An Improved Framework for Design Concept Generation

Based On Experiential and Intuitive Methods

by

Sumit Sunil Narsale

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved October 2014 by the
Graduate Supervisory Committee:

Jami Shah, Chair
Teresa Wu
Joseph Davidson

ARIZONA STATE UNIVERSITY

December 2014

ABSTRACT

Conceptual design stage plays a critical role in product development. However, few systematic methods and tools exist to support conceptual design. The long term aim of this project is to develop a tool for facilitating holistic ideation for conceptual design. This research is a continuation of past efforts in ASU Design Automation Lab. In past research, an interactive software test bed (Holistic Ideation Tool - version 1) was developed to explore logical ideation methods. Ideation states were identified and ideation strategies were developed to overcome common ideation blocks. The next version (version 2) of the holistic ideation tool added Cascading Evolutionary Morphological Charts (CEMC) framework and intuitive ideation strategies (reframing, restructuring, random connection, and forced connection).

Despite these remarkable contributions, there exist shortcomings in the previous versions (version 1 and version 2) of the holistic ideation tool. First, there is a need to add new ideation methods to the holistic ideation tool. Second, the organizational framework provided by previous versions needs to be improved, and a holistic approach needs to be devised, instead of separate logical or intuitive approaches. Therefore, the main objective of this thesis is to make the improvements and to resolve technical issues that are involved in their implementation.

Towards this objective, a new web based holistic ideation tool (version 3) has been created. The new tool adds and integrates Knowledge Bases of Mechanisms and Components Off-The-Shelf (COTS) into logical ideation methods. Additionally, an improved CEMC framework has been devised for organizing ideas efficiently.

Furthermore, the usability of the tool has been improved by designing and implementing a new graphical user interface (GUI) which is more user friendly. It is hoped that these new features will lead to a platform for the designers to not only generate creative ideas but also effectively organize and store them in the conceptual design stage. By placing it on the web for public use, the Testbed has the potential to be used for research on the ideation process by effectively collecting large amounts of data from designers.

DEDICATION

This thesis is dedicated to my mother Sujata Sunil Narsale and my father Sunil Vishnupant Narsale who worked very hard all their life to provide a good education for me and my brother Swapnil, whose love and support is always there for me. I also dedicate this thesis to all my family members and teachers back home whose care and affection was always there when I needed them. Special thanks to my school and college friends back in India. I would also like to thank my friends here in the U.S. who were always there to help me in the last 3 years.

ACKNOWLEDGEMENTS

My most sincere appreciation and gratitude go to my committee chair and advisor, Dr. Jami Shah, for his never-ending patience, guidance, inspiring mentorship and support in my professional as well as personal life. Without his help this work would not have been possible. I would like to thank Dr. Joseph Davidson and Dr. Teresa Wu, for their time, efforts and for being part of my committee.

I would also like to thank students in the Design Automation Laboratory - Prabath, Prashant, Samir, Shyam, Payam, Mahmoud, Pradeep, Lupin, Abraham and Nathan. They have been great colleagues to work with and have always been willing to discuss and offer advice on my ideas. I would also like to thank former students Manikandan Mohan and Ying Chen who developed the previous versions of the Ideation Testbed.

I wish to acknowledge the financial support for this research provided by the National Science Foundation Grants # 1045644,12010162.

TABLE OF CONTENTS

Page

CHAPTER

LIST OF TABLES

Table Page

Table Page

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

CHAPTER 1

PROBLEM STATEMENT

The conceptual design stage is a critical stage of product design in mechanical engineering, decisions made at this stage constrain the rest of the process. Despite this, little attention has been paid to the conceptual design stage. This is because most existing software tools assist designers in the later stages of product development, i.e., the detailed design stage.

Developing a computer tool for supporting the conceptual design stage will significantly contribute to the engineering design society. Such a tool should support activities such as Problem Formulation, Concept Generation, and Concept Evaluation. To carry out each of these activities, there exist many methods and, based on these methods, there exist some tools. However, such tools do not support a variety of diverse methods, so are very limited in scope.

At the Design Automation Lab at Arizona State University, a comprehensive ideation tool has been created, hereafter referred as holistic ideation tool. The first version of the holistic ideation tool (V.1) was based on the logical/experiential set of ideation methods. One of the significant contributions of V.1 was that the conceptual design process was divided into three different stages (Problem Formulation, Concept Generation, and Concept Evaluation). It was different from other existing tools because it was holistic, i.e., it provided access to multiple ideation methods. However, V.1 had very little organization for storing and documenting the ideas generated by the designer. In order to resolve this issue from V.1, a second version (V.2) provided the morphological chart as an organizational structure, and integrated the intuitive set of ideation methods in the tool. These two versions will be reviewed in the coming sections. Based on that we will see how a new improved version (V.3) was created as a part of my research.

PROBLEM STATEMENT: *The main objective of this thesis is to provide an holistic approach for conceptual design stage, for which the web based holistic ideation tool was created by integrating logical/experiential and intuitive ideation methods with an improved organizational framework to store and synthesize ideas.*

To achieve this objective, the following modifications are to be carried out :

1. An improved Organizational Framework for Holistic Ideation.
2. Addition of new ideation methods encompassing Mechanisms and Machine Elements.
3. An user-friendly GUI to improve the usability of the tool.

The modifications will be addressed through this thesis, but first we shall provide a background to the problem at hand, and give a detailed account of the previous versions of this tool.

CHAPTER 2

BACKGROUND

2.1 Conceptual Design

The design process typically consists of the following stages: pre-design (market studies, customer survey, identification of opportunities, competitive benchmarking), problem definition (design objectives, Tech specifications, QFD), conceptual design (data collection, research, functional synthesis, concept generation, concept evaluation, selection), embodiment design (product architecture, sizing critical components, material selection, geometric & parametric design, engineering analysis, trade-off studies, economic analysis, optimization), detail design (detailed layout, CAD models, engineering drawings, tolerance analysis, DfX), prototyping & testing. The majority of the cost of a new product design is committed at the conceptual design stage (Beitz & Pahl, 1996) because the decisions made at an early stage have a significant influence on the factors such as cost, performance, reliability, safety and environmental impact of a final product. In Conceptual Design stage, designers generate design alternatives or design concepts and evaluate them in order to determine their feasibility and fitness. More importantly, a good detailed design cannot come from a poorly conceived design concept (Liu & Hsu, 2000).

One should also take into account that during early phases of design, knowledge of most of the aspects like design requirements and constraints is usually imprecise or approximate and unknown in some of the cases. This project aims to help designers during the ideation process by providing a computer tool for supporting conceptual design.

2.2 Engineering Creativity

Neimark defines technological and scientific problem solving as creativity in a “narrow sense” (Neimark & Kagan, 1987). Creativity is defined by breaking free of fictitious barriers and constraints. The technological creativity lies at the intersection of novelty, functionality and feasibility. Often, we find designers finding creative solutions by combining the old ideas into novel ones, which must be valuable in some way. They concern original ideas (Boden, 1996) that not only did not happen before, but also could not have happened before. That is why during the conceptual design stage, an attempt to generate as many novel solutions as possible is very crucial in order to achieve a creative and novel design.

2.3 Holistic Ideation

There have been attempts made in the research community to find a systematic approach for the conceptual design stage. A majority of these show a one-dimensional approach for design ideation, i.e., following only one strategy, e.g., C-Sketch, TRIZ, Function-artifact morphology or Morphological Charts. Examples of one dimensional tools are: the Design Repository (Bohm, Stone, Simpson, & Steva, July 2008) which supports function based artifacts; DANE (Vattam, Helms, & Goel, 2010) which supports the use of analogies between nature and engineering; TRIZ Workbench (TRIZ, 2006).

However, the following facts make it difficult to use a single tool, which is based on a single approach, for all the scenarios: there is a wide variation in design problems, in artifact domains, and in designer expertise. Apart from that, a continuously changing ideation state of a designer requires different ideation strategies at different stages.

In the past, Bill Gordon who is the founder of Journal of Creative Behavior recognized that a designer goes through many different phases in creative problem solving (Gordon, 1961). And because of that the strategy needed to move forward in the conceptual design varies as a designer navigates through the design space. Synectics recognizes the need for changing strategies to match the needs at a given time. This method was developed 50 years ago and fails to take any advantage of information technologies, such as the Internet, computer assisted learning and searchable knowledge in data and knowledge bases. Following the spirit of Synectics, at DAL, we proposed a new holistic approach (Mohan, Chen, & Shah, 2011) to allow designers to use any combination of ideation methods and their ideation strategies to overcome creativity blocks encountered at different steps.

Mohan (Mohan M. , 2011) implemented the first version of the Holistic Ideation tool. The tool was later enhanced by Chen (Chen, 2012). We will call these two versions as V.1 and V.2 respectively.

2.4 Intuitive & Logical Methods

Intuitive ideation methods have been developed in order to remove perceived barriers to divergent thinking and increase the chances for conditions believed to be promoters of creativity (Shah J. J., 1998). They do not rely on technical or historical data. These methods do not guarantee a solution and solution depends heavily upon chance, human creativity and stimuli from interaction between people in groups. On the other hand, Logical methods involve step-by-step problem analysis, decomposition, and direct use of cataloged solutions (charts, tables, databases). These solution catalogs are based on science and engineering principles and past experience. Success of logical methods depends not only on technical expertise of the individuals but also on the quality/quantity of the information in catalogs, charts and other knowledge bases.

Figure 1 shows detailed classification of ideation methods. The Intuitive methods are classified into following categories: Reframing, Freeform, Progressive/Brain writing, Facilitated, Idea Morphing and Organizational. And the categories of the major logical methods are: History-based, Heuristic Principle, First Principle and Idea Morphing. Logical methods differ from each other in the type of archived knowledge and databases used.

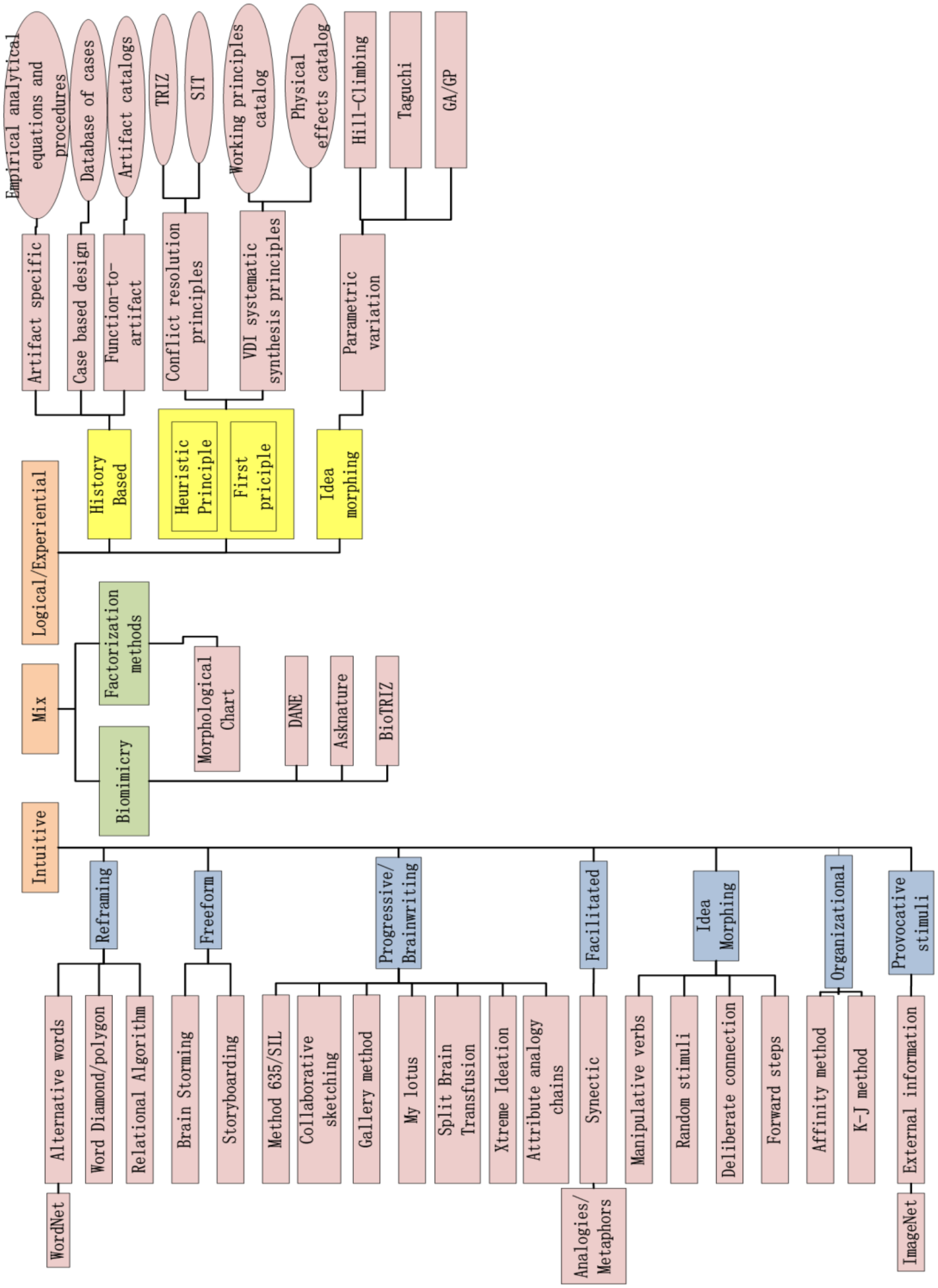


Figure 1: Ideation Methods Classification (Mohan, Chen, & Shah, 2011)

CHAPTER 3

HOLISTIC IDEATION TOOL V.1 REVIEW

In this chapter, we will review V.1, which was the first attempt at DAL to create a Holistic Ideation tool (Mohan M. , 2011). The following sections will discuss the background, framework, kinds of ideation methods implemented, and how these ideation methods were integrated. Furthermore, we will also review implementation, UI, user feedback and shortcomings of V.1.

3.1 Background

3.1.1 Ideation States

The term “ideation state” refers to the designers' current understanding of the design space and their current location in that design space (Mohan, Chen, & Shah, 2011). Examples of which include, source of difficulty (designer, problem, resource), nature of problem (technical, physical, economic) and complexity of problem (variables and relations involved, degree of coupling).

3.1.2 Characterization of Ideation States

In V.1, an ideation state is characterized by the following: the current focus of the problem solver, the factors that are blocking creativity, the level of satisfaction/dissatisfaction with ideas generated, and the types of problem (novelty, complexity, uncertainty). A set of indicators was used to characterize ideation states. These indicators are classified into process related, problem related, and outcome related categories (Table 1). The process of finding the position of the designer in the design space is called as the characterization of the ideation state.

Table 1: Indicators of Ideation States (Mohan, Chen, & Shah, 2011)

Problem			**Process**		**Outcome**			
Novelty	Complexity	Uncertainty	time	path	Quantity	quality	novelty	variety

3.1.3 Ideation Blocks and Unblocking Operations

Solving a design problem is an iterative process. In the conceptual design stage, a designer faces many impasses at various times due to various reasons. These impasses are defined as creativity blocks (Mohan, Chen, & Shah, 2011) (Dorst & Cross, 2001). In the classical Synectics method, there is an experienced facilitator present to monitor a groups' idea generation process. The task of the facilitator is to suggest appropriate ideation methods to the group. These ideation methods have different ideation strategies embedded in them. Ideation strategies help designers to overcome their mental blocks. Table 2 lists some of the ideation methods and certain cognitive mechanisms/strategies embedded in them.

Table 2: Ideation Strategies and Corresponding Ideation Methods (Mohan, Chen, & Shah, 2011)

Ideation method	***Ideation "mini" Strategy***
Brainstorming, K-J, PMI	Suspend judgment
Brainstorming, 635	Emphasize quantity over variety
Alternate Words, Action verbs, Physical effects database/ WP database	Shift frame of reference
Synectics	Use analogies and metaphors
C-Sketch, Gallery, 635, Brainstorming, Artifact catalogs	Apply provocative stimuli
Morph charts	Make random connections between sub solutions
Used whenever fixation is identified except for fixed time methods (C-sketch, Gallery,635)	Incubate (use SC thinking)
Synectics	Break rules; suspend constraints
Alternate words, hypernyms	Abstract the problem
Relational algorithm	Impose fictitious constraints
Artifact catalogs (based on functional decomposition)	Remove fictitious constraints
Database of cases, TRIZ, component catalogs	Look at an example solution

In previous research, the design community has identified and characterized some ideation blocks. Design Fixation is one of the blocking phenomenon. It refers to the inability to find new solutions or solution paths. There are other types of blocks, as well (Table 3). To overcome such blocks, specific ideation strategies may be used. Unblocking operations corresponding to different values of characterization measures are shown in the Table 3. Symbol "↑" means high, symbol "↓" means low, symbol "-" means "medium" and empty box indicates ongoing research work.

Table 3: Part of Blocking Phenomena, Tentative Characterization & Unblocking Operations (Mohan, Chen, & Shah, 2011)

Blocking Phenomena	**Tentative Characterization**									**Unblocking operations**
	Problem			**Process**		**Outcomes**				
	Nvlty	**Cmplx**	**Uncrt**	**Time**	**Path***	**Qty**	**Qlty**	**Nvlty**	**Var**	
Difficulty understanding the problem	↑	↑	↑	↓		↓	↓	↓	↓	Flexible problem representation, Use of analogies and metaphors ,Reframe problem
Unable to prioritize	↑	↑	↑	↓						TBD
Unmanageable complexity	–	↑	–	↓		↓	–	–	–	Work on a higher problem, Break rules, Decomposition
Design fixation	–	–	↓	↑		↓	–	↓	↓	Provocative stimuli (Random/focused), Random connections, Forced connections, Incubation

3.1.4 Ideation Strategies

The Ideation strategies are defined as the cognitive mechanisms believed to intrinsically promote ideation or to help designers overcome mental blocks (Mohan, Chen, & Shah, 2011). The V.1 recognizes ideation states at different times and helps to solve the design creativity blocks through different ideation strategies. These ideation strategies are embedded in different ideation methods which make holistic ideation rather a multi-dimensional approach for problem solving instead of a monolithic one.

3.2 Logical Ideation Methods

3.2.1 Function Decomposition

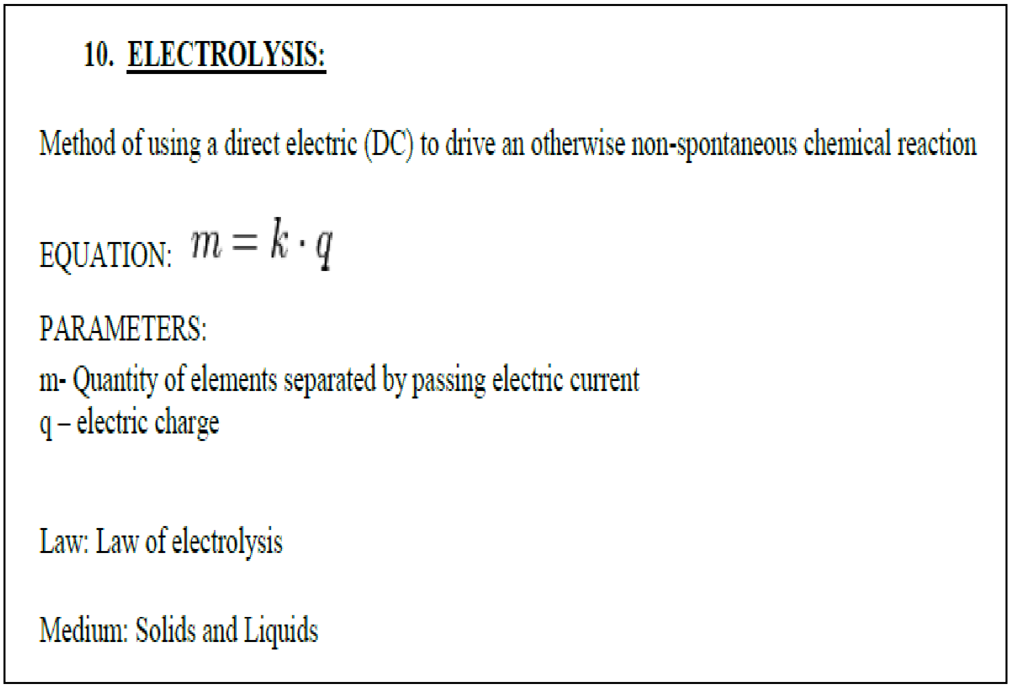
A function is defined as the intended input/output relationship of a system which performs a certain task (Beitz & Pahl, 1996). Functional decomposition here refers a process of breaking down high level functions into constituent parts, i.e., sub-functions. In this way, the original function can be decomposed and reconstructed. The result of this decomposition process is a structured representation of the functions with hierarchy and temporal relations. Functions are defined by an action and an object. Functions can also be modeled in terms of an input and an output flow (energy, material, and signal).

One tool for doing function decomposition is FunctionCAD, which was created by Oregon State University (Nagel, Perry, & Stone, 2009). FunctionCAD is an interactive functional modeler, which connects various functions by three types of flows (energy, material, and signal) (Hirtz, Stone, McAdams, Szykman, & Wood, 2002). In FunctionCAD, the user constructs a flow chart and specifies function types, which was taken from reconciled functional basis (RFB) ontology. RFB was developed as a combination of functional basis and United States National Institute of Standards Technology (NIST) function ontology (Szykman, Racz, & Sriram, 1999). V.1 includes FunctionCAD tool for doing function decomposition. It is up to the designer to select the functions and flows from the pre-loaded list. The advantage of using functions from a pre-defined ontology is that it can be used to index artifact repositories.

3.2.2 Physical Effects Catalog

There are certain physical laws which govern the physical quantities involved in a process. These laws govern the flow variables/physical variables. The most abstract level of representation is the physical effects as we can see in the genealogy tree defined by Shah et al. (Shah J. J.-H., 2003). All the artifacts exploit various physical effects (e.g. mechanical, thermal, biological, and electronic) to achieve one or more desired functions. Many such effects have been previously identified and formulated. Researchers have produced catalogs of physical effects. These physical effects correspond to a variety of mechanical functions, flow variables and physical parameters.

In V.1, a Physical Effect (PE) catalog is included, which define PEs in terms of a name, equation, parameters, physical law, medium, domain and description. It included 60 physical effects, 206 parameters, and 96 equations. Ideas and concepts generated at the physical effects level are highly varied and abstract. Therefore, the designer can explore different places in his design space by looking through PE catalog which represent fundamental abstract level of any idea. An example from the PE catalog in the Figure 2 shows the physical effect of electrolysis with its description, involved behavioral equations and related parameters.



10. ELECTROLYSIS:

Method of using a direct electric (DC) to drive an otherwise non-spontaneous chemical reaction

EQUATION: $m = k \cdot q$

PARAMETERS:
m- Quantity of elements separated by passing electric current
q – electric charge

Law: Law of electrolysis

Medium: Solids and Liquids

Figure 2: Example of Physical Effect

Usually, the use of physical effect follows function decomposition but it's not automated in V.1. Which means that the catalog does not contain any link to function, so the connection has to be made by the designers manually. Depending upon the factors like function or designer experience, one physical effect or a combination of many physical effects may be needed to fulfill a function.

There are two major advantages of the database design implemented in V.1. First, any of the attribute that describe a PE could be a starting point or breakthrough point to the other databases (i.e. working principle database, artifact database, etc). Second, multiple physical effects may share some common attributes, for example, the parameter “force” is associated with both Newton’s law and angular acceleration. By sharing common attribute (such as parameters, equations), the links between physical effects can be established through the parameters, which provides the basis for traversing from one physical effect to related ones.

3.2.3 Working Principle Catalog

Physical effects describe physical laws (“what it is”), while working principles focus on geometric and material characteristics (“how to do it”). A combination of physical effects and working principles are required to describe the behavior of a system/working structure. As Pahl and Beitz (Beitz & Pahl, 1996) described, working surfaces can be determined by type, shape, position, and size. Working motion can be type (translation or rotation), nature (regular or not), direction, and magnitude. However, these details are not sufficient to fully determine a working principle and hence material properties are also needed.

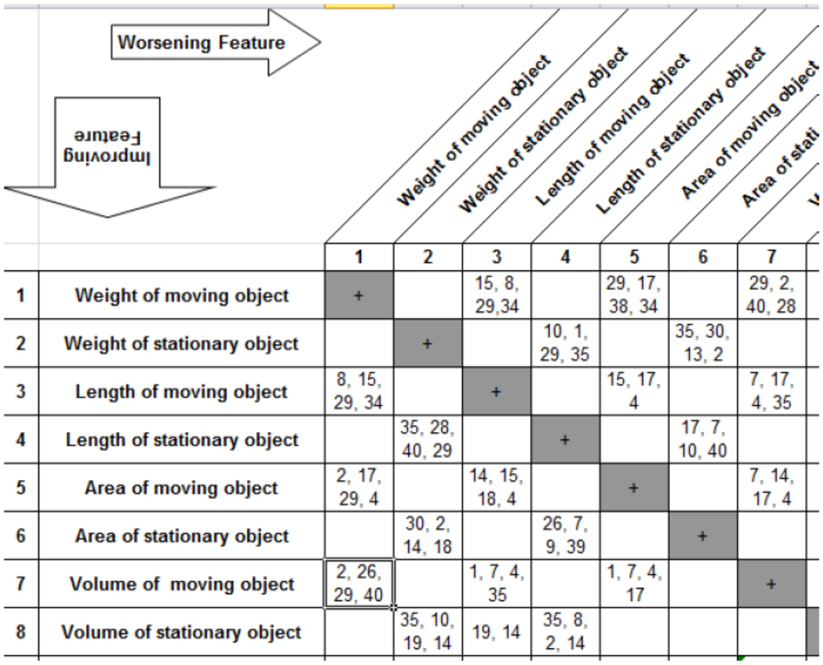
In V.1 a working principle catalog was implemented from a set of working principles (WP) described in VDI 2222 (VDI, 1997) and Pahl and Beitz (Beitz & Pahl, 1996). Working Principles are defined in the form of name, description, component, corresponding physical effect, materials, key physical variable, and geometry.

3.2.4 Artifact Catalog

Extensive research has been going on in UMR/Oregon State University since 1999 in the area of development of artifacts catalog. A tool developed by them is Design Repository (Bohm, Stone, Simpson, & Steva, July 2008). It has been included in V.1. The Artifact Catalog was initially used as a part of a design course in order to make contribution to the conceptual design research. Function structures were developed for each component and other attributes were constituted to the artifacts. Design Repository defined each artifact by input flow, output flow, output artifact, input artifact, support function and sub-function. This function artifact catalog, which contains 5,600 artifacts in the database, allows designers to search for artifacts by to functions.

3.2.5 TRIZ

The Theory of Inventive Problem Solving (TRIZ) (Altshuller, 2001)) was developed by Altshuller in the 1940s. It is a set of logical ideation methods. The most popular one is based on its contradiction matrix. TRIZ identified 39 parameters between which conflicts often arise. They also collected 40 so-called invention principles. The 39 X 39 contradiction matrix can be used to find invention principles to resolve conflict between parameter i and j. For using TRIZ, designers need to find a technical contradiction and after that match a TRIZ recommended principle which has been used in similar situation but perhaps in another application or domain. By using the TRIZ matrix shown in Figure 3, principles can be found for specific technical contradiction. Table 4 shows the list of TRIZ invention principles. Table 5 shows the list of TRIZ parameters. TRIZ helps designers to learn from past experience, and then use that for providing future design development by transferring his/her thinking patterns.



Improving Feature	Worsening Feature	Weight of moving object	Weight of stationary object	Length of moving object	Length of stationary object	Area of moving object	Area of stati	V
		1	2	3	4	5	6	7
1	Weight of moving object	+		15, 8, 29,34		29, 17, 38, 34		29, 2, 40, 28
2	Weight of stationary object		+		10, 1, 29, 35		35, 30, 13, 2	
3	Length of moving object	8, 15, 29, 34		+		15, 17, 4		7, 17, 4, 35
4	Length of stationary object		35, 28, 40, 29		+		17, 7, 10, 40	
5	Area of moving object	2, 17, 29, 4		14, 15, 18, 4		+		7, 14, 17, 4
6	Area of stationary object		30, 2, 14, 18		26, 7, 9, 39		+	
7	Volume of moving object	2, 26, 29, 40		1, 7, 4, 35		1, 7, 4, 17		+
8	Volume of stationary object		35, 10, 19, 14	19, 14	35, 8, 2, 14			

Figure 3: Partial TRIZ Matrix

Table 4: List of Invention Principles in TRIZ

Invention Principles	
1. Divide and Conquer	21. Speed up
2. Extract as needed	22. Turn a minus into Plus
3. Local Quality	23. Use Feedback
4. Asymmetry	24. Use Mediation
5. Consolidate	25. Generate Self service
6. Increase Universality	26. Copying
7. Nesting	27. Make disposable
8. Use counterweight	28. Replace Mechanical system
9. A priori counter action	29. Use Pneumatics and hydraulics
10. Pre-emptive action	30. Flexible shells and thin films
11. Compensation in advance	31. Porous materials
12. Equipotentiality	32. Color changes
13. Reverse action	33. Homogeneity
14. Change form	34. Rejection and Regeneration
15. Increase degree of flexibility	35. Transform parameters
16. Excessive or deficient action	36. Use phase Transformations
17. Change dimension	37. Thermal expansions
18. Use Mechanical vibration, Oscillation	38. Accelerate Oxidation
19. Periodic action	39. Inert environment
20. Steady Useful actions	40. Composite materials

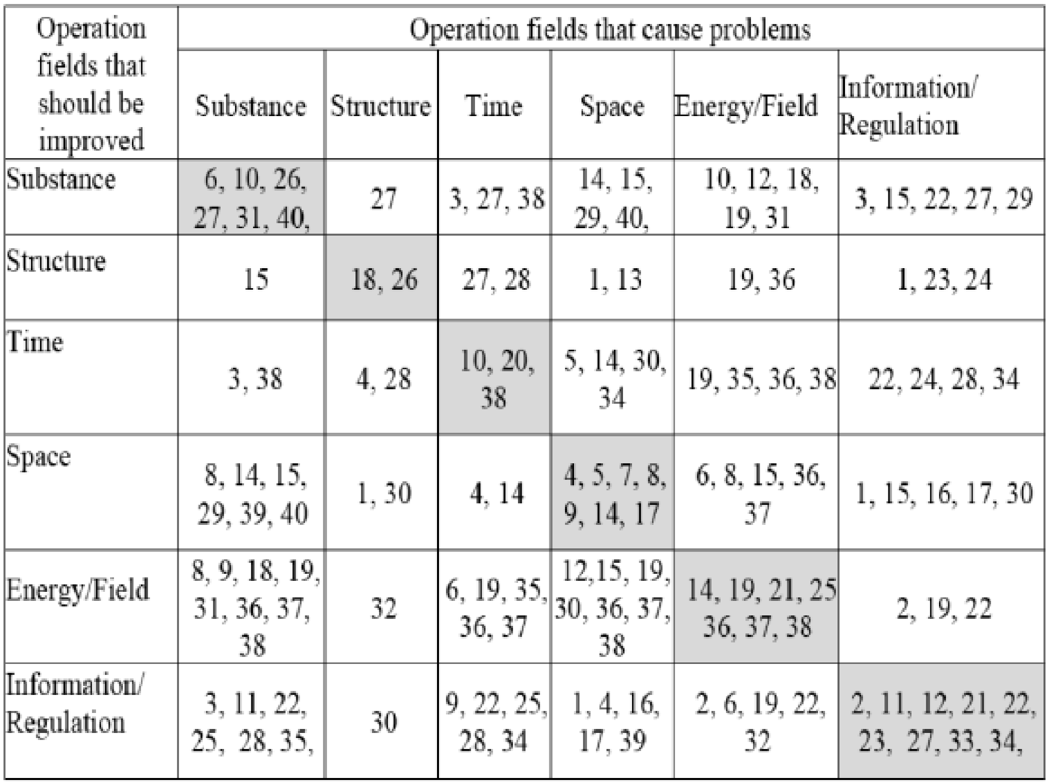
Table 5: List of Parameters in TRIZ

TRIZ Parameters	
1: Weight of moving object	21: Power
2: Weight of stationary	22: Loss of Energy
3: Length of moving object	23: Loss of substance
4: Length of stationary	24: Loss of Information
5: Area of moving object	25: Loss of Time
6: Area of stationary	26: Quantity of substance/the
7: Volume of moving object	27: Reliability
8: Volume of stationary	28: Measurement accuracy
9: Speed	29: Manufacturing precision
10: Force (Intensity)	30: Object-affected harmful
11: Stress or pressure	31: Object-generated harmful
12: Shape	32: Ease of manufacture
13: Stability of the object	33: Ease of operation
14: Strength	34: Ease of repair
15: Durability of moving object	35: Adaptability or versatility
16: Durability of non moving object	36: Device complexity
17: Temperature	37: Difficulty of detecting
18: Illumination intensity	38: Extent of automation
19: Use of energy by moving	39: Productivity
20: Use of energy by stationary	

3.2.6 Bio-TRIZ

Vincent and Bogatyreva (Bogatyreva O. S., 2004) developed Bio-TRIZ. As the name suggests, Bio-TRIZ was inspired from TRIZ and designs based on biological solutions. It acts as a bridge between biology and TRIZ. By doing this, Bio-TRIZ enables designers to implement natural principles for innovative design and technology. Bio-TRIZ establishes six fields of operation for the purpose of capability of comparing parameters from technological and biological domains. These six operational fields are: Substance, structure, energy, information, space, and time. These operational fields makes it possible to re-organize and condense the TRIZ classification (Contradiction Matrix) of both of the features used to generate the conflict statements and the Inventive Principles. This reorganized TRIZ matrix places the above mentioned 40 principles of TRIZ into a new order, which reflects the biological route through TRIZ conflicts. Thus, this new matrix is named Bio-TRIZ matrix. Bio-TRIZ reflects both logical and intuitive strategies by tracking past design experience though particular conflicts and seeking bio-inspiration (i.e. analogy). By particular pairs of conflicts in Bio-TRIZ, comparison of the types of solutions in technology and biology is possible. Table 6 shows how these 40 TRIZ principles are placed in Bio-TRIZ matrix.

Table 6: Six Fields of Operation Matrix (Bogatyreva O. &., March,2009)



Operation fields that should be improved	Operation fields that cause problems					
	Substance	Structure	Time	Space	Energy/Field	Information/ Regulation
Substance	6, 10, 26, 27, 31, 40,	27	3, 27, 38	14, 15, 29, 40,	10, 12, 18, 19, 31	3, 15, 22, 27, 29
Structure	15	18, 26	27, 28	1, 13	19, 36	1, 23, 24
Time	3, 38	4, 28	10, 20, 38	5, 14, 30, 34	19, 35, 36, 38	22, 24, 28, 34
Space	8, 14, 15, 29, 39, 40	1, 30	4, 14	4, 5, 7, 8, 9, 14, 17	6, 8, 15, 36, 37	1, 15, 16, 17, 30
Energy/Field	8, 9, 18, 19, 31, 36, 37, 38	32	6, 19, 35, 36, 37	12,15, 19, 30, 36, 37, 38	14, 19, 21, 25 36, 37, 38	2, 19, 22
Information/ Regulation	3, 11, 22, 25, 28, 35,	30	9, 22, 25, 28, 34	1, 4, 16, 17, 39	2, 6, 19, 22, 32	2, 11, 12, 21, 22, 23, 27, 33, 34,

3.3 V.1 Implementation

The logical ideation methods presented in Section 3.2 are supported in V.1. Table 7 represents how each ideation method is modeled and stored with which attributes. These methods are interconnected through physical parameters. How these different ideation methods are modeled and what entities are included in their representation will be discussed in this section. And the following subsections will provide details about the relation between entities.

When the designer is tackling a design problem, he/she is working with an ideation strategy. There are two scenarios in which a designer may want to change the ideation strategy. Firstly, the designer may change the ideation strategy because he/she is stuck. Secondly, a designer may change the ideation strategy just to try another method. While solving a design problem, the designer changes his/her ideation strategy several times. And as ideation strategies are embedded in different ideation methods, V.1 needed to have some relation between ideation methods to switch ideation strategies.

In previous research, physical parameters/variables were found to be an integral part of physical effects and working principles. Artifacts also comprise physical variables in the form of the flow variables that flow in and out of the components. After deeper analysis of TRIZ method, it was found that improving and worsening features of TRIZ can be also related to one or more physical variables. Similarly, each physical variable can correspond to one or more TRIZ parameter (e.g. ‘sigma – Stress’ is related to ‘Stress/Pressure’, ‘Strength’ and ‘Reliability’).

Table 7: Entities of Ideation Methods in V.1

Physical effect	a. Name b. Description c. Physical equations (E.g. F = m*a) d. Physical variables (E.g. ‘F – Force’, ‘m – mass’) e. Medium of occurrence (E.g. Solid, Liquid, Gas) f. Physical law involved (E.g. Newton’s law of motion)
Working Principle	a. Name b. Description c. Physical effects involved d. Related physical variables e. Materials (E.g. Steel alloy, Cast iron etc.) f. Graphical representation g. Functions it can fulfill (E.g. Mechanical energy to Electrical energy) h. Biological example (E.g. Translocation in plants is an example for ‘Flow of liquid’)
TRIZ/BioTRIZ	a. Improving feature (E.g. Strength, Reliability etc.) b. Worsening feature (E.g. Weight, Area etc.) c. Inventive principle (E.g. Segmentation, Asymmetry etc.)
Artifacts	a. Name b. Description c. Related functions d. Parent/Child artifact e. Failure (mode, type) – (E.g. Ductile fracture, wear etc.) f. Color g. Physical variables (In OSU design repository, it is mostly dimensions/weight)
Functions	a. Input/output flow variable (E.g. Mechanical energy, Solid material etc.) b. Function verb (E.g. Divide, convert, expand etc.)

In other words, this type of structure for databases created channels to traverse through multiple knowledge bases and give the user a way to look at more relevant information. The different types of inter-relations exploited are discussed in the following subsection.

3.3.1 Database Inter-Relations

Functions are related to physical effects. Flow variables of the function ontology were used to connect to physical effects by the use of physical parameters. As shown in Figure 4, each flow variable has several physical variables associated with it. Based on the relations between the flow variables and the physical variables, functions and physical effects were related.

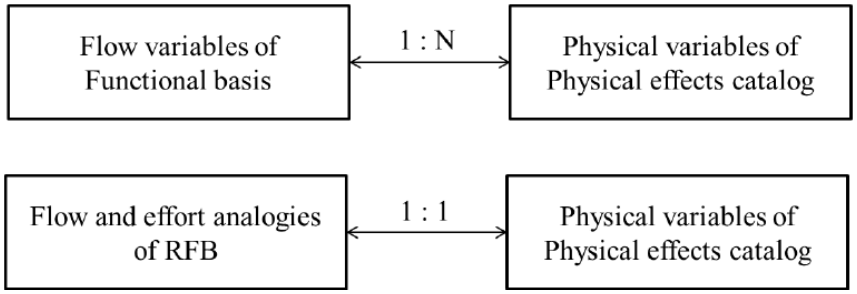
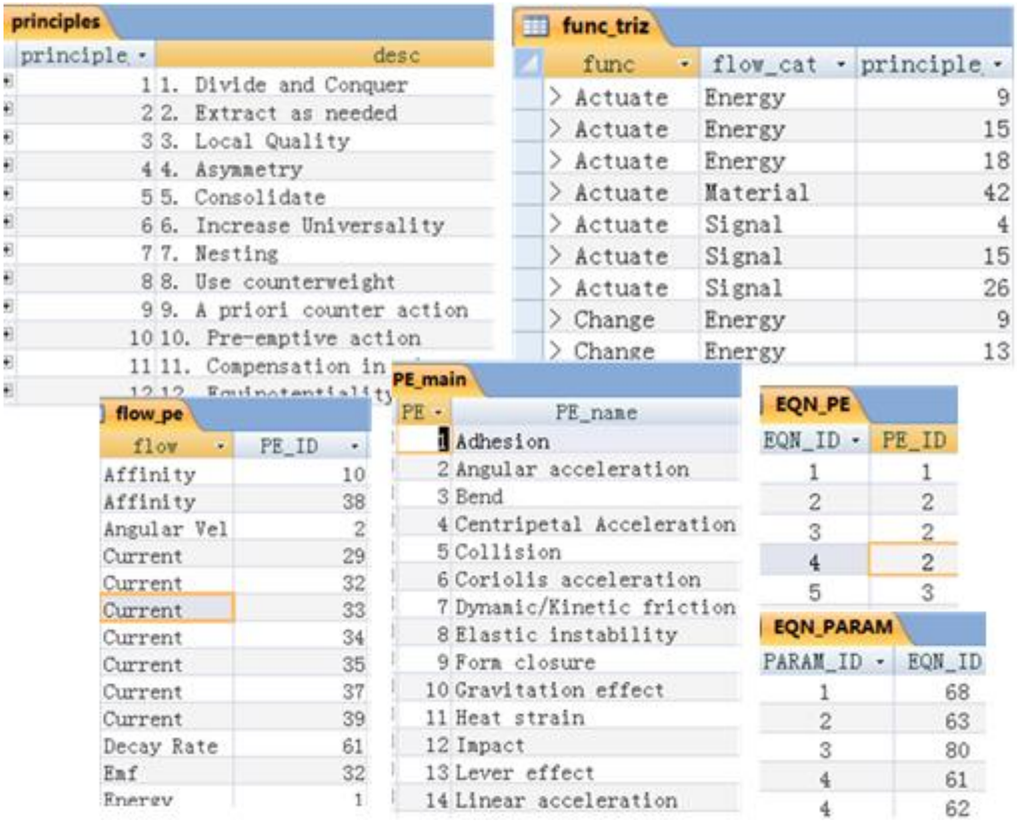

Flow variables of Functional basis
1 : N
Physical variables of Physical effects catalog
Flow and effort analogies of RFB
1 : 1
Physical variables of Physical effects catalog

Figure 4: Relation Between Function Definition and Physical Effects (Mohan M. , 2011)

The design research team in Oregon State University have developed a function based TRIZ (Nix, Sherrett, & Stone, 2011). They explored flow variables of functional basis and physical variables involved in TRIZ principles. Figure 5 indicates the mapping in between TRIZ principles and functional verbs in V.1. Moreover, the physical effects through their parameters had also been mapped to the improving and worsening parameters of TRIZ through flow variables. The 39 characteristics in TRIZ are related to one or more physical effects through parameters.



principles

principle	desc
1	1. Divide and Conquer
2	2. Extract as needed
3	3. Local Quality
4	4. Asymmetry
5	5. Consolidate
6	6. Increase Universality
7	7. Nesting
8	8. Use counterweight
9	9. A priori counter action
10	10. Pre-emptive action
11	11. Compensation in
12	12. Equipotentiality

func_triz

func	flow_cat	principle
Actuate	Energy	9
Actuate	Energy	15
Actuate	Energy	18
Actuate	Material	42
Actuate	Signal	4
Actuate	Signal	15
Actuate	Signal	26
Change	Energy	9
Change	Energy	13

flow_pe

flow	PE_ID
Affinity	10
Affinity	38
Angular Vel	2
Current	29
Current	32
Current	33
Current	34
Current	35
Current	37
Current	39
Decay Rate	61
Emf	32
Energy	1

PE_main

PE	PE_name
1	Adhesion
2	Angular acceleration
3	Bend
4	Centripetal Acceleration
5	Collision
6	Coriolis acceleration
7	Dynamic/Kinetic friction
8	Elastic instability
9	Form closure
10	Gravitation effect
11	Heat strain
12	Impact
13	Lever effect
14	Linear acceleration

EQN_PE

EQN_ID	PE_ID
1	1
2	2
3	2
4	2
5	3

EQN_PARAM

PARAM_ID	EQN_ID
1	68
2	63
3	80
4	61
4	62

Figure 5: Relating Flow Variables to TRIZ with Physical Effects and Parameters

Working principles are associated with related physical variables. And flow variables of functions are also related to physical variables. Hence, physical variables forms the bridge between functions and working principles. Figure 6 explains the relationship between functions and working principles.

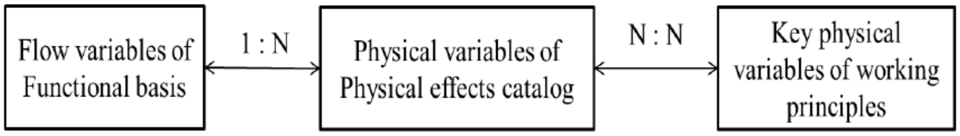

Flow variables of Functional basis
1 : N
Physical variables of Physical effects catalog
N : N
Key physical variables of working principles

Figure 6: Relating Function to Working Principles with Physical Variables

(Mohan M. , 2011)

Figure 7 is the ER diagram which illustrates the database structure used in the holistic ideation system to model different ideation methods. It also demonstrates how different databases are associated with each other and list of their respective attributes.

The ER diagram shows interaction between different database tables in holistic ideation. Energy, material, and signal flow are transformed into relevant physical variables and they flow through different ideation methods in V.1 to form the channels in knowledge base in order to provide the means for traversal.

V.1 was implemented using Matlab while the Microsoft Access tool was used to create and manage databases. As these databases are retained in the later versions of Holistic Ideation tool, these will be discussed later in the implementation chapter of new improved Holistic Ideation tool.

