change are views created by system boundaries. This interpretational aspect must be made explicit. From the above literature study and also after analyzing several natural and artificial entries we have found that the following seven constructs are required to depict the motion behavior of any entry completely. Seven elementary constructs are used:

1. *Parts:* A set of physical components and interfaces constituting the system and its environment of interaction.
2. *Organ:* The structural context necessary for a physical effect to be activated.
3. *Input:* The energy, information or material requirements for a physical effect to be activated; interpretation of energy/material parameters of a change of state in the context of an organ.
4. *Physical effect:* The laws of nature governing change.
5. *Physical phenomenon:* A set of potential changes associated with a given physical effect for a given organ and inputs
6. *State:* The attributes and values of attributes that define the properties of a given system at a given instant of time during its operation.
7. *Action:* An abstract description or high level interpretation of a change of state, a changed state, or creation of an input.

The relationships between these constructs are as follows: *parts* are necessary for <u>creating</u> *organs*. Organs and *inputs* are necessary for <u>activation</u> of *physical effects*. Activation of physical effects is necessary for <u>creating</u> *physical phenomena* and changes of *state*, changes of state are <u>interpreted</u> as *actions* or inputs, and <u>create or activate</u> parts. Essentially, there are three relationships: *activation, creation and interpretation*.

The causal description language is acronymed as the SAPPhIRE model, SAPPhIRE standing for State-Action-Part-Phenomenon-Input-oRgan-Effect, see Figure 2. Using the constructs and relationships of this model, behaviour of each system is described in the databases developed.

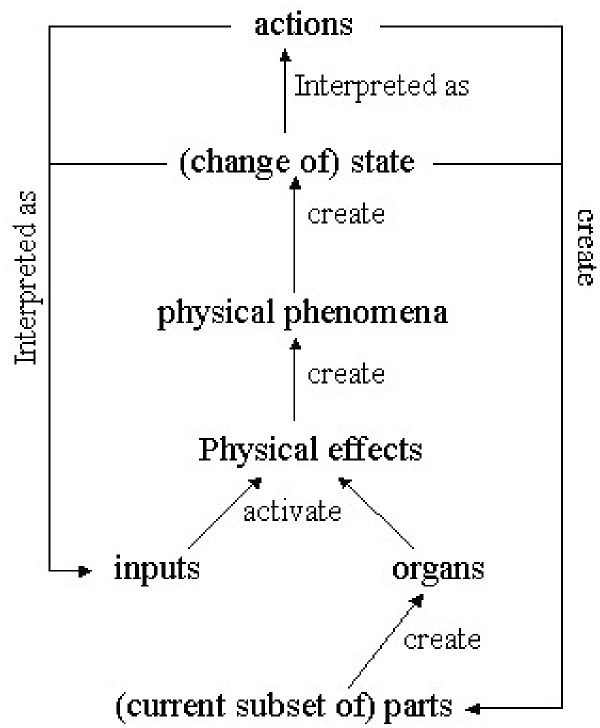

Figure 2. The SAPPhIRE model of causality

5 Implementation

The SAPPhIRE model of causality has been used to represent the entries in the databases, and reasoning procedures are developed to help browse and search for entries that are (analogically) relevant for solving a design problem. The concepts are implemented in software called IDEA-INSPIRE [25]. Designers can either 'browse' the database for inspiration or can directly go for 'solving problem', in case they have a particular design problem in mind.

5.1 Representation of Entries

Each entry in the databases is described, using the constructs of the SAPPhIRE model of causality (see Figure 2), in two forms:

1. A computer understandable form - this are text files containing a set of words, used to describe the entry and formatted in a particular way that can be read by the program easily and can be used for efficient searching.
2. A human understandable form - these are text files containing paragraphs, explaining the motion behaviour of the entry in sentence form.

In the computer understandable form, the content of each entry is divided into a list of actions, state, physical phenomena, inputs, physical effects, organs and parts; links between various subsets of these together constitute the entry. Details are as mentioned below:

Actions are described using a list of verbs, nouns and adjectives/ adverbs. For instance, the action of feeding is described using ‘feed’ as verb and with no specific noun or adjective as qualifiers:

ACTION: { A1 $ V < feed > A < > N < > $ }

A state change is also described using a phrase. In this case, one such state change is ‘from rest to reciprocating motion’:

STATE: { SS1 $ From rest to reciprocating motion $ }

Physical phenomena are also expressed in terms of verbs, nouns and adjectives. For instance, the physical phenomenon of ‘production of a chemical’ is described using verb ‘produce’ and noun ‘chemical’ with no specific adjectives:

PHYPHENOMENON: { PP1 $ V < produce > A < > N < chemical > $ }

Physical effects are described using the name of the effect, e.g., ‘Stimulus-Response effect’:

PHYEFFECT: { PE1 $ Stimulus-Response effect $ }

An input is represented using a verb, noun and/or an adjective. For instance, an input ‘electrical signal’ is described using no verb, the noun ‘signal’ and the adjective ‘electrical’:

INPUT: { I1 $ V < > A < electrical > N < signal > $ }

An organ is currently described with the help of a phrase that describes the organ. In the example case, one such organ is ‘the ability of the scent gland of Venus Flytrap to produce appropriate chemicals for scent emission’:

ORGAN: { O1 $ The ability of the scent gland to produce appropriate chemicals that emit the scent $ }

Parts are defined described with the help of a phrase that describes the part. In the example case, one such parts is ‘belt pair’:

PARTS: { P1$ Belt pairs $}

The links between these individual fragments of knowledge are represented using ordered lists. For instance, the above knowledge fragments are linked together by linking the action with physical phenomenon with the physical effect with the input with the state change with the organ with the part, and is represented as the following link:

LINK: A1 – SS1 – PP1 – PE1 – I1 – O1 – P1. (A1=action 1, SS1=state change 1, PP1=physical phenomenon 1, PE1= physical effect 1, I1=input 1, O1= organ 1, P1= part 1).

In the human understandable form of the entries, these links are expanded in paragraphs.

5.2 Reasoning

The goal of the search strategies is to support analogical reasoning at multiple levels of abstraction. Using these strategies, a designer should be able to either browse the entries for random stimulation, or systematically search through them with specific purposes. For searching for solutions a designer can give one or many constructs (action, physical principle etc.), or any combinations of them to the software as inputs. Presently the software can take three demands and three wishes. Each of these demands or wishes can be depicted by any of the seven constructs. Each of these constructs has a particular way of expressing - verb, noun and adjective or phrases (see Section 5.1 for details). A designer can go on selecting many combinations of these constructs in order to capture all aspects of the problem.

For instance, when a designer gives as a problem an action described using a verb, noun, and adjective (VNA), the program takes these as 'input' variables and search the computer understandable form of the entries for these variables. In each entry, actions are represented by verbs, nouns and adjectives and these are matched with the selected variables. If a direct match with the variables were not possible, it would search for synonyms of each variable in the entries and give a corresponding weightage. These matched entries are sorted in a descending order of importance. The potential for an entry to have various degrees of matching with the input, along with its potential for inspiring new solutions, gives rise to solutions of the following three types:

- Exact solutions: When all the constructs in an entry match with that of the given inputs, and a designer accepts this as it is as a potential solution to the problem.
- Partial solutions: When some of constructs in an entry match the given inputs, and a designer accepts this as it is as a potential partial solution to the problem.
- Inspirational solutions: When an entry with an exact or partial match with the given inputs triggers a designer to generate a new solution.

There are two strategies of analogical reasoning that are employed in the search process where a specific problem is defined by the designer

- Specification translation (this includes search types of simple and combination search)
- Jumping between the various behavioural levels (this use the complex search type).

These are explained in the following two sub-sections.

5.2.1 Translation of design input into analogical descriptions

The description of the mechanism looked for, its intended behaviour or constraints can be provided by a designer in terms of Verbs, Nouns and Adjectives. In order to implement the 'translation' of this input into analogical descriptions, clusters of equivalent words (synonyms) have been developed. Clustering of words is carried out beforehand for nouns, verbs and adjectives and stored in a database for use during translation. A *verb cluster* is divided into *generic verbs* and *specific verbs*.

When a designer gives the required input to the software by formulating the problem in terms of a combination of constructs, the software tries to find the best match of the entries in the descending order of importance, annotating each entry with an weightage, which shows how close the solution is to the desired solution.