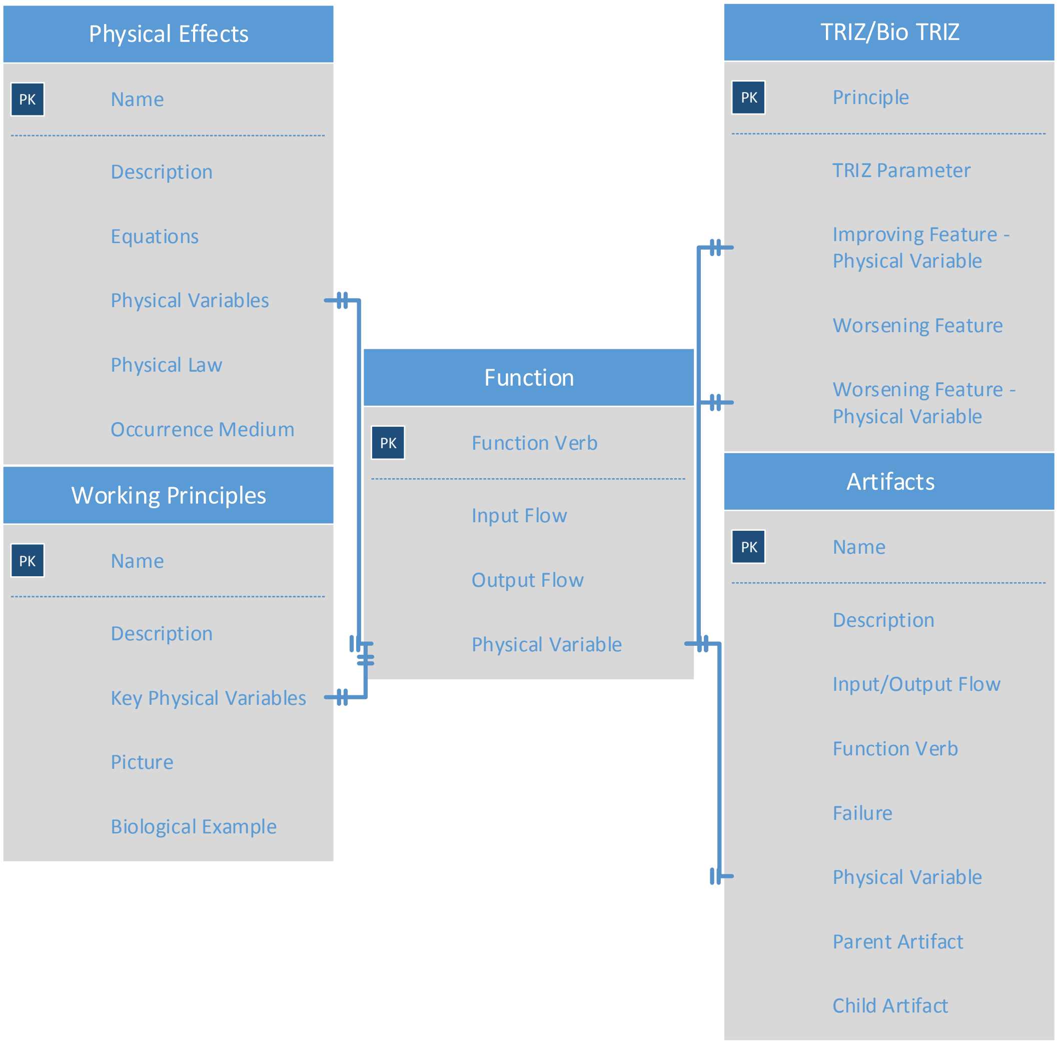
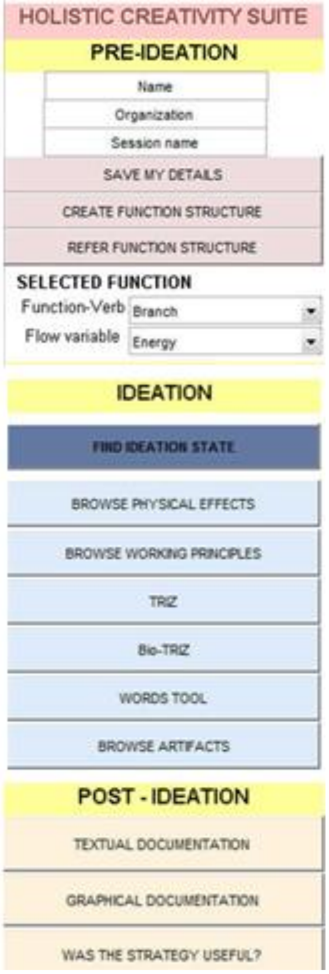


Figure 7: ER Diagram for Holistic Ideation with Experiential Methods

3.4 UI Development for V.1



Function-Verb: Branch

Flow variable: Energy

Figure 8: Main Menu of V.1

V.1 was organized into following three major stages: pre-ideation stage, ideation stage, and post-ideation stage (Figure 8). Pre-ideation includes the following tasks: documenting ideation state, function decomposition. Then the user can move to the ideation stage, which facilitates idea generation for each function. The designer can choose from a variety of ideation strategies based on his/her needs and preferences. In post ideation, solutions are evaluated by the user.

An user can not ideate without a function but in V.1 there was no connection between function decomposition and solutions. So, each tool could be used independently. Because of this when a designer is generating solutions in ideation stage, he/she can also document the ideas. There was no way provided to combine ideas in V.1. These pre-ideation, ideation, and post ideation processes repeat until solutions for all the functions/sub-functions are created. Figure 9 illustrates a generic flow of the general process through pre-ideation stage, ideation stage, and post-ideation stage. These are discussed in following sections.

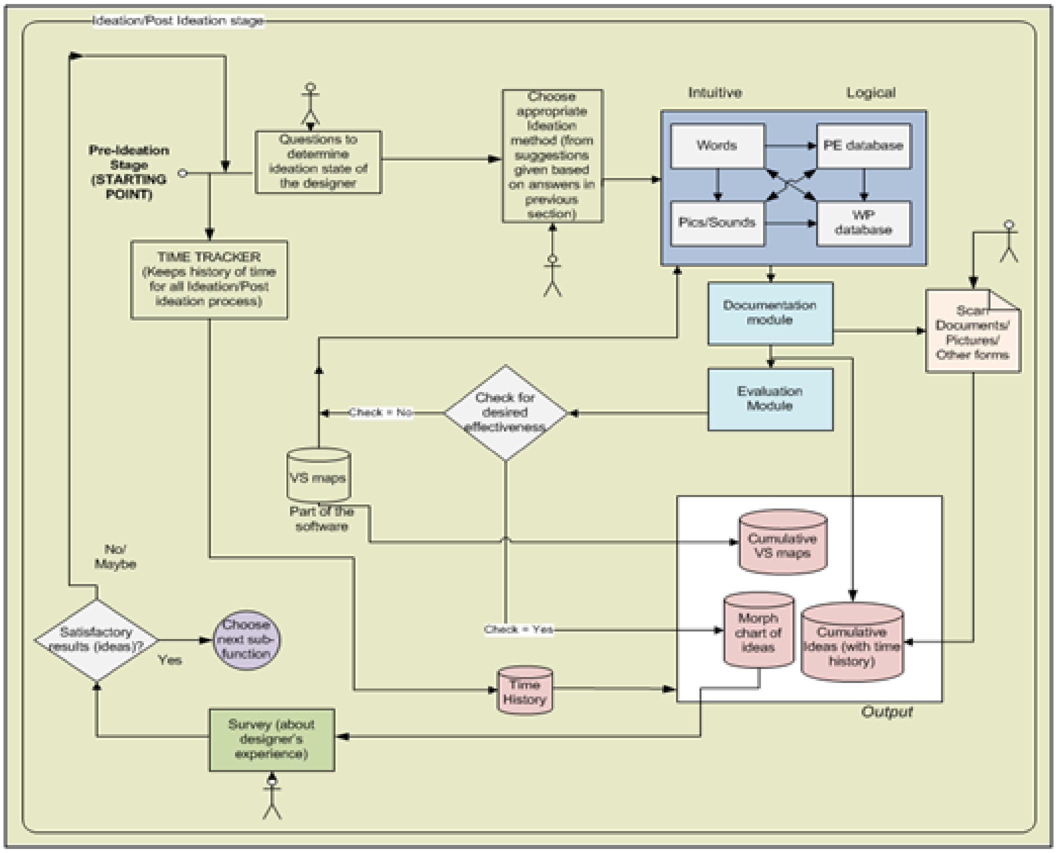
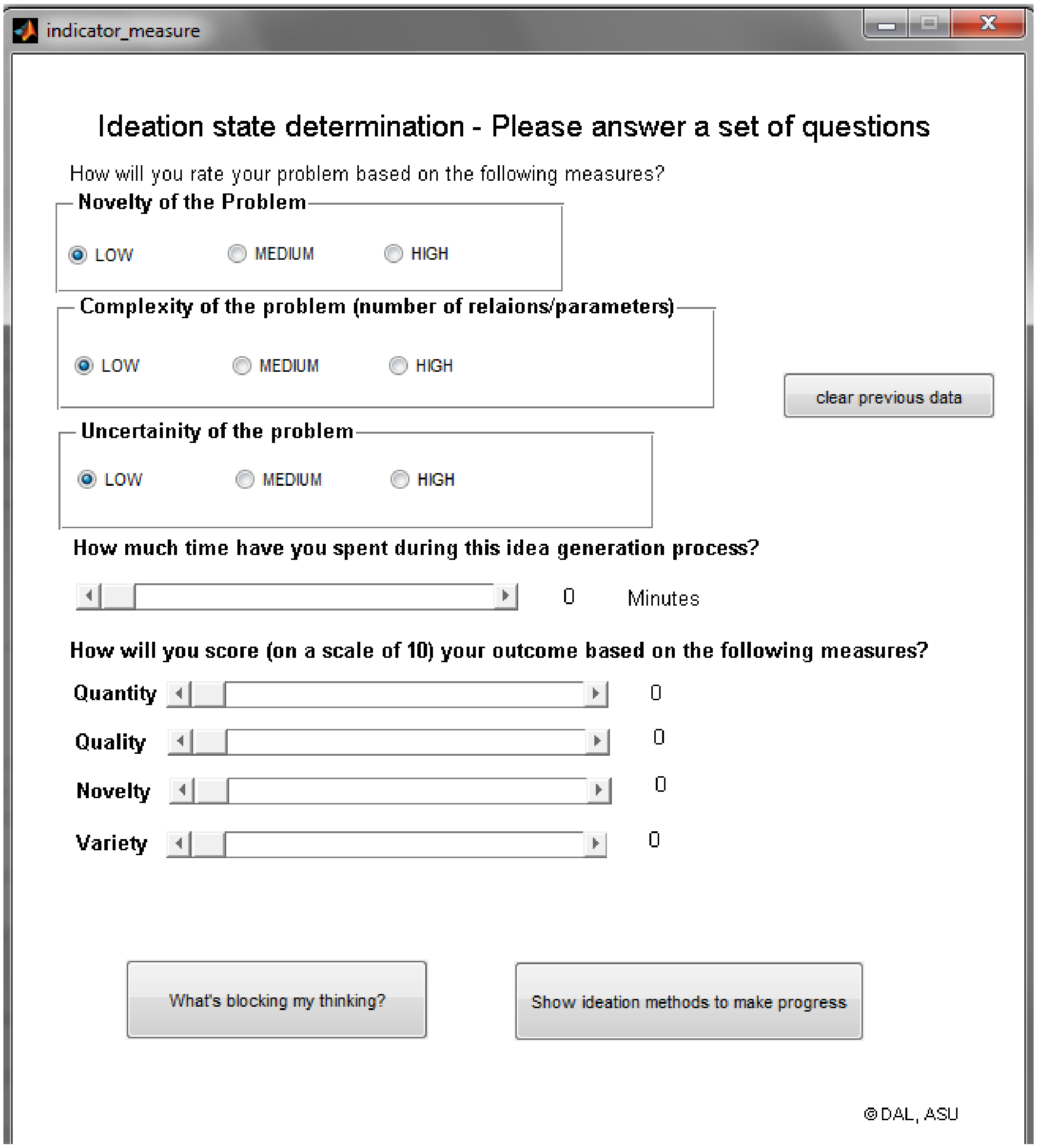

Ideation/Post Ideation stage
Pre-Ideation Stage (STARTING POINT)
Questions to determine ideation state of the designer
Choose appropriate Ideation method (from suggestions given based on answers in previous section)
Intuitive
Logical
Words
PE database
Pics/Sounds
WP database
TIME TRACKER (Keeps history of time for all Ideation/Post ideation process)
Documentation module
Scan Documents/ Pictures/ Other forms
Evaluation Module
Check for desired effectiveness
Check = No
VS maps
Part of the software
Cumulative VS maps
No/ Maybe
Satisfactory results (ideas)?
Yes
Choose next sub-function
Check = Yes
Morph chart of ideas
Cumulative Ideas (with time history)
Time History
Output
Survey (about designer's experience)

Figure 9: V.1 Task Flow (Mohan M. , 2011)

3.4.1 Ideation State Characterization Tool

As a part of Pre-Ideation, the user may document his/her ideation state, which includes the level of satisfaction with current solutions. Characteristics of the outcomes are defined by the effectiveness measures (Shah J. J., 1998) (Shah, Smith, & Vargas-Hernandez, 2003). These effectiveness measures are: Quantity, Quality, Novelty and Variety. When designers enter values for characterization measures, the corresponding ideation blocks are found by the tool. Some time in the future this information will be used to suggest ideation strategies, appropriate ideation methods. Now the designer has the option of freely choosing any ideation method according to his/her personal preference. Figure 10 shows the user interface for ideation state characterization. For now this data is being collected for future use.



indicator_measure

Ideation state determination - Please answer a set of questions

How will you rate your problem based on the following measures?

Novelty of the Problem

● LOW ○ MEDIUM ○ HIGH

Complexity of the problem (number of relaions/parameters)

● LOW ○ MEDIUM ○ HIGH

Uncertainity of the problem

● LOW ○ MEDIUM ○ HIGH

How much time have you spent during this idea generation process?

0 Minutes

How will you score (on a scale of 10) your outcome based on the following measures?

Quantity 0

Quality 0

Novelty 0

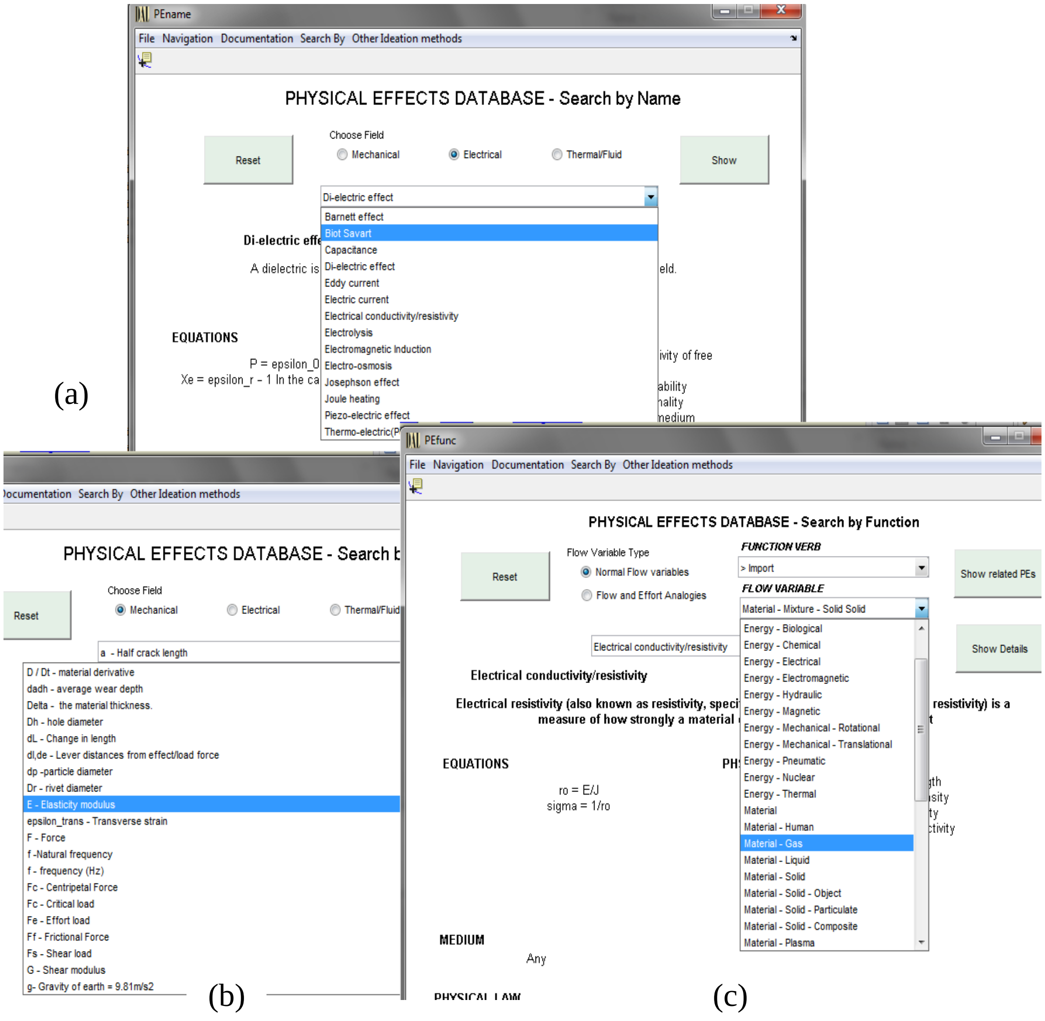
Variety 0

clear previous data
What's blocking my thinking?
Show ideation methods to make progress
© DAL, ASU

Figure 10: Ideation Characterization Tool

3.4.2 Idea Generation Tools

In the ideation stage, different ideation methods are provided to the user through different and specific user interfaces designed for each of the ideation methods. To start with, the PE database can be searched by the physical effect name, physical parameter or function. Figure 11 shows the user interface for physical effects.


PEname
File Navigation Documentation Search By Other Ideation methods
PHYSICAL EFFECTS DATABASE - Search by Name
Reset
Choose Field
○ Mechanical
● Electrical
○ Thermal/Fluid
Show
EQUATIONS
(a)
PEfunc
File Navigation Documentation Search By Other Ideation methods
PHYSICAL EFFECTS DATABASE - Search by Function
Reset
Flow Variable Type
● Normal Flow variables
○ Flow and Effort Analogies
FUNCTION VERB
FLOW VARIABLE
Show related PEs
Show Details
EQUATIONS
MEDIUM
PHYSICAL LAW
(c)
Documentation Search By Other Ideation methods
PHYSICAL EFFECTS DATABASE - Search b
Reset
Choose Field
● Mechanical
○ Electrical
○ Thermal/Fluid
(b)

(a)

Di-electric effect

Barnett effect
Biot Savart
Capacitance
Di-electric effect
Eddy current
Electric current
Electrical conductivity/resistivity
Electrolysis
Electromagnetic Induction
Electro-osmosis
Josephson effect
Joule heating
Piezo-electric effect
Thermo-electric(P

Di-electric effe

A dielectric is eld.

P = epsilon_0
Xe = epsilon_r - 1 In the ca

ivity of free
ability
nality
nedium

(b)

a - Half crack length

D / Dt - material derivative
dadh - average wear depth
Delta - the material thickness.
Dh - hole diameter
dL - Change in length
dl,de - Lever distances from effect/load force
dp -particle diameter
Dr - rivet diameter
E - Elasticity modulus
epsilon_trans - Transverse strain
F - Force
f -Natural frequency
f - frequency (Hz)
Fc - Centripetal Force
Fc - Critical load
Fe - Effort load
Ff - Frictional Force
Fs - Shear load
G - Shear modulus
g- Gravity of earth = 9.81m/s2

(c)

FUNCTION VERB: > Import

FLOW VARIABLE: Material - Mixture - Solid Solid

Energy - Biological
Energy - Chemical
Energy - Electrical
Energy - Electromagnetic
Energy - Hydraulic
Energy - Magnetic
Energy - Mechanical - Rotational
Energy - Mechanical - Translational
Energy - Pneumatic
Energy - Nuclear
Energy - Thermal
Material
Material - Human
Material - Gas
Material - Liquid
Material - Solid
Material - Solid - Object
Material - Solid - Particulate
Material - Solid - Composite
Material - Plasma

Electrical conductivity/resistivity

Electrical conductivity/resistivity

Electrical resistivity (also known as resistivity, speci resistivity) is a measure of how strongly a material t

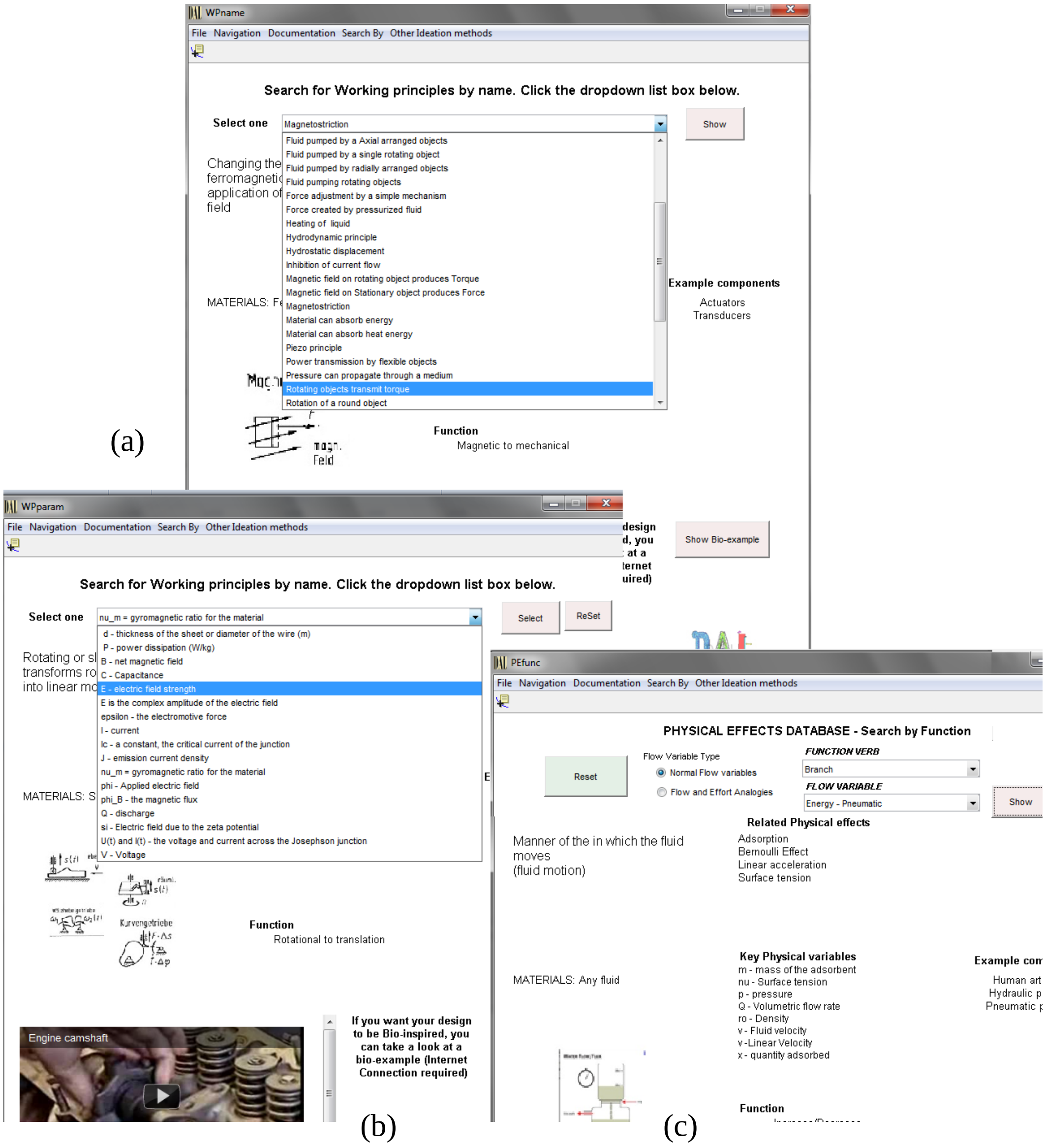
ro = E/J
sigma = 1/ro

PH
gth
sity
ty
ctivity

Any

Figure 11: Physical Effects Search by Name (a), by Physical Variables (b) and by Function (c)

The WP database can be searched with respect to name, physical variables and function. Figure 12 shows the user interface for working principles.


WPname

Search for Working principles by name. Click the dropdown list box below.

Select one: Magnetostriction

Changing the ferromagneti application of field

MATERIALS: F

Example components
Actuators
Transducers

Function
Magnetic to mechanical

WPparam

Search for Working principles by name. Click the dropdown list box below.

Select one: nu_m = gyromagnetic ratio for the material

Rotating or sl transforms ro into linear mo

MATERIALS: S

Function
Rotational to translation

Engine camshaft

If you want your design to be Bio-inspired, you can take a look at a bio-example (Internet Connection required)

PEfunc

PHYSICAL EFFECTS DATABASE - Search by Function

Flow Variable Type: ● Normal Flow variables ○ Flow and Effort Analogies

FUNCTION VERB: Branch

FLOW VARIABLE: Energy - Pneumatic

Manner of the in which the fluid moves (fluid motion)

Related Physical effects
Adsorption
Bernoulli Effect
Linear acceleration
Surface tension

MATERIALS: Any fluid

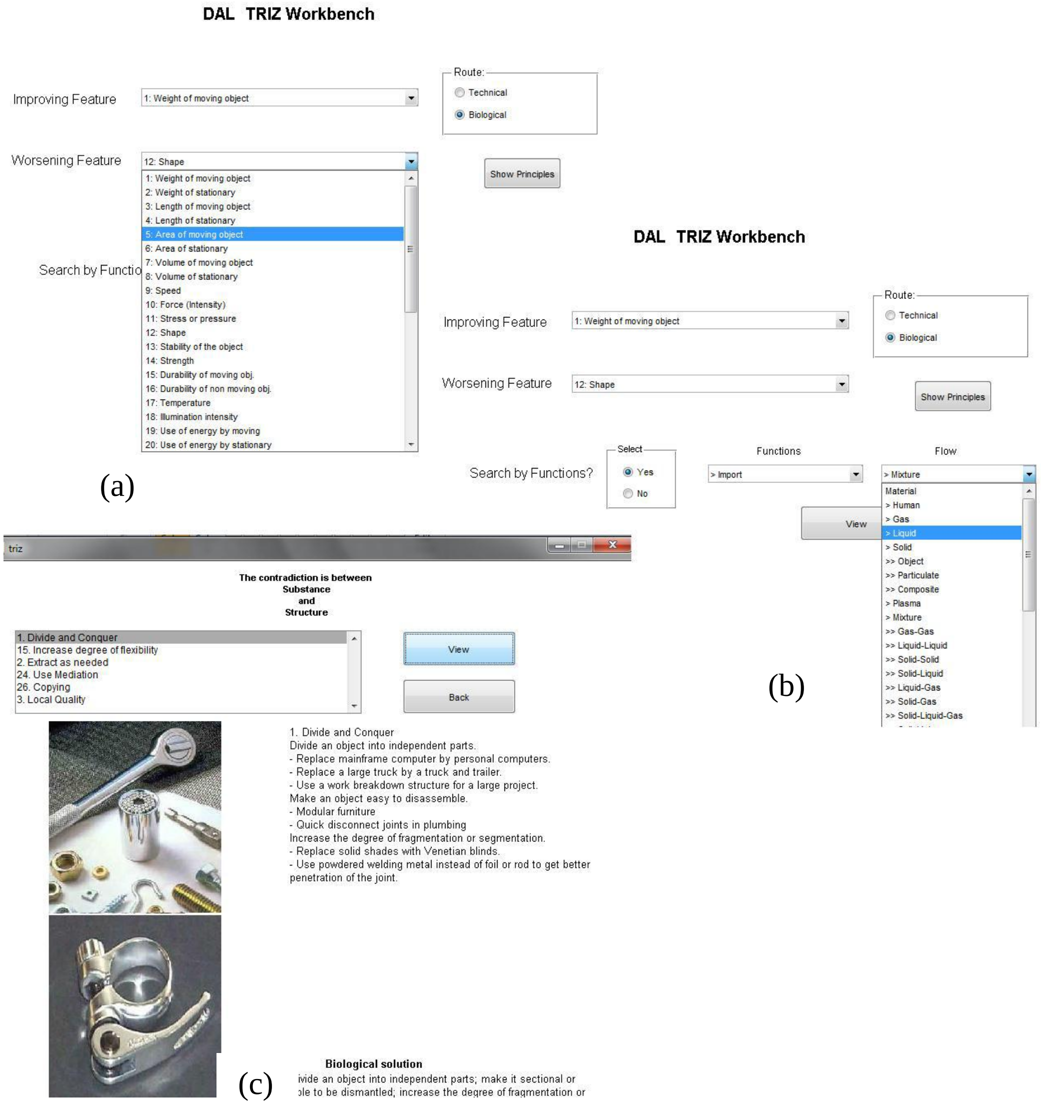
Key Physical variables
m - mass of the adsorbent
nu - Surface tension
p - pressure
Q - Volumetric flow rate
ro - Density
v - Fluid velocity
v -Linear Velocity
x - quantity adsorbed

Example com
Human art
Hydraulic p
Pneumatic p

Function
(a)
File Navigation Documentation Search By Other Ideation methods
Show
Fluid pumped by a Axial arranged objects
Fluid pumped by a single rotating object
Fluid pumped by radially arranged objects
Fluid pumping rotating objects
Force adjustment by a simple mechanism
Force created by pressurized fluid
Heating of liquid
Hydrodynamic principle
Hydrostatic displacement
Inhibition of current flow
Magnetic field on rotating object produces Torque
Magnetic field on Stationary object produces Force
Magnetostriction
Material can absorb energy
Material can absorb heat energy
Piezo principle
Power transmission by flexible objects
Pressure can propagate through a medium
Rotating objects transmit torque
Rotation of a round object
magn. field
design d, you at a ternet uired)
Show Bio-example
(b)
File Navigation Documentation Search By Other Ideation methods
Select
ReSet
d - thickness of the sheet or diameter of the wire (m)
P - power dissipation (W/kg)
B - net magnetic field
C - Capacitance
E - electric field strength
E is the complex amplitude of the electric field
epsilon - the electromotive force
I - current
Ic - a constant, the critical current of the junction
J - emission current density
nu_m = gyromagnetic ratio for the material
phi - Applied electric field
phi_B - the magnetic flux
Q - discharge
si - Electric field due to the zeta potential
U(t) and I(t) - the voltage and current across the Josephson junction
V - Voltage
Kurvengetriebe
(c)
File Navigation Documentation Search By Other Ideation methods
Reset
Show

Figure 12: Working Principles Search by Name (a), by Physical Parameters (b) and Function (c)

TRIZ/BioTRIZ had been implemented based on the database schema described previously. Figure 13 shows the user interface for TRIZ.



DAL TRIZ Workbench

Improving Feature: 1: Weight of moving object

Worsening Feature: 12: Shape

Route: ○ Technical ● Biological

Show Principles
Search by Functio
1: Weight of moving object
2: Weight of stationary
3: Length of moving object
4: Length of stationary
5: Area of moving object
6: Area of stationary
7: Volume of moving object
8: Volume of stationary
9: Speed
10: Force (Intensity)
11: Stress or pressure
12: Shape
13: Stability of the object
14: Strength
15: Durability of moving obj.
16: Durability of non moving obj.
17: Temperature
18: Illumination intensity
19: Use of energy by moving
20: Use of energy by stationary
(a)

DAL TRIZ Workbench

Improving Feature: 1: Weight of moving object

Worsening Feature: 12: Shape

Route: ○ Technical ● Biological

Search by Functions? Select: ● Yes ○ No

Functions: > Import

Flow: > Mixture

The contradiction is between
Substance
and
Structure

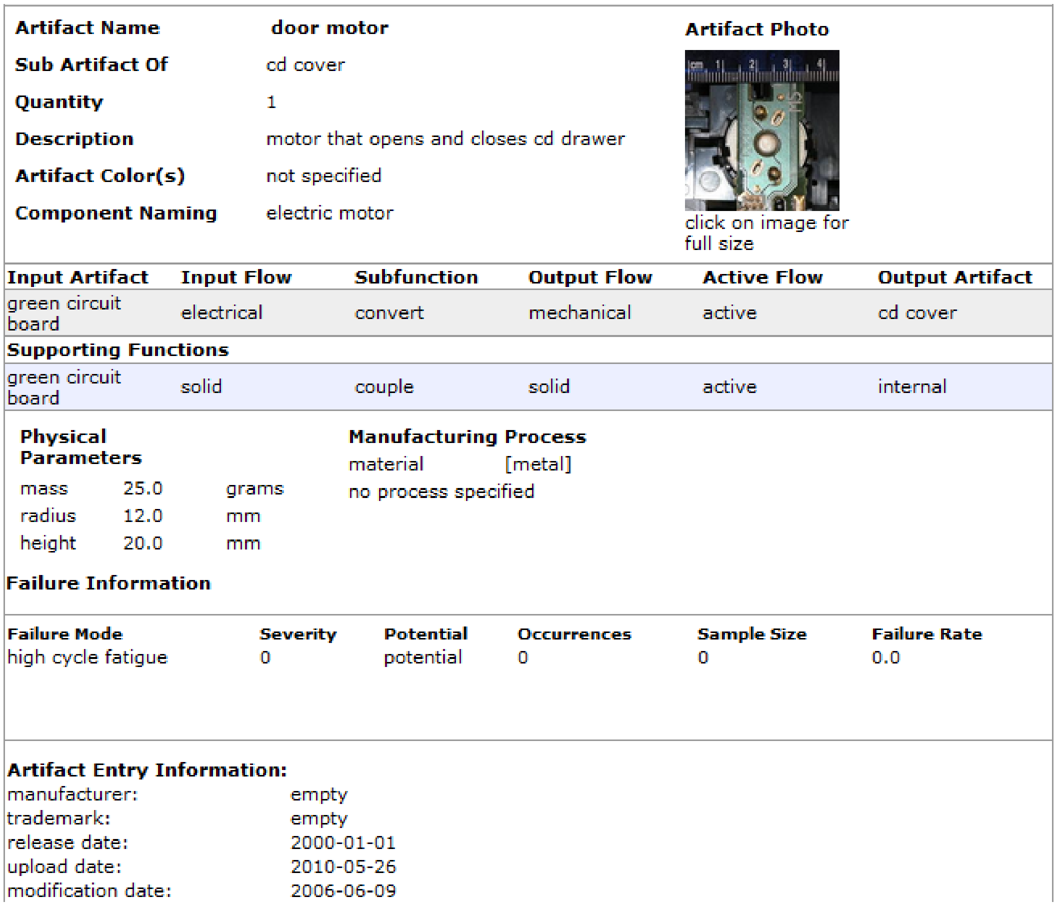
1. Divide and Conquer
Divide an object into independent parts.
- Replace mainframe computer by personal computers.
- Replace a large truck by a truck and trailer.
- Use a work breakdown structure for a large project.
Make an object easy to disassemble.
- Modular furniture
- Quick disconnect joints in plumbing
Increase the degree of fragmentation or segmentation.
- Replace solid shades with Venetian blinds.
- Use powdered welding metal instead of foil or rod to get better penetration of the joint.

Biological solution

ivide an object into independent parts; make it sectional or
ɔle to be dismantled; increase the degree of fragmentation or

Figure 13: TRIZ Normal search (a), Search by Function (b) and Example Window (c)

Artifact search can be done based on names and based on functions. The UI for artifact search tool is shown below in Figure 14.



Artifact Name	**door motor**
Sub Artifact Of	cd cover
Quantity	1
Description	motor that opens and closes cd drawer
Artifact Color(s)	not specified
Component Naming	electric motor

Artifact Photo

click on image for full size

Input Artifact	Input Flow	Subfunction	Output Flow	Active Flow	Output Artifact
green circuit board	electrical	convert	mechanical	active	cd cover
Supporting Functions					
green circuit board	solid	couple	solid	active	internal

Physical Parameters

mass	25.0	grams
radius	12.0	mm
height	20.0	mm

Manufacturing Process

material [metal]

no process specified

Failure Information

Failure Mode	Severity	Potential	Occurrences	Sample Size	Failure Rate
high cycle fatigue	0	potential	0	0	0.0

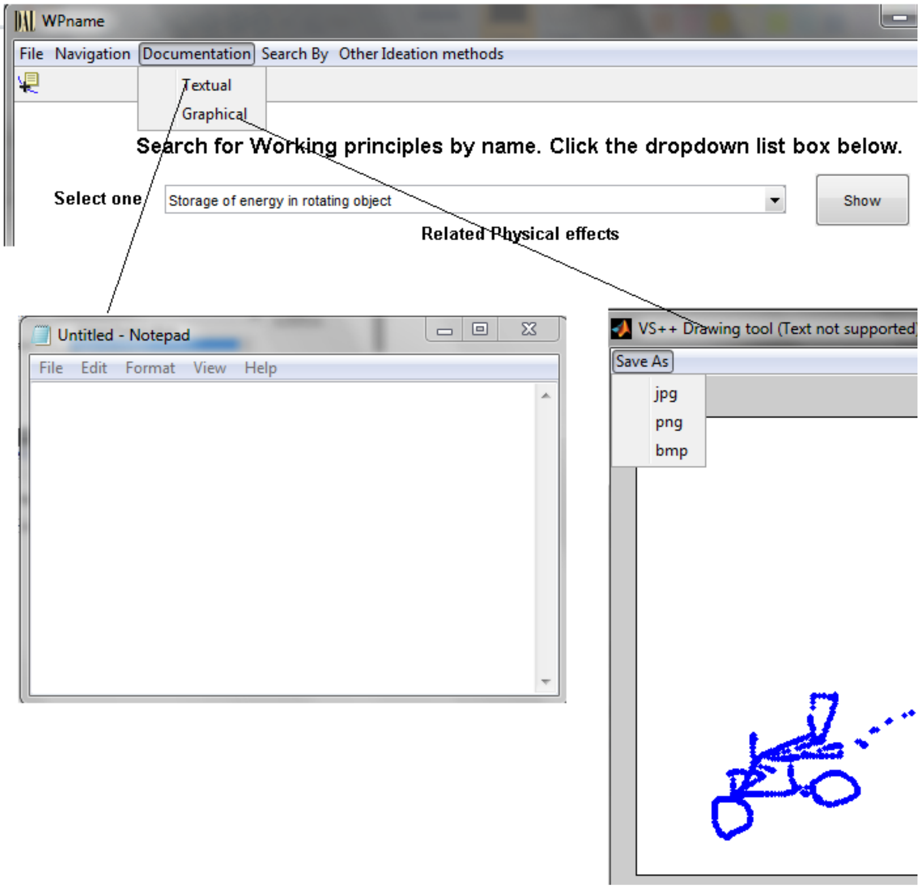
Artifact Entry Information:

manufacturer:	empty
trademark:	empty
release date:	2000-01-01
upload date:	2010-05-26
modification date:	2006-06-09

Figure 14: UI for Artifacts Search Tool

3.4.3 Post-Ideation Tools

The post-ideation UI is used for documentation of ideas generated in the conceptual design stage. User has to document the ideas manually. As shown in Figure 15, two different ways, textual and graphical documentation, are allowed. Textual documentation can be made in a text pad. On the other hand, graphical documentation is done by sketching in an in-built graphical editor.

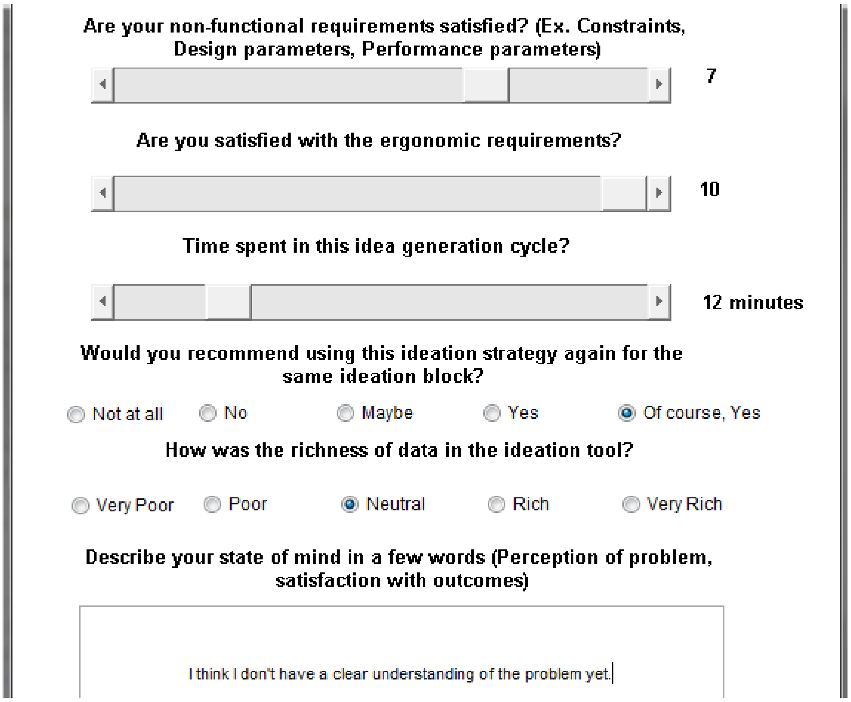


Search for Working principles by name. Click the dropdown list box below.

Storage of energy in rotating object

Figure 15: Textual and Graphical Documentation

The Post-ideation module also has a survey tool, which is shown in Figure 16. It is designed for collecting information about designers' feedback for the experience provided by the ideation tool. This survey tool collects information about how well the designers' functions were satisfied and also some details about the effectiveness of the ideation strategy used.



Are your non-functional requirements satisfied? (Ex. Constraints, Design parameters, Performance parameters)

7

Are you satisfied with the ergonomic requirements?

10

Time spent in this idea generation cycle?

12 minutes

Would you recommend using this ideation strategy again for the same ideation block?

○ Not at all ○ No ○ Maybe ○ Yes ● Of course, Yes

How was the richness of data in the ideation tool?

○ Very Poor ○ Poor ● Neutral ○ Rich ○ Very Rich

Describe your state of mind in a few words (Perception of problem, satisfaction with outcomes)

I think I don't have a clear understanding of the problem yet.

Figure 16: Satisfaction Survey to Collect Details about Effectiveness of Ideation Strategies

3.5 V.1 User Studies

User studies were conducted to test the effectiveness of V.1 (Mohan M. , 2011). The major problem mentioned in the feedback was regarding the UI of V.1. The following list shows a summary of the major issues reported by users:

1. Some of the users did not want to traverse through databases using physical variables because they thought it would block their creativity.

2. Some designers preferred dealing with multiple functions at a time which is not allowed by the tool.

3. As the list of ideation methods was given in a particular order in the main window of holistic ideation tool, users did tend to use them in a particular sequence (as shown to them in user interface).

4. Some users felt that it would be difficult to develop ideas from physical effects since they thought it would be hard for them to comprehend different effects and develop ideas from those.

5. Some users mentioned that it would be easier if FunctionCAD had the capability to write down artifacts or other ideas beside functions so that it would be easier to follow.

6. The functional verb/flow variables list was too long and difficult to search. The designers felt it would be better if the functions were shown at primary, secondary and tertiary levels.

7. Some users wanted stimulus from pictures and were looking for analogy all through the ideation stage which system could not provide. In short, the designers needed intuitive strategies.

8. Some users did not want to characterize their ideation state because they wanted to spend time to look at more information.

9. Some users mentioned that they should be allowed to take the survey at the end. If the survey is given in between, the user might lose focus and would not be able to generate effective ideas.

3.6 Shortcomings Of V.1

In this section V.1 shortcomings are grouped under four major categories: Lack of Intuitive Ideation Methods, Lack of Integration, Inflexible Process, and Organizational Framework Issues.

3.6.1 Lack of Intuitive Ideation Methods

V.1 was dependent on the logical ideation strategies. Solutions found can have high quality but low novelty. To understand more about all other strategies it is required to have a greater variety of intuitive and logical methods embedded in the holistic ideation tool.

3.6.2 Lack of Integration

As we add more intuitive and logical ideation strategies via respective ideation methods into the framework we need to find an effective way to integrate the new set of ideation methods with previous ones.

3.6.3 Inflexible Process

V.1 required the user to start with a functional decomposition of the problem in the pre-ideation stage. This limitation on starting point might be a constraint on creativity. The framework should provide a flexible decomposition which could support any starting point, i.e., requirements, features, function decomposition etc.

3.6.4 Organizational Framework Issues

There was no formal framework to organize and relate all the ideas explored. As conceptual design is a process where the designer needs to iterate through different steps many a times, he/she should also need to have some mechanism to view of all the ideas explored and any relationships between them. The framework should provide an organizational structure which can give a broader perspective of the solution space.

Basically all the included methods formed a loose federation with little inter-activity.

CHAPTER 4

HOLISTIC IDEATION TOOL V.2 REVIEW

To overcome the shortcomings from V.1 and improve the Holistic Ideation tool, a second version (V.2) was developed and implemented by Chen (Chen, 2012). The intuitive ideation methods were added to the tool. Moreover, morphological charts were improved and implemented in V.2 to organize solutions effectively. In this chapter, we will review V.2 with respect to the ideation strategies embedded, ideation methods implemented, organizational framework, implementation, UI, user feedback and shortcomings in V.2. This will set the stage for my own work.

4.1 Background

V.2 added intuitive ideation methods, which will be discussed in this section. They include: Reframing, Analogical reasoning, Restructuring, Random connection and Deliberate connection (Chen, 2012).

4.1.1 Reframing

In design thinking, one strategy to find new solutions is to change the problem formulations. Reframing involves change in functions, objectives, specification and constraints. In order to understand reframing, first we need to understand what the word "frame" means in the context of engineering design.

In engineering design, a frame could mean a combination of three things: a certain perception of a problem or situation, the adoption of a terminology, and a way of reasoning that allows the designer to think and develop a set of possible actions (Schon, 1984). In the Conceptual design, “Reframing” can be seen as a change in problem formulation. Once a situation is experienced by considering it outside its original frame, changing the viewpoint could be involved in reframing. In other words, reframing is a situation in a different frame which could fit the primary situation equally well or even better, thereby altering its partial or entire meaning (Weakland, Fisch, & Watzlawick).

4.1.2 Restructuring

Restructuring is the transformation from one representational form to another at the same relative abstraction level, while preserving the subject system’s external behavior (functionality and semantics) (Chikofsky & James, 1990). There is a difference between reframing and restructuring: restructuring involves changing the relationship between components (e.g. function structure) defined by problems, and reframing involves change in functions, objectives, specification and constraints.

Previous research has shown that the creative potential of a design is related with the problem restructuring. Some studies (Akin, 1994) (Duncker, 1945) indicated that in order to be productive in engineering problem solving, one of the actions is to restructure the problem.

The V.2 allows the designer to sub-divide a problem into smaller sub-problems. After generating sub-solutions for each of the sub-problems, the designer can combine sub-solutions to synthesis full solutions.

4.1.3 Analogical Reasoning

Analogical reasoning is a cognitive process of transferring information or meaning from a particular subject (the analogue or source) to another particular subject (the target). The fundamental properties of analogical reasoning are the relational similarity and structural similarity. In the same domain, analogical reasoning maps the causal structure between the source objects (i.e. products, designs) to the target design problem which is being solved. However, if enabled by a supporting system of relations or representations of situations, analogical reasoning could be a mapping of knowledge from different domains (Hey, Linsey, Agogino, & Wood, 2012).

Analogical reasoning is an important ideation strategy in the field of engineering design. One possible example is Holyoak's research (Gick & Holyoak, 1983), under the aid of similarity, they demonstrated that analogy can help people to solve a difficult problem in engineering design.

4.1.4 Random & Deliberate Connection

One way to explore the solution space is by combining the elements of existing solutions randomly or deliberately. Random connection means associating one concept/object/idea/solution with another one without thinking about the relation between them at all. This connection could be conceptual, geometrical or topological.

Random connections are used to break designers out of the fixated response by stimulating divergent thinking (Grossman & Wiseman, 1993) (Parnes, 1987). A deliberate connection, on the other hand, means to link two concepts/objects/ideas/solutions together by careful selection rather than randomly connecting them.

The purpose of random and the deliberate connections is to facilitate a possibility relating two unrelated concepts which could lead to new ideas. Random connection and deliberate connection finds commonalities between two or more seemingly unrelated concepts to generate creativity. In this process, by making the association between two concepts/objects/ideas/solutions, the designer could discover that how he/she can open mind to a new and unexpected idea. He/she can achieve that by thinking about the attributes and descriptions of seemingly unrelated objects and images. For example: the invention of rollerblade. It is apparent that some designer conceptually linked a pair of roller and ice skates and in-turn invented a new product (Chen, 2012).

4.1.5 Provocative Stimuli

Provocative stimuli are any external stimuli for the designers by providing a change of reference (DeBono, 1984) (Osborn, 1979). They may represent the information and knowledge in any form, such as function, behavior or component. Textual information and graphical information could also act as a provocative stimuli.

A provocative stimulus is defined as external inputs, which act as a catalyst in idea generation (Shah, Vargas, Summers, & Kulkarni, 2001).

Provocative stimuli may help designers combine multiple concepts in unexpected ways. Also, it was found that while many design modifications were misinterpreted from the original intent, the misinterpretations served as launching pads for new design solutions (Shah, Vargas, Summers, & Kulkarni, 2001). Research on the components of C-Sketch was conducted at the DAL (Shah J. J., 1998) which indicated that with the help of provocative stimuli, which is an ideation strategy embedded in C-Sketch ideation method, designers developed new concepts by combining two or more concepts in unexpected ways. In C-Sketch the designer was excited by provocative stimuli by looking at the sketches given to him from a previous designer. These provocative stimuli provided the designers with new solution directions and new frames of reference from the previous designers.

4.2 Intuitive Ideation Methods

4.2.1 WordNet

V.2 implemented the Wordnet database to facilitate reframing. Fundamentally, WordNet (WordNet, Princeton University) is a network of meaningfully related words and can be navigated through the web browser. In WordNet, Synsets are interlinked using conceptual-semantic and lexical relations. It can be used for searching alternative ways to formulate the problem. In short, WordNet is a very large lexical database of English words such as nouns, verbs, adjectives and adverbs which are grouped into sets of cognitive synonyms (Synsets). Each Synset expresses a distinct concept.

WordNet superficially resembles a thesaurus, in a way that it groups words together based on their meanings. However, there are some important distinctions. First, WordNet interlinks not just word forms - strings of letters - but specific senses of words. As a result of this interlinking, words that are found in close proximity to one another in the network are semantically disambiguated.

The main relation among words in WordNet is synonymy. For example, the relation between words shut and close or car and automobile. WordNet has 117,798 Synsets which are linked to other Synsets by means of a small number of "conceptual relations". We will discuss these relations one by one below.

1. Hypernomy/Hyponomy :

The most frequently encoded relation among Synsets is the super-subordinate relation (also called Hypernymy, Hyponymy). This relation links general Synsets like {furniture, piece of furniture} to specific ones like {bed} and {bunk-bed}. Thus, WordNet states that the category furniture includes bed, which in turn includes bunk-bed; conversely, concepts like bed and bunkbed make up the category furniture. All noun hierarchies ultimately go up the root node {entity}. Hyponymy relation is transitive: if an armchair is a kind of chair, and if a chair is a kind of furniture, then an armchair is a kind of furniture.

2. Meronymy :

Meronymy can be seen as the part-whole relation held between Synsets like {chair} and {back, backrest}, {seat} and {leg}. Parts are inherited from their super-ordinates, i.e., if a chair has legs, then an armchair has legs as well. Parts are not inherited “upward” as they may be a characteristic of specific kinds of things rather than the class as a whole, i.e., chairs and kinds of chairs have legs, but not all kinds of furniture have legs.

Verb Synsets are arranged into hierarchies as well; verbs towards the bottom of the trees (Troponyms) express increasingly specific manners characterizing an event, as in {communicate}-{talk}-{whisper}. The specific manner expressed depends on the semantic field; volume (as in the example above) is just one dimension along which verbs can be elaborated. Others are speed (move-jog-run) or intensity of emotion (like-love-idolize). Verbs describing events that necessarily and unidirectionally entail one another are linked: {buy}-{pay}, {succeed}-{try}, {show}-{see}, etc.

3. Antonomy :

Adjectives are organized under Antonymy. Words which are direct antonyms like up-down, wet-dry, etc reflect the strong semantic contrast of their members. Each of these polar adjectives are also linked to a number of “semantically similar” adjectives. For example, dry is linked to parched, arid and bone etc. Semantically similar adjectives are “indirect antonyms” of the central member of the opposite pole. Relational adjectives ("Pertainyms") point to the nouns they are derived from (criminal-crime). Table 8 shows the wordnet database statistics.

Table 8: WordNet 3.0 Database Statistics

	Unique Strings	Synsets	Word-Sense Pairs
Nouns	117798	82115	146312
Verbs	11529	13767	25047
Adjectives	21479	18156	30002
Adverbs	4481	3621	5580
Totals	155287	117659	206941

Woednet can be used for reframing in V.2. Reframing through Wordnet can work as long as there is some text that can be used to search related information. All the information in solution space or problem space can be used as an input to find alternative description of problems and solutions.

4.2.2 Relational Algorithm

This is another method to facilitate problem reformulation. The basic idea behind the Relational-Algorithm method is to take two phrases of the problem statement and insert one “relation” word to reframe a new perspective of the problem. In Table 9, these 37 relation words are listed. In V.2 the users can experiment with constructing new phrases to redefine the original statement.

Table 9: Relational Algorithm Words

Above	Among	By	Near	Toward	Upon
About	Because	Beyond	Into	Over	Up
Across	Behind	Down	In	Through	When
After	Below	From	Off	Throughout	While
Against	Beneath	During	On	Till	Within
Along	Between	Except	Opposite	Under	With
Without					

4.2.3 Word Diamond (Polygon)

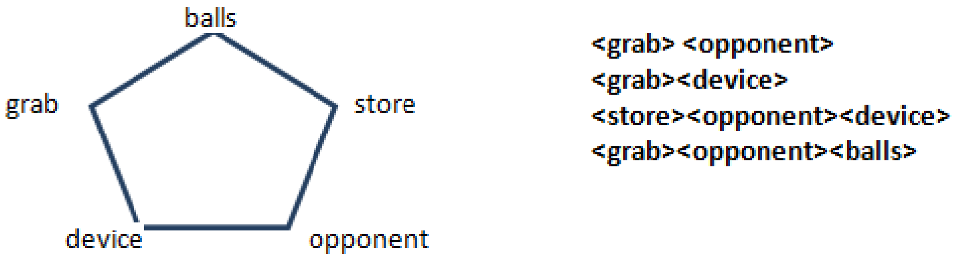
Yet another technique for reformulation is Word Diamond. It helps towards transforming the problem statements and discover new possibilities. The original word diamond method suggests that the designer should select 4 keywords from the problem and arrange them at the vertices of a “diamond”. V.2 has modified this to allow the users to select any number of words. It is good to include both verbs and nouns. The selected words then gets combined at random to see if new functions are revealed.

For example:

Problem - Design a device which can grab and store more golf balls than the opponents.

Key words: balls, store, grab, device, opponent

First, these 5 words are linked as a pentagon (Figure 17). And then some words are picked randomly and put in a random order. The new generated problem statements may lead the designer to new directions. V.2 provides you a platform to make use of the word diamond method.



<grab> <opponent>
<grab><device>
<store><opponent><device>
<grab><opponent><balls>

Figure 17: One Example of Word Diamond

4.3 Organizational Framework

One of the important changes in V.2 was the new framework to organize ideas and navigate through ideation methods. In V.1, only the ideation methods to find ideas were included but an effort towards making a framework to store ideas was not done. In V.2 one of the tasks was to include intuitive ideation strategies such as reframing and restructuring, which help designers to facilitate creativity. These intuitive ideation strategies are discussed in above sections. Implementing these intuitive strategies is not as straight forward as implementing logical ideation methods. It was evident quite clearly that if you want to have intuitive strategies like reframing and restructuring embedded in the tool, first of all the tool needed to give the user a structure to organize ideas. He/she then can use this structure to restructure and combine his ideas. And using this framework he/she can reframe the problem at hand. Thus the new framework was needed to provide the above mentioned qualities. To do that, it needed to have following capabilities:

1. Designers should be allowed to navigate through different ideation methods at any time.
2. While navigating through ideation methods designers should be able to navigate and relate multiple ideas generated during the process.
3. Designers should be given maximum freedom to decompose and formulate the problem; avoid possible constraint which could arise due to the organizational framework.
4. Designers should be able to store solutions in any way they pleases to store.
5. Multiple framing of a single solution should be allowed to coexist.

Morphological Chart is the basis of organizational structure used in V.2. It forms the foundation of organizing the ideas and let designer navigate intuitive and logical ideation methods supported by the tool.

4.3.1 Morphological Chart

The root of the term morphology comes from ancient Greek, i.e., morphe which means shape or form. The general definition of morphology is "the study of form or pattern", i.e., the shape and arrangement of parts of an object, and how these "conform" to create a whole. The "objects" in question can be physical objects (e.g. an organism, geography or ecology), social objects (an organization or other social system) or mental objects (e.g. linguistic forms, concepts or systems of ideas). In contemporary scientific world the term morphology is associated with a number of scientific disciplines where formal structure is a central issue. For example, in biology it deals with the form and structure of organisms, while in geology it deals with the characteristics, configuration and evolution of rocks and land forms. Certainly the presence of this idea is ubiquitous.

In engineering design, morphological charts were first used by Zwicky (Zwicky & Wilson, 1967). It consists of decomposing a problem into sub-problems for design ideation, generating all solutions to each sub problem independently and then combining the sub-solutions randomly to obtain complete solutions. All this information is then stored in a chart called the Morphological Chart. This chart has rows to represent each sub-problem and in each row there are one or more set of sub-solutions to that problem. Thus (i, j) in the chart is the j'th sub-solution to the i'th sub-problem. One candidate complete solution is the union of one sub-solution from each row. Of course, if the combinations are made at random, some union solutions may be incompatible and will be discarded.

To use the morphological chart, a function decomposition could be the starting point. Alternately, design attributes and requirements could also be used for problem decomposition. Various functions, sub-functions, features or requirements of a product can be established through a pre-analysis and can be entered in the left-most column. This left-most column is called as the problem space. All the possible solutions/means/ideas are listed corresponding to their features/functions in rows corresponding to the problem they address. These solutions are usually concrete and specific. Some of them could also be specified to a category (i.e. parameter). These solutions could be known ones that come from existing solutions such as analogous products or these could be novel ones which are generated during the design process. Figure 18 is a part of a morphological chart. It shows how sub-functions are listed in columns and solutions are listed in rows. The design problem is to design a manned transportation device for vegetables. The listed sub-functions include "human power", "stearing", "transmition", "surprise", "learning effect", "acceleration" and so on.

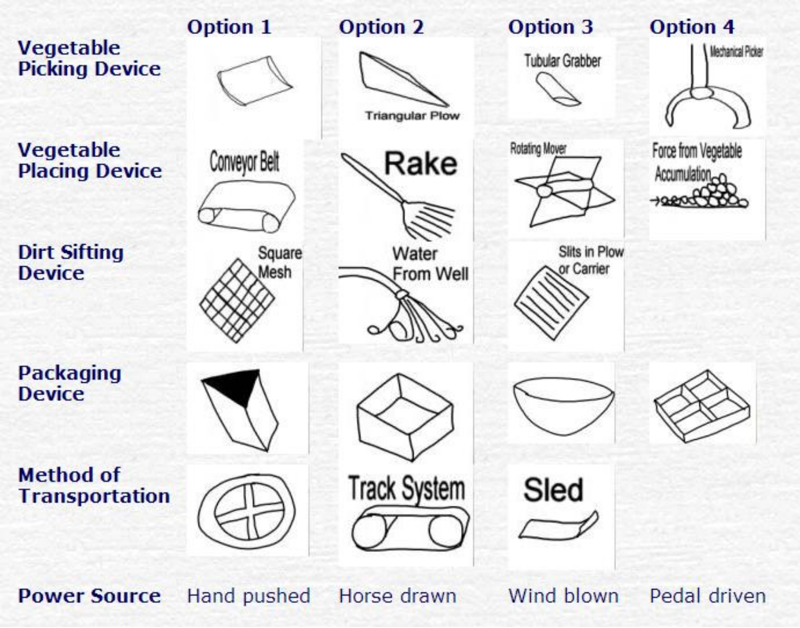


Figure 18: One Example of Morphological Chart

(http://www.eng.fsu.edu/~haik/design/idea_generation.htm)

The main objective achieved by the morphological chart is to widen the search of new complete solutions based on combination of the previously known sub-solutions. The process of forming a morphological chart is not a static one but rather it is an evolutionary approach: sub-problems and sub-solutions evolve in parallel until the final morphological chart is made.

In a fully fledged out morphological chart, the number of possible configurations is simply the product of the number of solutions under each feature (i.e. sub-functions). However, only trying to examine or read all possible configurations would take a good deal of time and effort. For example, in Figure 18, the morphological chart contains 4*4*3*4*3=576 possible complete solutions. That is where we can tap the computational power available to us today. A computer tool can facilitate the tracking of the combinations. With the help of computers we can devise a step in the analysis-synthesis process which can reduce the total set of possible configurations to the feasible ones in the total problem space. This will generate a smaller set of configurations.

Before going to the specific design of the V.2 organizational framework, We will first summarize the enhancements needed in the traditional morphological charts:

1. Any of the sub-problems can be further decomposed, generating a morphological chart of its own (This will add a cascading nature to the traditional morphological charts).

2. Sub-solutions can be generated by the designer either with any ideation aids or on his own.
3. The sub-problems represent a particular framing and the candidates represent a structure. Reframing and restructuring can be facilitated by morphological charts. Rows can be reformulated using different function or attributes for reframing. And if the morphological chart could be arranged and modified, it amounts to restructuring (Thus it will be an evolutionary morphological chart).

4.3.2 Cascading Evolutionary Morphological Charts

In order to match the requirements of the framework, the traditional morphological chart was tweaked and called as Cascading Evolutionary Morphological Charts (CEMC). In comparison with the classical morphological chart one of the new features of CEMC is the cascading property where further decomposition of any (sub) problem is allowed. The term “Cascading” indicates that it could support further decomposition of the design problem and the term “Evolutionary” indicates the reframing and restructuring would be embedded in the modified morphological charts so that the sub-problems and sub-solutions evolve in parallel until the final chart is made (Chen, 2012). Cascading Evolutionary Morphological Charts can be used as an organizational framework for holistic ideation for following reasons:

1. They allow complex problems to be decomposed into sub-problems that are more likely to be solved. This way the designer can see the hierarchy

he/she has in his/her problem decomposition in this format and this important information does not get evaporated during the ideation process unlike with traditional morphological charts.

2. The cascading morphological structure is not solely created for functions. The flexibility of the tool allows the decomposition by various aspects of the design such as feature, material, requirement and so on.
3. Any of the sub-problems can be further decomposed, generating a morphological chart of its own.
4. Sub-solutions can be generated by the designer with or without any ideation aids.
5. Any combination of ideation methods can be used for sub-solutions.
6. The designer can make deliberate or random connections between sub-solutions.
7. The solutions can be combined to generate complete solutions.

It is evident that major difference between traditional morphological charts and cascading evolutionary morphological charts is in heirarchial information the later preserves through the multilayer decomposition attribute. We will discuss that in detail below.

4.3.3 Multilayer Problem Decomposition

As Figure 19 shows, the morphological chart consists of two parts. The left part is the problem space and the right part is the solution space. The left part contains sub-problems which are features, (sub)functions, requirements or anything could achieve the target design. The right side is the (sub) solutions for each feature (function, requirement or anything) in a row.

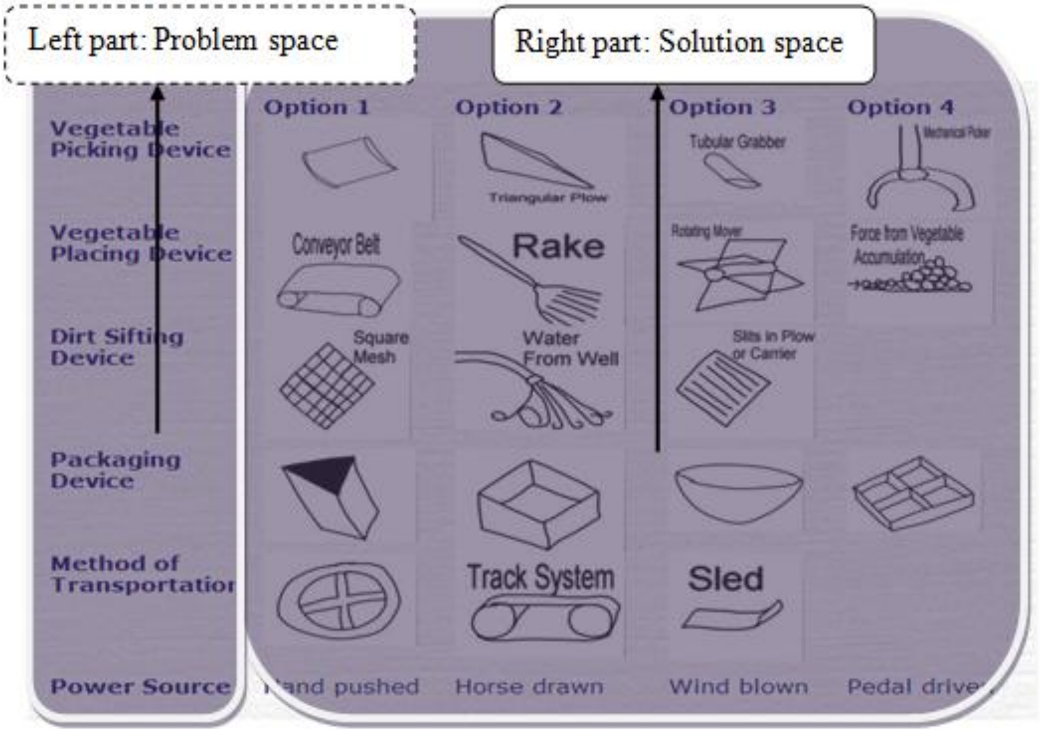


Figure 19: Problem Space and Solution Space

In the traditional morphological charts, the first step is to formulate the design problem through problem decomposition in a single chart. The new CEMC organizational framework allows the designer to choose any basis for decomposition. The designer can fill the problem space with not only functions but also features, requirements, etc. of the design target. The various functions, sub-functions, features, requirements, morphology, or properties of a new design could be established through a pre-analysis (Chen, 2012).

One of the new features of CEMC is the cascading property where further decomposition of any (sub) problem is allowed. In the CEMC , a series (hierarchy) of morphological charts can be created from any sub-problem listed in the base/parent morphological chart. Based on one of the sub-problem of the original CEMC, a new morphology list or morphology structure could be explored and the user can create a new CEMC based on that. Figure 20 is an example of one of the possible problem decomposition of a CEMC for the design problem: "Design a mechanical device to create burritos."

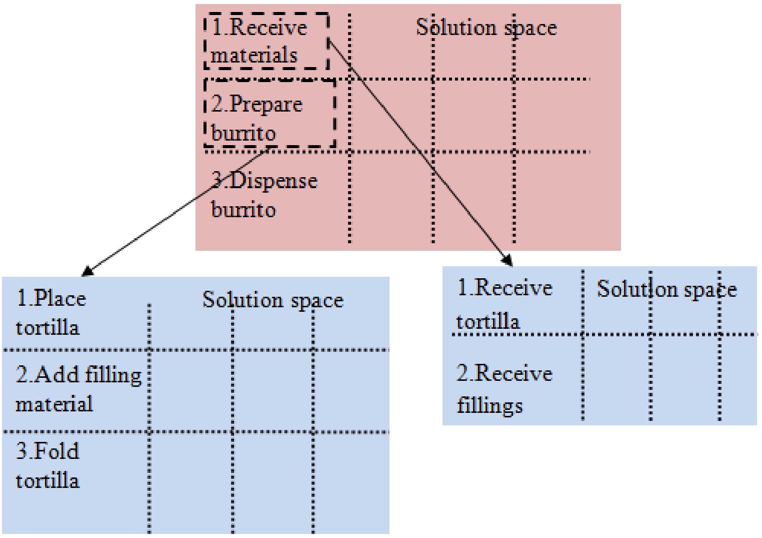


Figure 20: Multilayer Problem Decomposition of CEMC **(Chen, 2012)**

In above CEMC, one sub-problem (Receive Materials) is decomposed into two sub-problems (Receive tortilla and Receive fillings) to create second level CEMC. Consequently, Figure 21 shows the same CEMC in a tree format to shed light on the hierarchy stored in the decomposition of the problem.

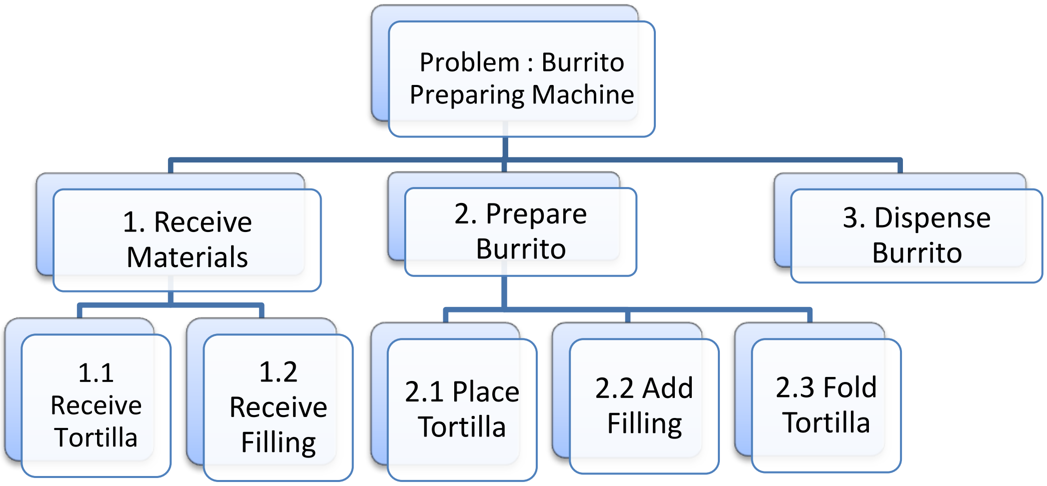


Figure 21: CEMC in Tree Format

4.4 Database Schema For V.2

V.2 was developed using Java, while Microsoft Access tool was used to create and manage databases. The general process of making a Cascading Evolutionary Morphological Chart in V.2 is as follows:

1. Problem decomposition;

2. Creating morphology list and morphology structure (reframing and restructuring of the organized problem form is possible at any time);

3. Generating ideas or solutions by using ideation methods;

4. Composing the sub-solutions into complete solutions;

5. Store the plausible complete solutions.

V.2 retained the same databases for all the logical ideation methods used in the V.1, i.e., PE, WP, Artifact and TRIZ/Bio-TRIZ database. In the V.1 these databases were implemented as independent tables. While populating any two related PE or WP, the information was fetched through queries dynamically at runtime.

The Database Repositories play a significant role in cross relating intuitive and logical ideation methods. For that purpose PE and WP databases are tweaked and joint tables were created. Figure 22 shows the relationship diagrams for the design repositories. The database "WP_name_desc" contains the information of the working principles. The database "PE_main" contains the information of physical effects. Databases "PE-main" and "WP_name_desc" are related by mapping WP_ID to PE_name. Working principles (database "WP_name_desc") are related to the components (database "WP_comp_table") by mapping WP_ID to comp_ID. Functions in RFB (database "func") are related to the working principles (database "WP_comp_table") through mapping both working principles and functions to flow (database "flow"). As the data schemas displays, the physical effect database, the working principle database and the RFB database are related together. The system supports the cross relating by keywords.

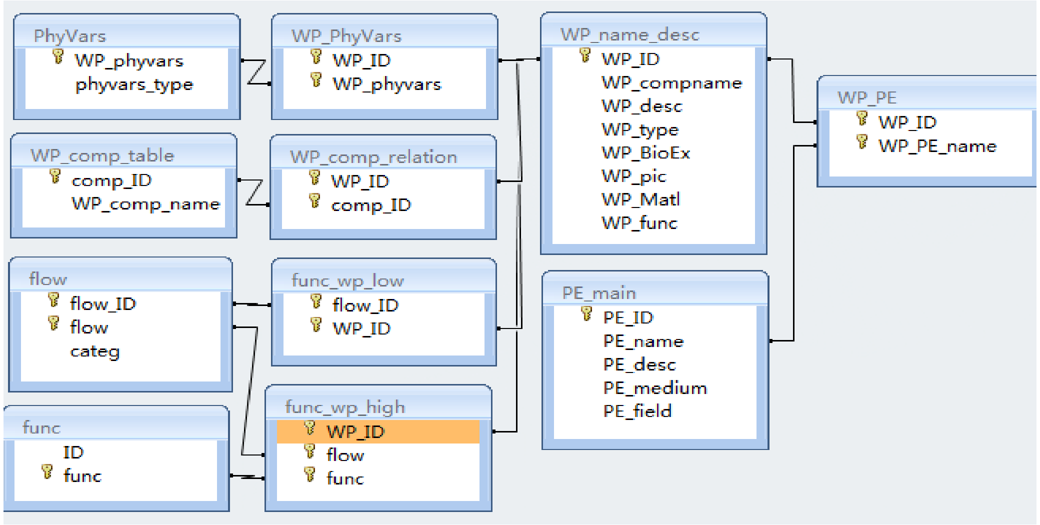


Figure 22: Database Schema for Cross Relating Repositories

In V.2, the information of the sub-problems and sub-solutions are stored in the databases (Figure 23). Problem information is stored in the form of sub-problem name, the name of the design, and a short description. Sub-solution information is stored as sub-problem name, information types (text or graph) and information content (words or pictures address). The complete solutions are defined as the combination of sub-solutions.

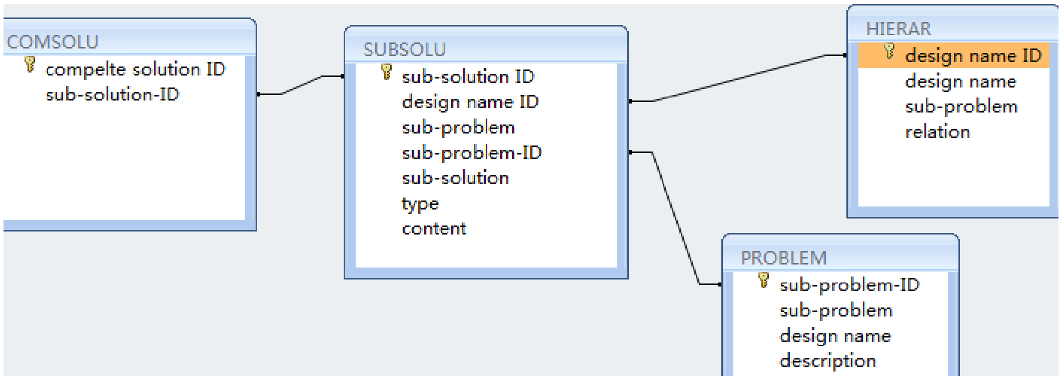
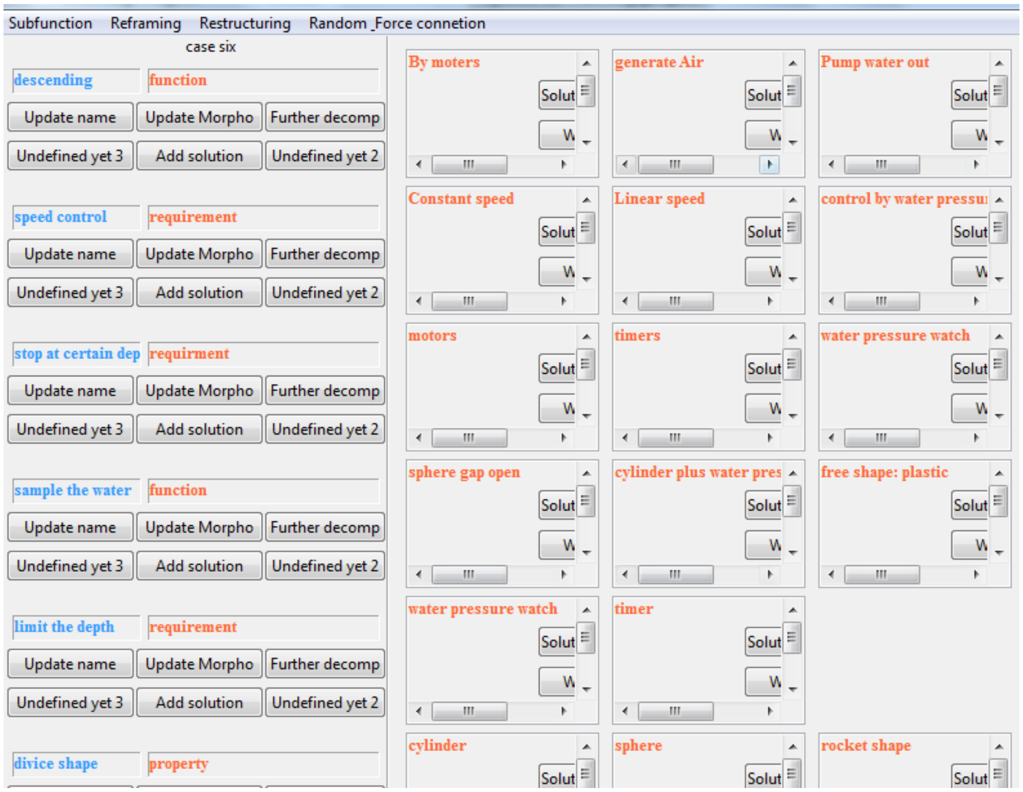


Figure 23: Database Schema for Morphological Chart

4.5 UI Development For V.2

4.5.1 Problem Decomposition

The V.2 main user interface (UI) of a CEMC was divided into two different parts: problem area (morphology list) and solution area. In V.2, all intuitive methods are available at anytime during problem decomposition without losing any design data. As shown in Figure 24 and 25, the morphology list present on the left hand side allows the user to specify and update the name of the (sub) feature and the short description. Each (sub) feature has a row to list the solutions horizontally on the right hand side. There is no limit of the number of the (sub) features and the solutions. The UI could be extended by the user and maximum and minimum size depends on the resolution of the computer.



case six

descending | function

speed control | requirement

stop at certain dep | requirment

sample the water | function

limit the depth | requirement

divice shape | property

By moters

generate Air

Pump water out

Constant speed

Linear speed

control by water pressu

motors

timers

water pressure watch

sphere gap open

cylinder plus water pres

free shape: plastic

water pressure watch

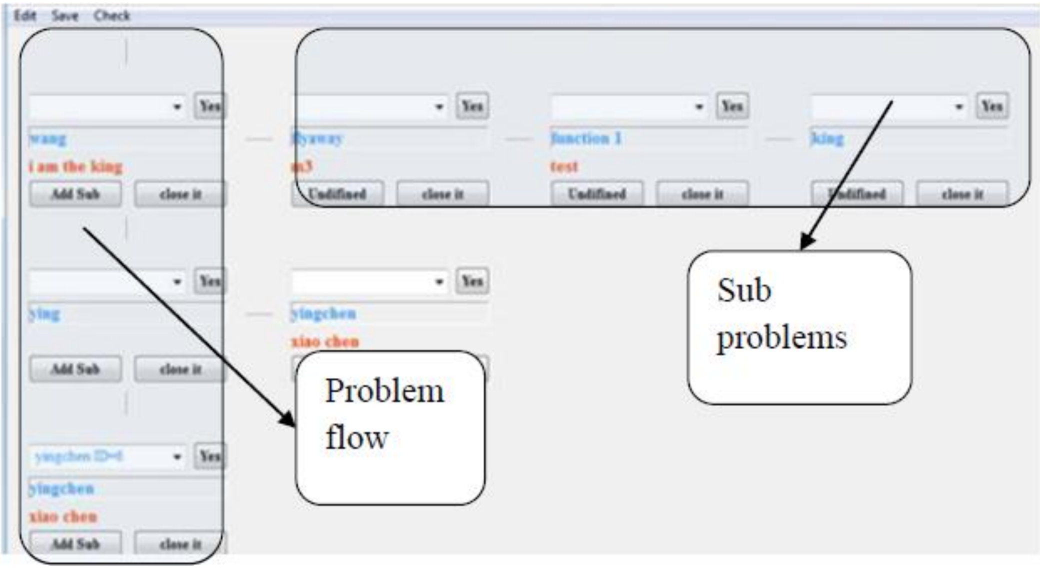
timer

cylinder

sphere

rocket shape

Figure 24: UI for Morphological Chart



wang

i am the king

flyaway

m3

function 1

test

king

ying

yingchen

xiao chen

yingchen ID=4

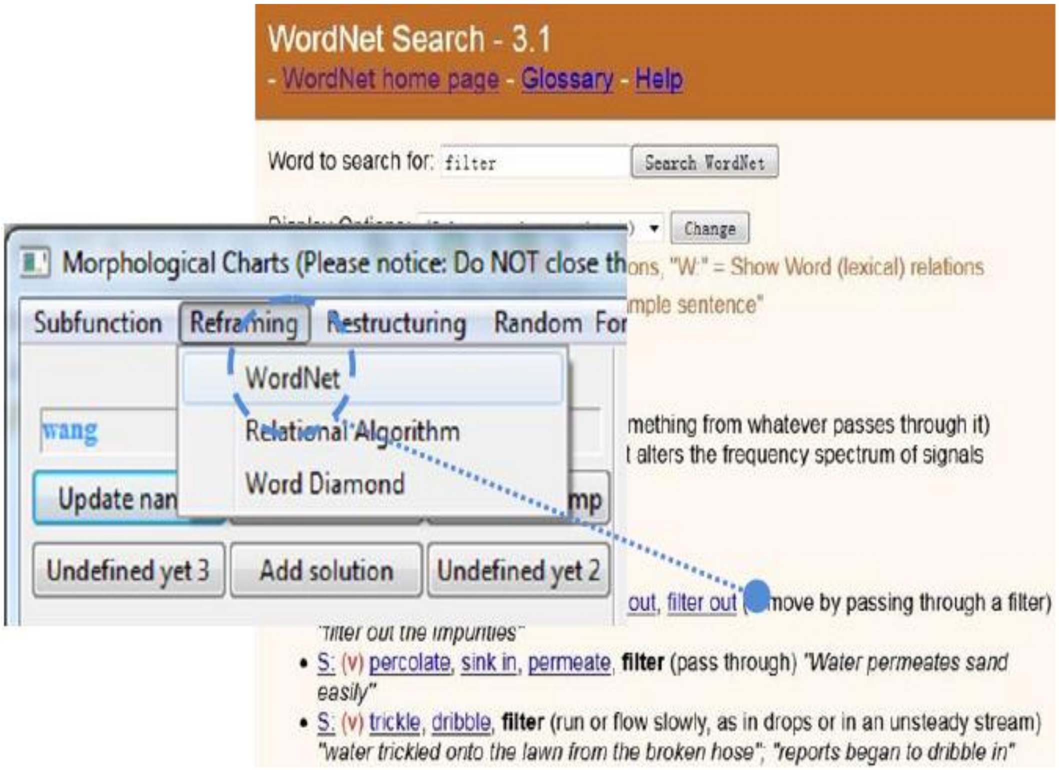
yingchen

xiao chen

Figure 25: UI for Morphology Tree

4.5.2 WordNet

As discussed above, in WordNet, Synsets are interlinked by means of conceptual-semantic and lexical relations. The resulting network of meaningfully related words and concepts can be navigated with an internet browser. As shown in Figure 26, under the top toolbar “reframing”, the button for opening WordNet tool is available. While reframing “filter”, if we consider the “filter” as a function, the outputs shown are “filtrate”, “separate out”, “filter out”, which are all functions. Some functions are followed by behavior explanation, such as “filter out (remove by passing through a filter)”.



ons, "W:" = Show Word (lexical) relations

mple sentence"

mething from whatever passes through it)

t alters the frequency spectrum of signals

out, filter out (move by passing through a filter)

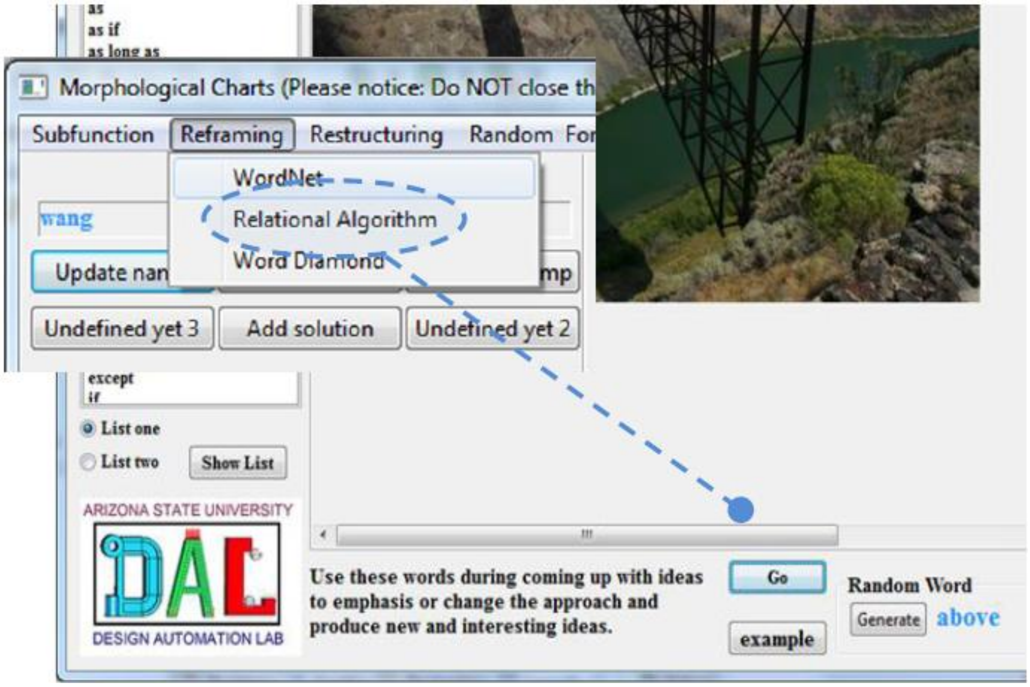
"filter out the impurities"

- S: (v) percolate, sink in, permeate, **filter** (pass through) *"Water permeates sand easily"*
- S: (v) trickle, dribble, **filter** (run or flow slowly, as in drops or in an unsteady stream) *"water trickled onto the lawn from the broken hose"; "reports began to dribble in"*

Figure 26: Reframing by WordNet

4.5.3 Relational Algorithm

As shown in Figure 27, V.2 supports the Relational-Algorithm ideation method by providing the designer the list of 37 words. Sketches and pictures are attached with these words to generate stimuli for the reframing of the design problems. These pictures are pre-selected and attached with these 37 words.

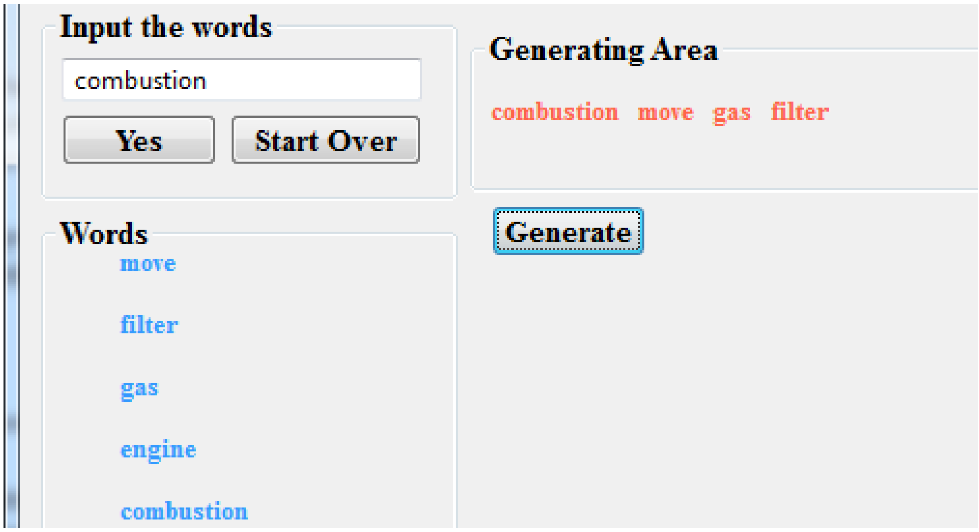


Use these words during coming up with ideas to emphasis or change the approach and produce new and interesting ideas.

Figure 27: Relational-Algorithm UI

4.5.4 Word Diamond

The Word Diamond ideation method provided in V.2 lets a designer decide how many words he/she want to use. These words are taken as input by the system. After using a random number generator algorithm the tool gives out these words in a random order as an output (Figure 28).



Input the words: combustion

Figure 28: Word Diamond UI

4.5.5 Documentation tool

The designer can make either one sketch or attach multiple images for each solution. He/she is also provided with a textual documentation tool (Figure 29) which can be used at any time in the design process. The available image file formats are ".jpg", ".png", and ".gif". Moreover, the framework provides the option to attach multiple images. All the images that are attached by designers with the solutions are copied and stored in the database.

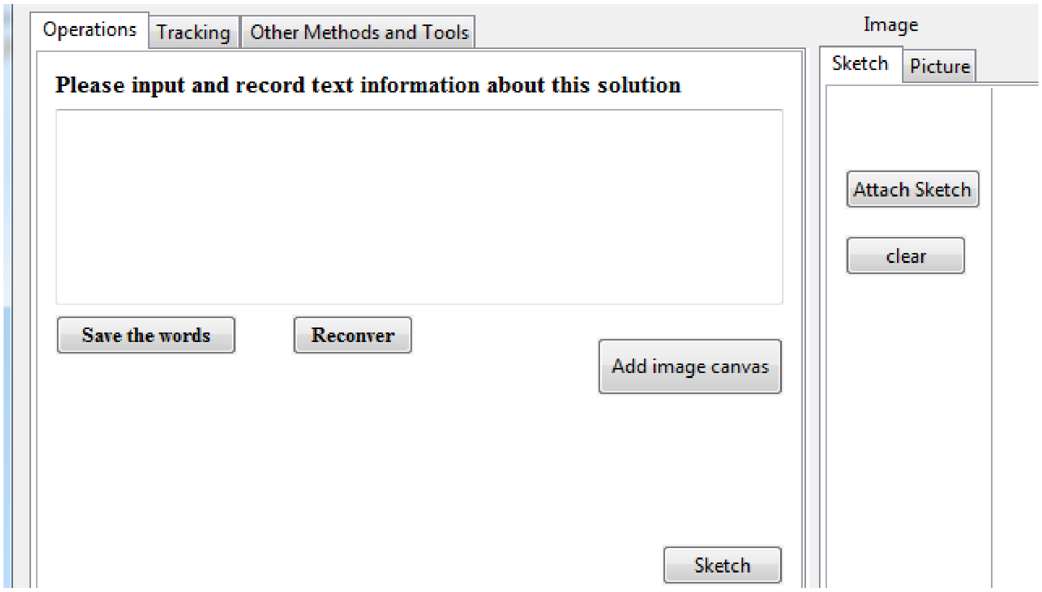

Please input and record text information about this solution

Figure 29: UI for Documenting Each Solution

4.6 V.2 User Reviews

User studies were conducted to test the effectiveness of the V.2 (Chen, 2012). Following are some of the major issues reported by users :

1. Provocative stimuli of cross-relating and Bio-analogy were useful for generating novel solutions. However, those generated ideas could be the solutions for any sub-problems in the design.

2. If there was a guided pre-analysis provided by the system before the problem decomposition, the design process would be faster.

3. Searching by keywords is useful and it would be better if the function information, behavior information and component information could be shown at the same time.

4. Reframing is not useful after the overview of the design problem has already been structured by the designer.

5. Analogy is very useful when there are some pictures of some existing components.

6. High level provocative stimuli with animation would be better for users to generate solutions and ideas. For information at behavior level, animation could positively contribute towards novelty than physical equations.

7. In the provocative stimuli of cross-relating of logical database, it will be better to present relative information automatically by listing all the working principles which are related to certain parameters.

8. It will be more helpful if one could construct the morphological chart in 3-D, which means presenting the morphology tree in 2-D and listing all the solutions of each feature in the third dimension.

9. It would be better if the hierarchy of problem decomposition could be converted into morphology list automatically.

10. Tracking through keywords is easier than tracking by selecting physical effect parameters and working principle parameters, because tracking by keyword does not require design discipline and experience.

11. In the reframing by WordNet, it is a constraint to allow the designer to use only one word for exploring.

12. Reframing by Word Diamond was not helpful. Because all the words were known, some random combinations of the words are meaningless and some random combinations are not even feasible.

4.7 Shortcomings Of V.2

User studies revealed many shortcomings in the content and the user experience of V.2. We will discuss those below.

4.7.1 Lack of Ideation Methods

Even though the V.2 has ideation methods from both logical as well as intuitive ideation set of methods, it can still benefit from some more ideation methods. To understand more about other strategies it will be required to have additional intuitive and logical methods embedded in the holistic ideation tool.

4.7.2 Lack of Integration

As we add more intuitive and logical strategies via respective methods into the framework we need to find an effective way to integrate the new set of ideation methods with the previous ones.

4.7.3 Organizational Framework

V.2's organizational framework was based on CEMC but it was not mature as it was the first try in that direction. The organizational framework needed to improve. Though V.2 was less restrictive than V.1, it had constraints like the user not being able to search the database without sticking to a specific function.

4.7.4 User Interface

The user interface for V.2 was a little bit confusing because too many buttons were given on the same page. This might hinder creativity as the user should feel at ease with UI when he/she tries to do any activity such as searching the knowledge base, constructing the morphological chart, etc. Multiple level hierarchy was present but implementation and UI made it very difficult to traverse through that hierarchy. Also, multiple views of multiple sections of the CEMC at the same time were not possible.

4.7.5 Lack of Animation and Images

Both V.1 and V.2 concentrated on ideation methods from logical as well as intuitive ideation method sets. However while implementing these ideation methods textual information was given higher preference over graphical. And whatever images were present, for the description of some of the methods, lacked quality. Also video animations of any kind were not implemented in the V.1 and V.2.

4.8 Summary And New Tasks

Chapters 3 and 4 reviewed the previous two versions of the Holistic Ideation Tool. After carefully studying both the versions and their respective shortcomings, the following aspects of Holistic Ideation tool were selected for improvement:

1. Addition of the New Ideation Methods such as Mechanisms and Machine Elements.

2. Improving Existing Organizational Framework to provide new features such as automatic solution population, categorization of ideation methods.

3. Improving UI Aesthetics.

CHAPTER 5

NEW IMPROVED HOLISTIC IDEATION TOOL

Chapters 3 and 4 presented the previous versions (V.1 and V.2) of the holistic ideation tool. The new version (V.3) of the holistic ideation tool was created as a part of my research at DAL. In V.3 new ideation methods were added and organizational framework was improved. These improvements in V.3 will be discussed in detail in the following sections. This chapter will also discuss the UI walk-thru of V.3.

5.1 Addition Of New Ideation Methods

Many mechanical design problems involve devices to transfer or convert motion. In such types of problems designers face the challenge of either coming up with a new mechanism or using existing mechanism types that could serve as a solution for the problem at hand. The designer could also synthesize a new mechanism as a combination of the existing mechanisms or he/she could resort to off the shelf components to create a solution. The previous versions did not incorporate any module to help designers with the common mechanisms or machine elements. To fill that void, the "mechanisms" and the "machine elements" ideation modules were added to the tool. To achieve this many independent electronic and non-electronic databases of the mechanisms and the machine elements along with their classification scheme were studied. After that a new database schema was designed to create and integrate the "Mechanisms" and the "Machine Elements" ideation modules in V.3.