5.2.2 Searching entries having similar characteristics

Since each entry is a linked network of actions, organs, inputs, effects, phenomena and parts, analogical solutions can be reached if it is possible to search for, say, entries that share the same principles in the entries that fulfil the required actions, or entries that have analogical parts to those that have the required intended action. For search of these kinds, one needs to be able to construct complex search queries with multiple, intermediate and final search points of specified types and with specified input types and values.

This can be achieved by a multiple search with the following form:

For a given input type, find all output of given type for all intermediate outputs of given types. For instance, one such query is: for a given action (input), find all entries (output) that provide actions (intermediate output type) that use the same effects (intermediate output type) as are used by the entries that provide the input action:

Given action -> effects used -> actions -> entries.

Such complex search problems are likely to help 'discover' solutions that are more difficult to immediately associate with the design problem (e.g., the input action) at hand and yet are analogically relevant as potential ideas for solving the problem.

5.3 Example

We have developed three different search types- simple, combination and complex- to aid designers to search in various ways the databases and the entries previously selected. Let us take an example to explain this. Let a designer is interested in solving this given problem: *Design an aid that can enable people with disabled upper limbs to eat food.*

For this problem, the designer can describe the problem as: put solid food in the mouth slowly, then, his search becomes a **simple search** like,

a. Action: V= feed, N = solid, A= slow or,

b. Action: V= consume, N = solid, A= slow

For case a., the software will find the set of entries having the closest required combinations of the action, some of the entries with their corresponding weightages are shown below:

Entry name: Dodder, weightage =57.1;Entry name: Aardvark, weightage =57.1;Entry name: Crawler, weightage =7.14, etc.

Now, one cansee a video display of the motion of Dodders, or read the text provided in the Dodder entry. The physical phenomena related to Dodders is described as follows. *Tendrils of the dodder wrap the host plant. Suckers penetrate the sap area of host plant. Sap of the host plant is absorbed by the suckers.*

From this entry one can get an idea of a machine, which will suck the food (in liquid or semi solid form), and then pump it to the mouth. This is an example of a *partial solution*, since the designer needs to modify the entry to develop the idea into a solution. Similarly, sucking liquid food by a straw can be another potential solution; this would be an example of an *inspirational solution* if the designer conceived this with the Dodder idea as a trigger. Using the crawler entry, a designer could conceive design of a simple motorised system which employs a crawler to take solid food from the plate to the mouth- this is an example of an *exact solution*, since the designer can directly use the solution as shown in the entry to solve the current problem.

Alternatively, a designer may think of designing a device that will first take the food in a container, move close to the mouth, and transfer it to the mouth. Then the problem needs to be formulated using a **combination search** like,

a. Action: (V= hold, N = solid, A =quick) + (V= move, N =solid, A= slow) + (V= push N=solid, A=slow or,

b. Action: (V= get, N = solid, A =slow) + (V=swallow, N =solid, A= slow).

For case b, the entries found by the software includes the following with their weightages:

Entry name: Alpine Nut cutter, weightage =32.1; Entry name: Bee Orchid, weightage =7.14;

Once a simple or a combinational search is over, a designer is offered a list of entries with corresponding weightages. Taking these entries as inputs, the designer can initiate another search within these entries, or search remaining entries having similar characteristics as those for any particular construct of a set of selected entries - this latter is called **complex search**.

For example, if a designer takes the three entries from the simple search (Dodder, Aardvark, Crawler) and searches the entire database for entries that have the same physical phenomena as used in these entries, the designer will receive other new entries, e.g., honey bee, flipping mechanism, etc. Complex search can be initiated with various sets of initial entries and for various combinations of constructs whose values are to be used for matching.

6 Evaluation

Evaluation of the software has been carried out in two phases. In the fist phase three designers were requested to solve individually two problems of their choice from a pool of problems given, first without using the IDEA-INSPIRE software and subsequently by using the software. In the second phase three designers (different from those in the first phase) individually solved a single design problem provided by the funding agency ISRO (see acknowledgement for details of ISRO), in the same format (first without, and then with IDEA-INSPIRE). Some solutions, which the designers have generated with, and without the help of the software for the given problem (designing a solar array deployment mechanism), have been shown in Figure 3 and 4. All six designers have an undergraduate degree in engineering and have formal product design training, and four of these designers (two in the first phase and the other two in the second phase) have more than two years of professional experience in design, in a design firm. The results are summarized in Tables 2 and 3.