5.1.1 Machine Element and Mechanism Databases Review

The survey of the tools for conceptual design revealed that there are no existing tools which use functional indexing of mechanisms and machine elements suitable for conceptual design stage. There are many traditional books which give detail description of various kinds of Mechanisms or Components off-the-shelf (COTS) machine elements. Also, there are some online resources created by the research community. In order to use those in conceptual design process we need to integrate them in the holistic ideation tool. We will review current databases in the next section.

5.1.1.1 COTS Databases Review

Various mechanical engineering texts were reviewed in search of a comprehensive machine elements database. For example, traditional texts like Mechanical Engineering Design, Shigley (Shigley, 2003) have a comprehensive set of COTS. The problem encountered with this textbook was the indexing of machine elements. It was very rudimentary; the indexing was based on types of COTS, i.e., all the different springs (tension, compression, torsion) are classified into one general group "Springs". This type of classification could not help the user in conceptual design stage as this classification failed to organize COTS according to achieved function. The same classification problem was prevalent in digital databases.

ThomasNet (THOMASNET) is an online platform to connect buyers and sellers in the manufacturing and industrial world. They have categorized COTS according to a single hierarchical classification scheme, i.e., all different types of bearings are classified into a single category of Bearings. The sub-classes used to classify COTS from a category are based on the type of COTS. For example, Bearings are sub classified into Ball Bearings, Linear Bearings, Roller Bearings, and Mounted Bearings. They do not classify COTS according to achieved functions. Figure 30 shows a partial view of the different categories in which different machine elements are organized by them.

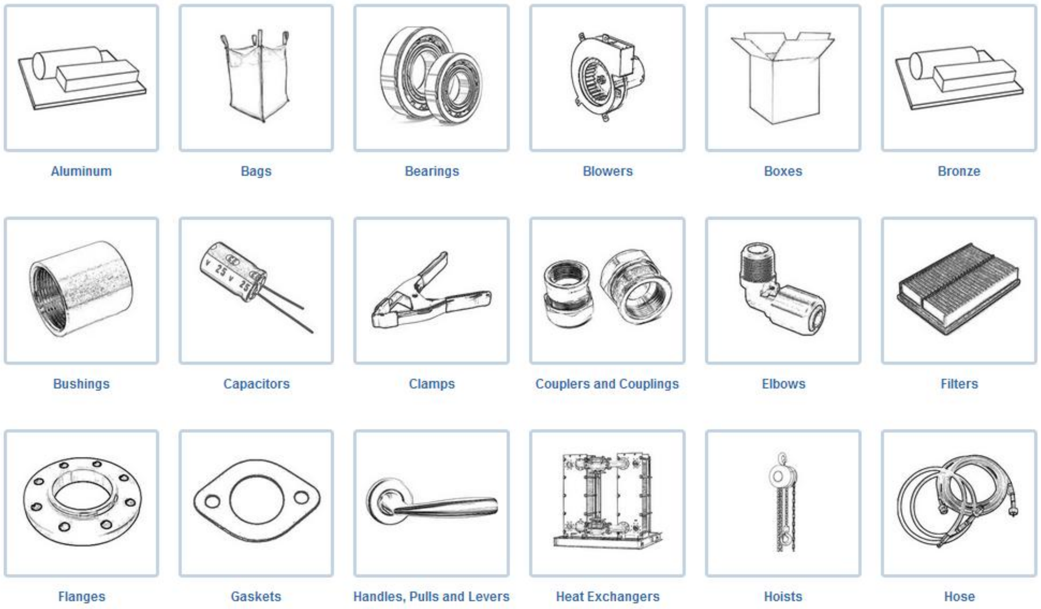


Figure 30: ThomasNet COTS Classification

Shah (Shah J. , Advanced Product Design Methodology, 2011) provided a classification scheme for the COTS as shown in Figure 31, which classifies the COTS according to the functions achieved, e.g., Speed Reduction: Belt Drives, Gear Drives.

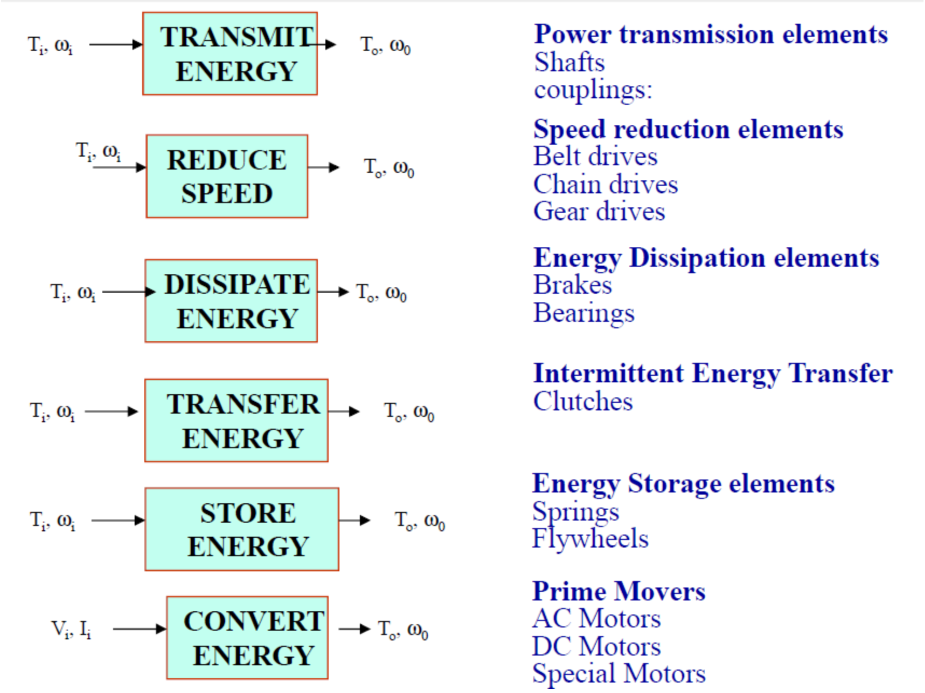
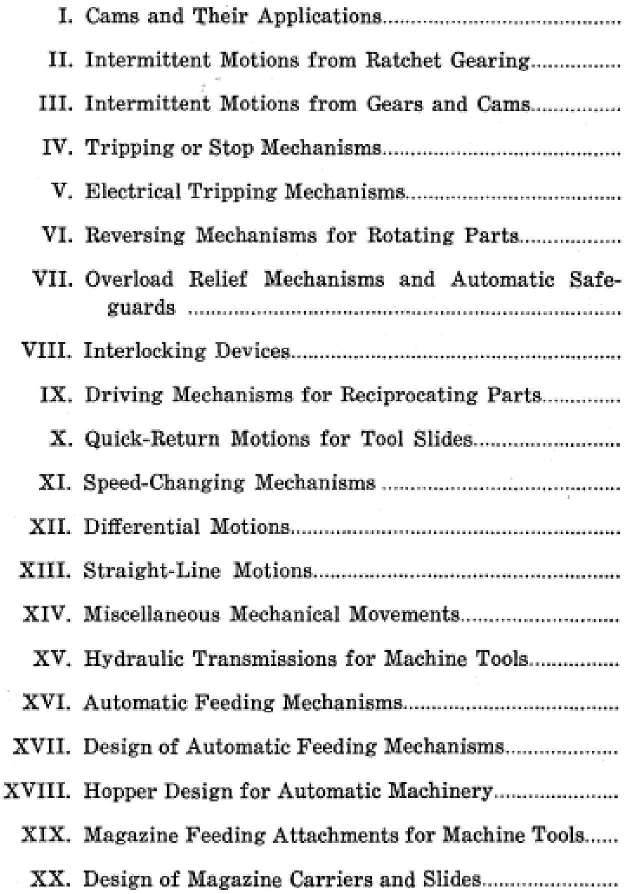


Figure 31: Function Based COTS Classification

In the next section current electronic and non-electronic databases for the mechanisms will be reviewed.

5.1.1.2 Mechanisms Databases Review

Texts such as the Ingenious Mechanisms (Horton & Newell, 2004), Mechanisms in modern engineering design (Artobolevskiĭ, 1979-82) have listed more than 2000 examples of various mechanisms in detail. The Ingenious Mechanisms (Horton & Newell, 2004) have classified mechanisms into 20 groups and the main theme in the classification scheme can be found as common motions involved, e.g., straight line motions, differential motions. Figure 32 shows the list of these 20 groups.



I. Cams and Their Applications..

II. Intermittent Motions from Ratchet Gearing................

III. Intermittent Motions from Gears and Cams................

IV. Tripping or Stop Mechanisms..

V. Electrical Tripping Mechanisms....................................

VI. Reversing Mechanisms for Rotating Parts.................

VII. Overload Relief Mechanisms and Automatic Safeguards ....

VIII. Interlocking Devices..

IX. Driving Mechanisms for Reciprocating Parts.............

X. Quick-Return Motions for Tool Slides.........................

XI. Speed-Changing Mechanisms ....

XII. Differential Motions..

XIII. Straight-Line Motions..

XIV. Miscellaneous Mechanical Movements..........................

XV. Hydraulic Transmissions for Machine Tools................

XVI. Automatic Feeding Mechanisms....................................

XVII. Design of Automatic Feeding Mechanisms...................

XVIII. Hopper Design for Automatic Machinery.....................

XIX. Magazine Feeding Attachments for Machine Tools......

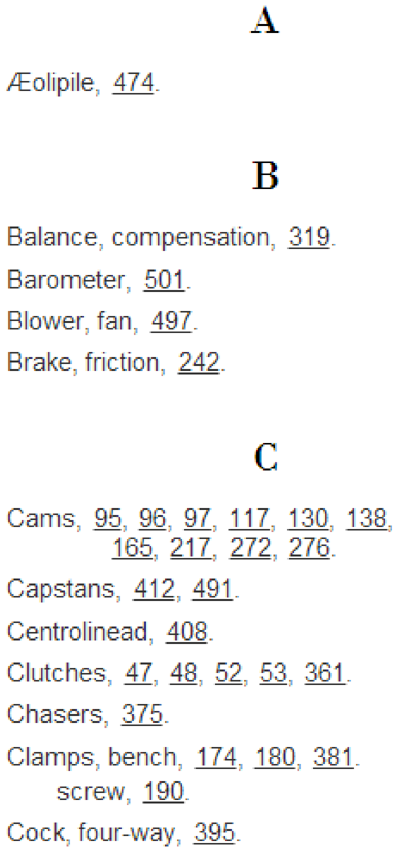
XX. Design of Magazine Carriers and Slides.......................

Figure 32: Motion Based Classification of Mechanisms

The classification scheme used in the Modern Engineering Design (Artobolevskiĭ, 1979-82) books have used structural features and functional features to list mechanisms in different groups. This type of function based classification can be used in the conceptual design stage.

The 507 Mechanical Movements (Keveney) and the Kinematic Models for Design Digital Library (KMODDL) (A Cornell University Web site) have created a digital database of the mechanisms.

The 507 Mechanical Movements (Keveney) online resource has a comprehensive list of mechanisms. They have organized mechanisms by names. Currently this database provides images of the mechanism and the textual information regarding the working of the mechanism. They have recently started to add video simulations of each mechanism. Figure 33 shows the example of the indexing of mechanisms.



A

Æolipile, 474.

B

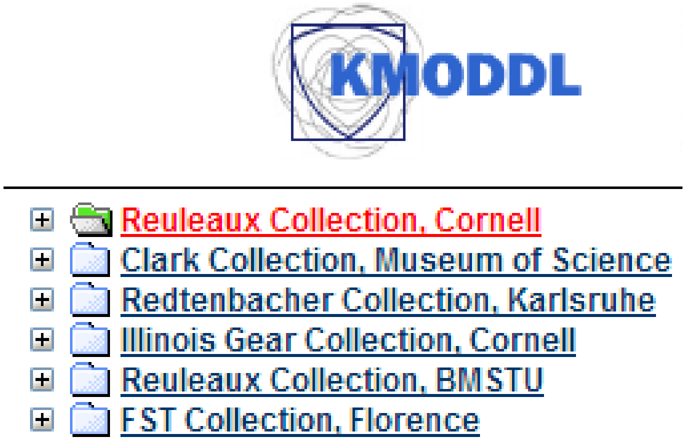
Balance, compensation, 319.
Barometer, 501.
Blower, fan, 497.
Brake, friction, 242.

C

Cams, 95, 96, 97, 117, 130, 138,
165, 217, 272, 276.
Capstans, 412, 491.
Centrolinead, 408.
Clutches, 47, 48, 52, 53, 361.
Chasers, 375.
Clamps, bench, 174, 180, 381.
screw, 190.
Cock, four-way, 395.

Figure 33: Indexing of Mechanisms by 507 Mechanical Movements Database

The KMODDL (A Cornell University Web site) online resource also has a comprehensive set of the mechanisms documented digitally in a web based database. Though the organization of the mechanisms suffers from the same problems discussed above, the advantage for the designer with this database is that the database provides the detailed information about working of the mechanisms. Along with the description it also shows an image and a working video of the mechanism. Figure 34 shows the classification scheme adopted by the KMODDL to organize the mechanisms into different groups, which is not very conducive for using in the conceptual design stage. The mechanisms are grouped into 6 major categories (Figure 34).

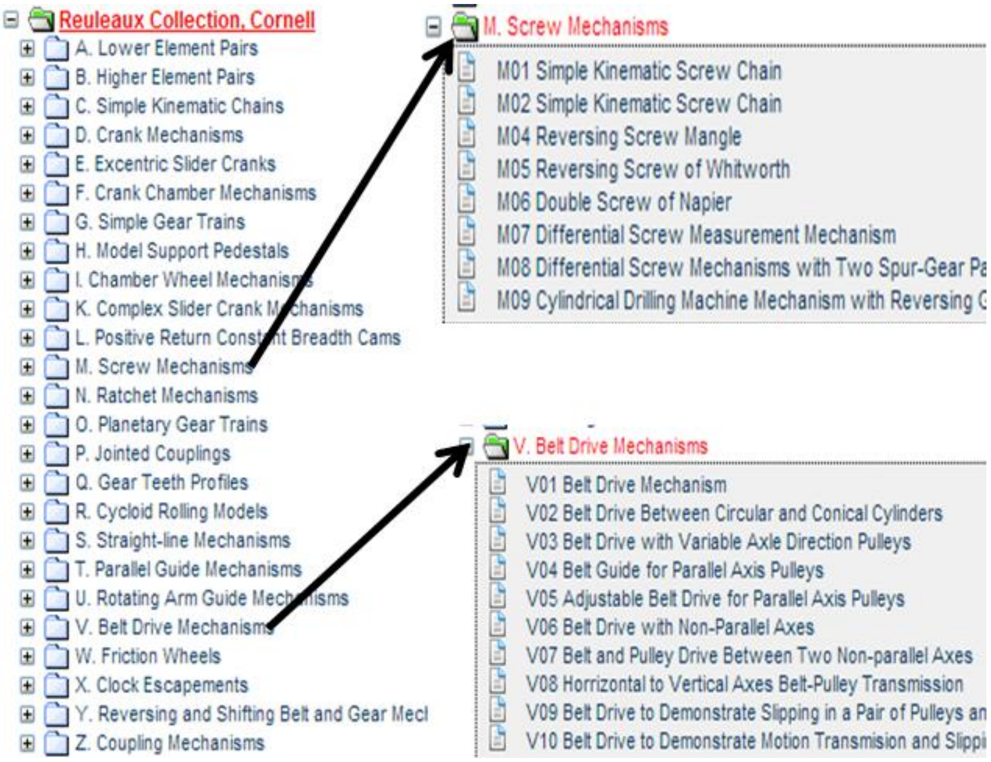


KMODDL

⊞ Reuleaux Collection, Cornell
⊞ Clark Collection, Museum of Science
⊞ Redtenbacher Collection, Karlsruhe
⊞ Illinois Gear Collection, Cornell
⊞ Reuleaux Collection, BMSTU
⊞ FST Collection, Florence

Figure 34: Highest Level of Classification Used in KMODDL Library

Each of these categories represent the source of the list. Each of these categories then sub-divides the mechanisms into different categories based on the similar machine components involved in different mechanisms as illustrated in Figure 35.



Reuleaux Collection, Cornell

- A. Lower Element Pairs
- B. Higher Element Pairs
- C. Simple Kinematic Chains
- D. Crank Mechanisms
- E. Excentric Slider Cranks
- F. Crank Chamber Mechanisms
- G. Simple Gear Trains
- H. Model Support Pedestals
- I. Chamber Wheel Mechanisms
- K. Complex Slider Crank Mechanisms
- L. Positive Return Constant Breadth Cams
- M. Screw Mechanisms
- N. Ratchet Mechanisms
- O. Planetary Gear Trains
- P. Jointed Couplings
- Q. Gear Teeth Profiles
- R. Cycloid Rolling Models
- S. Straight-line Mechanisms
- T. Parallel Guide Mechanisms
- U. Rotating Arm Guide Mechanisms
- V. Belt Drive Mechanisms
- W. Friction Wheels
- X. Clock Escapements
- Y. Reversing and Shifting Belt and Gear Mecl
- Z. Coupling Mechanisms

M. Screw Mechanisms

- M01 Simple Kinematic Screw Chain
- M02 Simple Kinematic Screw Chain
- M04 Reversing Screw Mangle
- M05 Reversing Screw of Whitworth
- M06 Double Screw of Napier
- M07 Differential Screw Measurement Mechanism
- M08 Differential Screw Mechanisms with Two Spur-Gear Pa
- M09 Cylindrical Drilling Machine Mechanism with Reversing G

V. Belt Drive Mechanisms

- V01 Belt Drive Mechanism
- V02 Belt Drive Between Circular and Conical Cylinders
- V03 Belt Drive with Variable Axle Direction Pulleys
- V04 Belt Guide for Parallel Axis Pulleys
- V05 Adjustable Belt Drive for Parallel Axis Pulleys
- V06 Belt Drive with Non-Parallel Axes
- V07 Belt and Pulley Drive Between Two Non-parallel Axes
- V08 Horrizontal to Vertical Axes Belt-Pulley Transmission
- V09 Belt Drive to Demonstrate Slipping in a Pair of Pulleys an
- V10 Belt Drive to Demonstrate Motion Transmision and Slippi

Figure 35: Component Based Classification Scheme for Mechanisms

After reviewing the existing COTS and Mechanisms databases the main question was how to make this data useful to a designer in the conceptual design stage. The adopted representation for implementing the COTS and the Mechanisms in the new holistic ideation tool is discussed in following sections.

5.1.1.3 Shortcomings in Current Databases

In light of the usability in the conceptual design stage as an ideation module the following shortcomings were found in the current databases:

1. Some of the databases have a small list of the mechanisms/COTS.

2. The non-function based categorization used for classification of the mechanisms/COTS is not helpful for a designer working at the conceptual design stage.

3. Some databases did not have any image or working videos of the mechanisms/COTS.

After reviewing the current databases we can see that the knowledge about mechanisms/COTS is available in many databases but the organizational structure, indexing and searching methods are lacking. So, what is needed is a smart front end and a new database design to support that front end.

Accordingly, a new database structure meeting those requirements was also designed to store the mechanisms and the COTS. This new database structure enabled us to implement ideation methods for searching mechanisms and COTS. The details will be discussed in the following sections.

5.1.2 Representation for Mechanisms and COTS

The fundamental need behind the creation of this new database schema was to classify different mechanisms/COTS effectively, so that it can be used in the conceptual design stage. At this stage, the mechanism/COTS type selection is more appropriate rather than sizing and geometrical design. Most of the databases reviewed provide searching by the name or category. This type of search capability is necessary but not sufficient. One should also be able to search by the functional requirement. In the conceptual design stage, a designer depending upon his/her domain knowledge may want to search mechanisms/COTS according to the following categories:

1. A designer may want to search by the name or category of the mechanisms/COTS if he/she already knows for which mechanisms/COTS to look.

2. A designer may want to search by the function.

3. A designer with low domain knowledge may not know what is the name of the mechanism/COTS or under what category it might belong but he/she may know what kind of input or output characteristics to look for.

5.1.2.1 Fundamental Difference and Similarity between Mechanisms and COTS

Before tackling the issue of creating a suitable classification scheme for storing the mechanisms/COTS into a database, we need to look at the fundamental difference as well as similarities between a mechanism and a machine element.

In kinematics a mechanism/kinematic linkage is a means of transmitting, controlling or constraining relative movement (Hunt, 1978). The central theme of representing the mechanisms is multiple machine elements connected together by joints. The term mechanism is a applied to the combination of geometrical bodies which constitute multiple machine element (Ham, 1958).

Despite the fact that a truly rigid body does not exist, many engineering components are considered rigid because their deformations and distortions are negligible compared to their relative movements. The similarity between the machine elements and the mechanisms is that:

1. They both are combinations of rigid bodies,

2. The relative motions among the rigid bodies are definite.

For instance, Figure 36 shows that the cross section of a "power cylinder" in a diesel engine. It consists of many machine elements such as cylinder, piston, pins, bearings. However, the mechanism used in this particular combination of machine elements is a "slider crank mechanism" as shown in Figure 37.

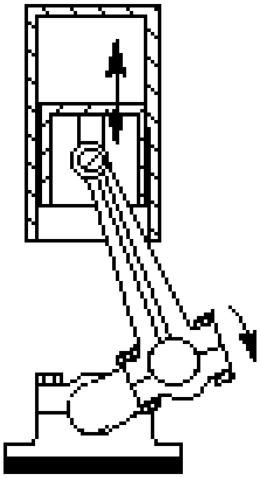


Figure 36: Cross Section of Power Cylinder in Diesel Engine

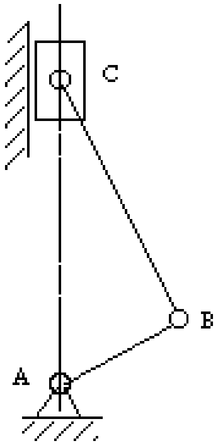


Figure 37: Slider Crank Mechanism.

Because of such differences between the COTS and mechanisms we created separate databases for COTS and mechanisms. Furthermore, because of the similarities between them, same classification scheme with minor differences is used for both databases.

5.1.2.2 Mechanisms and COTS Database

As discussed above, there are differences between mechanisms and COTS. Because of this, the representation used for both databases have some differences which will be discussed below.

Each device in the device library is described by five major categories of attributes: INPUT, OUTPUT, RELATION, FUNCTION and DESCRIPTIVE. These major categories are divided into sub categories in order to fully describe a mechanism/COTS. Because of the similarities in mechanisms and COTS, both databases use the same high level structure, i.e., both databases use same major categories described above. Furthermore, the first four categories of both databases have the same sub categories. Table 10, 11, and 12 shows the major categories, sub categories, and their explanation which are common for both databases.

Table 10: Input and Output Category

INPUT And OUTPUT	INPUT/OUTPUT TYPE	This attribute describes the type of input/output motion for the mechanism/COTS e.g. rotation, translation
	INPUT/OUTPUT SPEED	This attribute describes the speed of the input/output motion for the mechanism/COTS e.g. constant, intermittent
	INPUT/OUTPUT VELOCITY DIRECTION	This attribute describes information related to velocity direction of input/output motion for the mechanism/COTS e.g. constant, oscillate

Table 11: Relation Category

RELATION	RELATION BETWEEN INPUT AND OUTPUT LINE OF MOTION	This attribute describes the geometrical relation between axes of input and output line of motion ex. coincident, non coplanar
	REVERSIBILITY	This attribute describes if the input and output lines of motions are reversible or not.

Table 12: Function Category

FUNCTION	Describes the major function performed by the mechanism

On the other hand, because of the differences between mechanisms and COTS, the DESCRIPTIVE category of COTS is different than of mechanisms.

Mechanisms have following sub categories under the DESCRIPTIVE category: 1. Name, 2. Group, 3. Components Involved, 4. Dimension, 5. DOF, 6.Visualization.

Machine Elements have following sub categories under the DESCRIPTIVE category: 1. Name, 2. Group, 3. Machine Element Category, 4. Related Physical Effect, 5. Visualization.

In the Mechanisms database the sub category of "Group" contains Mechanisms of same type, however in COTS database machine elements are grouped together based on the similar function performed by them. Table 13, and 14 shows the major category "DISCRIPTIVE", which is different for both the databases.

Table 13: Descriptive Category for Mechanism Database

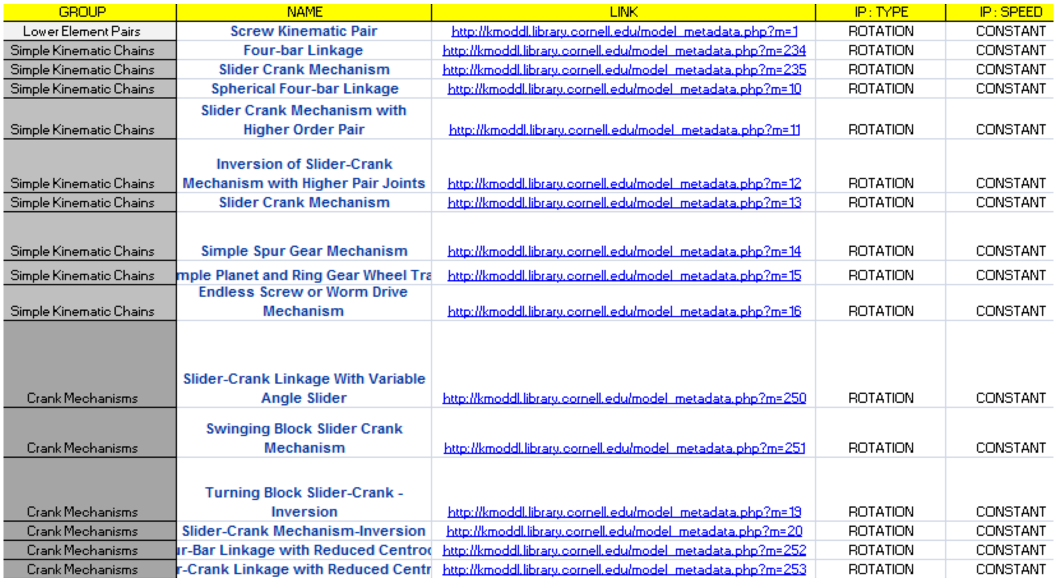
DISCRIPTIVE	NAME	Describes name of the mechanism
	GROUP	Describes the group to which the mechanism belongs ex. simple gear trains, ratchet mechanisms
	COMPONENTS INVOLVED	Describes major components involved in the mechanism ex. cam, spur gears
	DIMENSION	Describes spatial dimension of the mechanism ex. planar, linear
	DOF	Describes degrees of freedom
	VISUALIZATION	This provides a link to KMODDL web page where image and video simulation of the mechanisms are provided

Table 14: Descriptive Category for COTS Database

DESCRIPTIVE	NAME	Describes name of the machine element
	GROUP	Describes the group to which the machine element belongs ex. Couplings
	MACHINE ELEMENT CATEGORY	This describes the machine element category the given machine element belongs ex. Fastening Elements, Energy Storage Elements.
	VISUALIZATION	Provides an image representation of the machine element
	RELATED PHYSICAL EFFECT	Describes the physical effect embedded in machine element ex. COTS - Flexure coupling : PE - Bend

Each of the mechanism/COTS included in the database is actually a discrete representation of a general solution concept rather than any particular physical form (Kota & Chiou, 1992). Many other variations may exist in the physical implementation that makes use of the same working principle (Li, Tan, & Chan, 1996). For example, "rack-and-opinion" is a solution concept for achieving a conversion between translation and rotation. The solution concept itself does not preclude, say, using multiple racks or multiple pinions. The design suggested by the method should be treated as a suggestion for applying a certain working principle, and not the actual physical form. The suggested mechanism/COTS should be developed further into concept variants for evaluation.

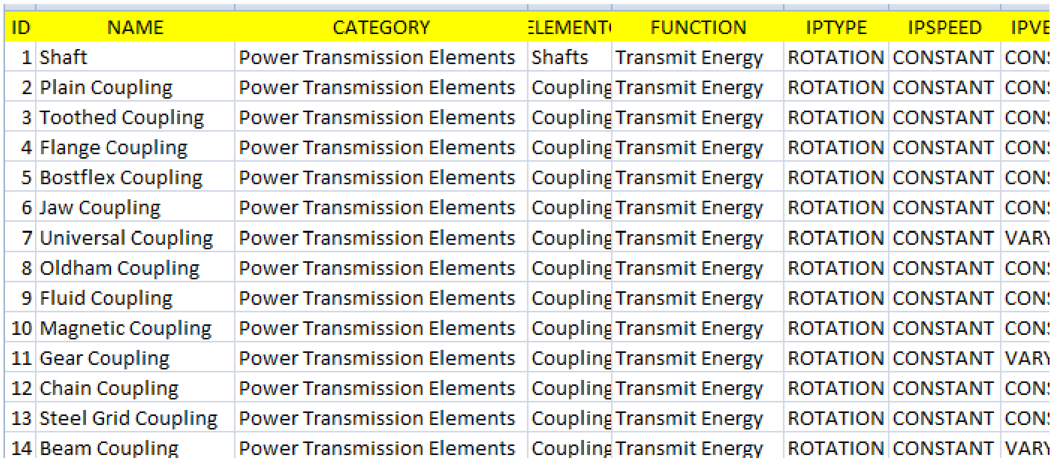
Using the representation discussed above, the mechanisms and COTS databases were created. For the mechanisms database the KMODDL digital library was used as the source. All the mechanisms stored in the source library were one by one transferred into the Holistic Ideation Mechanisms database. After removing mechanisms that were repeated, the current database contains 387 different mechanisms. Figure 38 shows excerpt from new Mechanism catalog.



GROUP	NAME	LINK	IP : TYPE	IP : SPEED
Lower Element Pairs	Screw Kinematic Pair	http://kmoddl.library.cornell.edu/model_metadata.php?m=1	ROTATION	CONSTANT
Simple Kinematic Chains	Four-bar Linkage	http://kmoddl.library.cornell.edu/model_metadata.php?m=234	ROTATION	CONSTANT
Simple Kinematic Chains	Slider Crank Mechanism	http://kmoddl.library.cornell.edu/model_metadata.php?m=235	ROTATION	CONSTANT
Simple Kinematic Chains	Spherical Four-bar Linkage	http://kmoddl.library.cornell.edu/model_metadata.php?m=10	ROTATION	CONSTANT
Simple Kinematic Chains	Slider Crank Mechanism with Higher Order Pair	http://kmoddl.library.cornell.edu/model_metadata.php?m=11	ROTATION	CONSTANT
Simple Kinematic Chains	Inversion of Slider-Crank Mechanism with Higher Pair Joints	http://kmoddl.library.cornell.edu/model_metadata.php?m=12	ROTATION	CONSTANT
Simple Kinematic Chains	Slider Crank Mechanism	http://kmoddl.library.cornell.edu/model_metadata.php?m=13	ROTATION	CONSTANT
Simple Kinematic Chains	Simple Spur Gear Mechanism	http://kmoddl.library.cornell.edu/model_metadata.php?m=14	ROTATION	CONSTANT
Simple Kinematic Chains	mple Planet and Ring Gear Wheel Tra	http://kmoddl.library.cornell.edu/model_metadata.php?m=15	ROTATION	CONSTANT
Simple Kinematic Chains	Endless Screw or Worm Drive Mechanism	http://kmoddl.library.cornell.edu/model_metadata.php?m=16	ROTATION	CONSTANT
Crank Mechanisms	Slider-Crank Linkage With Variable Angle Slider	http://kmoddl.library.cornell.edu/model_metadata.php?m=250	ROTATION	CONSTANT
Crank Mechanisms	Swinging Block Slider Crank Mechanism	http://kmoddl.library.cornell.edu/model_metadata.php?m=251	ROTATION	CONSTANT
Crank Mechanisms	Turning Block Slider-Crank - Inversion	http://kmoddl.library.cornell.edu/model_metadata.php?m=19	ROTATION	CONSTANT
Crank Mechanisms	Slider-Crank Mechanism-Inversion	http://kmoddl.library.cornell.edu/model_metadata.php?m=20	ROTATION	CONSTANT
Crank Mechanisms	ır-Bar Linkage with Reduced Centroc	http://kmoddl.library.cornell.edu/model_metadata.php?m=252	ROTATION	CONSTANT
Crank Mechanisms	r-Crank Linkage with Reduced Centr	http://kmoddl.library.cornell.edu/model_metadata.php?m=253	ROTATION	CONSTANT

Figure 38: Mechanism Catalog (Partial View)

For the COTS database the entries were taken from the textbook Mechanical Engineering Design, Shigley (Shigley, 2003) and Advanced Product Design Cots database by Shah, (Shah J. , Advanced Product Design Methodology, 2011). All the COTS were stored in the source library were one by one transferred into the Holistic Ideation COTS database. The current database contains 85 different machine elements types. Figure 39 shows an excerpt from new COTS catalog.



ID	NAME	CATEGORY	ELEMENT	FUNCTION	IPTYPE	IPSPEED	IPVE
1	Shaft	Power Transmission Elements	Shafts	Transmit Energy	ROTATION	CONSTANT	CON
2	Plain Coupling	Power Transmission Elements	Coupling	Transmit Energy	ROTATION	CONSTANT	CON
3	Toothed Coupling	Power Transmission Elements	Coupling	Transmit Energy	ROTATION	CONSTANT	CON
4	Flange Coupling	Power Transmission Elements	Coupling	Transmit Energy	ROTATION	CONSTANT	CON
5	Bostflex Coupling	Power Transmission Elements	Coupling	Transmit Energy	ROTATION	CONSTANT	CON
6	Jaw Coupling	Power Transmission Elements	Coupling	Transmit Energy	ROTATION	CONSTANT	CON
7	Universal Coupling	Power Transmission Elements	Coupling	Transmit Energy	ROTATION	CONSTANT	VARY
8	Oldham Coupling	Power Transmission Elements	Coupling	Transmit Energy	ROTATION	CONSTANT	CON
9	Fluid Coupling	Power Transmission Elements	Coupling	Transmit Energy	ROTATION	CONSTANT	CON
10	Magnetic Coupling	Power Transmission Elements	Coupling	Transmit Energy	ROTATION	CONSTANT	CON
11	Gear Coupling	Power Transmission Elements	Coupling	Transmit Energy	ROTATION	CONSTANT	VARY
12	Chain Coupling	Power Transmission Elements	Coupling	Transmit Energy	ROTATION	CONSTANT	CON
13	Steel Grid Coupling	Power Transmission Elements	Coupling	Transmit Energy	ROTATION	CONSTANT	CON
14	Beam Coupling	Power Transmission Elements	Coupling	Transmit Energy	ROTATION	CONSTANT	VARY

Figure 39: COTS Catalog (Partial View)

5.2 Improved Organizational Framework

Feedback from the users was a good way to understand some usability issues regarding V.1 and V.2. Following are the areas that needed improvement to increase the usability of the Holistic Ideation Tool.

5.2.1 Freedom Vs Constraint : Starting Point for Holistic Ideation

Though V.2 is built on the basis of V.1, the starting point for the conceptual design provided by both the versions is different. The V.1 required starting point for conceptual design to be the function decomposition for which it provided Function CAD tool. However, the user review on V.1 indicated that starting with the function decomposition for all the problems was a constraint. To remove this constraint V.2 did not provide any option as a starting to the designers, i.e., the designers were allowed to start by filling CEMC with anything from function, requirement, material, etc. The V.2 did not provide any tool for doing function decomposition. Because of this some of the users reported that they felt being lost while finding solutions. Consequently, there were some set of users for whom the usability of the tool decreased in V.1 as well as V.2.

To overcome this dilemma, the user should be provided with both the option and he/she should be allowed to choose any option according to his/her personal preference. Ideally the Holistic Ideation tool is able to let the designers start with any starting point as suggested by V.2 and give them a tool to start with the function decomposition if they choose to.

In V.3, we have solved this problem by providing the designer with both the options. A designer can start working by doing function decomposition using a Function CAD tool which is designed at DAL to integrate with the Holistic Ideation Tool. On other hand, if a designer feels constrained with this idea, he/she has a choice of starting from either CEMC or any ideation method.

5.2.2 Grouping of Ideation Methods

In V.1 and V.2, the ideation process was grouped into three stages:

1. Pre-Ideation

2. Ideation

3. Post-Ideation

Because of the grouping of the ideation methods under these labels, many times users were misled to follow the ideation methods one by one. To avoid this usability issue, in V.3, we have organized the methods under six different tool sets: *Problem Formulation Methods, Re-formulation, Standard Solutions, Generative methods, Gateways to external resources, Solution Synthesis and Process Monitoring.*

1. *Problem Formulation Methods:* This tool set contains modules "Function CAD" and "Morph Chart". These modules can be used in problem formulation stage. The Function CAD module in V.3 can be used for doing the function decomposition of the problem. This module is based on OSU's Function CAD tool and is based on a standardized ontology as discussed above. The CEMC module is retained from V.2 for providing the organizational framework for the tool. As discussed in review of V.2, CEMC does not require organization based only on function. A problem/sub-problem can be any aspect of the problem e.g. requirement, material, function, etc. The CEMC serves not only as a decomposition strategy - by organizing problems/sub-problems - but also as a way to organize solutions/sub-solutions. Because of this, CEMC can be regarded as the Design Workspace.

2. *Re-formulation*: For assisting the designers with problem re-formulation, in V.3, we have included "Word Diamond", "WordNet", and "Relational Algorithm" modules which are inherited from V.2. As discussed

above, these methods can be used for reframing and restructuring the problem. The CEMCs can have conjunctions and disjunctions, because of which they can be used to allow flexible and dynamic framing using the re-formulation methods.

3. *Standard Solutions*: Modules grouped under this set include "Mechanisms", "Machine Elements (COTS)", and "Artifacts" databases. While, Artifacts database is inherited from V.1, Mechanisms and COTS databases are designed as a part of my work. Usability of this tool set depends upon domain expertise of the user. The three separate knowledge bases (KB) contains: off the shelf electro-mechanical machine elements (83), linkages and mechanisms (387) and OSU Design Repository (artifact database with 5600 components) (Bohm, Stone, Simpson, & Steva, July 2008). The mechanisms knowledge base is linked to Cornel’s KMODDL (A Cornell University Web site), which contains animations of physical or virtual mechanisms. Each of these can be browsed or searched in various ways: by name, function, category, or in the case of mechanisms, by motion input/output types and constraints. The Standard Solutions KBs may lead to feasible and well established solutions.

4. *Generative methods*: In some cases a designer may fail to find a standard solution. In this case, designer can synthesize solutions from first principles, or using analogies from other domains, which can result in novel and creative solutions. Following Pahl & Beitz (Beitz & Pahl, 1996) and VDI2221 (VDI, 1997) systematic design, we have included several Physical Effects (PE) and Working Principles (WP) in our knowledge bases. Along with PE and WP modules, a second set of method for providing analogical reasoning is also

included. The designer can use the standard 39x39 TRIZ contradiction matrix for resolving conflicts between physical variables by finding up to 4 “invention principles” collected by Altshuller from various patents (Altshuller, 2001). The tool provides explanations of each principle and illustrations of its applications. Bio-TRIZ (Bogatyreva O. &., March,2009), which is based on conflict resolution principles from nature is also included in this tool set. PE,WP and TRIZ databases in V.3 are retained from V.1.

5. *External Resources*: As a holistic ideation system demands more and more back end knowledge encoded into the system, there is a need to find other resources. Other research communities have come up with diverse resources. We use these resources to leverage our efforts by providing gateways to external tools such as Imagenet® (ImageNet) and AskNature® (AskNature). While, ImageNet® uses Wordnet® structure and can be used for provocative stimuli in the form of pictures, the AskNature® can be a source of inspiration from nature. Furthermore, the Wikipedia® and the Google® search engine can be used for providing general search options.

6. *Solution Synthesis*: CEMC is used as a work space in the V.3. A designer can generate solutions for sub-functions independently and then combine them using the CEMC to get novel combinations. Adding a solution into the morph chart can be done manually (including graphical and textual description by the user) or they can come from one of the KBs; in the latter case the source of the inspiration is recorded by the system automatically. The designer can synthesize

complete design concepts from entries in a CEMC by combining solutions or system can randomly generate complete solutions for the designer.

7. *Process Monitoring:* Two modules are included in V.3 for the purpose of process monitoring (e.g. Find Ideation State, and Was the Strategy Useful ? module). These modules are retained from V.1. Using the "Find Ideation State" module the tool can characterize designers ideation state by asking him a standard set of questions. The other module is feedback module, which stores information such as the problem being pursued, main issues of concern, the level of satisfaction with solutions, the effectiveness of different tools used, and rationale for process being followed.

5.2.3 Disjunctions in CEMC

In the function decomposition, one function could be replaced by another function. A set of (sub) functions can be equivalent with other sets of (sub) functions. It is called "OR" relation or Disjunctions in a tree. Technically the CEMC way of organization provides the option that the user can replace any (sub) problem with another one. The user also can replace any sets of (sub) problem with another sets. As indicated in Figure 40, sub-problem F1 can be further decomposed in different ways, thus different morphology trees can be created.

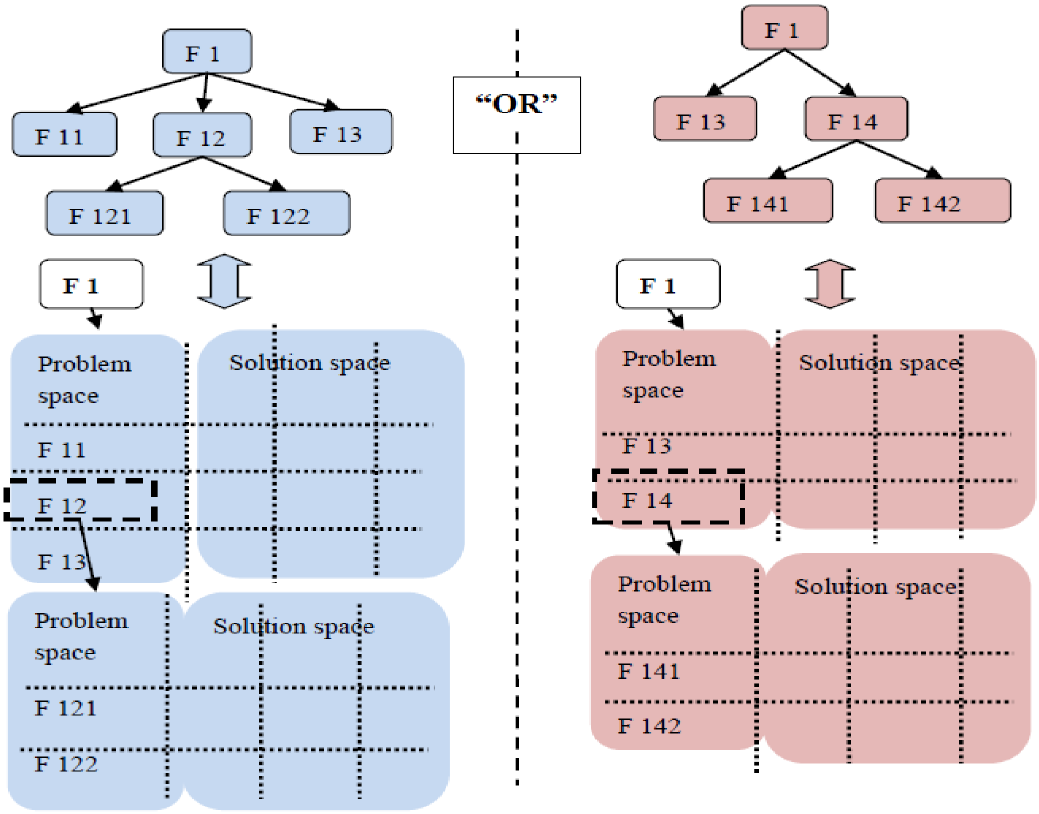


Figure 40: Disjunction in CEMC

But "Disjunctions" were not possible in V.2. In new Holistic Ideation tool we have added this functionality to make use of "Disjunctions", so that a designer can decompose a single problem in multiple different ways. This can be used by an user by creating multiple different high level CEMC for a single design problem. "Disjunctions" are implemented by designing the database with many to one relation in between CEMC and Problem tables.

5.2.4 Creating Solution Sets

The V.2 allowed designers to store multiple solutions for any problem. But, the designers were not provided a way to form a solution set for a parent problem by combining solutions from sub problems. Again, the CEMC structure technically provides this capacity to a designer but this was not implemented in the V.2. In V.3 we have addressed this problem and a designer can combine solution sets by combining different solutions.

5.2.5 Handling Deep Morphological Trees

The CEMC can be used efficiently as an organization framework for ideation because it can also store the function hierarchy. V.2 implemented CEMC in a way that multiple and different views of the same morphological or sub morphological chart was not possible. Many designers often failed to understand the hierarchy of their function decomposition once they created a deep function decomposition or took breaks in between multiple sessions. This issue often significantly reduced the usability of the tool. In V.3, therefore, we have resolved this main issue by allowing a designer to see multiple and different views of the same morphological chart or a sub morphological chart. This feature is implemented by providing different links for different CEMC which can be opened in individual tabs in a browser.

5.2.6 Auto-Population of Solutions.

In V.1, there is no structure to populate the ideas generated by user. So, V.2 provided the CEMC based workspace where one can store his/her ideas. Though it could be automated, the designer had to do it manually in V.2. This might cause hindrance in the flow of thoughts of the designer. In V.3, we have provided designers with the functionality to populate the solutions for problems automatically. The system can remember what problem a designer is working on and stores the new idea automatically in an appropriate position in the solution space. This feature is implemented by providing access to the knowledge bases through a pre-selected problem by user. Because of this pre-selection the system automatically links the newly created solution to the problem selected by the user.

5.2.7 Improving UI Aesthetics and Usability

In V.1 and V.2, UI appearances was given little attention. The most common feedback from users indicated that previous versions were not user friendly and lacked user friendly UI. In order to improve the tool usability, much effort were needed to improve this aspect of the tool.

Studies on UI appearance show that there is a strong correlation between a systems perceived aesthetics and perceived usability. Even though, the actual usability of two systems providing same functionality is same, perceived usability increases with perceived aesthetics of any system (Tractinsky, Katz, & Ikar, 2000).

To increase the usability of the tool a new UI was designed and implemented in V.3. As a major change, V.3 used whole area that is provided by the screen. The V.1 and V.2 were windows form based tools and failed to use the screen area to their advantage. The user feedback for those tools strongly suggested the need to improve the UI. Web based V.3 gave us very high control on "What to present", "How to present" and "Where to present".

To achieve this, we have implemented the "slicing and dicing" techniques used in multidimensional database based software tools (Russell & Cohn, 2012). These techniques enable users to easily and selectively extract and view the data from different points of view. It helps the user to visualize data and gather information specific to a dimension. Consequently, the user in V.3 can see all the parent and sub morph charts at the same time, which is not possible in V.2.

5.2.8 Studying Cognitive Process

A crucial moment in the conceptual design stage is when a stimulus inspires a designer to come up with a novel solution. To study ideation process one approach could be to study and document point of inspiration. There was no mechanism provided to record this point in detail.

While documenting a point of inspiration, one should take following things into account:

1. When it happened - Time

2. For what problem it happened - Problem

3. What entity from knowledge base provided the stimulus - Source

The V.3 automatically documents all three things whenever this happens with a designer while he/she is working on the problem.

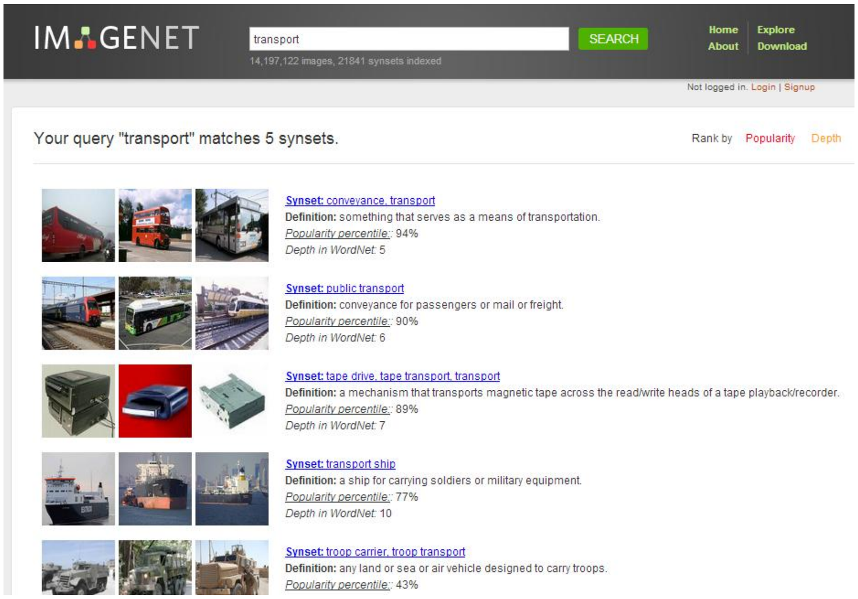
In the next chapter, we will discuss how all the above mentioned improvements are implemented.

5.3 External Resources

We understand that a tool such as the holistic ideation tool gets more and more efficient with the extra amount of knowledge brought into the system. There is a continuous need to search and add other resources to improve the tool. For this purpose the tool provides gateways to external tools, such as ImageNet (ImageNet) and AskNature (AskNature). These tools are retained from V.2 and they are implemented in V.3 in the same manner . ImageNet uses a hierarchical structure like WordNet and can be used for provocative stimuli in the form of pictures. The AskNature can also provide a source of inspiration by showing various examples from nature. Wikipedia (Wikipedia) and Google (Google) can be used for doing general searches.

5.3.1 ImageNet

The ImageNet tool provides an online image database for pictures. It uses the same hierarchical structure as WordNet where each node of the hierarchy is depicted by many images. One of the advantage is that the pictures are organized and the users can view related pictures of each node while surfing through the database. Currently the ImageNet have an average of over five hundred images per node. The ImageNet was derived from the WordNet and aims to provide pictures in place of words. These pictures could be used as “provocative stimuli” to overcome fixation and search in new directions. The ImageNet is an external implementation and can be accessed just by clicking on the ImageNet module button in our tool. Figure 41 shows different results generated by the ImageNet website for a query of "transport".



14,197,122 images, 21841 synsets indexed

Not logged in. Login | Signup

Your query "transport" matches 5 synsets.

Rank by Popularity Depth

Synset: conveyance, transport
Definition: something that serves as a means of transportation.
Popularity percentile:: 94%
Depth in WordNet: 5

Synset: public transport
Definition: conveyance for passengers or mail or freight.
Popularity percentile:: 90%
Depth in WordNet: 6

Synset: tape drive, tape transport, transport
Definition: a mechanism that transports magnetic tape across the read/write heads of a tape playback/recorder.
Popularity percentile:: 89%
Depth in WordNet: 7

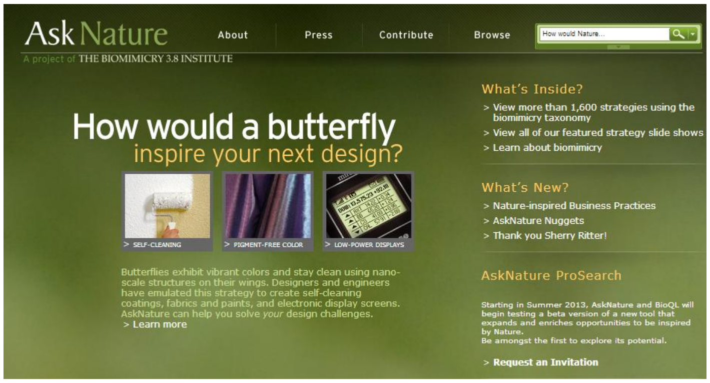
Synset: transport ship
Definition: a ship for carrying soldiers or military equipment.
Popularity percentile:: 77%
Depth in WordNet: 10

Synset: troop carrier, troop transport
Definition: any land or sea or air vehicle designed to carry troops.
Popularity percentile:: 43%

Figure 41: ImageNet Web Site

5.3.2 AskNature

The AskNature (Figure 42) is an online tool from the biomimicry community which can be used as an inspiration source. This website provides a tool for biologists to share their knowledge about any organism from nature, so a designer, architect, engineer, or chemist can find eco-friendly solutions to their problems. The AskNature can be called as the platform where biology and design meet, so bio-inspired solutions can be created. It is based on the design principle of bio-mimicry, which seeks sustainable solutions by emulating nature’s time-tested patterns and strategies, e.g., a solar cell inspired by a leaf. The fundamental idea behind bio-mimicry is that, nature, which is imaginative by necessity, has already solved many of the problems we are trying to solve, e.g., energy, food production, climate control, non-toxic chemistry, transportation, packaging, and a whole lot more.

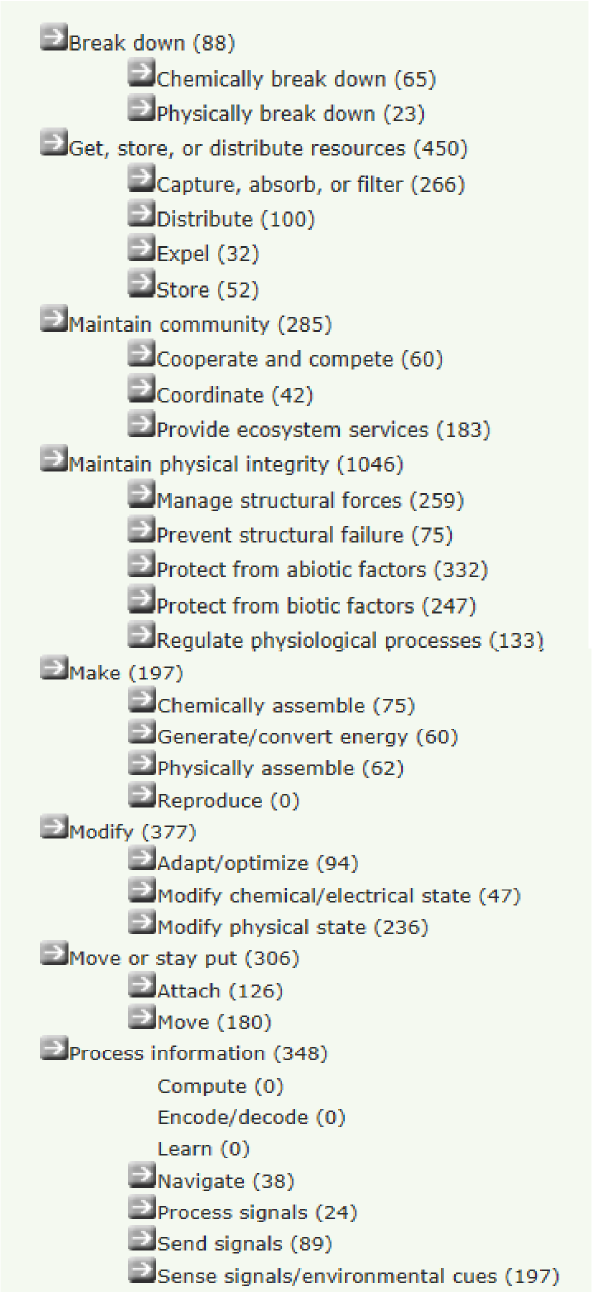


Butterflies exhibit vibrant colors and stay clean using nano-scale structures on their wings. Designers and engineers have emulated this strategy to create self-cleaning coatings, fabrics and paints, and electronic display screens. AskNature can help you solve *your* design challenges.

Starting in Summer 2013, AskNature and BioQL will begin testing a beta version of a new tool that expands and enriches opportunities to be inspired by Nature.
Be amongst the first to explore its potential.

Figure 42: AskNature Website

The AskNature uses its own bio-mimicry ontology as shown in Figure 43. This can be used to get inspiration for new ideas from nature. This external implementation can be accessed by just clicking on the AskNature module button in the holistic ideation tool.



- Break down (88)
  - Chemically break down (65)
  - Physically break down (23)
- Get, store, or distribute resources (450)
  - Capture, absorb, or filter (266)
  - Distribute (100)
  - Expel (32)
  - Store (52)
- Maintain community (285)
  - Cooperate and compete (60)
  - Coordinate (42)
  - Provide ecosystem services (183)
- Maintain physical integrity (1046)
  - Manage structural forces (259)
  - Prevent structural failure (75)
  - Protect from abiotic factors (332)
  - Protect from biotic factors (247)
  - Regulate physiological processes (133)
- Make (197)
  - Chemically assemble (75)
  - Generate/convert energy (60)
  - Physically assemble (62)
  - Reproduce (0)
- Modify (377)
  - Adapt/optimize (94)
  - Modify chemical/electrical state (47)
  - Modify physical state (236)
- Move or stay put (306)
  - Attach (126)
  - Move (180)
- Process information (348)
  - Compute (0)
  - Encode/decode (0)
  - Learn (0)
  - Navigate (38)
  - Process signals (24)
  - Send signals (89)
  - Sense signals/environmental cues (197)

Figure 43: AskNature Ontology

By using the sharing keywords (e.g. functions, subjects, objects, verbs) different cases in the AskNature tool can be searched. For example, the designer who is interested in a verb (e.g. accelerate), which may come from physicals effects, working principles, or any information during the design process, can take the verbs to Asknature to search bio-cases for analogical reasoning in bio domain (case: “Wing structure allows rapid acceleration: dragonfly”).

5.3.3 Google and Wikipedia

The Holistic Ideation tool also provide a gateway to the Google and Wikipedia websites. Google, which is search engine, and Wikipedia, which is a online encyclopedia project, can be used for general searches.

5.4 Software Process Flow

The Holistic Ideation Tool primarily has two parts: 1. The front-end User Interface (UI), and 2. The back-end Databases. As the CEMC organizational framework is retained from the V.2, the general software process flow has remained same. Figure 44 indicates process of making a Cascading Evolutionary Morphological Chart. The general design flow supported is as follow:

1. Problem decomposition.

2. Creating a CEMC and populating it with problems/sub-problems. Reframing and restructuring of this CEMC is possible at any time.

3. Generating ideas or solutions by using one or more ideation methods.

4. Combining the sub-solutions into one or more complete solutions.

5. Storing all the solution combinations generated by the user.

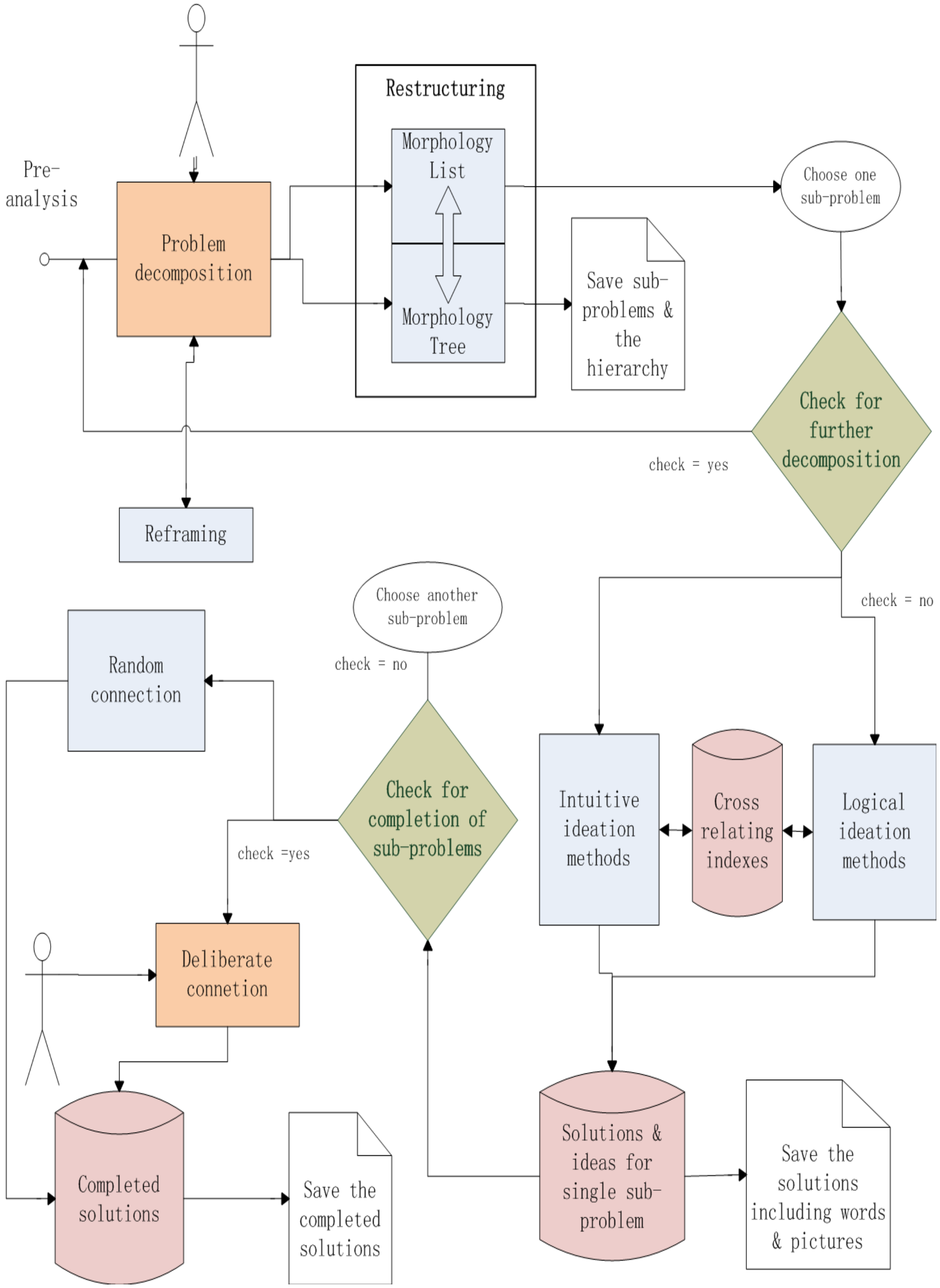

Pre-analysis
Problem decomposition
Restructuring
Morphology List
Morphology Tree
Save sub-problems & the hierarchy
Choose one sub-problem
Check for further decomposition
check = yes
check = no
Reframing
Choose another sub-problem
check = no
Random connection
Check for completion of sub-problems
check =yes
Intuitive ideation methods
Cross relating indexes
Logical ideation methods
Deliberate connetion
Completed solutions
Save the completed solutions
Solutions & ideas for single sub-problem
Save the solutions including words & pictures

Figure 44: Typical Process Flow

In the V.3 a designer can start with the problem decomposition. In this stage he/she can formulate the problem using the Function CAD tool for function decomposition. After this, he/she can also reformulate any problem using the reformulation modules such as Word Diamond, WordNet, and Relational Algorithm. After this, the designer can either restructure the CEMC until he/she feels satisfied or he/she can choose a single problem and start working on it. Once the designer chooses a problem, he/she can check if further decomposition is possible or not. If further decomposition is possible the designer can build the CEMC accordingly, otherwise he/she can start working on a problem. To do so, he/she can use the Find Ideation State module for process monitoring, which can suggest a suitable ideation method. The designer is also given the freedom to choose any ideation method or multiple ideation methods according his/her preference. Once the designer finds one or more solutions for all the leaf problems (i.e. a problem with no sub-problem), he/she can start thinking about combining feasible solution sets. With this combinations the designer can create solutions for the parent problems. This process can be continued till all the high level problems have at least one solution. At this stage the designer is ready to form the complete solutions for the problem. The designer can generate any number of complete solutions either by forced connection or by random connection. This complete solutions can be then stored into the system database. The system encodes any data stored by a designer and nobody except the designer is allowed view the data in decoded form. This way we protect the intellectual property right of the designer. Any session can be revisited and changed at any later time.

5.5 User Interface

In this section, we will discuss the UI for Holistic Ideation Tool - Version 3. UI is developed to support all the ideation methods and CEMC organizational structure. The V.3 is a web based tool and can be accessed through any browser with "http://www.ideationspace.com/ " address. As the tool offers many modules, we will discuss all the different aspects of the tool in following order:

1. Project Homepage

2. Organization of Modules.

3. Function Decomposition Tool.

4. CEMC Organizational Framework.

5. New Ideation Methods.

6. Logical Ideation Methods Retained from V.1.

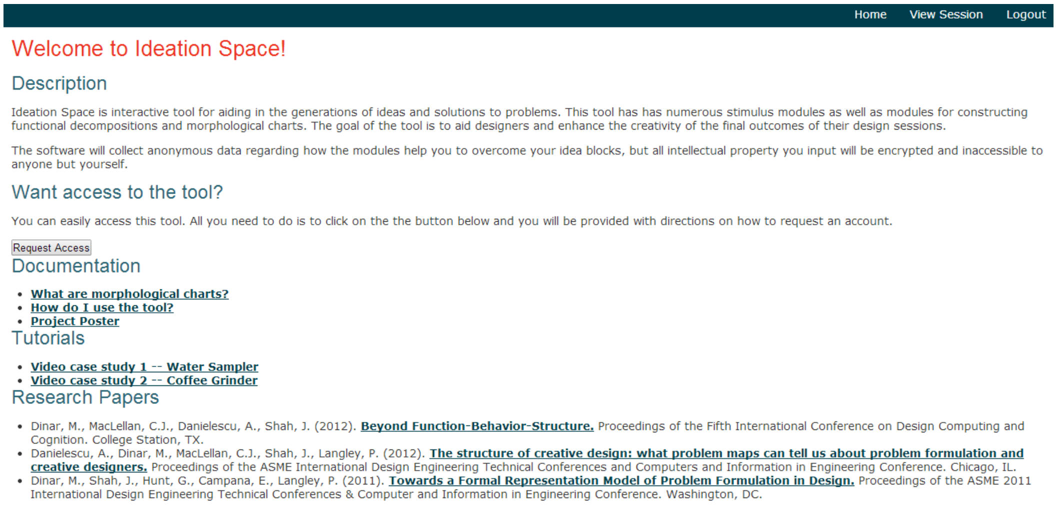
7. Intuitive Ideation Methods Retained and V.2.

8. External Resources.

9. Process Monitoring Tools.

5.5.1 Project Homepage

Figure 45 below shows the homepage of the Ideation website. In order to use the tool, the users need to set up an account with the tool. All the related information about how to create an account is described on the homepage. After creating the account, the users can create any number of sessions which can be used to work on multiple problems (Figure 46).



Welcome to Ideation Space!

Description

Ideation Space is interactive tool for aiding in the generations of ideas and solutions to problems. This tool has has numerous stimulus modules as well as modules for constructing functional decompositions and morphological charts. The goal of the tool is to aid designers and enhance the creativity of the final outcomes of their design sessions.

The software will collect anonymous data regarding how the modules help you to overcome your idea blocks, but all intellectual property you input will be encrypted and inaccessible to anyone but yourself.

Want access to the tool?

You can easily access this tool. All you need to do is to click on the the button below and you will be provided with directions on how to request an account.

Documentation

- **What are morphological charts?**
- **How do I use the tool?**
- **Project Poster**

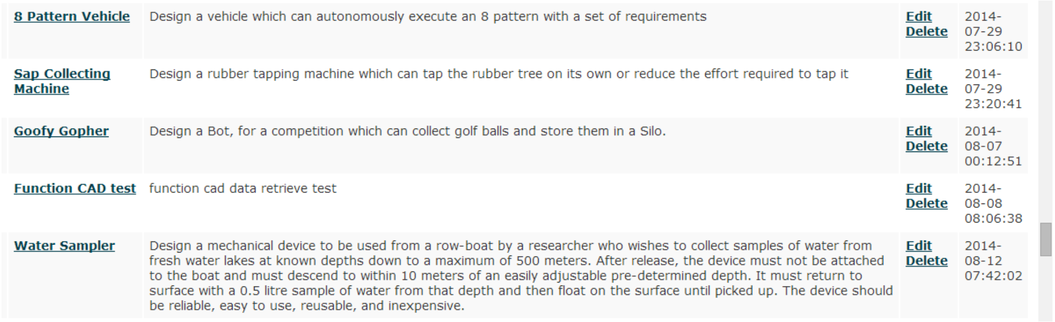
Tutorials

- **Video case study 1 -- Water Sampler**
- **Video case study 2 -- Coffee Grinder**

Research Papers

- Dinar, M., MacLellan, C.J., Danielescu, A., Shah, J. (2012). **Beyond Function-Behavior-Structure.** Proceedings of the Fifth International Conference on Design Computing and Cognition. College Station, TX.
- Danielescu, A., Dinar, M., MacLellan, C.J., Shah, J., Langley, P. (2012). **The structure of creative design: what problem maps can tell us about problem formulation and creative designers.** Proceedings of the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Chicago, IL.
- Dinar, M., Shah, J., Hunt, G., Campana, E., Langley, P. (2011). **Towards a Formal Representation Model of Problem Formulation in Design.** Proceedings of the ASME 2011 International Design Engineering Technical Conferences & Computer and Information in Engineering Conference. Washington, DC.

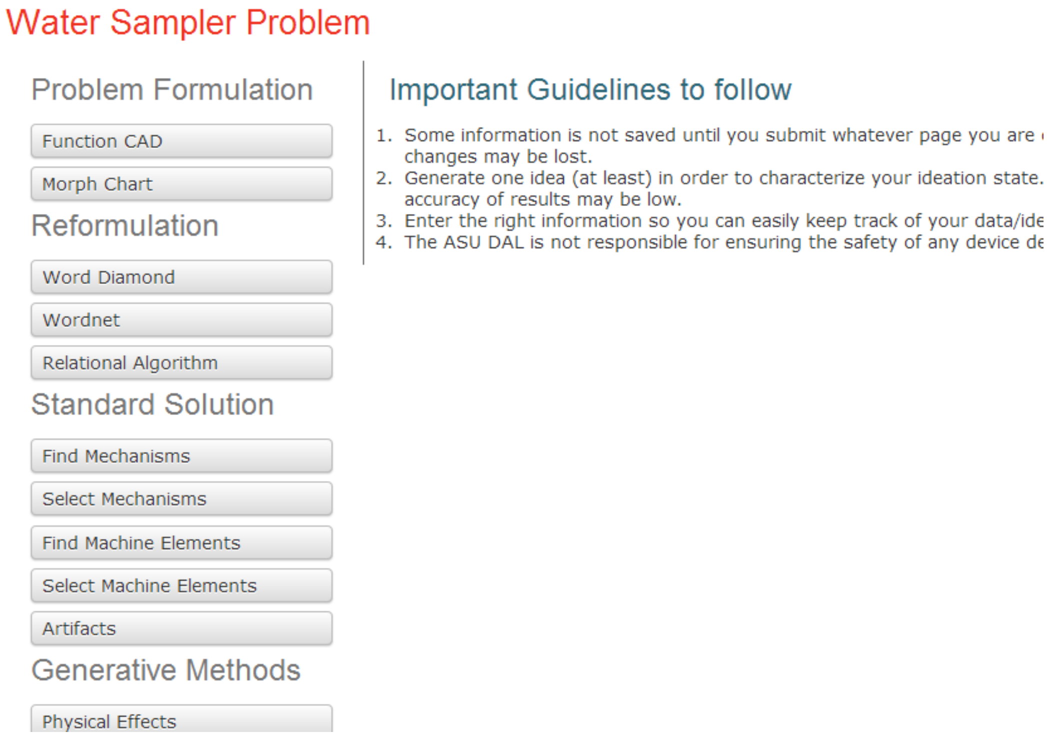
Figure 45: Ideation Space Homepage



8 Pattern Vehicle	Design a vehicle which can autonomously execute an 8 pattern with a set of requirements	**Edit** **Delete**	2014-07-29 23:06:10
Sap Collecting Machine	Design a rubber tapping machine which can tap the rubber tree on its own or reduce the effort required to tap it	**Edit** **Delete**	2014-07-29 23:20:41
Goofy Gopher	Design a Bot, for a competition which can collect golf balls and store them in a Silo.	**Edit** **Delete**	2014-08-07 00:12:51
Function CAD test	function cad data retrieve test	**Edit** **Delete**	2014-08-08 08:06:38
Water Sampler	Design a mechanical device to be used from a row-boat by a researcher who wishes to collect samples of water from fresh water lakes at known depths down to a maximum of 500 meters. After release, the device must not be attached to the boat and must descend to within 10 meters of an easily adjustable pre-determined depth. It must return to surface with a 0.5 litre sample of water from that depth and then float on the surface until picked up. The device should be reliable, easy to use, reusable, and inexpensive.	**Edit** **Delete**	2014-08-12 07:42:02

Figure 46: Multiple Session Per User

Each of the different sessions can be used to work on a different problems. The Figure 47 shows the homepage for any design problem/session. This is the place where users are given access to all the ideation methods of the tool.



Important Guidelines to follow

1. Some information is not saved until you submit whatever page you are
changes may be lost.
2. Generate one idea (at least) in order to characterize your ideation state.
accuracy of results may be low.
3. Enter the right information so you can easily keep track of your data/ide
4. The ASU DAL is not responsible for ensuring the safety of any device de

Figure 47: Project Homepage

5.5.2 Organization of Modules

As discussed previously, in V.3, we have organized the modules under seven different tool sets: Problem Formulation Methods, Re-formulation, Standard Solutions, Generative methods, Gateways to external resources, Solution Synthesis and process monitoring (Figure 48). Table 15 shows different modules contained in this tool sets.

Table 15: Tool Set and Ideation Modules Relation

TOOL SET	**IDEATION MODULES**
1. Problem Formulation Methods:	Function CAD
	Morph Chart
2. Re-formulation:	Word Diamond
	WordNet
	Relational Algorithm
3. Standard Solutions:	Find Mechanisms
	Select Mechanisms
	Find Machine Elements
	Select Machine Elements
	Artifacts
4. Generative methods:	Physical Effects
	Working Principles
	TRIZ
	Bio-TRIZ
5. External Resources:	ImageNet
	AskNature
	Google
	Wikipedia
6. Solution Synthesis:	CEMC
7. Process Monitoring:	Find Ideation State
	Was the Strategy Useful?

In the next section we will present software implementation of all the ideation modules mentioned in table 15.

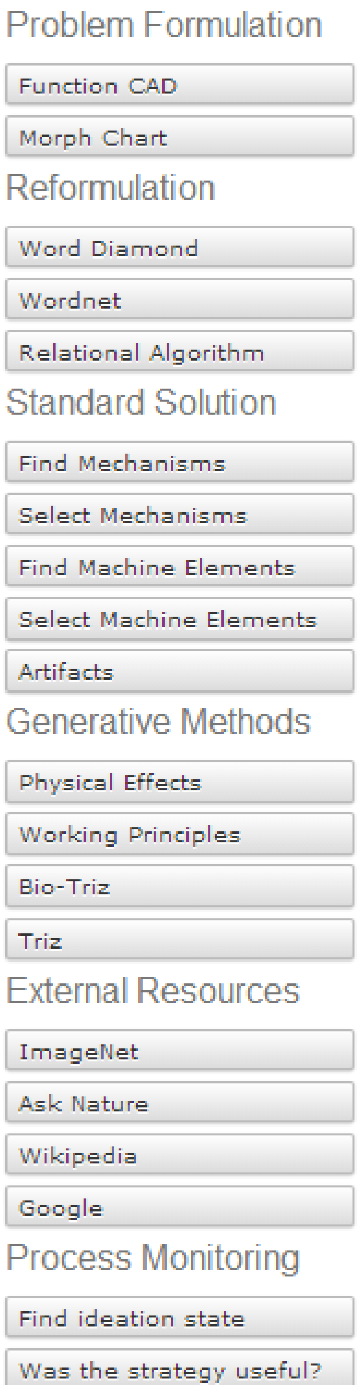


Figure 48: New Organization of Methods

5.5.3 Function Decomposition Tool

The Function CAD could be the key part of pre-ideation stage in our holistic ideation framework. The Function CAD module in V.3 implementation is based upon the Function CAD tool from Oregon State University. Figure 49 shows the UI for Function CAD tool provided in V.3.

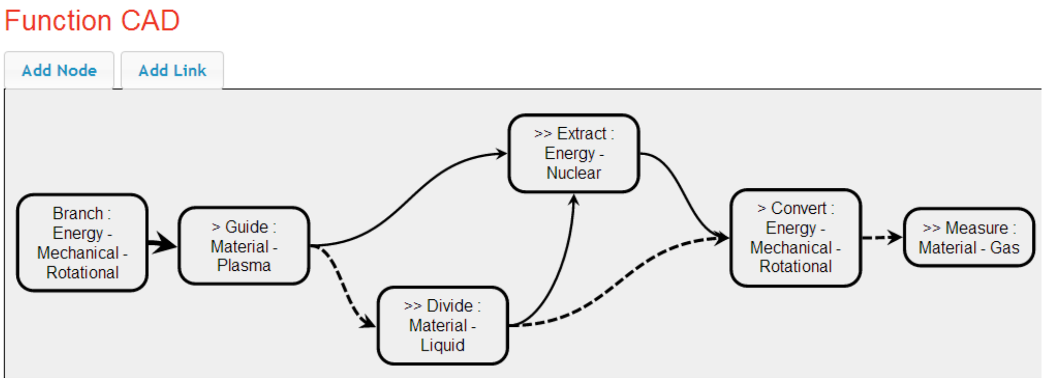
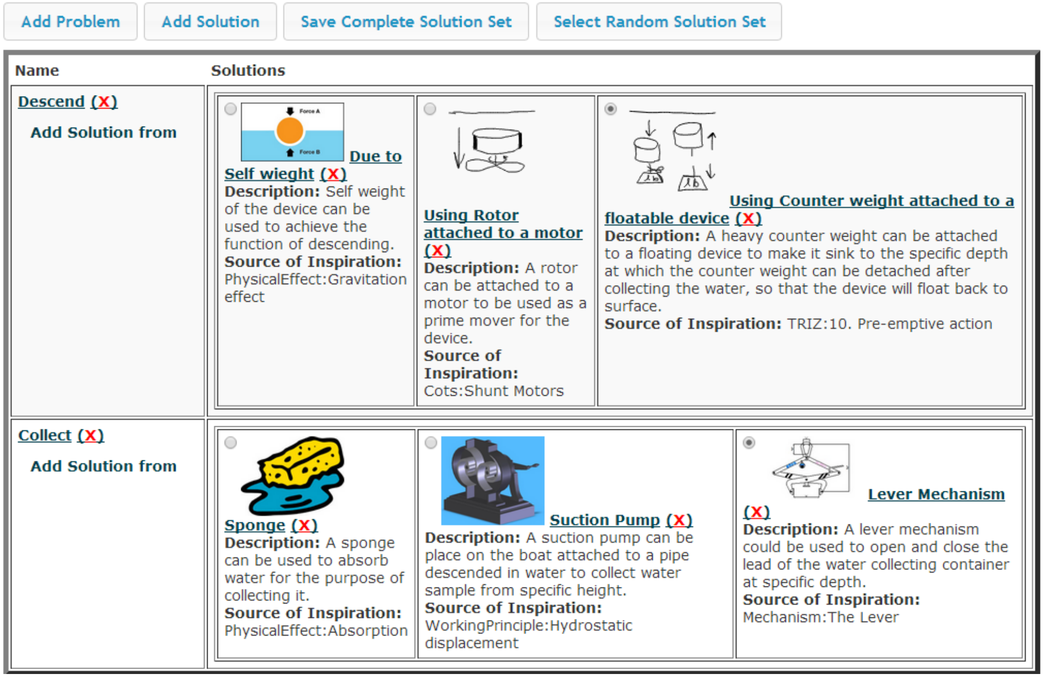

Function CAD
Add Node
Add Link
Branch : Energy - Mechanical - Rotational
> Guide : Material - Plasma
>> Extract : Energy - Nuclear
>> Divide : Material - Liquid
> Convert : Energy - Mechanical - Rotational
>> Measure : Material - Gas

Figure 49: Example of a Function Structure Drawn Using Function CAD Module

5.5.4 CEMC Organizational Framework.

The main interface for the CEMC is divided into two different parts: problem space and solution space. In V.3, even during problem decomposition, all the ideation methods are available to use at any time. As Figure 50 shows, problems are stored in a single column on the left hand side, while the solutions for each of the problem are stored on the right hand side of the CEMC in a row. The framework allows the user to add and edit the name and description of any problem and solution at any time.

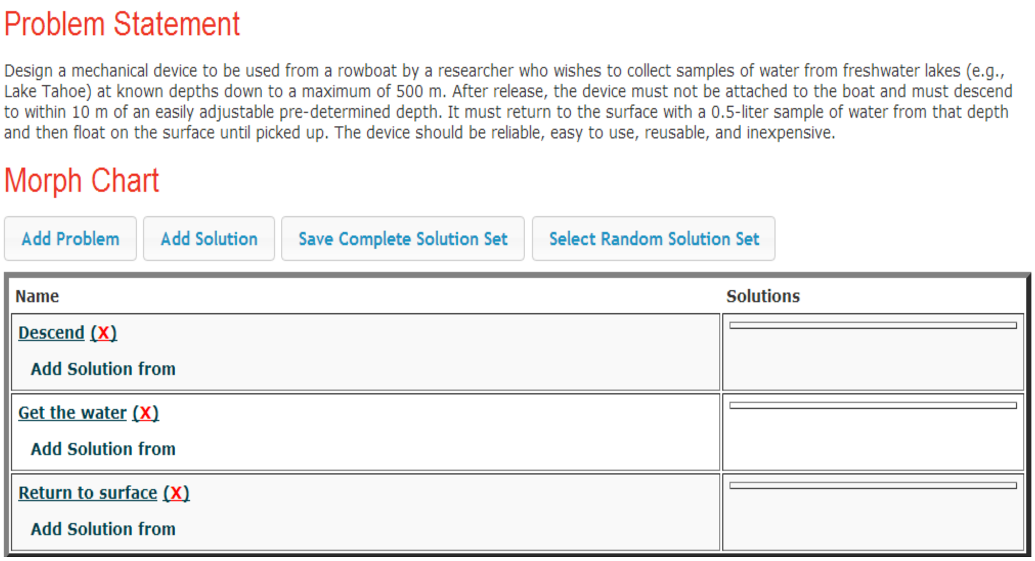
For instance, in Figure 50, "Descend" and "Collect" are the problems added by a designer. "Due to Self Weight", "Using Rotor attached to a motor", and "Using Counter weight attached to a floatable device" are solutions generated and documented by the designer for fulfilling the problem "Descend". "Sponge", "Suction Pump", and "Lever Mechanism" were added by the designer as the solutions corresponding to the "Collect" problem.



Name	Solutions		
Descend (X) Add Solution from	○ Due to Self wieght (X) **Description:** Self weight of the device can be used to achieve the function of descending. **Source of Inspiration:** PhysicalEffect:Gravitation effect	○ Using Rotor attached to a motor (X) **Description:** A rotor can be attached to a motor to be used as a prime mover for the device. **Source of Inspiration:** Cots:Shunt Motors	● Using Counter weight attached to a floatable device (X) **Description:** A heavy counter weight can be attached to a floating device to make it sink to the specific depth at which the counter weight can be detached after collecting the water, so that the device will float back to surface. **Source of Inspiration:** TRIZ:10. Pre-emptive action
Collect (X) Add Solution from	○ Sponge (X) **Description:** A sponge can be used to absorb water for the purpose of collecting it. **Source of Inspiration:** PhysicalEffect:Absorption	○ Suction Pump (X) **Description:** A suction pump can be place on the boat attached to a pipe descended in water to collect water sample from specific height. **Source of Inspiration:** WorkingPrinciple:Hydrostatic displacement	● Lever Mechanism (X) **Description:** A lever mechanism could be used to open and close the lead of the water collecting container at specific depth. **Source of Inspiration:** Mechanism:The Lever

Figure 50: Example of Single Layered CEMC

This is noteworthy that the CEMC example shown in Figure 50 is single layered morphological chart, i.e., designer have not added any low level sub problems under any of the high level parent problems. As mentioned earlier, the V.3 supports creation of a cascading morph chart through selection of a parent problem for any of the sub-problem. Any multilayer problem decomposition can be converted into a CEMC in V.3 by adding all the problems into the tool one by one. The order in which the user adds these problems does not matter, unless the parent problem for each of the sub problem is selected appropriately. Figure 51 shows a problem decomposition with only high level problems (Descend, Get the water, and Return to Surface) added into CEMC. The problems and solution can be added by clicking the "Add Problem" or "Add Solution" buttons shown on top of the CEMC.



Problem Statement

Design a mechanical device to be used from a rowboat by a researcher who wishes to collect samples of water from freshwater lakes (e.g., Lake Tahoe) at known depths down to a maximum of 500 m. After release, the device must not be attached to the boat and must descend to within 10 m of an easily adjustable pre-determined depth. It must return to the surface with a 0.5-liter sample of water from that depth and then float on the surface until picked up. The device should be reliable, easy to use, reusable, and inexpensive.

Morph Chart

Name	Solutions
Descend (X) Add Solution from	
Get the water (X) Add Solution from	
Return to surface (X) Add Solution from	

Figure 51: CEMC with High Level Problems

Once the high level parent problems are added, the users can add the low level child problems. While adding the child/sub problems, the user has to specify the parent problem. Figure 52 shows the fully formed CEMC. If a sub-problem is added to a parent problem, a green color plus sign (➕) is displayed in front of the name of the parent problem. The sub morphological chart for that problem can be easily accessed by clicking on the "➕" sign. Any problem or solution can be edited/updated by clicking on their respective names in the CEMC. On the other hand, to delete one, a designer can just click on the red cross (X) sign displayed in front of the name of the problem or solution.

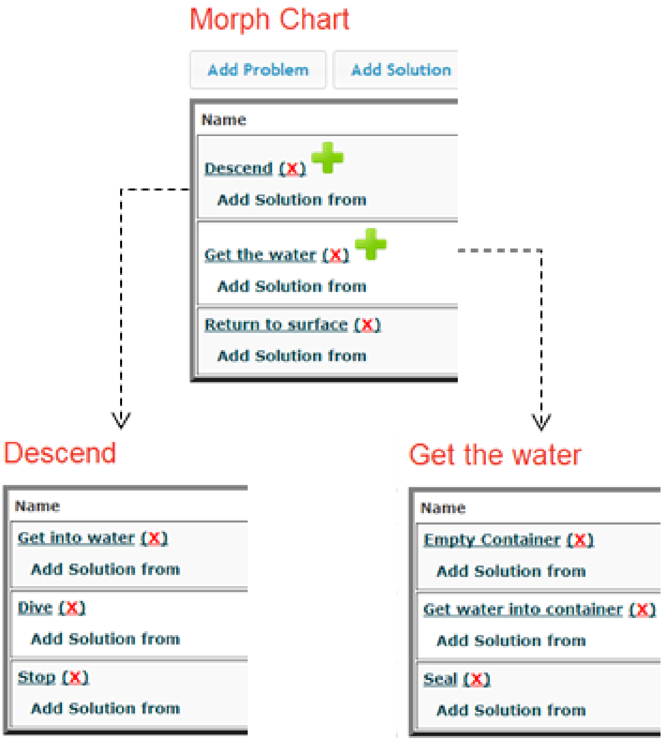


Figure 52: Multilayer CEMC Example

After creating the CEMC with all the problems and sub-problems, a user can start finding solutions for any of the problem, though it is advised to start with leaf problems, i.e., a problem containing no sub problems. The user can use any ideation method to find a solution for a particular problem. To do that, the V.3 provides two ways: Manual and Automatic. A solution can be manually added in the CEMC by selecting any ideation method from session homepage (Figure 47) and freely looking through the knowledge base for a stimulus. In this method, system does not know what specific function a designer is working on, so if the designer finds a solution he/she has to add it to the CEMC manually. On the other hand, a solution can be automatically added to CEMC by searching the knowledge base for a particular problem. This can be done by selecting the ideation methods from the list of methods provided by hovering the mouse over "Add Solution From" button given below each problem name. By this method, system knows what problem the designer is working on and the solution found gets added automatically to the respective problem. Figure 53, shows the manual and automatic addition of the solution to the CEMC.

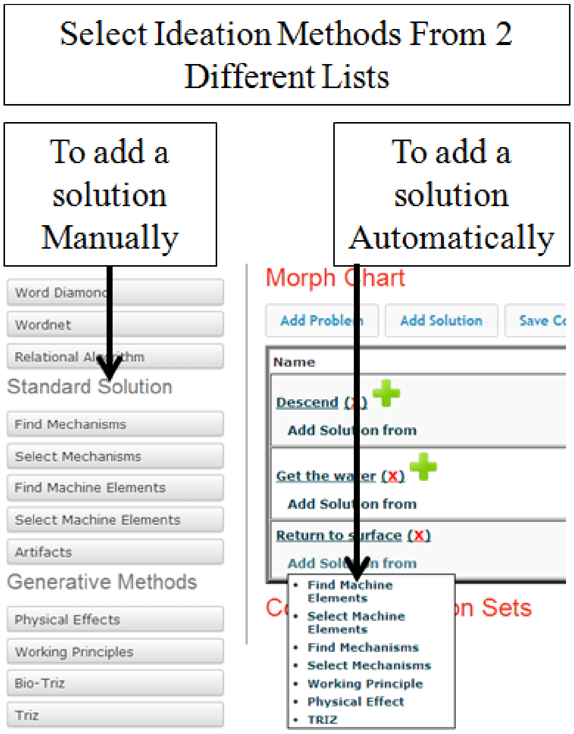
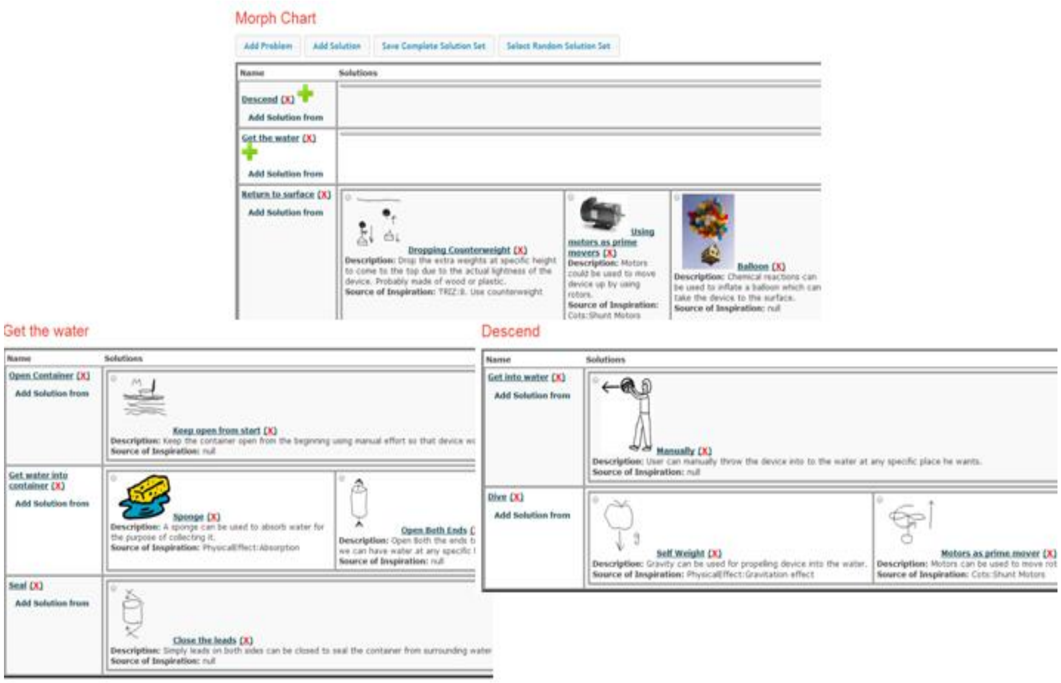


Figure 53: Automatic and Manual Method for Solution Addition

Using these two methods, a designer can populate the CEMC with solutions. Figure 54 shows CEMC with solutions added for leaf problems.