The idea was to see if the intervention made a substantial difference in the number and kind of solutions generated. The number of inspiring ideas, triggered by the entries from the software, and the entries (in the database) that can be used directly as a solution, were noted down by the designers. We found that by using the software, each designer received/ created additional ideas for solving each problem than (s)he did without using the software.

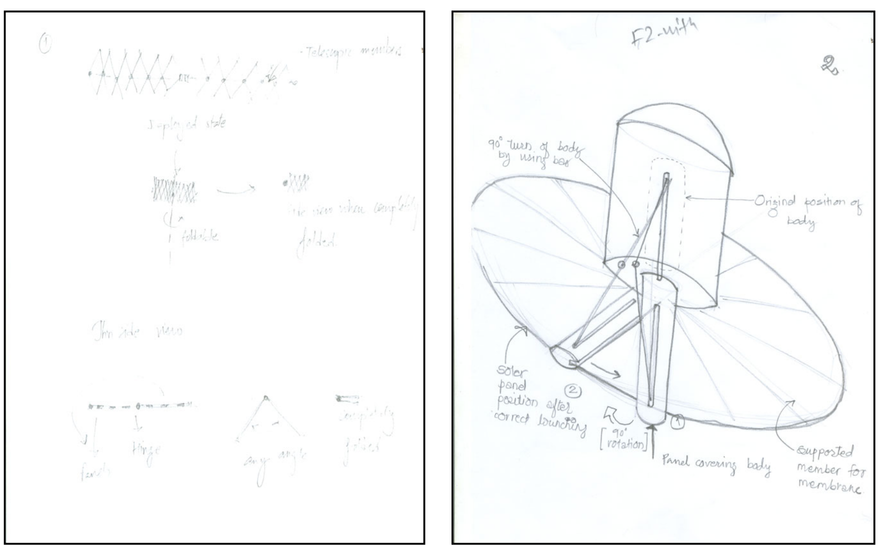

Figure 3. Shows the type of solutions that the designers have produced without(left) and with software(right).

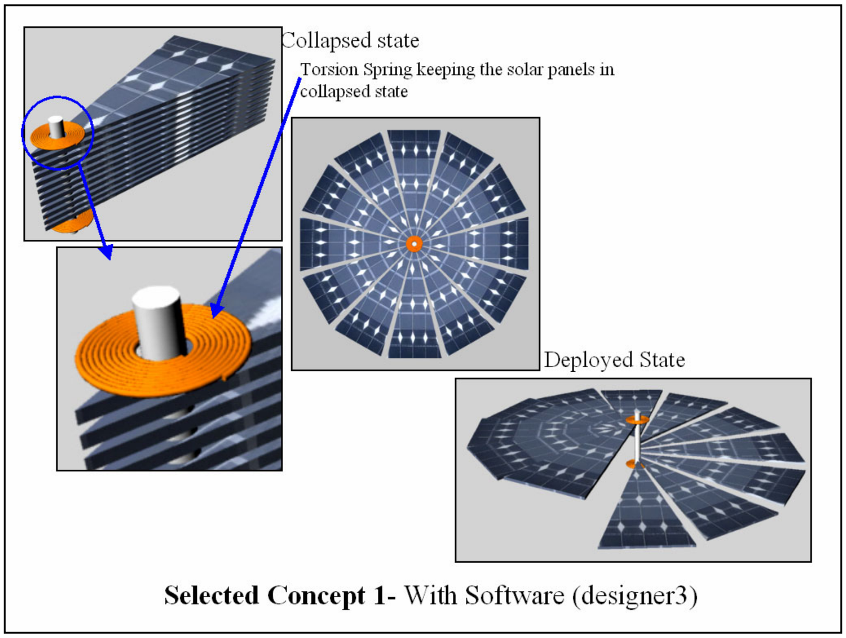

Figure 4. Shows one of the concepts modelled using a 3d modelling software. This idea has been generated with the aid of the software.