Morph Chart

Name	Solutions		
Descend (X) Add Solution from			
Get the water (X) Add Solution from			
Return to surface (X) Add Solution from	Dropping Counterweight (X) **Description:** Drop the extra weights at specific height to come to the top due to the actual lightness of the device. Probably made of wood or plastic. **Source of Inspiration:** TRIZ:8. Use counterweight	Using motors as prime movers (X) **Description:** Motors could be used to move device up by using rotors. **Source of Inspiration:** Cots:Shunt Motors	Balloon (X) **Description:** Chemical reactions can be used to inflate a balloon which can take the device to the surface. **Source of Inspiration:** null

Get the water

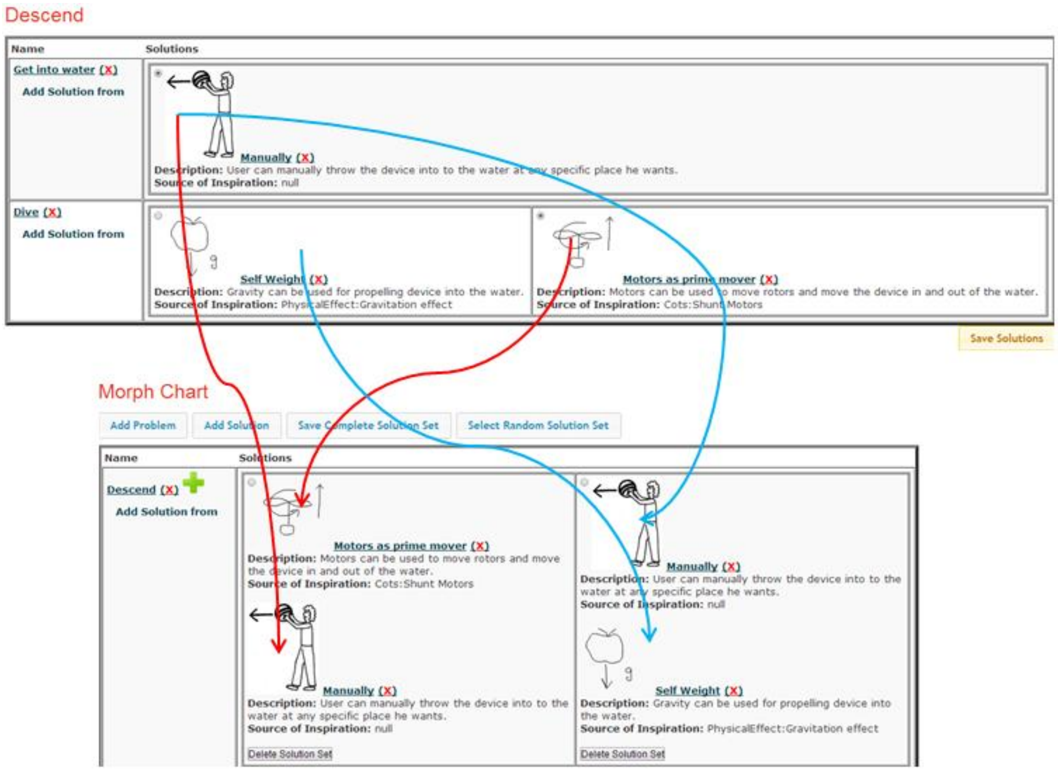
Name	Solutions	
Open Container (X) Add Solution from	Keep open from start (X) **Description:** Keep the container open from the beginning using manual effort so that device wo **Source of Inspiration:** null	
Get water into container (X) Add Solution from	Sponge (X) **Description:** A sponge can be used to absorb water for the purpose of collecting it. **Source of Inspiration:** PhysicalEffect:Absorption	Open Both Ends (**Description:** Open Both the ends t we can have water at any specific l **Source of Inspiration:** null
Seal (X) Add Solution from	Close the leads (X) **Description:** Simply leads on both sides can be closed to seal the container from surrounding water **Source of Inspiration:** null	

Descend

Name	Solutions	
Get into water (X) Add Solution from	Manually (X) **Description:** User can manually throw the device into to the water at any specific place he wants. **Source of Inspiration:** null	
Dive (X) Add Solution from	Self Weight (X) **Description:** Gravity can be used for propeling device into the water. **Source of Inspiration:** PhysicalEffect:Gravitation effect	Motors as prime mover (X) **Description:** Motors can be used to move rot **Source of Inspiration:** Cots:Shunt Motors

Figure 54: CEMC with Solutions Added to Leaf Problems

The solutions to a parent problem can be added by combining solution sets from solutions stored for sub-problems. This can be done by selecting solutions through checkbox given in front of the name of the solution and clicking "Save Solutions" button provided on bottom right corner of each morphological chart. Figure 55 and 56 shows solution sets created for parent problem "Descend" and "Get the Water".



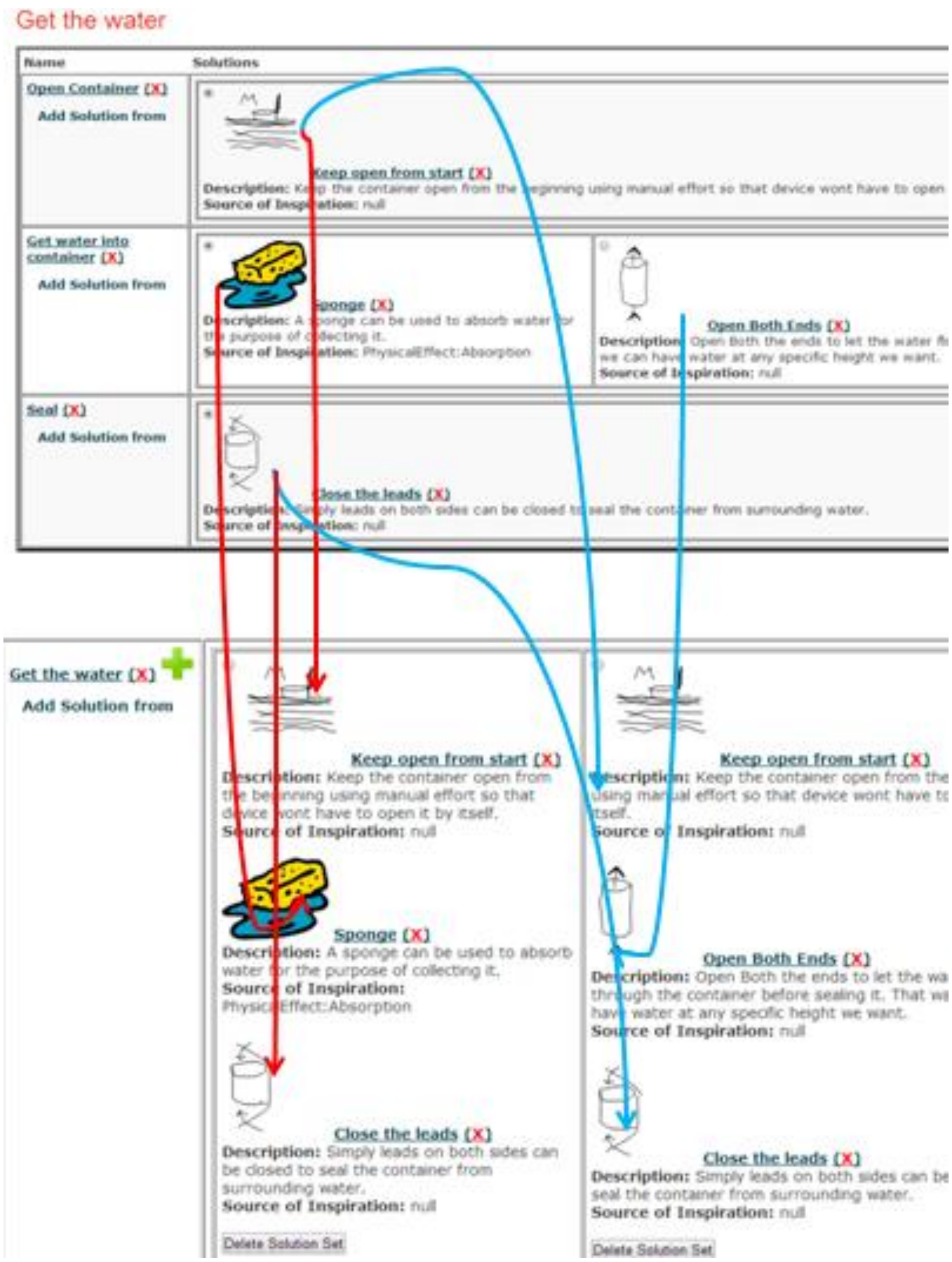
Descend

Name	Solutions
Get into water (X) Add Solution from	Manually (X) Description: User can manually throw the device into to the water at any specific place he wants. Source of Inspiration: null
Dive (X) Add Solution from	Self Weight (X) Description: Gravity can be used for propelling device into the water. Source of Inspiration: PhysicalEffect:Gravitation effect Motors as prime mover (X) Description: Motors can be used to move rotors and move the device in and out of the water. Source of Inspiration: Cots:Shunt Motors

Morph Chart

Name	Solutions	
Descend (X) Add Solution from	Motors as prime mover (X) Description: Motors can be used to move rotors and move the device in and out of the water. Source of Inspiration: Cots:Shunt Motors Manually (X) Description: User can manually throw the device into to the water at any specific place he wants. Source of Inspiration: null Delete Solution Set	Manually (X) Description: User can manually throw the device into to the water at any specific place he wants. Source of Inspiration: null Self Weight (X) Description: Gravity can be used for propelling device into the water. Source of Inspiration: PhysicalEffect:Gravitation effect Delete Solution Set

Figure 55: Solution Sets for Parent Problem "Descend"



Get the water

Name	Solutions	
Open Container (X) Add Solution from	Keep open from start (X) **Description:** Keep the container open from the beginning using manual effort so that device wont have to open **Source of Inspiration:** null	
Get water into container (X) Add Solution from	Sponge (X) **Description:** A sponge can be used to absorb water for the purpose of collecting it. **Source of Inspiration:** PhysicalEffect:Absorption	Open Both Ends (X) **Description:** Open Both the ends to let the water fl we can have water at any specific height we want. **Source of Inspiration:** null
Seal (X) Add Solution from	Close the leads (X) **Description:** Simply leads on both sides can be closed to seal the container from surrounding water. **Source of Inspiration:** null	

Get the water (X)

Add Solution from

Keep open from start (X)
Description: Keep the container open from the beginning using manual effort so that device wont have to open it by itself.
Source of Inspiration: null

Sponge (X)
Description: A sponge can be used to absorb water for the purpose of collecting it.
Source of Inspiration: PhysicalEffect:Absorption

Close the leads (X)
Description: Simply leads on both sides can be closed to seal the container from surrounding water.
Source of Inspiration: null

Delete Solution Set

Keep open from start (X)
Description: Keep the container open from the using manual effort so that device wont have to itself.
Source of Inspiration: null

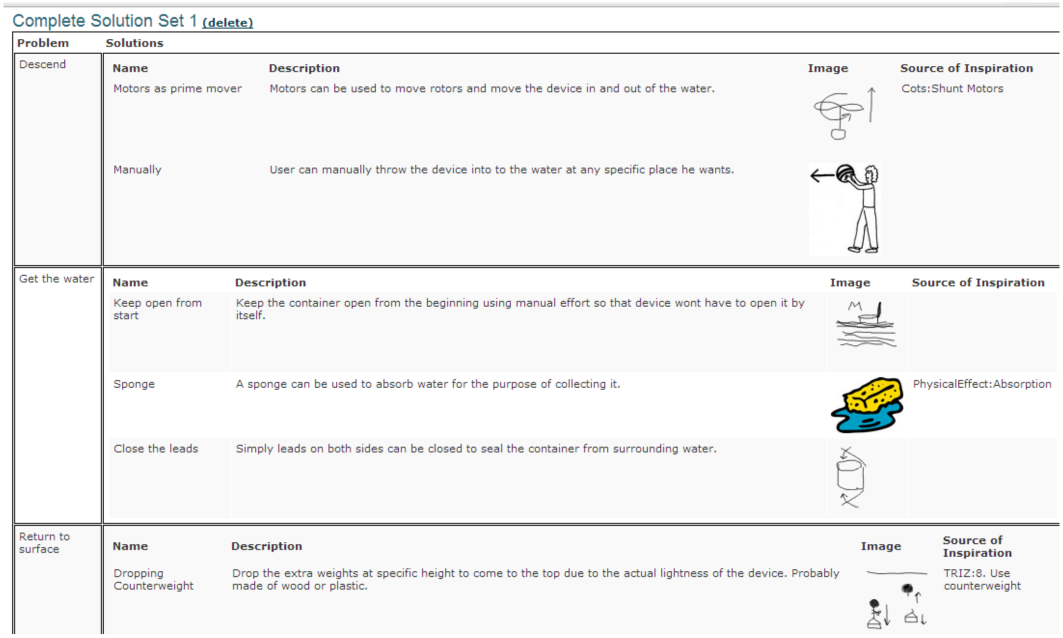
Open Both Ends (X)
Description: Open Both the ends to let the wa through the container before sealing it. That we have water at any specific height we want.
Source of Inspiration: null

Close the leads (X)
Description: Simply leads on both sides can be seal the container from surrounding water.
Source of Inspiration: null

Delete Solution Set

Figure 56: Solution Sets for Parent Problem "Get the water"

After filling the CEMC with solutions, the user can create complete solution sets. These complete solution sets can be created by two ways: Forced Connection or Random Connection. In forced connection method, a user has to select the combination of solutions for each of the problem manually. This can be done by clicking the checkboxes given in front of the name of a solution. On the other hand, tool can create random solution sets for user by randomly connecting solutions from every row with each other. This can be done by clicking the "Select Random Solution Set" button provided on top of the CEMC. Both these processes give designer a complete solution set for the design problem. If the user thinks that the complete solution set is feasible, he/she can store it in the database by clicking the "Save Complete Solution Button" provided over the CEMC. Figure 57 shows an example of a complete solution set saved by the designer.

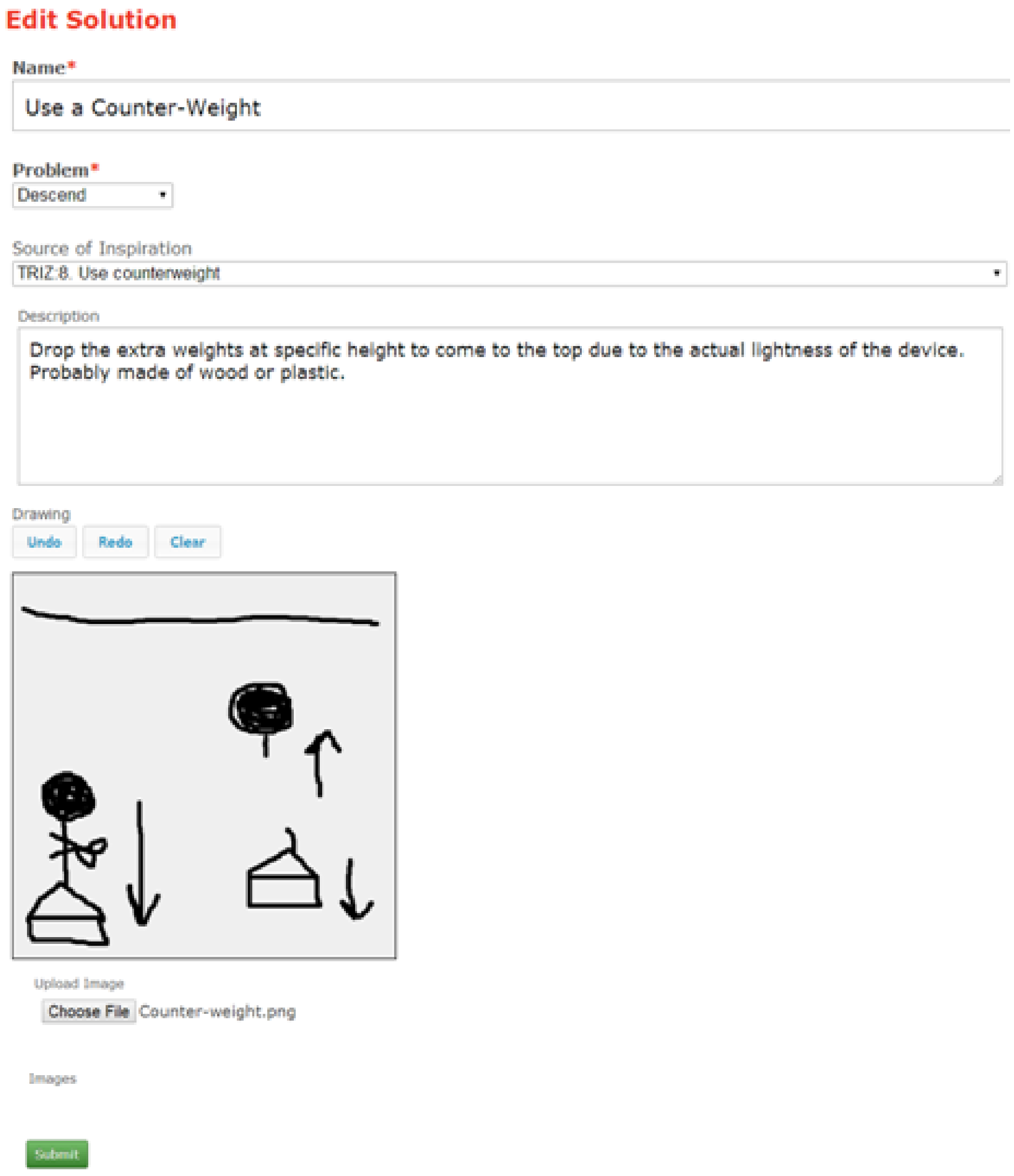


Complete Solution Set 1 (delete)

Problem	Solutions			
	Name	Description	Image	Source of Inspiration
Descend	Motors as prime mover	Motors can be used to move rotors and move the device in and out of the water.		Cots:Shunt Motors
	Manually	User can manually throw the device into to the water at any specific place he wants.		
Get the water	Keep open from start	Keep the container open from the beginning using manual effort so that device wont have to open it by itself.		
	Sponge	A sponge can be used to absorb water for the purpose of collecting it.		PhysicalEffect:Absorption
	Close the leads	Simply leads on both sides can be closed to seal the container from surrounding water.		
Return to surface	Dropping Counterweight	Drop the extra weights at specific height to come to the top due to the actual lightness of the device. Probably made of wood or plastic.		TRIZ:8. Use counterweight

Figure 57: Example of Complete Solution Set

Whenever a solution is added to the CEMC, it can be documented in following ways: 1. Textual documentation, 2. Graphical Documentation, and 3. Source of Inspiration. For textual documentation a text box is provided where description of the problem can be written. For graphical documentation, user can either use the sketching tool provided by V.3 or he/she can upload images. While only one sketch is allowed per solution, there is no limit on number of images attached with the solution. The "Source of Inspiration" provides the list of all the entries from all the knowledge bases. When the designers use the automatic population method for finding a solution, "Source of Inspiration" field gets automatically filled by the system. This field indicates the entity in the KB that inspired the designer to come up with the solution. It can be changed at any time by the designer. Figure 58 shows the documentation tools provided by V.3.



Edit Solution

Name*

Use a Counter-Weight

Problem*

Descend

Source of Inspiration

TRIZ:8. Use counterweight

Description

Drop the extra weights at specific height to come to the top due to the actual lightness of the device. Probably made of wood or plastic.

Drawing

Upload Image

Counter-weight.png

Images

Figure 58: Documentation Tools

5.5.5 New Ideation Methods

In this section, we will discuss how all the ideation methods UI is designed and how can each ideation method be used to find a solution.

5.5.5.1 Mechanisms and COTS

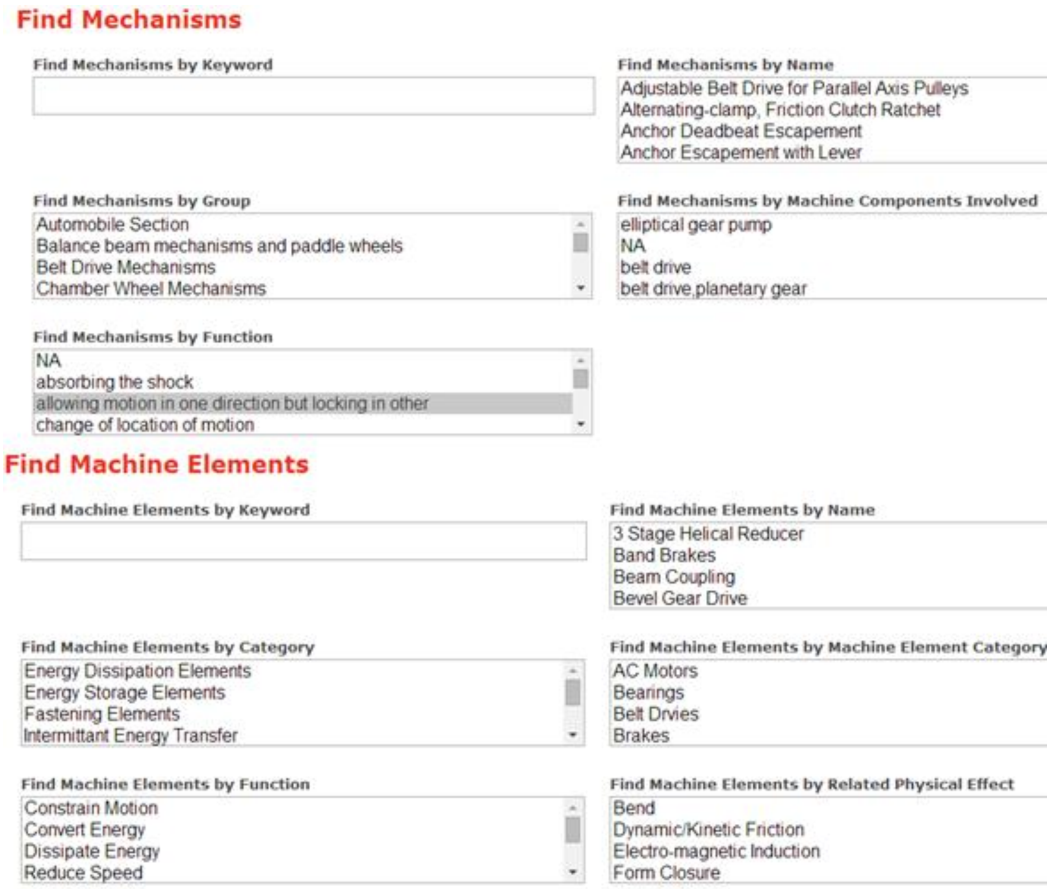
In the previous chapter we discussed the need of the mechanisms and COTS modules in Holistic Ideation Tool. To serve that need, not only the representation of the mechanisms and COTS databases is decided but also the UI for these methods is designed. The fundamental need behind the creation of the Mechanisms and COTS module was to use them in the conceptual design stage, because many design problems deal with mechanism type selection or using COTS. At this stage, mechanism/COTS type selection and search through the function requirement are the most important things. As discussed in the previous chapter, in the conceptual design stage, a designer may want to search the mechanisms/COTS databases through three different aspects. To provide these search options for the designer, the mechanisms/COTS databases were modeled with five categories (i.e. INPUT, OUTPUT, RELATION, FUNCTION, DISCRIPTIVE). Table 16 shows the relation between the search options needed and related category in the mechanisms and COTS databases.

Table 16: Relation Between Needed Search Options and Database Categories

SEARCH OPTIONS NEEDED	CATEGORY PROVIDED IN DATABASES
1. A designer might want to search by functional requirement.	FUNCTION
2. A designer might want to search by name or category of the mechanisms/COTS if he/she already knows for what mechanisms/COTS to look for.	DESCRIPTIVE
3. A designer with low domain knowledge may not know what is the name of the mechanism/COTS or under what category it might belong, but know what kind of mechanism/COTS to look. In that case he/she might want to search by characteristic properties of the mechanisms/COTS (input and output types).	INPUT, OUTPUT and RELATION

The UI was designed to provide search options mentioned in Table 16. The user can search the mechanisms or COTS by options 1 or 2 listed in Table 16 with the help of the modules "Find Mechanisms" and "Find Machine Elements". Or he/she can search mechanisms or COTS by option 3 in Table 16 with the help of the modules "Select Mechanisms" and "Select Machine Elements". Because of the fundamental similarities between the mechanisms and COTS, some of the search options provided in "Select Mechanisms" and "Select Machine Elements" UI are also similar, e.g., Search by Input (Type, Speed, and Direction), Search by Output (Type, Speed, and Direction), Search by Function. So, data for following options is fetched from the shared tables between the mechanisms and COTS databases: 1. Input/Output Type, 2. Input/Output Speed, 3. Input/Output Velocity Direction, 4. Relation between in Input and Output Line of Motion. However, though modules "Find Mechanisms" and "Find Machine Elements" provide similar "Search by Function" option to the user, the data for that is fetched from the individual mechanisms and COTS databases.

Figure 59 shows the UI for "Find Mechanisms" and "Find Machine Elements" modules. Multiple options are provided to search into databases and the list for each option is populated from the list of possible values for that option (see Tables 20 and 21).



Find Mechanisms

Find Mechanisms by Keyword

Find Mechanisms by Name
Adjustable Belt Drive for Parallel Axis Pulleys
Alternating-clamp, Friction Clutch Ratchet
Anchor Deadbeat Escapement
Anchor Escapement with Lever

Find Mechanisms by Group
Automobile Section
Balance beam mechanisms and paddle wheels
Belt Drive Mechanisms
Chamber Wheel Mechanisms

Find Mechanisms by Machine Components Involved
elliptical gear pump
NA
belt drive
belt drive,planetary gear

Find Mechanisms by Function
NA
absorbing the shock
allowing motion in one direction but locking in other
change of location of motion

Find Machine Elements

Find Machine Elements by Keyword

Find Machine Elements by Name
3 Stage Helical Reducer
Band Brakes
Beam Coupling
Bevel Gear Drive

Find Machine Elements by Category
Energy Dissipation Elements
Energy Storage Elements
Fastening Elements
Intermittant Energy Transfer

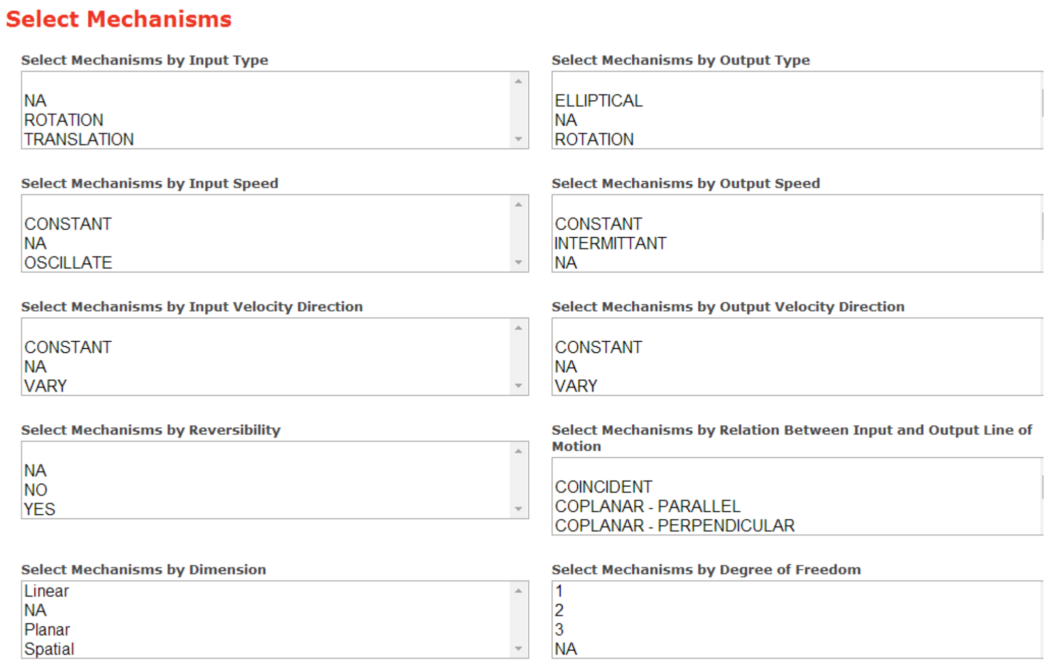
Find Machine Elements by Machine Element Category
AC Motors
Bearings
Belt Drvies
Brakes

Find Machine Elements by Function
Constrain Motion
Convert Energy
Dissipate Energy
Reduce Speed

Find Machine Elements by Related Physical Effect
Bend
Dynamic/Kinetic Friction
Electro-magnetic Induction
Form Closure

Figure 59: UI for Find Mechanisms and Find Machine Elements Module

Figure 60 and 61 shows the UI for "Select Mechanisms" and "Select Machine Elements" modules. The list for each of the option is populated from the list of possible values for that option (see tables 17, 18, and 19).



Select Mechanisms

Select Mechanisms by Input Type
NA
ROTATION
TRANSLATION

Select Mechanisms by Output Type
ELLIPTICAL
NA
ROTATION

Select Mechanisms by Input Speed
CONSTANT
NA
OSCILLATE

Select Mechanisms by Output Speed
CONSTANT
INTERMITTANT
NA

Select Mechanisms by Input Velocity Direction
CONSTANT
NA
VARY

Select Mechanisms by Output Velocity Direction
CONSTANT
NA
VARY

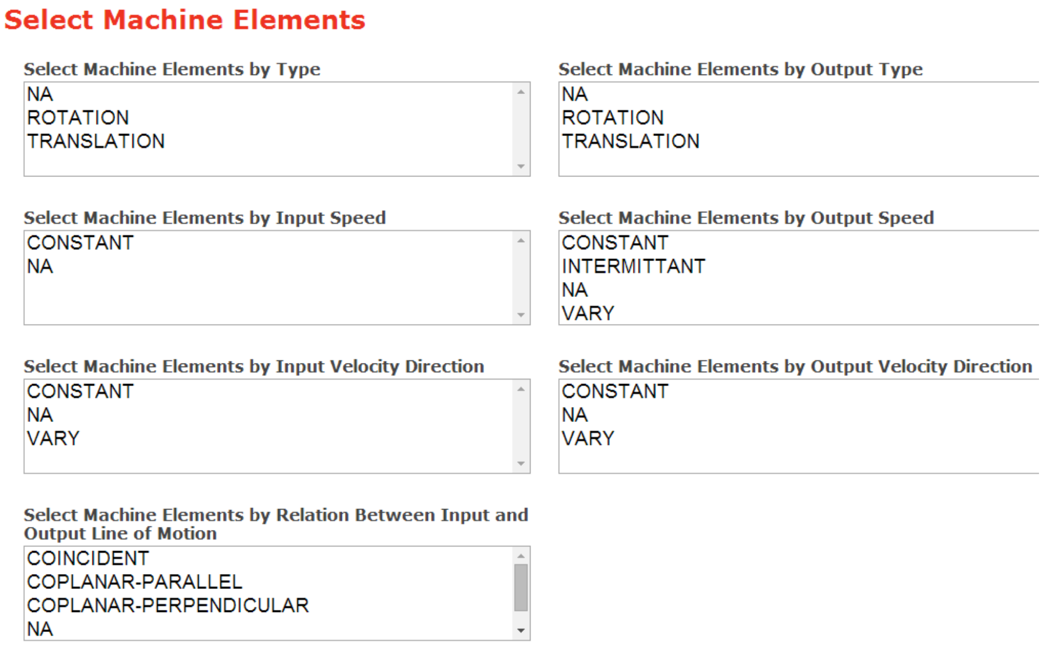
Select Mechanisms by Reversibility
NA
NO
YES

Select Mechanisms by Relation Between Input and Output Line of Motion
COINCIDENT
COPLANAR - PARALLEL
COPLANAR - PERPENDICULAR

Select Mechanisms by Dimension
Linear
NA
Planar
Spatial

Select Mechanisms by Degree of Freedom
1
2
3
NA

Figure 60: UI for Select Mechanisms Module



Select Machine Elements

Select Machine Elements by Type
NA
ROTATION
TRANSLATION

Select Machine Elements by Output Type
NA
ROTATION
TRANSLATION

Select Machine Elements by Input Speed
CONSTANT
NA

Select Machine Elements by Output Speed
CONSTANT
INTERMITTANT
NA
VARY

Select Machine Elements by Input Velocity Direction
CONSTANT
NA
VARY

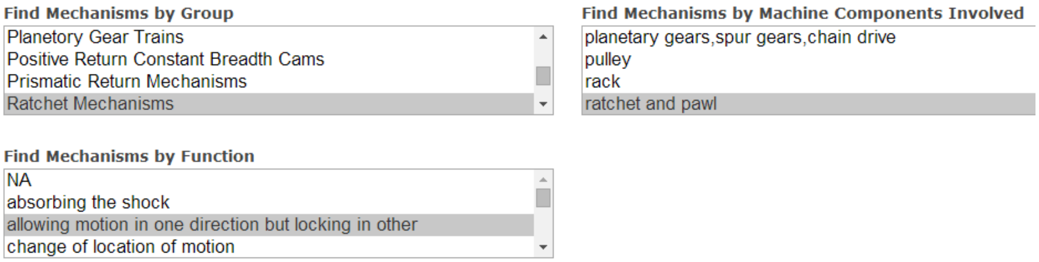
Select Machine Elements by Output Velocity Direction
CONSTANT
NA
VARY

Select Machine Elements by Relation Between Input and Output Line of Motion
COINCIDENT
COPLANAR-PARALLEL
COPLANAR-PERPENDICULAR
NA

Figure 61: UI for Select Machine Elements Module

In all the four modules shown above, a user can select multiple values for multiple options. If a designer does not choose any option, then system generates an output with a list of all the mechanisms/COTS. If a user makes any selection, then this full list is selectively reduced depending upon the multiple values chosen by the designer for different options. The output list is generated by clicking on the "Submit" button provided below the options.

Figure 63 shows the sample output list of mechanisms for specific set of values selected by the designer as input (Figure 62). The system provided 17 different mechanisms from "Ratchet Mechanisms" group which contains "ratchet and pawl" components and perform the function of "allowing motion in one direction but locking in the other direction".



Find Mechanisms by Group

Planetory Gear Trains
Positive Return Constant Breadth Cams
Prismatic Return Mechanisms
Ratchet Mechanisms

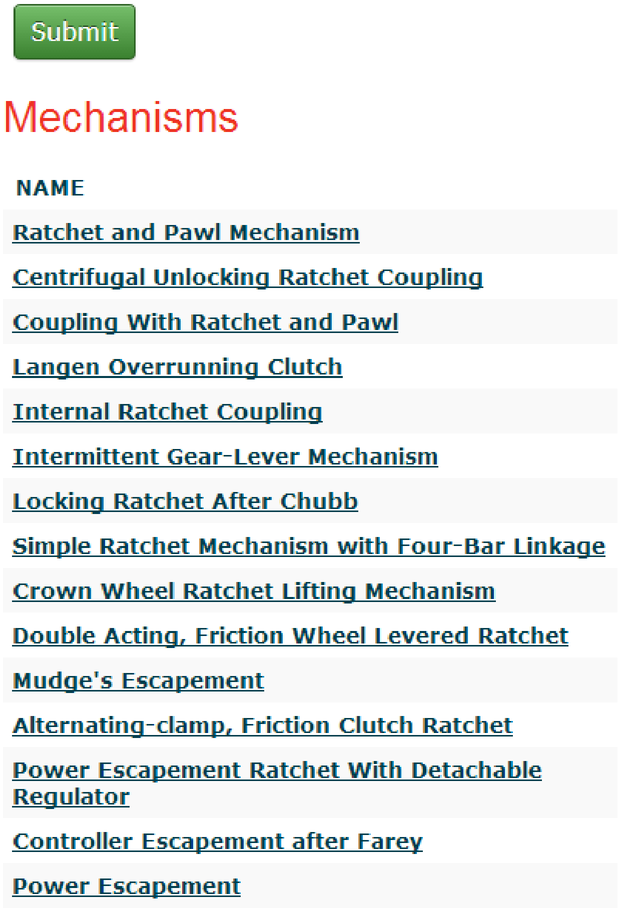
Find Mechanisms by Machine Components Involved

planetary gears,spur gears,chain drive
pulley
rack
ratchet and pawl

Find Mechanisms by Function

NA
absorbing the shock
allowing motion in one direction but locking in other
change of location of motion

Figure 62: A Set of Input Values to the Find Mechanisms Module

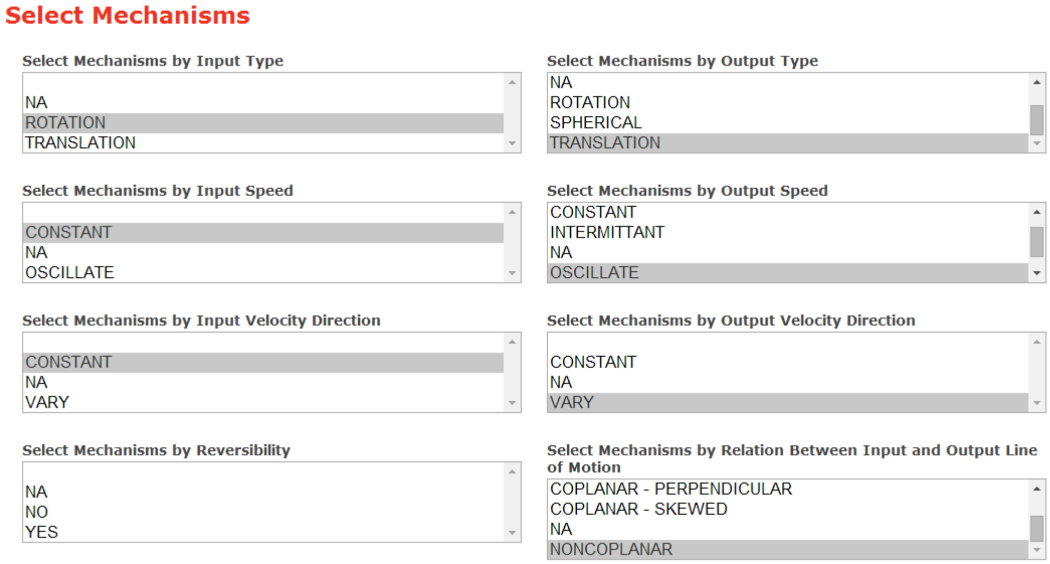


Mechanisms

NAME
Ratchet and Pawl Mechanism
Centrifugal Unlocking Ratchet Coupling
Coupling With Ratchet and Pawl
Langen Overrunning Clutch
Internal Ratchet Coupling
Intermittent Gear-Lever Mechanism
Locking Ratchet After Chubb
Simple Ratchet Mechanism with Four-Bar Linkage
Crown Wheel Ratchet Lifting Mechanism
Double Acting, Friction Wheel Levered Ratchet
Mudge's Escapement
Alternating-clamp, Friction Clutch Ratchet
Power Escapement Ratchet With Detachable Regulator
Controller Escapement after Farey
Power Escapement

Figure 63: Partial Output List of Mechanisms from the Tool

Figure 65 shows the output list of mechanisms for the specific set of input values selected by the designer as shown in Figure 64. The system provided 8 different mechanisms which can convert "Rotation" type of input motion with "Constant" speed and "Constant" velocity direction to "Translation" type of output motion with "Oscillating" speed at "Varying" velocity direction under the constraint of "Non-Coplanar" axes of input and output line of motion.



Select Mechanisms

Select Mechanisms by Input Type

NA
[✓ ROTATION]
TRANSLATION

Select Mechanisms by Output Type

NA
ROTATION
SPHERICAL
[✓ TRANSLATION]

Select Mechanisms by Input Speed

[✓ CONSTANT]
NA
OSCILLATE

Select Mechanisms by Output Speed

CONSTANT
INTERMITTANT
NA
[✓ OSCILLATE]

Select Mechanisms by Input Velocity Direction

[✓ CONSTANT]
NA
VARY

Select Mechanisms by Output Velocity Direction

CONSTANT
NA
[✓ VARY]

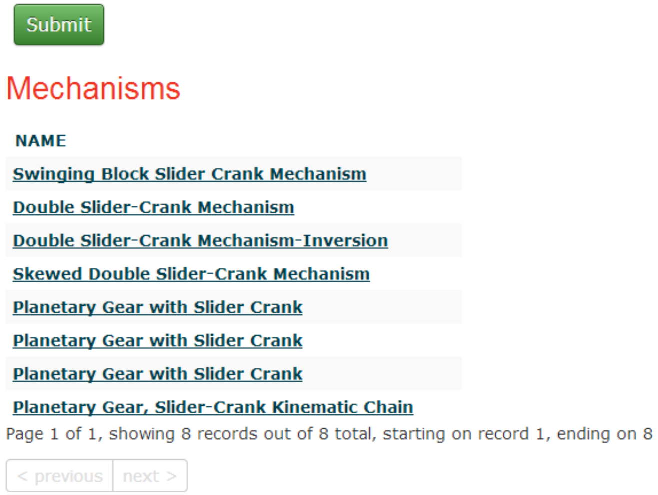
Select Mechanisms by Reversibility

NA
NO
YES

Select Mechanisms by Relation Between Input and Output Line of Motion

COPLANAR - PERPENDICULAR
COPLANAR - SKEWED
NA
[✓ NONCOPLANAR]

Figure 64: A Set of Input Values to the Select Mechanisms Module



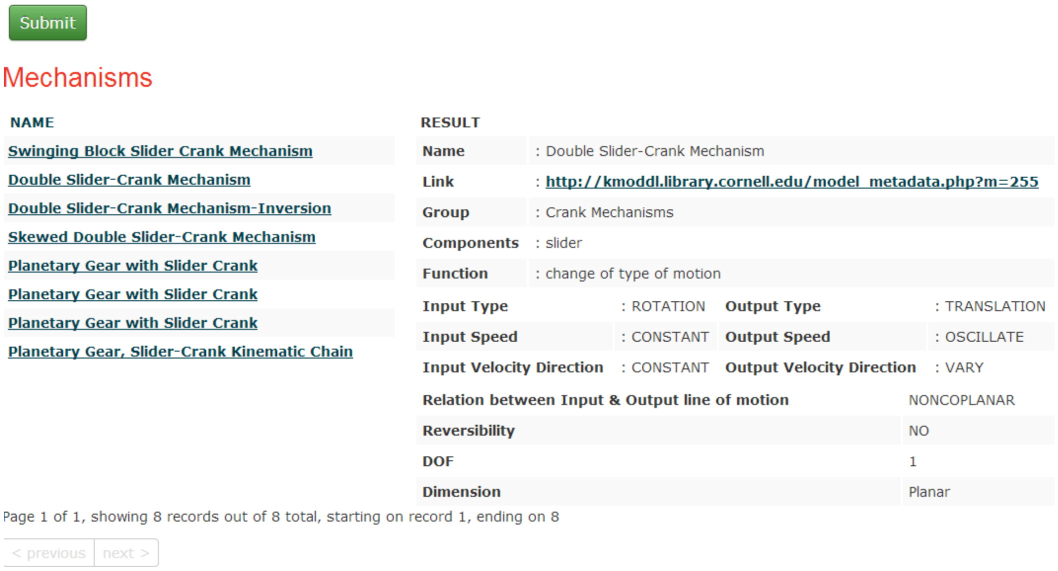
Mechanisms

NAME
Swinging Block Slider Crank Mechanism
Double Slider-Crank Mechanism
Double Slider-Crank Mechanism-Inversion
Skewed Double Slider-Crank Mechanism
Planetary Gear with Slider Crank
Planetary Gear with Slider Crank
Planetary Gear with Slider Crank
Planetary Gear, Slider-Crank Kinematic Chain

Page 1 of 1, showing 8 records out of 8 total, starting on record 1, ending on 8

Figure 65: Output List of Mechanisms from the Tool

Details about any mechanism from the output list can be seen by clicking on the name of the mechanism. Figure 66 shows the UI for output of mechanisms.



Mechanisms

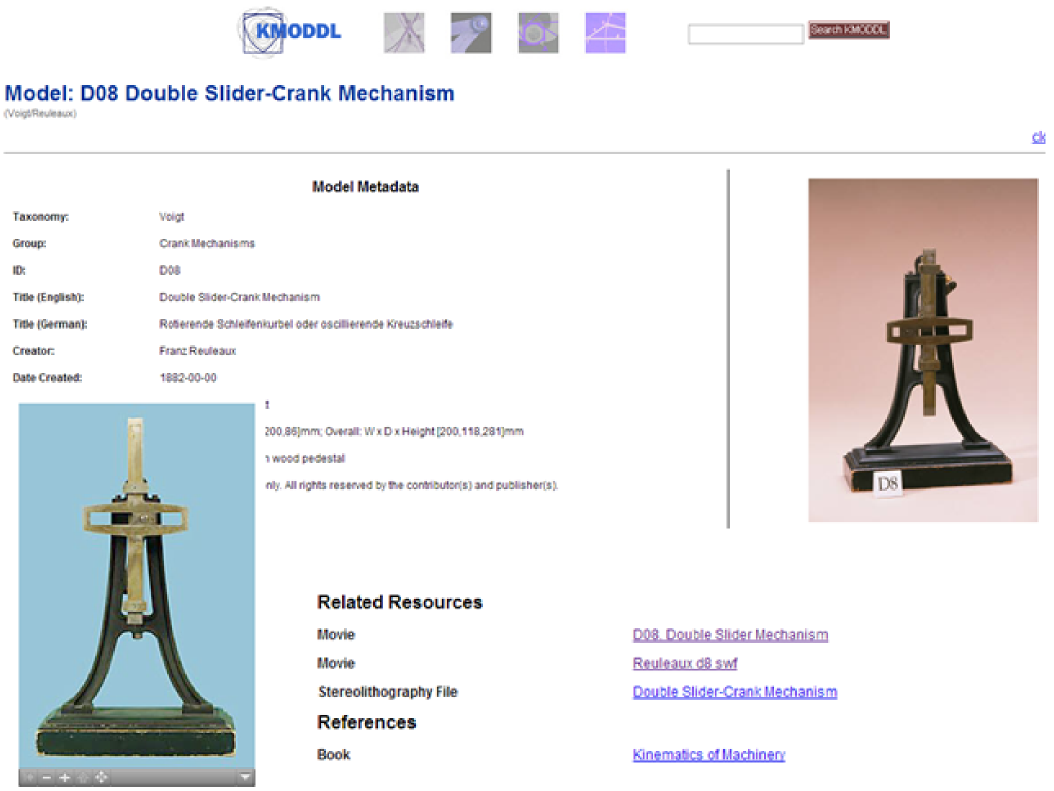
NAME
Swinging Block Slider Crank Mechanism
Double Slider-Crank Mechanism
Double Slider-Crank Mechanism-Inversion
Skewed Double Slider-Crank Mechanism
Planetary Gear with Slider Crank
Planetary Gear with Slider Crank
Planetary Gear with Slider Crank
Planetary Gear, Slider-Crank Kinematic Chain

RESULT			
Name	: Double Slider-Crank Mechanism		
Link	: http://kmoddl.library.cornell.edu/model_metadata.php?m=255		
Group	: Crank Mechanisms		
Components	: slider		
Function	: change of type of motion		
Input Type	: ROTATION	Output Type	: TRANSLATION
Input Speed	: CONSTANT	Output Speed	: OSCILLATE
Input Velocity Direction	: CONSTANT	Output Velocity Direction	: VARY
Relation between Input & Output line of motion			NONCOPLANAR
Reversibility			NO
DOF			1
Dimension			Planar

Page 1 of 1, showing 8 records out of 8 total, starting on record 1, ending on 8

Figure 66: UI for Mechanisms Information

A designer can click on the "Link" option to see the video simulation and detailed information about that mechanism on KMODDL website (Figure 67). This website provide details such as reference books, image and video simulation. It also provides the designer a stereo-lithography file of the given mechanism in case the designer want to 3D print the mechanism to understand its working more intuitively.



Model: D08 Double Slider-Crank Mechanism

(Voigt/Reuleaux)

Model Metadata

Taxonomy:	Voigt
Group:	Crank Mechanisms
ID:	D08
Title (English):	Double Slider-Crank Mechanism
Title (German):	Rotierende Schleifenkurbel oder oscillierende Kreuzschleife
Creator:	Franz Reuleaux
Date Created:	1882-00-00

t

200,86]mm; Overall: W x D x Height [200,118,281]mm

ı wood pedestal

nly. All rights reserved by the contributor(s) and publisher(s).