7 Statistics of the evaluation of the software

What is promising is that the software, with a limited number of data entries (about 280) accounts for an average of about 47 % of all the ideas generated in phase 1. In phase 2 we see that the average ratio of the number of ideas generated with and without software is 165%, which is considerably higher (with a larger database of about 400 entries); this illustrates the efficacy of the support in enhancing designers performance in creating a wide variety of ideas for solving a given problem (see Tables 2 and 3). Experiments conducted in phase 2 also indicate that exploration of ideas does enhance one's capability create large number of solutions. We think that, by exploring a larger number of entries, designers are getting triggered with newer ideas with varied characteristics, which is helping them to generate a non-linearly larger number of other ideas. In-depth research in this regard is under progress.

Table 2. Results of phase 1 shows the number and % of design solutions generated by designers

	Solutions without software's aid	Solutions with software's aid	
		Direct solutions	Inspired solutions
Type of person providing solutions: Designer 1			
Problem 1	8 (8/9*100) 88.8 %	--	1 (11.2 %)
Problem 2	9 (60 %)	6 (40 %)	--
Type of person providing solutions: Designer 2			
Problem 1	5 (41.7 %)	--	7 (58.3 %)
Problem 2	8 (72.7 %)	--	3 (27.3 %)
Type of person providing solutions: Designer 3			
Problem 1	3 (23.1 %)	6 (46.2 %)	4 (30.7 %)
Problem 2	3 (33.3%)	--	6 (66.7%)

Table 3. Results of phase 2: In this phase we combined direct and inspired solution from the software, but checked how many entries the designers are watching (exploring).

	Without using idea-inspire			With using idea-inspire					
	No of ideas generated (G1)	No of ideas selected (S1)	Ratio of S1/G1	No of ideas generated (G2)	No of ideas selected (S2)	Ratio of S2/G2	No of entries explored (E)	Ratio of G2/E	Ratio of G2/G1
D1	9	4	44%	17	6	35%	60	28%	188% (17/9)
D2	6	4	66%	5	3	60%	40	12%	083% (5/6)
D3	8	3	37%	18	6	33%	60	30%	225% (18/8)
average	7.6	3.6	49%	13.3	5	42%	53	23%	165%

8 Results and conclusions

Two databases with entries from natural and artificial systems have been developed, consisting of several entries in each of them. Entries in each database are carefully selected in order to demonstrate the motion characteristics of systems in different media (e.g. land, water etc.) and in different ways to execute the same task (like one can move by leaping, hopping etc). The information collected contains details about these systems in the form of written, diagrammatic, pictorial, video and animation data, and their varied motions.

Using the constructs and relationships of the SAPPhIRE model, behaviour of each system is described in the databases developed. Various analogical search strategies are developed to search for entries with potential to solve a given problem. The concepts are implemented in a piece of software called IDEA-INSPIRE.

Comparative experiments using several design problems with designers indicate substantial potential for the approach in inspiring generation of novel solutions.

9 Further work

Further analyses of the relative efficacy of the search processes are to be carried out with a larger number of experienced designers, taking more design problems from industry. Understanding of the mechanism of exploration and triggering process is also an important task and this work is in progress. Also, we are in the process of increasing the number of entries in both the databases, in order to make the software more effective.

Besides problem solving, the software does has educational value. One can learn about various animals, plants and mechanical devices in depth by using this software. We have not yet conducted any design experiment to ascertain its efficacy in aiding learning over other standard methods such as compendia or encyclopaedia CDs.

10 Acknowledgements

Major part of the work described in this paper is the outcome of a project on Biomimetics, funded by ISRO (Indian Space Research Organization), Bangalore, India. Development and testing of the model and the software have been carried out in-house in the Innovation, Design Study and Sustainability Laboratory (IDeAS lab), Centre for Product Design and Manufacturing, Indian Institute of Science, Bangalore. The authors acknowledge help from S Phaneendra and Santosh Jagtap in various phases of development of the software, and from several designers, both internal and external, for testing the software.

References

[1] Larry W., Zimmerman.P.E and Glen D Hart, 1988, Value Engineering -a practical approach for owners, designers and contractors, CBS Publishers and Distributors.

[2] Murakami Tamotsu, & Nakajima Naomasa, (1997). Mechanism Concept retrieval Using Configuration Space. *Research in Engineering Design, 9*: 9-11.

[3] Kletke Marilyn, G. Mackay Jane, M. Barr Steve, H. & Jones Beata. (2001). Creativity in the organization: the role of individual creative problem solving and computer support. *Int. J. Human-Computer Studies, 55,* pp. 217-237.