Related Resources

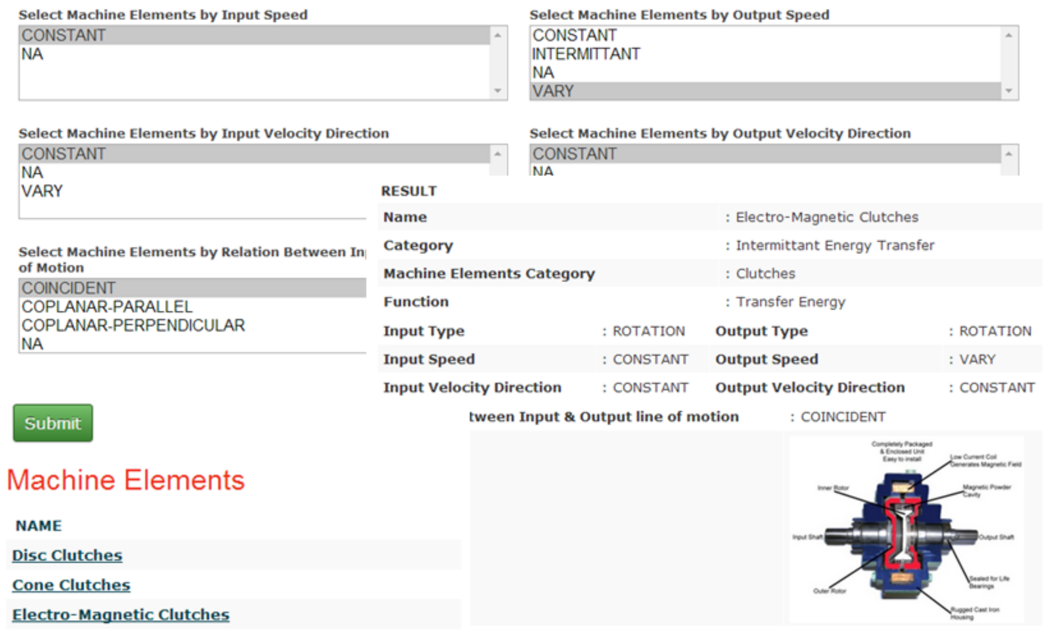
Movie	D08. Double Slider Mechanism
Movie	Reuleaux d8 swf
Stereolithography File	Double Slider-Crank Mechanism

References

Book	Kinematics of Machinery

Figure 67: Detailed Information about Mechanisms by KMODDL

Similarly, Figure 68 shows the output list of three COTS (Disc Clutch, Cone Clutch, and Electro Magnetic Clutch), for the input values (Input Speed-Constant, Output Speed-Vary, Input Velocity Direction-Constant, Output Velocity Direction-Constant, and Relation between input and output line of motion-Coincident) given to the tool using "Select Machine Elements" module.



Select Machine Elements by Input Speed
CONSTANT
NA

Select Machine Elements by Output Speed
CONSTANT
INTERMITTANT
NA
VARY

Select Machine Elements by Input Velocity Direction
CONSTANT
NA
VARY

Select Machine Elements by Output Velocity Direction
CONSTANT
NA

Select Machine Elements by Relation Between In
of Motion
COINCIDENT
COPLANAR-PARALLEL
COPLANAR-PERPENDICULAR
NA

Machine Elements

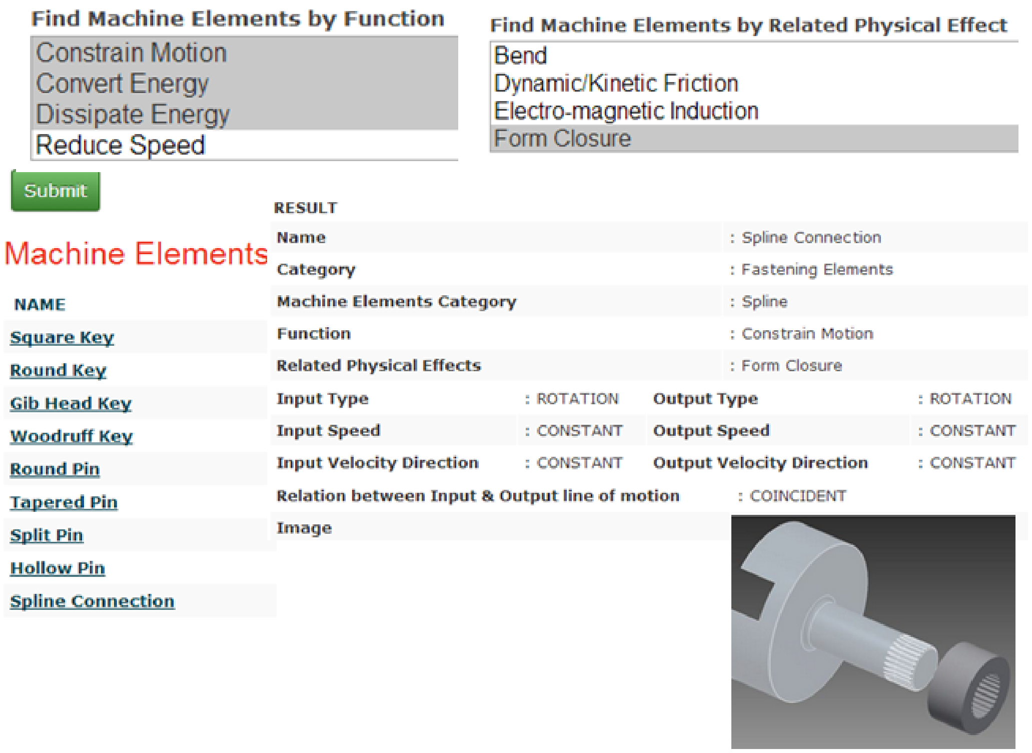
NAME
Disc Clutches
Cone Clutches
Electro-Magnetic Clutches

RESULT

Name	: Electro-Magnetic Clutches		
Category	: Intermittant Energy Transfer		
Machine Elements Category	: Clutches		
Function	: Transfer Energy		
Input Type	: ROTATION	Output Type	: ROTATION
Input Speed	: CONSTANT	Output Speed	: VARY
Input Velocity Direction	: CONSTANT	Output Velocity Direction	: CONSTANT
tween Input & Output line of motion	: COINCIDENT		

Figure 68: Example of UI for Select Machine Elements Module

Figure 69 shows the UI for "Find Machine Elements" module. The system outputs all the COTS in the database if user does not select any option. In this case the system generated 9 results for the input of COTS performing "Constrain Motion", "Convert Energy" *OR* "Dissipate Energy" function *AND* embedding the physical effect of "Form Closure".



Find Machine Elements by Function

- Constrain Motion
- Convert Energy
- Dissipate Energy
- Reduce Speed

Find Machine Elements by Related Physical Effect

- Bend
- Dynamic/Kinetic Friction
- Electro-magnetic Induction
- Form Closure

Machine Elements

NAME

- Square Key
- Round Key
- Gib Head Key
- Woodruff Key
- Round Pin
- Tapered Pin
- Split Pin
- Hollow Pin
- Spline Connection

RESULT

Name	: Spline Connection		
Category	: Fastening Elements		
Machine Elements Category	: Spline		
Function	: Constrain Motion		
Related Physical Effects	: Form Closure		
Input Type	: ROTATION	**Output Type**	: ROTATION
Input Speed	: CONSTANT	**Output Speed**	: CONSTANT
Input Velocity Direction	: CONSTANT	**Output Velocity Direction**	: CONSTANT
Relation between Input & Output line of motion	: COINCIDENT		
Image			

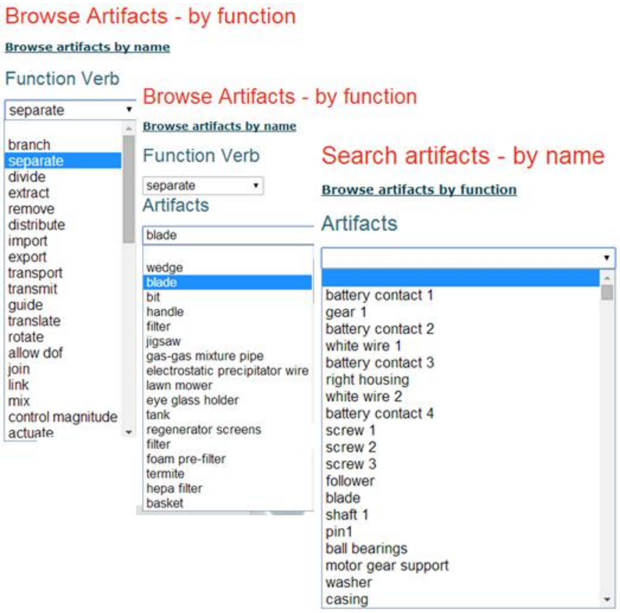
Figure 69: Example of UI for Find Machine Elements Module

5.5.6 Logical Ideation Methods Retained from V.1

The V.3 has retained logical ideation methods from V.1, UI for which is discussed in this section. These methods are retained with the same database structure which is discussed in previous sections. Various database queries to implement these ideation methods are kept same as discussed in Mohan's work (Mohan M. , 2011). However, to improve the usability of the tool the UI has been changed. In this section we will present the UI designed for these ideation methods.

5.5.6.1 Artifacts

The Artifacts can be searched based on names or functions. The new input and output UI are shown in Figure 70 and 71 respectively.



Browse Artifacts - by function

Browse artifacts by name

Function Verb

separate

Browse Artifacts - by function

Browse artifacts by name

Function Verb

separate

Artifacts

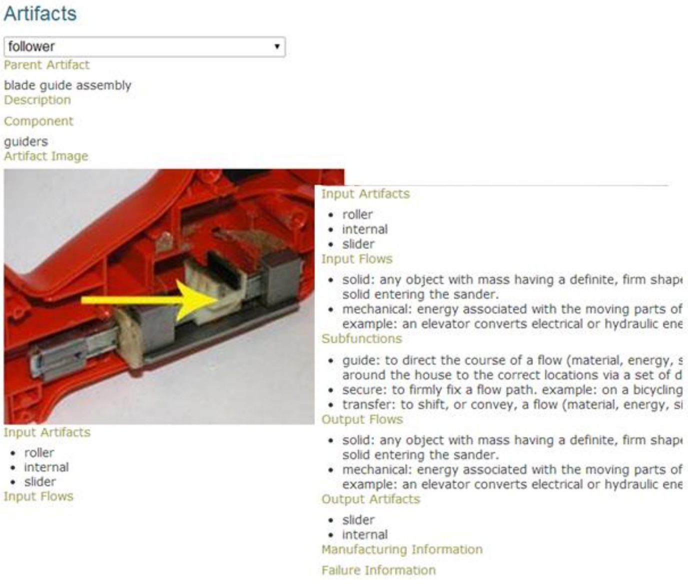
blade

Search artifacts - by name

Browse artifacts by function

Artifacts

Figure 70: Input UI for Artifact Module



Artifacts

follower

Parent Artifact

blade guide assembly

Description

Component

guiders

Artifact Image

Input Artifacts

- roller
- internal
- slider

Input Flows

- solid: any object with mass having a definite, firm shap solid entering the sander.
- mechanical: energy associated with the moving parts of example: an elevator converts electrical or hydraulic ene

Subfunctions

- guide: to direct the course of a flow (material, energy, s around the house to the correct locations via a set of d
- secure: to firmly fix a flow path. example: on a bicycling
- transfer: to shift, or convey, a flow (material, energy, si

Output Flows

- solid: any object with mass having a definite, firm shap solid entering the sander.
- mechanical: energy associated with the moving parts of example: an elevator converts electrical or hydraulic ene

Output Artifacts

- slider
- internal

Manufacturing Information

Failure Information

Input Artifacts

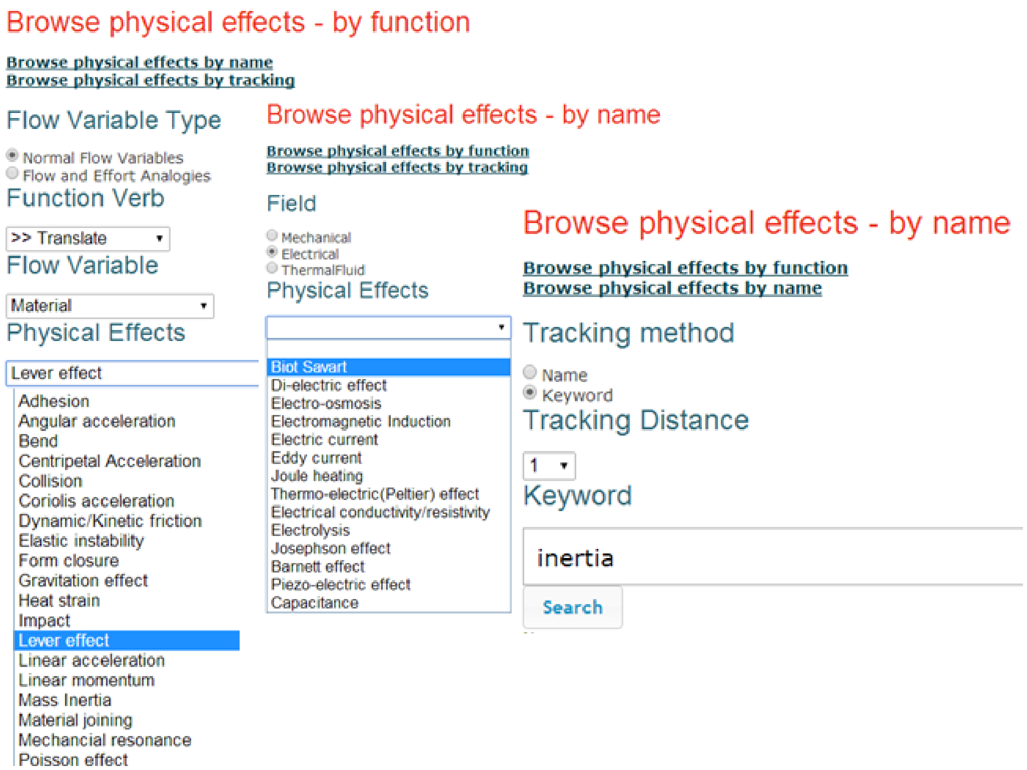
- roller
- internal
- slider

Input Flows

Figure 71: Output UI for Artifact Module

5.5.6.2 Physical Effects

The physical effects database can be searched through the physical effect's name, function or with a keyword. The following snapshot shows the input and output UI for the physical effects module (Figure 72 and 73). As shown in Figure 73, the related working principle are linked with any physical effect to allow traversal from PE knowledge base to WP knowledge base. In the case of searching by keyword, "Tracking Distance" indicates how close or far related physical effect is with a particular keyword.



Browse physical effects - by function

Browse physical effects by name
Browse physical effects by tracking

Flow Variable Type

● Normal Flow Variables
○ Flow and Effort Analogies

Function Verb

>> Translate ▾

Flow Variable

Material ▾

Physical Effects

Lever effect

Adhesion
Angular acceleration
Bend
Centripetal Acceleration
Collision
Coriolis acceleration
Dynamic/Kinetic friction
Elastic instability
Form closure
Gravitation effect
Heat strain
Impact
Lever effect
Linear acceleration
Linear momentum
Mass Inertia
Material joining
Mechancial resonance
Poisson effect

Browse physical effects - by name

Browse physical effects by function
Browse physical effects by tracking

Field

○ Mechanical
● Electrical
○ ThermalFluid

Physical Effects

▾

Biot Savart
Di-electric effect
Electro-osmosis
Electromagnetic Induction
Electric current
Eddy current
Joule heating
Thermo-electric(Peltier) effect
Electrical conductivity/resistivity
Electrolysis
Josephson effect
Barnett effect
Piezo-electric effect
Capacitance

Browse physical effects - by name

Browse physical effects by function
Browse physical effects by name

Tracking method

○ Name
● Keyword

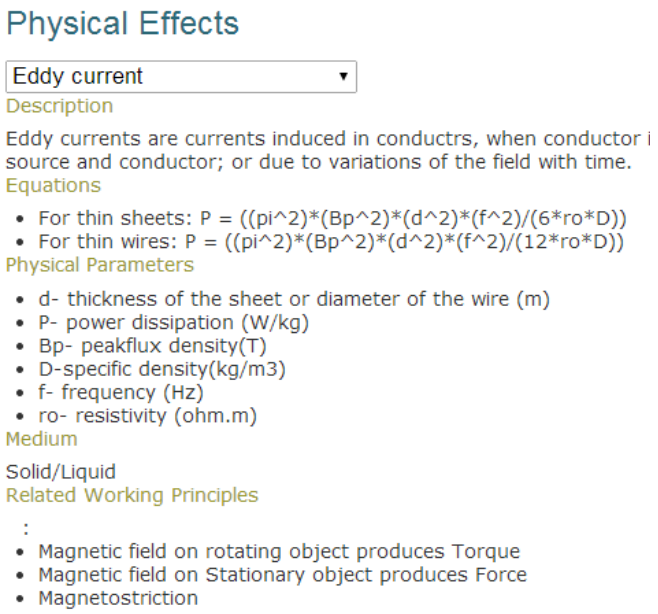
Tracking Distance

1 ▾

Keyword

inertia

Figure 72: Input UI for Physical Effect Module



Physical Effects

Eddy current ▾

Description

Eddy currents are currents induced in conductrs, when conductor i
source and conductor; or due to variations of the field with time.

Equations

- For thin sheets: P = ((pi^2)*(Bp^2)*(d^2)*(f^2)/(6*ro*D))
- For thin wires: P = ((pi^2)*(Bp^2)*(d^2)*(f^2)/(12*ro*D))

Physical Parameters

- d- thickness of the sheet or diameter of the wire (m)
- P- power dissipation (W/kg)
- Bp- peakflux density(T)
- D-specific density(kg/m3)
- f- frequency (Hz)
- ro- resistivity (ohm.m)

Medium

Solid/Liquid

Related Working Principles

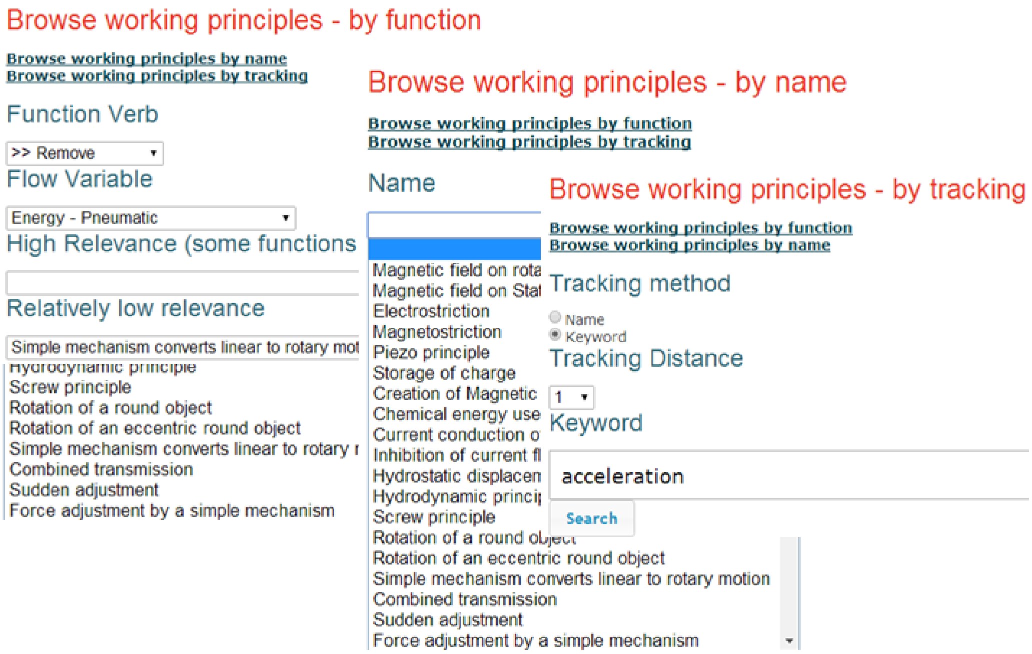
:

- Magnetic field on rotating object produces Torque
- Magnetic field on Stationary object produces Force
- Magnetostriction

Figure 73: Output UI for Physical Effect Module

5.5.6.3 Working Principle

The working principles can be searched with respect to working principle's name, function or with a keyword. The following snapshot shows the input and output UI for the working principles module (Figure 74 and 75). As shown in Figure 75, related physical effects are listed with any working principle to allow traversal from the WP knowledge base to the PE knowledge base. When searching with related function, the tool generates two lists: highly relevant working principles and working principles with low relevance with the given function. In the case of searching by keyword "Tracking Distance" indicates how close or far related working principle is with a particular keyword. Furthermore, each working principle also gives a related biological example from AskNature tool. This example can be accessed by clicking on the link provided in the output UI for the working principle module (Figure 75).



Browse working principles - by function

Browse working principles by name
Browse working principles by tracking

Function Verb

>> Remove

Flow Variable

Energy - Pneumatic

High Relevance (some functions

Relatively low relevance

Simple mechanism converts linear to rotary mot
Hydrodynamic principle
Screw principle
Rotation of a round object
Rotation of an eccentric round object
Simple mechanism converts linear to rotary r
Combined transmission
Sudden adjustment
Force adjustment by a simple mechanism

Browse working principles - by name

Browse working principles by function
Browse working principles by tracking

Name

Magnetic field on rota
Magnetic field on Stat
Electrostriction
Magnetostriction
Piezo principle
Storage of charge
Creation of Magnetic
Chemical energy use
Current conduction o
Inhibition of current fl
Hydrostatic displacem
Hydrodynamic princi
Screw principle
Rotation of a round object
Rotation of an eccentric round object
Simple mechanism converts linear to rotary motion
Combined transmission
Sudden adjustment
Force adjustment by a simple mechanism

Browse working principles - by tracking

Browse working principles by function
Browse working principles by name

Tracking method

○ Name
● Keyword

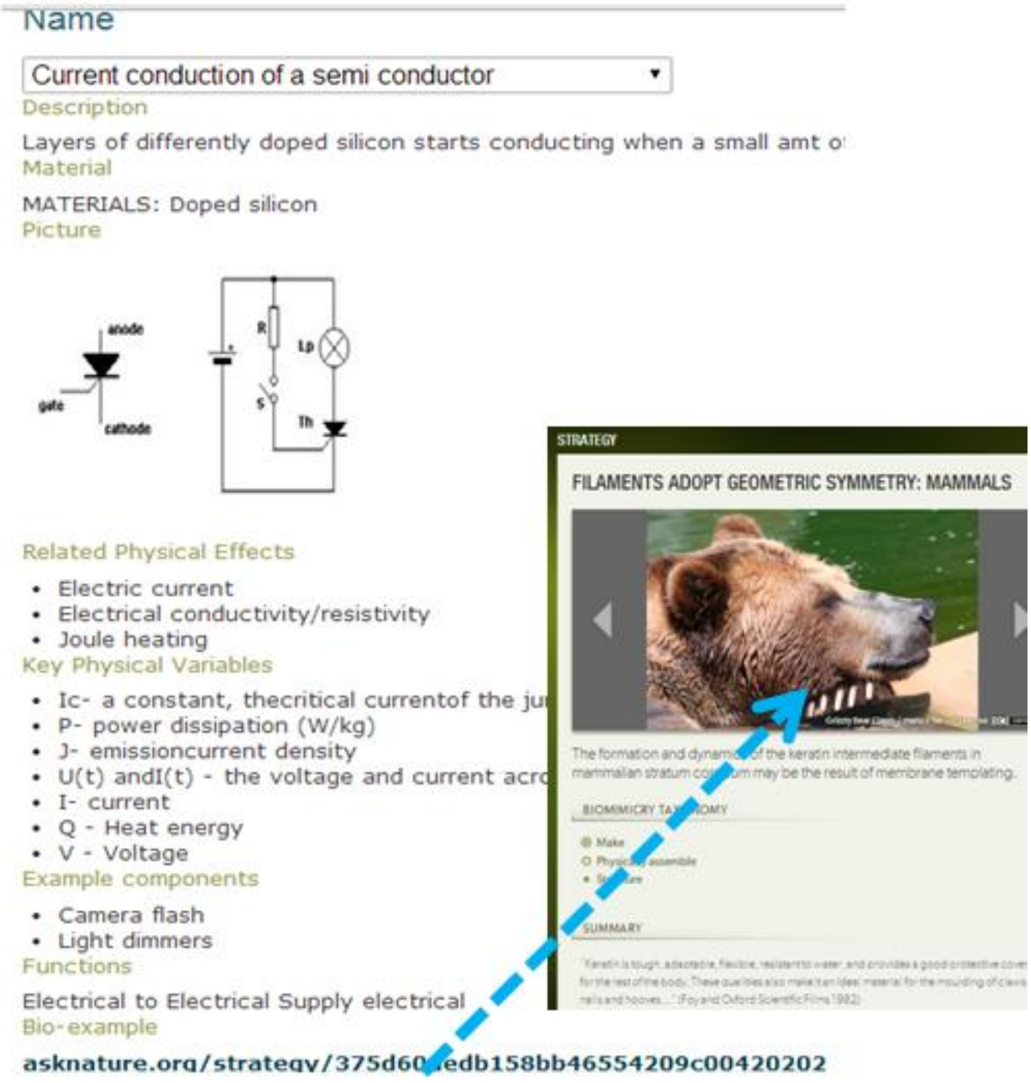
Tracking Distance

1

Keyword

acceleration

Figure 74: Input UI for Working Principle Module



Name

Current conduction of a semi conductor ▾

Description

Layers of differently doped silicon starts conducting when a small amt o

Material

MATERIALS: Doped silicon

Picture

Related Physical Effects

- Electric current
- Electrical conductivity/resistivity
- Joule heating

Key Physical Variables

- Ic- a constant, thecritical currentof the ju
- P- power dissipation (W/kg)
- J- emissioncurrent density
- U(t) andI(t) - the voltage and current acro
- I- current
- Q - Heat energy
- V - Voltage

Example components

- Camera flash
- Light dimmers

Functions

Electrical to Electrical Supply electrical

Bio-example

asknature.org/strategy/375d60_edb158bb46554209c00420202

STRATEGY

FILAMENTS ADOPT GEOMETRIC SYMMETRY: MAMMALS

The formation and dynamics of the keratin intermediate filaments in mammalian stratum corneum may be the result of membrane templating.

BIOMIMICRY TAXONOMY

- ◉ Make
- ○ Physically assemble
- • Structure

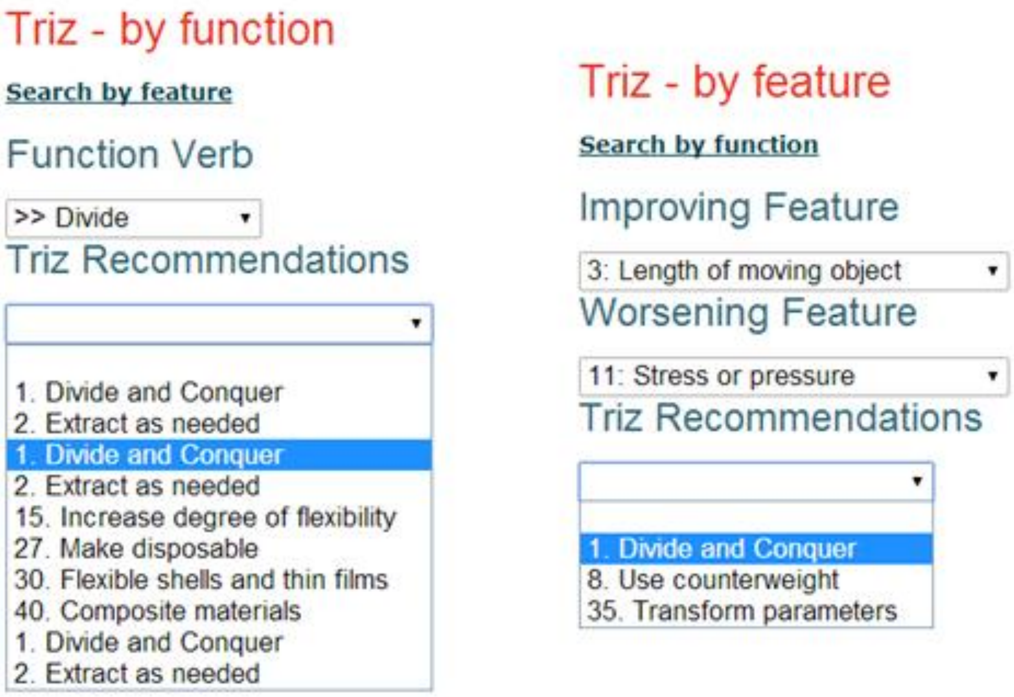
SUMMARY

"Keratin is tough, adaptable, flexible, resistant to water, and provides a good protective cover for the rest of the body. These qualities also make it an ideal material for the moulding of claws, nails and hooves...." (Foy and Oxford Scientific Films 1982)

Figure 75: Output UI for Working Principle Module

5.5.6.4 TRIZ/Bio-TRIZ

The TRIZ/Bio-TRIZ databases can be searched by either the traditional way (e.g. by improving and worsening parameters) or by related functions. Figure 76 and 77 shows the input and output UI for the TRIZ/BioTRIZ module. In BioTRIZ module, the user is given a biological solution related to the TRIZ principle.



Triz - by function

Search by feature

Function Verb

>> Divide

Triz Recommendations

1. Divide and Conquer
2. Extract as needed
1. Divide and Conquer
2. Extract as needed
15. Increase degree of flexibility
27. Make disposable
30. Flexible shells and thin films
40. Composite materials
1. Divide and Conquer
2. Extract as needed

Triz - by feature

Search by function

Improving Feature

3: Length of moving object

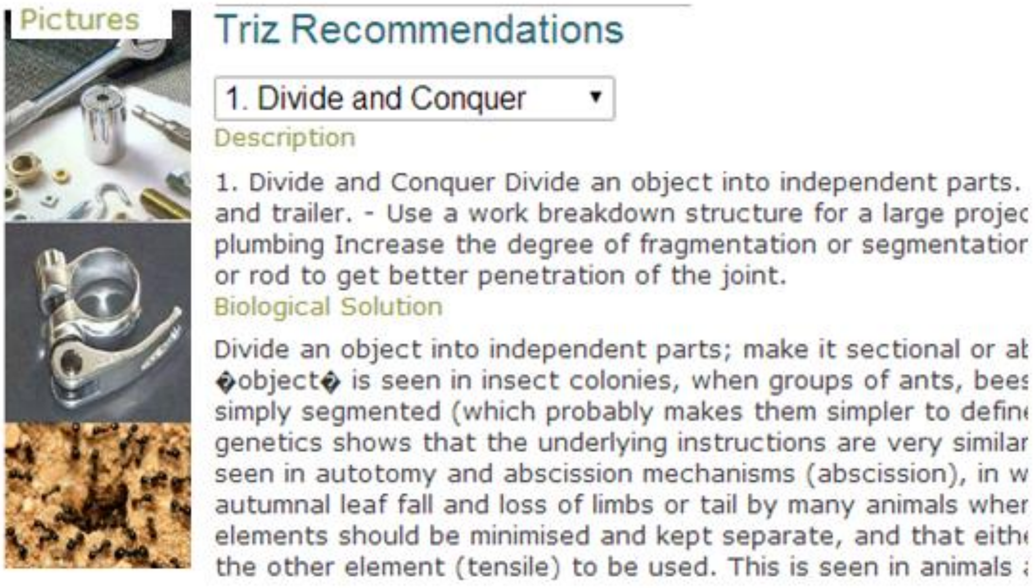
Worsening Feature

11: Stress or pressure

Triz Recommendations

1. Divide and Conquer
8. Use counterweight
35. Transform parameters

Figure 76: Input UI for TRIZ/Bio-TRIZ Module



Pictures

Triz Recommendations

1. Divide and Conquer

Description

1. Divide and Conquer Divide an object into independent parts.
and trailer. - Use a work breakdown structure for a large projec
plumbing Increase the degree of fragmentation or segmentatior
or rod to get better penetration of the joint.

Biological Solution

Divide an object into independent parts; make it sectional or ab
�object� is seen in insect colonies, when groups of ants, bees
simply segmented (which probably makes them simpler to define
genetics shows that the underlying instructions are very similar
seen in autotomy and abscission mechanisms (abscission), in w
autumnal leaf fall and loss of limbs or tail by many animals wher
elements should be minimised and kept separate, and that eithe
the other element (tensile) to be used. This is seen in animals a

Figure 77: Output UI for TRIZ/BioTRIZ Module

5.5.7 Intuitive Ideation Methods Retained from V.2

Despite the fact that, V.3 has retained intuitive ideation methods (e.g., Word Diamond, WordNet and Relational Algorithm) from V.2, the UI has been changed to improve usability. Details about the new UI will be presented in following sections.

5.5.7.1 Word Diamond

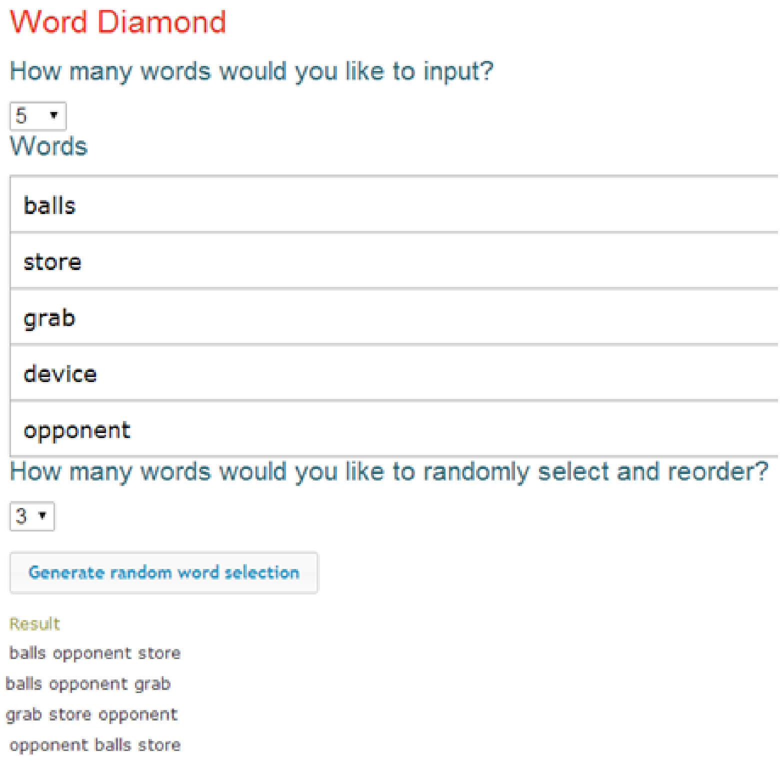
When using the Word Diamond module the V.3 allows designer to select any number of words. It is good to include both verbs and nouns as the input words. The selected words then get combined at random to see if the new functions are revealed. Figure 78 shows the UI for "Word Diamond" module.

For example:

Problem - Design a device which can grab and store more golf balls than the opponents.

Key words chosen by a designer as the input: balls, store, grab, device, opponent

These 5 words will be put in multiple random orders by the tool.



Word Diamond

How many words would you like to input?

5 ▾

Words

balls

store

grab

device

opponent

How many words would you like to randomly select and reorder?

3 ▾

Result

balls opponent store

balls opponent grab

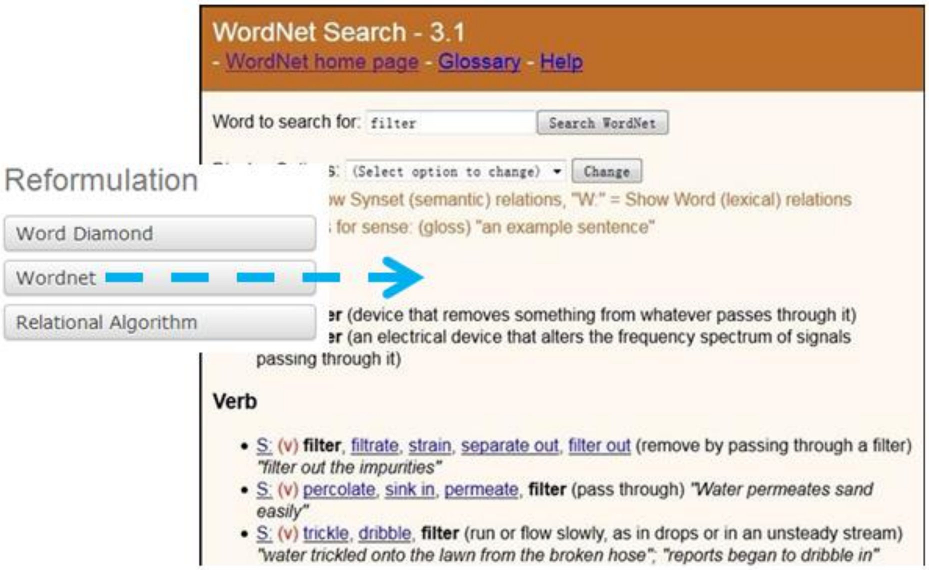
grab store opponent

opponent balls store

Figure 78: Word Diamond UI

5.5.7.2 WordNet

The V.3 has provided the link to the WordNet tool created by the Princeton University. It can be accessed by clicking on the "WordNet" module button (Figure 79). When reframing the word "filter", if we consider the “filter” as a function, the outputs shown by the WordNet tool are “filtrate”, “separate out”, “filter out”. Some functions are also followed by behavior explanation, such as “filter out (remove by passing through a filter)”.



WordNet Search - 3.1
- WordNet home page - Glossary - Help

Word to search for: filter

S: (Select option to change)

w Synset (semantic) relations, "W:" = Show Word (lexical) relations

for sense: (gloss) "an example sentence"

er (device that removes something from whatever passes through it)
er (an electrical device that alters the frequency spectrum of signals passing through it)

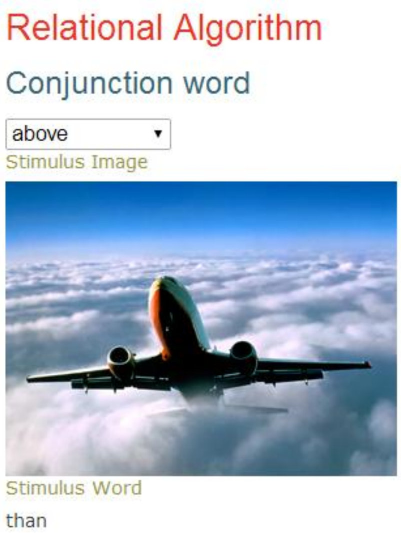
Verb

- S: (v) **filter**, filtrate, strain, separate out, filter out (remove by passing through a filter) *"filter out the impurities"*
- S: (v) percolate, sink in, permeate, **filter** (pass through) *"Water permeates sand easily"*
- S: (v) trickle, dribble, **filter** (run or flow slowly, as in drops or in an unsteady stream) *"water trickled onto the lawn from the broken hose"; "reports began to dribble in"*

Figure 79: WordNet Tool UI

5.5.7.3 Relational Algorithm

As shown in Figure 80, the V.3 has provided the Relational-Algorithm ideation module, in which the designer is provided the list of 37 words. Sketches and pictures are attached with these words to generate stimuli for the reframing of the design problems. These pictures are pre-selected and attached with certain words. Figure 80 shows the output by the Relational Algorithm module for the input "above".



Relational Algorithm

Conjunction word

above

Stimulus Image

Stimulus Word

than

Figure 80: Relational Algorithm UI

5.5.8 External Resources

In V.3, we have provided user with the external resources which can be accessed by clicking on their respective names as shown in Figure 81.

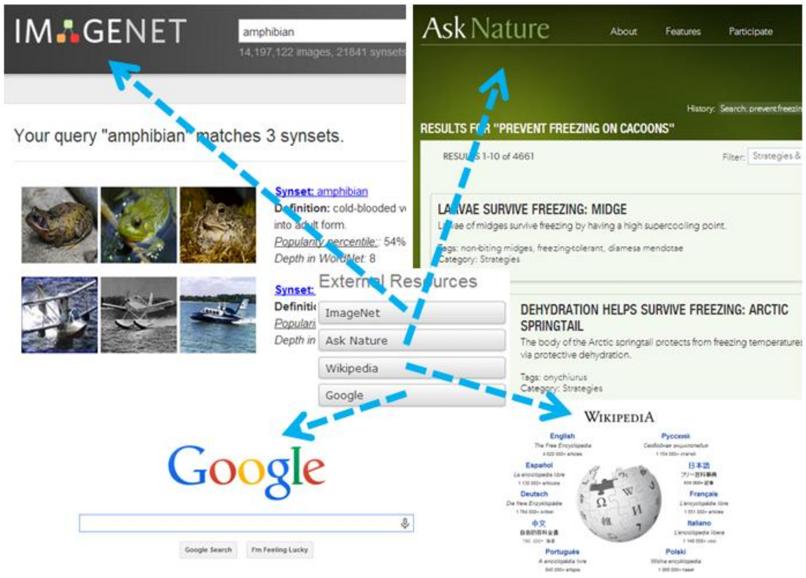
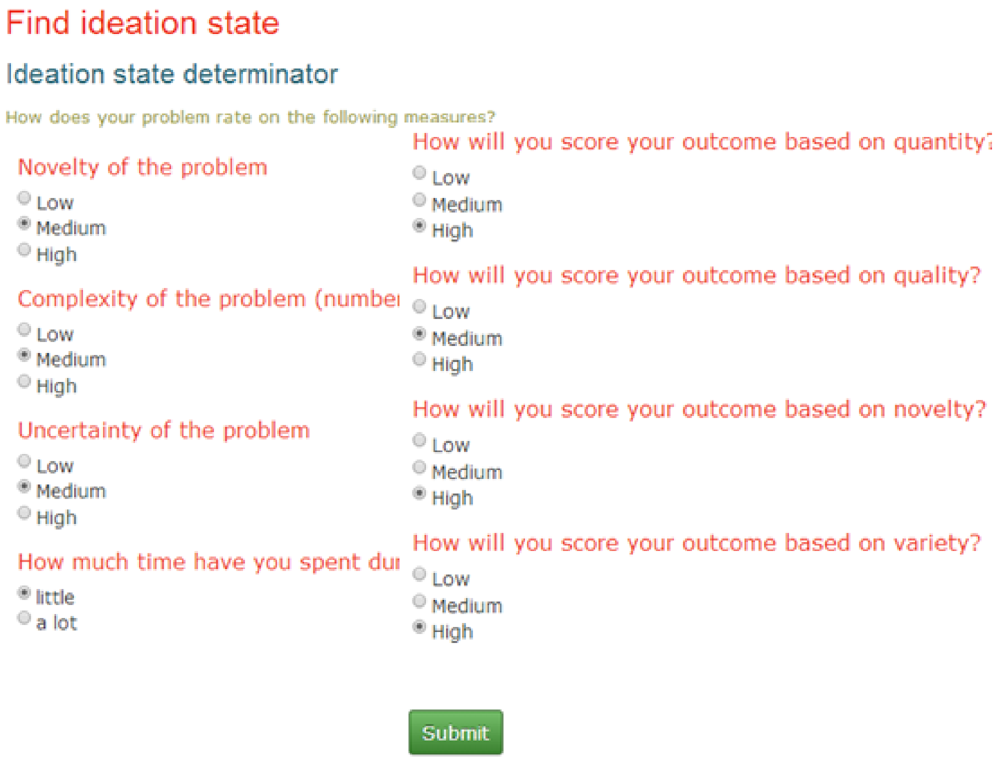

IM GENET
amphibian
14,197,122 images, 21841 synsets
Your query "amphibian" matches 3 synsets.
Synset: amphibian
Definition: cold-blooded v
into adult form.
Popularity percentile: 54%
Depth in WordNet: 8
Synset:
Definiti
Populari
Depth in
Ask Nature
About
Features
Participate
History: Search: prevent freezin
RESULTS FOR "PREVENT FREEZING ON CACOONS"
RESULTS 1-10 of 4661
Filter: Strategies &
LARVAE SURVIVE FREEZING: MIDGE
Larvae of midges survive freezing by having a high supercooling point.
Tags: non-biting midges, freezing-tolerant, diamesa mendotae
Category: Strategies
DEHYDRATION HELPS SURVIVE FREEZING: ARCTIC SPRINGTAIL
The body of the Arctic springtail protects from freezing temperatures via protective dehydration.
Tags: onychiurus
Category: Strategies
External Resources
ImageNet
Ask Nature
Wikipedia
Google
Google
Google Search
I'm Feeling Lucky
WIKIPEDIA
English
Русский
Español
日本語
Deutsch
Français
中文
Italiano
Português
Polski

Figure 81: External Resources in V.3

5.5.9 Process Monitoring Tools

5.5.9.1 Find Ideation State

To characterize the ideation state of the user, the tool takes several values for the indicator measures. Once the user inputs the values for the characterization measures corresponding blocks are identified. Then appropriate ideation methods mitigating corresponding blocks are suggested. The input UI for ideation characterization is shown in Figure 82. This tool is retained from V.1.



Find ideation state

Ideation state determinator

How does your problem rate on the following measures?

Novelty of the problem

○ Low
● Medium
○ High

Complexity of the problem (numbe

○ Low
● Medium
○ High

Uncertainty of the problem

○ Low
● Medium
○ High

How much time have you spent du

● little
○ a lot

How will you score your outcome based on quantity?

○ Low
○ Medium
● High

How will you score your outcome based on quality?

○ Low
● Medium
○ High

How will you score your outcome based on novelty?

○ Low
○ Medium
● High

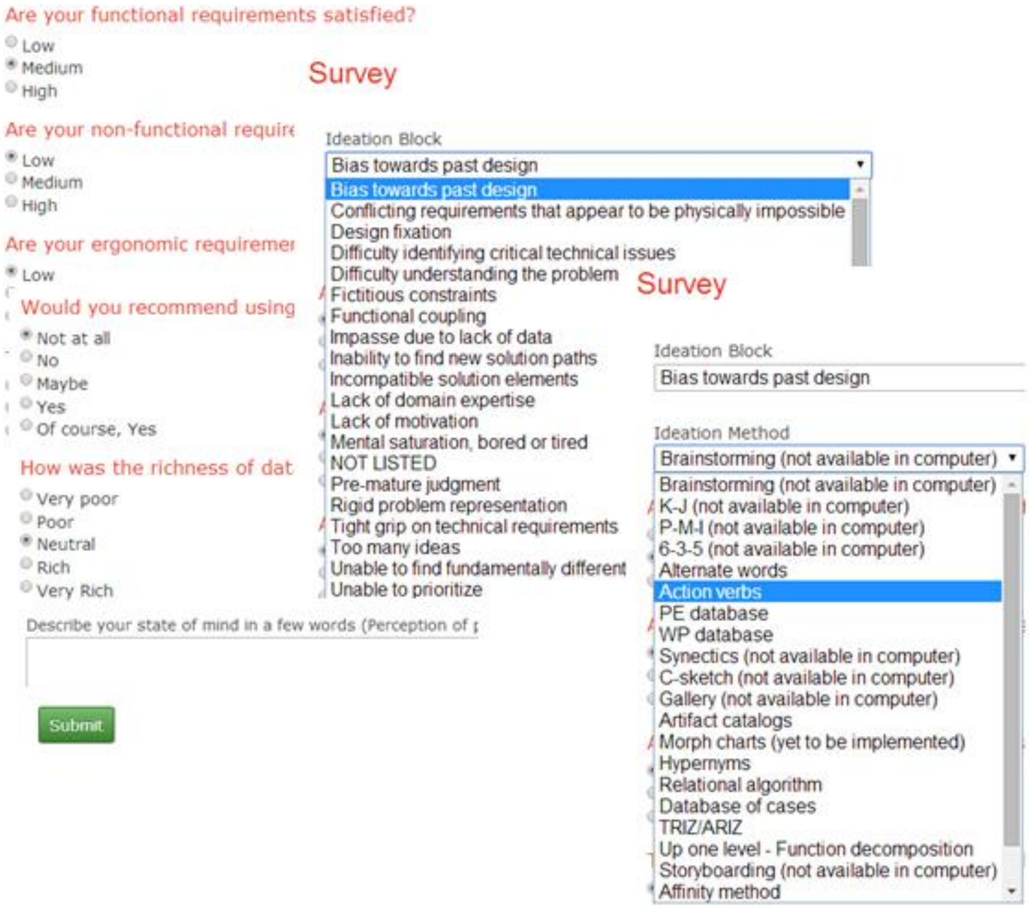
How will you score your outcome based on variety?