[4] MacCrimmon, K.R. & Wagner, C. (1994). Stimulating ideas through creative software. *Management Science, 40*, pp.1514-1532.

[5] Watson, D. L.(1989). Enhancing creative productivity with the Fisher associated lists, *the Journal of Creative Behavior, 23*,pp. 51-58.

[6] Vogel, S. (1998). *Cat's Paws and Catapults: Mechanical Worlds of Nature and People*. WW Norton, NY.

[7] French, M.J. (1998). *Invention and Evolution: Design in Nature and Engineering*, CUP, Cambridge.

[8] Bhatta, S. Goel, A. & Prabhakar, S. (1994). Innovation in Analogical Design: A Model-Based Approach. *Proc. AI in Design, Kluwer Academic Publishers, Dordrecht, The Netherlands*, pp. 57–74.

[9] Gordon, W.J.J. (1961). *Synthesis: The Development of Creative Capacity*, HAPPER & ROW, NY.

[10] Chandrasekaran, B. (1994). *Functional representation and causal processes*, in *Advances in Computers*, M.C.Yovits, Editor., Academic Press. p. 73-143.

[11] Qian, L. and Gero, J.S. (1996). Function-Behavior- Structure Paths and Their Role in Analogy- Based Design. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing, Vol. 10, No. 4, Sept.,* pp. 289–312.

[12] Umeda et al., Y., (1996). Supporting Conceptual Design Based on the Function-Behavior-State Modeler, *AIEDAM, Vol. 10, No. 4, Sept.* pp. 275–288.

[13] Goel, A. K. (1997). Design, Analogy and Creativity. *AI in Design*.

[14] Deng, Y.M. (2002) Function and Behavior Representation in Conceptual Mechanical Design, AI EDAM, 16, 343-262.

[15] Chakrabarti, A. (2001). Towards hybrid methods for synthesis. *International Conference on Engineering Design, ICED 01 GLASGOW*, pp. 379-386.

[16] Chakrabarti, A. and Blessing, L. (1996). Special Issue: Representing functionality in design. *Artificial Intelligence for Eng. Design, Analysis and Manufacturing, 10,* 251-253.

[17] Chakrabarti, A and Bligh, T.P. (1993). Towards a Theory for Functional Reasoning. *Proc. of the International Conference in Engineering Design (ICED93), The Hague, 1*, 1-8.

[18] Pahl, G. and Beitz, W. (1988). *Engineering Design: A Systematic Approach*. Springer.

[19] Chakrabarti, A. Johnson, A.L. & Kiriyama, T. (1997). An Approach to Automated Synthesis of Solution Principles for Micro-Sensor Designs. *Proc. Intl. Conference in Engineering Design, Vol. 2, Tampere,* pp-125-128.

[20] Hubka, V., (1976) Theorie der Konstruktionsprozesse, Berlin, Springer-Verlag.

[21] Andreasen, M. M. (1980), "Syntesemetoder pa Systemgrundlag", Ph.D, Thesis, Lund Technical university, Lund, Sweden.

[22] Chakrabarti, A. & Bligh Thomas, P. (2001). A Scheme for Functional reasoning in Conceptual Design. *Design Studies, Vol 22, No.6,* pp. 493-517.

[23] Yoshioka, M., & Tomiyama, T. (1997). Pluggable Metamodel Mechanism: A Framework of an Integrated Design Object Modelling Environment, in A. Bradshaw and J. Counsell (eds.): *Computer Aided Conceptual Design '97, Proceedings of the 1997 Lancaster International Workshop on Engineering Design CACD '97*, Lancaster University, Lancaster, UK, pp. 57-70

[24] Chakrabarti, A. & Regno, R. (2001). A New Approach to Structure sharing. *International Conference on Engineering Design, ICED 01 GLASGOW*, pp. 155-162.

[25] IDEA-INSPIRE (2004) Idea-Inspire User's Manual, Centre for product Design and Manufacturing, Indian Institute of Science, Bangalore.

Corresponding author:
Amaresh Chakrabarti,
Innovation Design Study and Sustainability Laboratory, Centre for Product Design and Manufacturing,
Indian Institute of Science, Bangalore – 560012.India.
Tel: 091-80-22932922,
E-mail: ac123@cpdm.iisc.ernet.in
URL: http://cpdm.iisc.ernet.in/ac.htm