○ Low
○ Medium
● High

Figure 82: Input UI for Find Ideation State Module **(Mohan M. , 2011)**

5.5.9.2 Feedback Module

The survey tool is used to collect information about how well the designers' functions were satisfied by the ideation methods used. It also asks details about the effectiveness of the ideation method used. The designer can also describe his/her state of mind in this UI. Figure 83 shows the UI for the feedback module "Was the Strategy Useful?".



Survey

Are your functional requirements satisfied?

○ Low
● Medium
○ High

Are your non-functional require

● Low
○ Medium
○ High

Are your ergonomic requiremer

● Low

Would you recommend using

● Not at all
○ No
○ Maybe
○ Yes
○ Of course, Yes

How was the richness of dat

○ Very poor
○ Poor
● Neutral
○ Rich
○ Very Rich

Describe your state of mind in a few words (Perception of p

Ideation Block: Bias towards past design

Survey

Ideation Block: Bias towards past design

Ideation Method: Brainstorming (not available in computer)

Bias towards past design
Conflicting requirements that appear to be physically impossible
Design fixation
Difficulty identifying critical technical issues
Difficulty understanding the problem
Fictitious constraints
Functional coupling
Impasse due to lack of data
Inability to find new solution paths
Incompatible solution elements
Lack of domain expertise
Lack of motivation
Mental saturation, bored or tired
NOT LISTED
Pre-mature judgment
Rigid problem representation
Tight grip on technical requirements
Too many ideas
Unable to find fundamentally different
Unable to prioritize
Brainstorming (not available in computer)
K-J (not available in computer)
P-M-I (not available in computer)
6-3-5 (not available in computer)
Alternate words
Action verbs
PE database
WP database
Synectics (not available in computer)
C-sketch (not available in computer)
Gallery (not available in computer)
Artifact catalogs
Morph charts (yet to be implemented)
Hypernyms
Relational algorithm
Database of cases
TRIZ/ARIZ
Up one level - Function decomposition
Storyboarding (not available in computer)
Affinity method
Submit

Figure 83: Feedback Module UI

CHAPTER 6

COMPUTER IMPLEMENTATION

V.3 of the Holistic Ideation tool created as a part of my research work is a web based tool which is implemented in Cake-Php and uses SQLite tool for managing databases. It can be used on any system through any internet browser (preferably: Google Chrome). V.2 was derived from V.1 and retained the logical ideation methods from it. Similarly, though V.3 is an independent tool, it is derived from V.1 and V.2 versions of the Holistic Ideation tool. Because of this some part of it is retained from the previous versions and that part is implemented the same way as it was implemented in the previous versions. The following sections will discuss the back end databases and implementation of V.3.

6.1 Database Schema

In this section, we will present the relationship diagrams for the CEMC and each of the design repositories.

6.2.1 ER Diagram and Database Schema for CEMC

The ER diagram for CEMC is shown in Figure 84. In V.3 a "USER" can have one or more "USER SESSIONS". Each "USER SESSION" can be used to store one and only one "MORPH CHART". This "MORPH CHART" can contain one or more "PROBLEMS". Each "PROBLEM" is allowed to have multiple "CHILD PROBLEMS" but only one "PARENT PROBLEM". Each "PROBLEM" can also have multiple "SOLUTIONS" attached to it which can be combined to create "SOLUTION SETS". And finally, the "COMPLETE SOLUTION" can have one or more "SOLUTION SETS". Each of these entities are designed to store their respective attributes such as name, description, etc.

Using this model, information about the problems, sub-problems, solutions, sub-solutions, solution sets, and complete solutions is stored. Figure 85 shows the database schema for CEMC. The "users" table stores the information about the user. Each user is then connected to entity from "sessions" table. Each session can have "morph chart problems" by id, name and root. Each of the morph chart problem can be connected to entity from "morph chart solutions" table. Solutions are also connected to entities from "morph chart solutions sets" table. It contains the information related to which solution set the morph chart solution is a part of.

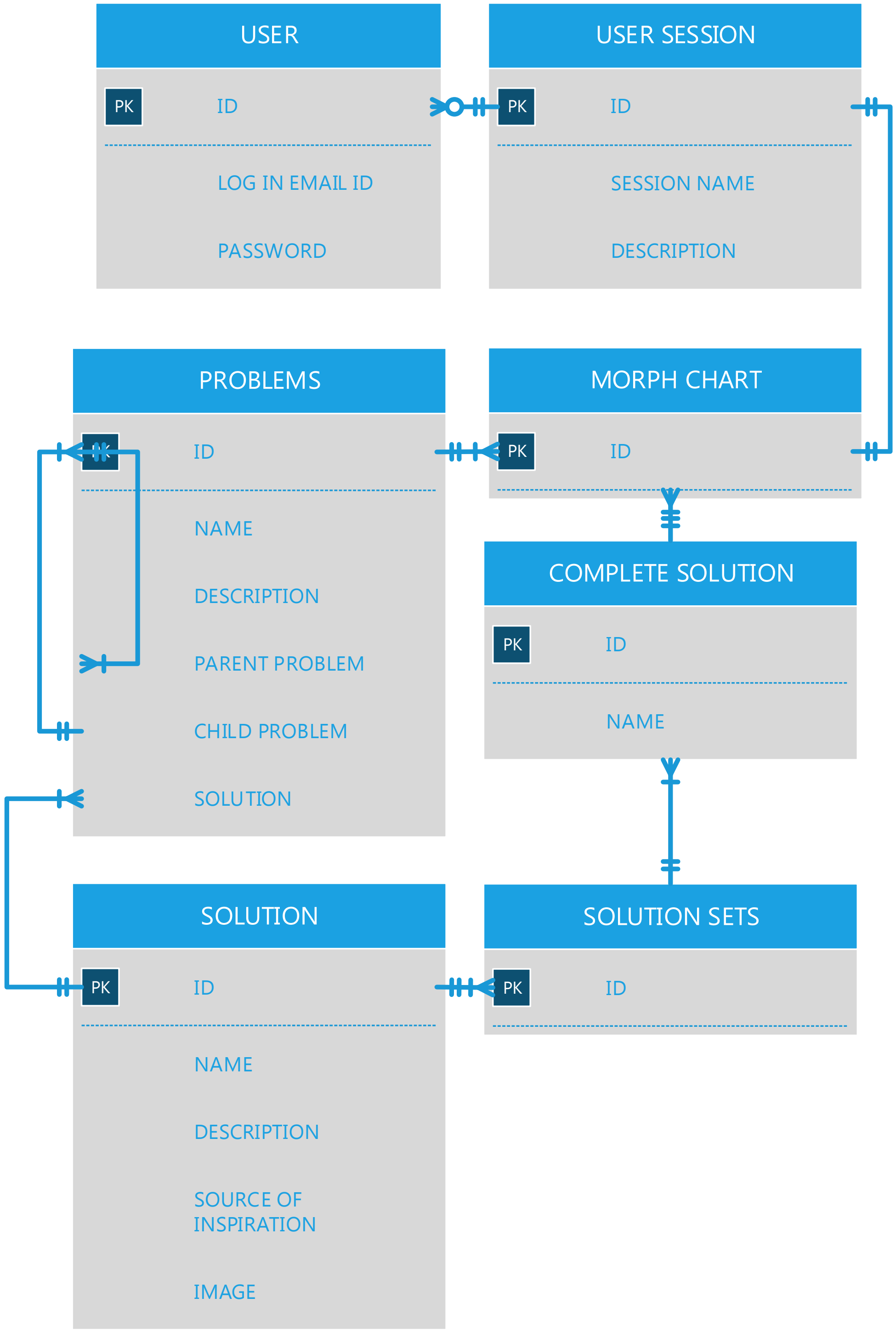


Figure 84: ER Diagram for CEMC

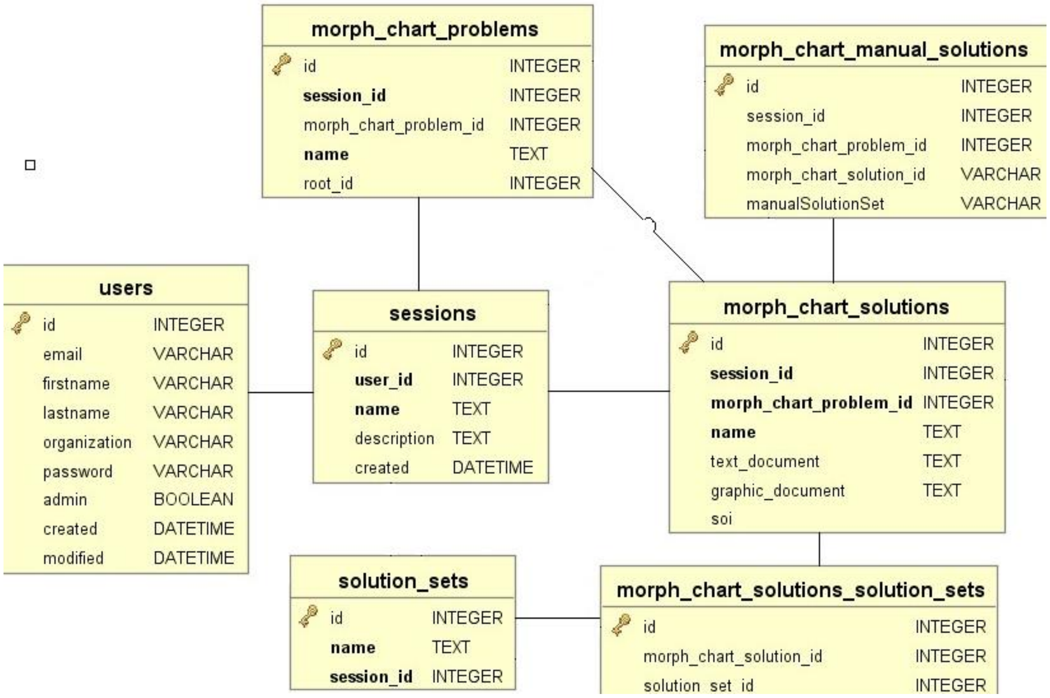
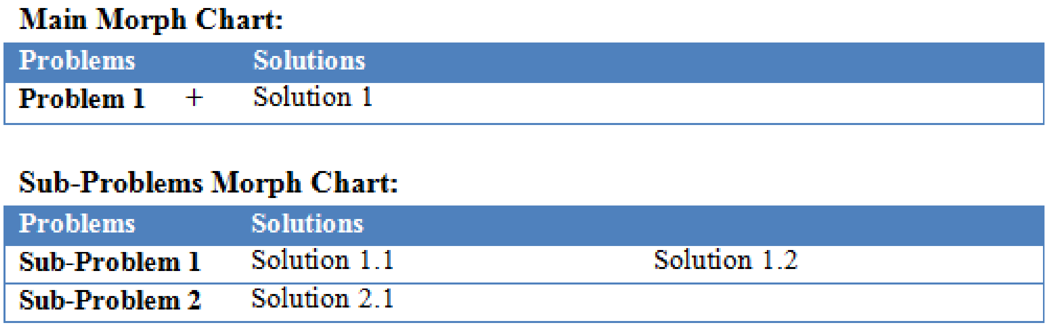


Figure 85: Database Schema for CEMC

The data structure of the CEMC can be visualized as a Tree. The contents of the CEMC can be displayed using any basic Tree traversal algorithm. For example, Consider a Morph Chart shown in Figure 86.



Main Morph Chart:

Problems		Solutions
Problem 1	+	Solution 1

Sub-Problems Morph Chart:

Problems	Solutions	
Sub-Problem 1	Solution 1.1	Solution 1.2
Sub-Problem 2	Solution 2.1	

Figure 86: Sample Morph Chart

The above morph chart can be represented as a tree structure as shown in Figure 87.

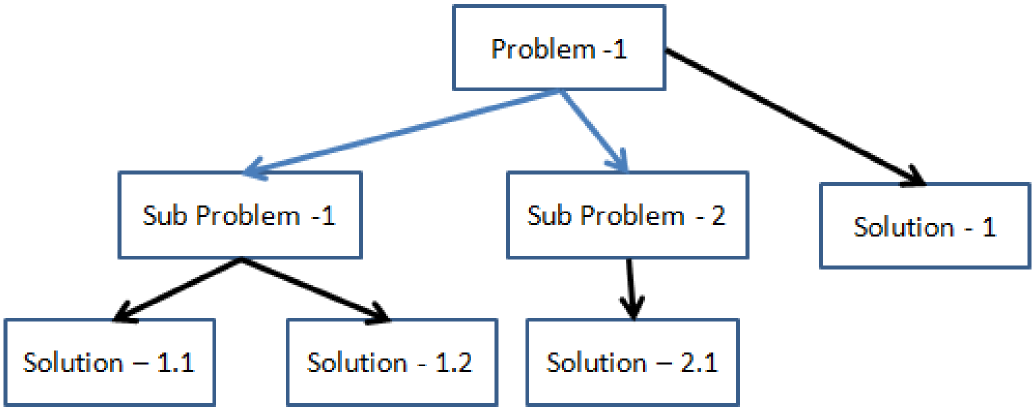


Figure 87: Morph Chart in Tree Format

6.2.2 Mechanisms and COTS Database

The two newly added databases are the Mechanisms database and the COTS database. In previous chapters we discussed the representation used for these databases. Some of the attributes are common in the definition of the mechanisms and COTS (i.e. Input, Output, Relation, and Funtion). Some of these attributes such as Input, Output, and Relation are given values from pre-defined set of values. For these attributes the list of all possible values is shown in table 17 and 18. These pre-defined set of values are same for both Mechanisms and COTS databases.

Table 17: All Possible Values for INPUT and OUTPUT Attributes for Mechanisms and COTS Databases

INPUT/OUTPUT		
TYPE	***SPEED***	***VELOCITY DIRECTION***
ROTATION	CONSTANT	CONSTANT
TRANSLATION	INTERMITTANT	VARY
CYLINDRICAL	VARY	
HELICAL	OSCILLATE	
NA	ZERO	
SPHERICAL	FREE	
PLANAR	NA	
HYPOCYCLOIDAL		
EPICYCLOIDAL		
CONCHOIDAL		
LEMINISCATE		

Table 18: All Possible Values for RELATION Attributes Mechanisms and COTS Databases

RELATION	
RELATION BETWEEN INPUT AND OUTPUT LINE OF MOTION	***REVERSIBILITY***
COINCIDENT	YES
COPLANAR - PARALLEL	NO
COPLANAR - PERPENDICULAR	
COPLANAR - SKEWED	
NON COPLANAR	
NA	

"NA" value for any of the above attribute means that category is not applicable for that specific mechanism/COTS.

The attributes under the Descriptive category are different for mechanisms and COTS. For the mechanism database, there are several attributes under the category of "Descriptive" as discussed in previous chapter. Of these attributes Dimension, degree of freedom (DOF), and Group attributes values are selected from a pre-defined set of values. These values are listed in table 19 and 20.

Table 19: All Possible Values for DISCRIPTIVE Attributes for Mechanisms Database

DESCRIPTIVE	
DIMENSION	***DOF***
LINEAR	1
PLANAR	2
SPATIAL	3

Table 20: All Possible Values for DISCRIPTIVE Attributes for Mechanisms Database

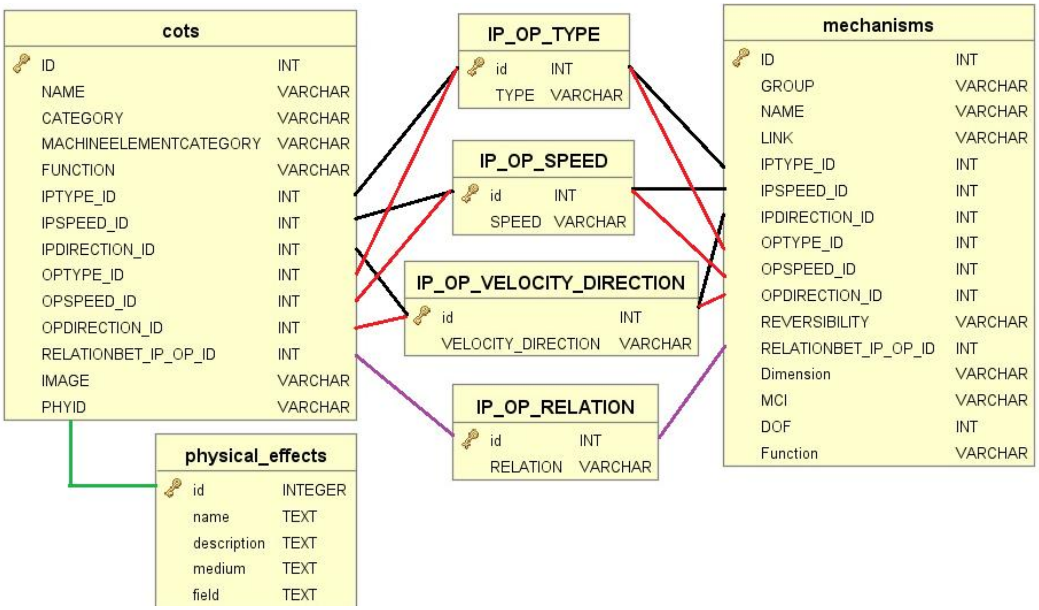
DESCRIPTIVE	
GROUP	
1. Lower Element Pairs	25. Ratchet Mechanisms
2. Higher Element Pairs	26. Planetary Gear Trains
3. Simple Kinematic Chains	27. Jointed Couplings
4. Crank Mechanisms	28. Gear Teeth Profiles
5. Eccentric Slider Cranks	29. Cycloid Rolling Models
6. Crank Chamber Mechanisms	30. Straight-line Mechanisms
7. Simple Gear Trains	31. Parallel Guide Mechanisms
8. Model Support Pedestals	32. Rotating Arm Guiding Mechanisms
9. Gear Chains	33. Gear Mechanisms
10. Gear Collection	34. Belt Drive Mechanisms
11. Chamber Wheel Mechanisms	35. Water Handling and Power Devices
12. Positive Return Constant Breadth Cams	36. Steam Power Group, Engines, Turbines
13. Crank Transmissions	37. Coupling Mechanisms
14. Prismatic Return Mechanisms	38. Simple Machines and Drives
15. Heart-Shaped Cams	39. Universal Joints, Out of Line Drives, Eccentrics
16. Balance beam mechanisms and paddle wheels	40. Countershafts, Straight-Line Motions, Cams
17. Revolute Return Mechanisms	41. Gearing Devices
18. Engaging and Disengaging Gears	42. Household Devices
19. Steam Engines with Valve Control	43. Automobile Section
20. Miscellaneous	44. Clock Escapements
21. Linkages	45. Friction Wheels
22. Screw Mechanisms	46. Complex Slider Crank Mechanisms
23. Clock Escapements, Power Hammers and Punch	47. Reversing and Shifting Belt and Gear Mechanisms
24. Ratchet Wheels, Drives	

Similarly, for COTS, values for the attributes of Machine Element Category, Related Physical Effect, and Group are selected from a pre-defined set of values, which are shown in table 21.

Table 21: All Possible Values for DISCRIPTIVE Attributes for COTS Database

DESCRIPTIVE				
MACHINE ELEMENT CATEGORY	***RELATED PHYSICAL EFFECT***	***GROUP***		
Power Transmission Elements	All the Physical Effects from PE Database	Shafts	Gear Drives	Flywheels
Fastening Elements		Couplings	Gear Box	DC Motors
Speed Reduction Elements		Keys	Brakes	AC Motors
Energy Dissipation Elements		Pins	Bearings	Special Motors
Intermittent Energy Transfer		Spline	Clutches	
Energy Storage Elements		Belt Drives	Springs	
Prime Movers		Chain Drives	Flywheels	

Figure 88, shows the database schema for Mechanisms and COTS databases. The new databases are implemented using the representation form discussed above. The common attributes in both the databases are implemented as independent tables named "IP OP TYPE", "IP OP SPEED", "IP OP VELOCITY DIRECTION", and "IP OP RELATION" respectively. The uncommon attributes can be accessed through the different columns embedded in the Mechanisms and COTS tables named "cots" and "mechanisms". The COTS database is connected with the rest of the knowledge base through related physical effects from the "physical effects" table, which can serve as a transition point for a designer to look at the rest of the knowledge base. All the attributes in the following database schema are mapped from the Tables 10 to 14.



cots	
ID	INT
NAME	VARCHAR
CATEGORY	VARCHAR
MACHINEELEMENTCATEGORY	VARCHAR
FUNCTION	VARCHAR
IPTYPE_ID	INT
IPSPEED_ID	INT
IPDIRECTION_ID	INT
OPTYPE_ID	INT
OPSPEED_ID	INT
OPDIRECTION_ID	INT
RELATIONBET_IP_OP_ID	INT
IMAGE	VARCHAR
PHYID	VARCHAR

physical_effects	
id	INTEGER
name	TEXT
description	TEXT
medium	TEXT
field	TEXT

IP_OP_TYPE	
id	INT
TYPE	VARCHAR

IP_OP_SPEED	
id	INT
SPEED	VARCHAR

IP_OP_VELOCITY_DIRECTION	
id	INT
VELOCITY_DIRECTION	VARCHAR

IP_OP_RELATION	
id	INT
RELATION	VARCHAR

mechanisms	
ID	INT
GROUP	VARCHAR
NAME	VARCHAR
LINK	VARCHAR
IPTYPE_ID	INT
IPSPEED_ID	INT
IPDIRECTION_ID	INT
OPTYPE_ID	INT
OPSPEED_ID	INT
OPDIRECTION_ID	INT
REVERSIBILITY	VARCHAR
RELATIONBET_IP_OP_ID	INT
Dimension	VARCHAR
MCI	VARCHAR
DOF	INT
Function	VARCHAR

Figure 88: Database Schema for COTS and Mechanisms

6.2.3 Physical Effect Database

The representation form and hence the database schema for Physical Effects are retained from previous versions. No changes have been made to database from V.1. The database schema for the physical effects database is shown in Figure 89. Physical Effects are represented using:

1. Name - Stored as an attribute in table "PE main"

2. Description - Stored as an attribute in table "PE main"

3. Related physical equations - Stored as an attribute in table "PE main"

4. Related physical parameters - Stored as an attribute in table "PARAM main"

5. Occurrence medium - Stored as an attribute in table "PE main"

The "PE main" table contains the attributes to store name, description, occurrence medium and field of the PE. The equation corresponding to the PE is stored in the "EQN main" table. The "EQN PE" tables acts as a joint table for connecting the equations to related PEs. The "PARAM main" table stores parameters which are joined to equations via "EQN PARAM" table. The "flow pe" table joins the PEs to related flow entity stored in "flow" table.

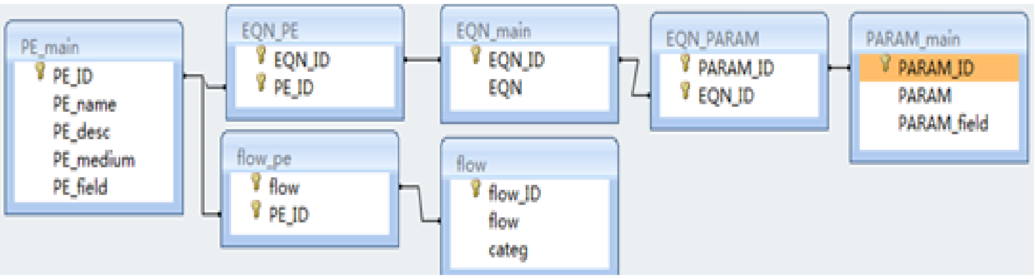


Figure 89: Database Schema for Physical Effects

6.2.4 Working Principle Database

For Working Principles also, the representation form and hence the database schema is retained from previous versions. No changes have been made to database from V.1. Figure 90 shows the database schema for WP. Working principles are defined using :

1. Name - Stored as an attribute in table "WP name desc"

2. Description - Stored as an attribute in table "WP name desc"

3. Related physical effects/variables - Stored as an attribute in table "WP PhyVars"

4. Intended functions - Stored as an attribute in table "WP name desc"

5. Graphical representation - Stored as an attribute in table "WP name desc"

6. Biological examples - Stored as an attribute in table "WP name desc"

The WP table "WP name desc" contains information regarding name, description, related physical effects/variables, intended functions, graphical representation and biological examples for any particular working principle. The WPs are connected to related physical variables stored in "PhyVars" table via the joint table "WP PhyVars". Related WPs and PEs are connected through the joint table "WP PE".

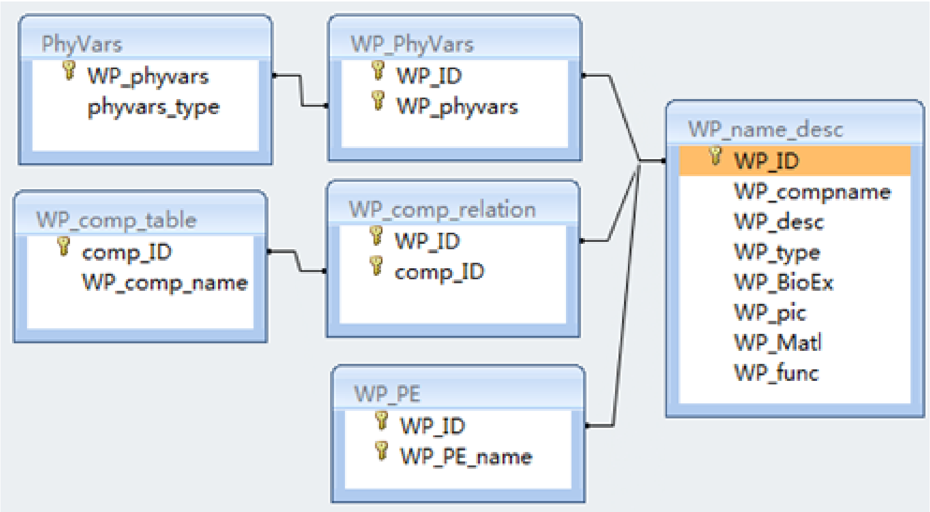


Figure 90: Database Schema for Working Principles

6.2.5 TRIZ/Bio-TRIZ Database

As we discussed in previous section, based on the contradicting parameters and the TRIZ/BioTRIZ matrix, appropriate principles are stored. The database are retained from V.1 without any change. The database schema shown in Figure 91. The "triz matrix" table contains id of TRIZ invention principles and TRIZ parameters. The related functions and flows from "func" and "flow" tables are connected through a connector table "func triz". The "bio triz" table contains attributes for related bio principles which are connected to the "triz matrix" table though the connector tables named "imp param" and "worsen params".

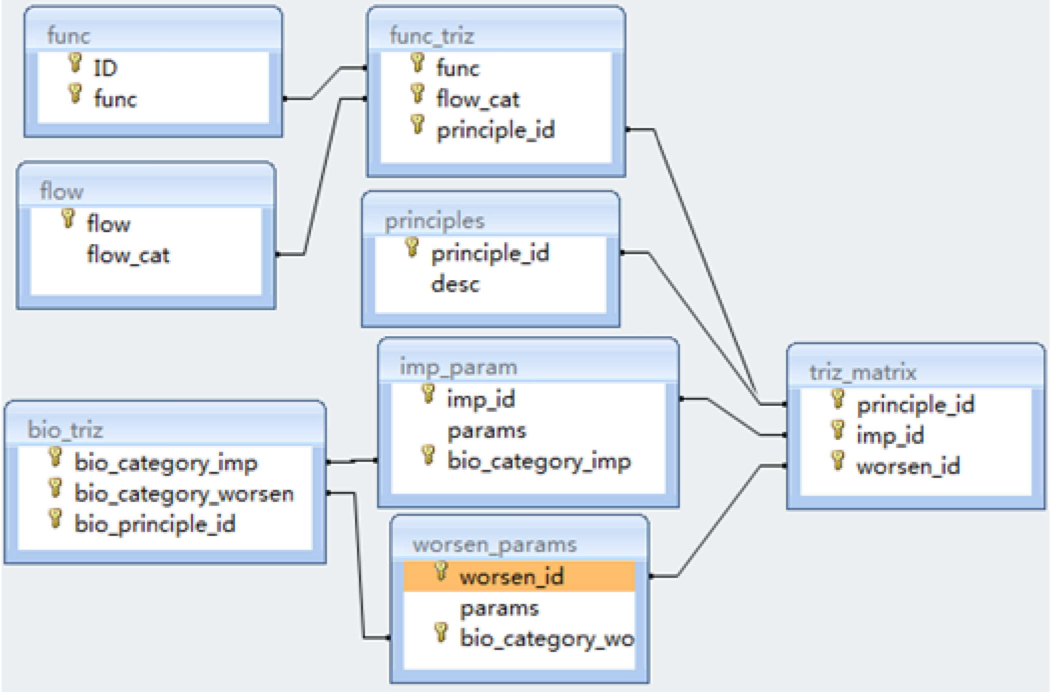


Figure 91: Database Schema for TRIZ/Bio-TRIZ

6.2.6 Artifacts Database

In the V.1 review section, we discussed about how artifacts are defined. The following Figure 92 shows the design repository schema taken from the work of Robert Stone et al. (Bohm, Stone, Simpson, & Steva, July 2008). This database stores the useful information attributes related to artifacts such as the function it performs, the failure modes, performance characteristics, material, parent and child components etc. The shaded circles (e.g. material, media, failure, function flow, manufacturing process, etc.) imply property of higher importance whereas the un-shaded circles (e.g. failure rating, sound, texture, parameter metric, etc.) represent additional non essential details. This function artifact catalog allow designers to search for artifacts according to functions. We have imported this database without any change.

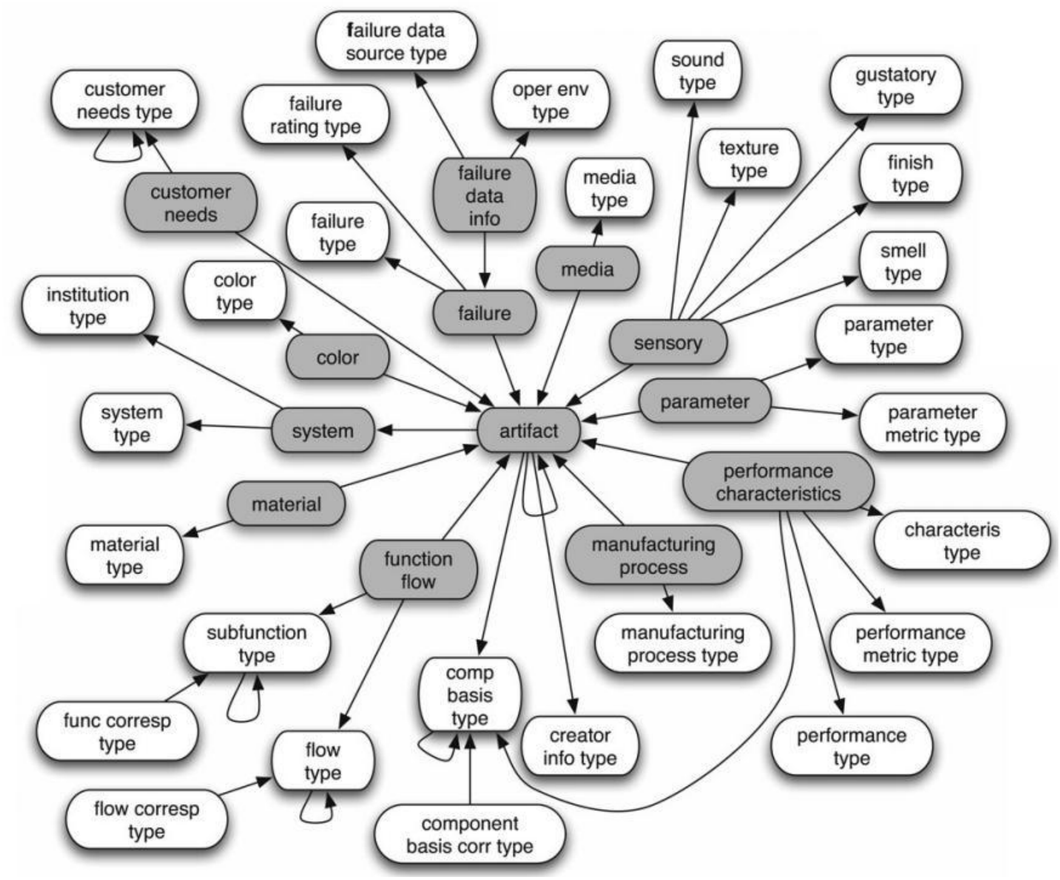


Figure 92: Database Schema for Artifacts Database **(Bohm, Stone, Simpson, & Steva, July 2008)**

6.2.7 Integration of All Databases

In previous sections, we learned that the "function" was decided to be the common denominator of all the knowledge bases. Because of this, function is used to form the common link between all the independent tables and connect them. Figure 93 shows the overall database schema for all the knowledge bases and how they are connected with each other by using "functions". TRIZ database table "principle" which contains information related to TRIZ invention principles is connected to function table "funcs" via the connector table "function trizs". WP table "working principles" is connected to the "funcs" table through the connector table "high function working principles" table. The PE table "physical effects" and WP table "working principles" are connected by a joint table "physical effects working principles". While the COTS table "cots" is directly connected with related PEs from "physical effects" using a foreign key id. The names of the database tables are created to be in accordance with the naming convention used by CakePhp environment on which is the V.3 is built.

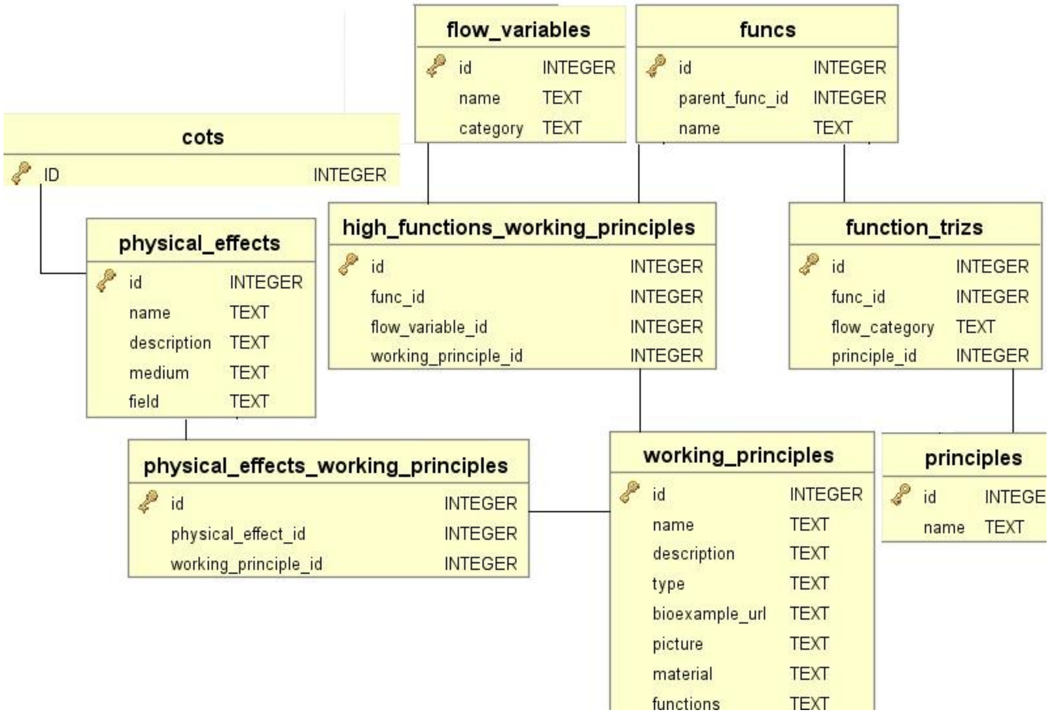


Figure 93: Integration of All Databases

6.2 MVC Software Framework

V.3 is developed as a website (www. Ideationspace.com) using CakePhp framework (CakePhp). CakePhp uses a Model-View-Controller (MVC) structure for software development. CakePhp is a rapid application development framework designed on top of the MVC file structure. It gives advantage of using 3000+ plugins implemented by open source community. We have integrated several such plugins such as Grid Plugin for displaying nested morphological charts, Inline Drawing Plugin for rendering sketches, efficient caching based Search Plugin for querying the back end databases.

MVC is a software architectural patter for implementing software tools which has multiple user interfaces built on databases. It divides a software application into three different parts (Model, View and Controller) which are interconnected. This three part structure separates internal representations of information from the ways that information is presented to the user. Figure 94 shows how these components collaborate with each other and the user. View files creates the UI of the tool which is seen by the user. When user clicks on any button he/she uses the functions written in Model files which then updates the Model files which are connected to the back end databases. The Model files fetches or changes the appropriate data from the database to update the View files. This leads to updating the UI seen by the user.

The central component of the MVC structure is Model. It captures the behavior of the application independent of the user interface. Model directly manages the databases. A View can be any output representation of information such as a morphological chart and knowledge base. It allows multiple views of the same information. The Controller accepts input from the user and converts them to commands for Model.

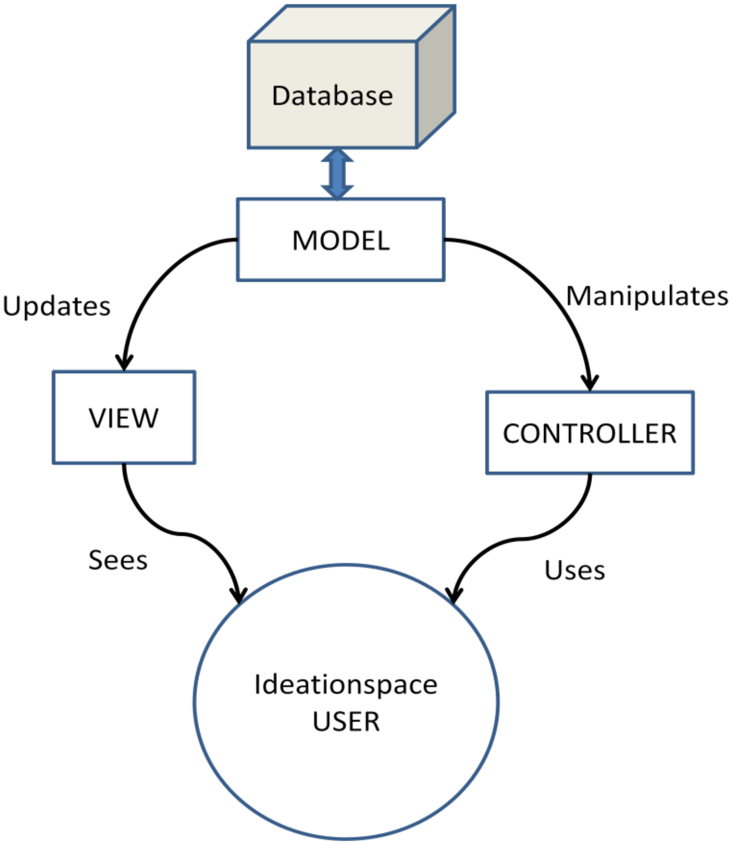


Figure 94: Collaboration of MVC Components

6.2.1 Interactions Between MVC

In addition to dividing the application into three kinds of components, the MVC structure also defines the interactions between them. Each Model file has respective View and Controller file and vice versa. These files do not exist independently. The respective MVC files has interactions between each other. These interactions can be summarized as following:

1. A Controller sends commands to the Model to update the databases (e.g. adding a problem or solution to the CEMC).

2. A Model notifies its associated Views and Controllers when there has been a change in the databases or change in the data fetched from the database (e.g. fetching different PEs, WPs, Artifacts, Mechanisms, COTS etc. according to the command from Controller).

3. A View requests information from the Model and uses it to generate the desired output presentation to the user (e.g. showing sub morphological chart or new ideation method page to the user).

6.2.2 Functions Performed by Different MVC Files

In this section we will discuss different MVC files and their functions.

6.2.2.1 Model Files

Each ideation method has its own Model files. These model files fetches the appropriate data from the databases according to the users' needs. The Model file contains classes which are pre-defined representations of data from the database. These classes get instantiated to select appropriate data from the database according the commands received from Controller files. As the V.3 uses large databases mostly these commands includes multiple queries described in the section 6.4. These queries are used by model files to fetch the data from back end databases. Table 22 lists important Model files and their functions.

Table 22: Model Files and Their Function

Model Files	Function
User.php	Fetches or edits the data for user sessions and log in details.
MorphChartProblem.php MorphChartSolution.php SolutionSet.php	Fetches and edits the data related to CEMC. Fetches and edits morph chart problem, sub problem, solution and solution sets related information.
Artifact.php BioTriz.php Cot.php Mechanism.php PhysicalEffect.php WorkingPrinciple.php TrizMatrix.php	These model files does not edits any data. These can only fetch the appropriate data from the database in pre-defined form according to the queries submitted by the Controller files.

6.2.2.2 View Files

Each page from the user interface of V.3 require a View file. These View files takes the data fetched from the database by Model files as an input and renders the web page as the output. Each ideation method is rendered using its own View file. Also, CEMC has its own View files to show the morph chart on the screen. Because V.3 UI comprises of large number of pages only important View files and their respective functions are listed in Table 23.

Table 23: View Files and Their Function

View Files	Function
MorphChartProblem.ctp MorphChartSolution.ctp SolutionSet.ctp	Renders a CEMC into tabular format using Grid and Sketching Plugins.
Artifact.ctp BioTriz.ctp Cot.ctp Mechanism.ctp PhysicalEffect.ctp WorkingPrinciple.ctp TrizMatrix.ctp	Renders any page in pre defined format depending upon the data received from the respective Model files.

6.2.2.3 Controller Files

Controller files store the logic of the software. When user clicks on any button on the UI, a call to the appropriate controller file is made where all the logic is run. After that these controller files sends appropriate commands to the model files. As V.3 is dependent on databases most of these commands take the form of a query call. Table 24 shows some of the important Controller files and their respective function.

Table 24: Controller Files and Their Function

Model Files	Function
MorphChartProblem.php MorphChartSolution.php SolutionSet.php	Commands Model files to fetch the appropriate CEMC related data from the database as per the user's request (as per the button clicked by the user).
Artifact.php BioTriz.php Cot.php Mechanism.php PhysicalEffect.php WorkingPrinciple.php TrizMatrix.php	Commands Model files to fetch the appropriate data from the knowledge bases as per the user's request (as per the button clicked by the user).

6.3 Algorithms for the Use Cases

While working on V.3 a user can perform many different tasks, e.g., adding a sub problem, adding a sub solution, automatic solution population, creating solution sets, browsing any ideation method. These are called as use cases. Algorithms and pseudo code for these cases are listed below. The actual implementation code is documented in Appendix C.

6.3.1 Adding a problem/sub problem

Morph chart contains the set of problems and their respective set of sub problems. This use case is for adding the new problem or sub problem in existing morph chart.

Algorithm:

1. Accept the name of the problem and its parent problem name as an input from the user.

2. If the problem has no parent then set the parent problem value to NULL and thus add the new problem in the highest hierarchy in the CEMC database.

3. If the problem has a valid parent then add it as the sub problem for the selected parent problem.

Pseudo Code:

```
function AddProblem(NameOfProblem, NameOfParentProblem)
{
    If (NameOfParentProblem is not NULL)
    {
        Look for the NameOfParentProblem in the "MorphChartProblems"
        table and fetch it's corresponding id.

        Add the new entry into table "MorphChartProblems" with the
        parent id fetched above.
    }
    Else
    {
        Add the new entry into table "MorphChartProblems" with the
        parent  id as null.
    }
}
```

6.3.2 Adding a Solution Manually

Morph chart contains the set of problems and their respective set solutions. This use case is for manually adding the new solution in existing morph chart.

Algorithm:

1. Accept the name solution and its parent problem name as an input from the user.

2. If the solution has no parent then show a error message.

3. If the solution has a valid parent problem then add it as the solution for the selected parent problem.

Pseudo Code:

```
function AddProblem(NameOfSolution, NameOfParentProblem)
{
        If (NameOfParentProblem is not NULL)
        {
                Look for the NameOfParentProblem in the "MorphChartProblems"
                table and fetch it’s corresponding id.

                Using that id add the new entry into table "MorphChartSolution"
        }
        Else
        {
                Show an error message.
        }
}
```

6.3.3 Automatic Solution Population

Solutions can also be added automatically to respective parent problems. This use case is for automatically adding the new solution in existing morph chart.

Algorithm:

1. When user chooses any method from the dropdown list given below the problem button temporarily store the problem id in variable "A".

2. Show the web page for the ideation method selected from the dropdown menu. Also show a add solution button in respective page.

3. If the user stores any solution take the name as an input and store it as a solution for the problem with id stored in "A".

4. In case of user not submitting any solution before closing the respective window clear "A".

Pseudo Code:

```
function AutomaticSolutionPopulation(SelectedProblemName)
{
        A =  problem id from "MorphChartProblem" table for the problem with
        the name "SelectedProblemName".

        Show the web page for selected ideation method with add solution button
        at the bottom.

        If (User adds a solution)
        {
                Using A add the new entry into table "MorphChartSolution"
        }
        Else
        {
                A = NULL;
        }
}
```

6.3.4 Adding Solution Sets

Solutions sets can be added to respective parent problems by combining solutions from sub problems. This use case is for creating solution sets.

Algorithm:

1. When user selects any combination of solutions from checkboxes given in front of the solution name, create a string variable "SET" and store the respective ids (with a single space in between two id numbers) as a string in variable "SET". Also store the parent problem id in "A".

2. If user clicks on save solution button save the "SET" in "SolutionSets" table for parent problem id "A"

3. In case of user not submitting any solution set before closing the respective window clear "A" and "SET".

Pseudo Code:

```
function CreateSolutionSets(ParentProblemId)
{
        A =  ParentProblemId
        SET  = NULL

        If (User clicks checkboxes)
        {
                Fetch the solution ids from the table "MorphChartSolutions" and
                store it in "SET" as "x y z" (where x,y and z are respective solution
                ids)
        }

        If (User clicks on add solution set button)
        {
                Store the "SET" value in the table "SolutionSets" where parent
                problem id is equal to "A"
        }

        A = NULL;
        SET = NULL;
}
```

6.3.5 Displaying the CEMC

CEMC is displayed on the web page in a tabular format. This use case is for displaying the CEMC.

Algorithm:

1. For the given user session id print each of the problem in the "MorphChartProblems" table.

2. If the problem has sub problems output the link to the sub morph chart.

3. For each solution for the selected problem output the solution details from the "MorphCahrtSolutions" table.

4. For each solution set for the selected problem output the solution set details from the "SolutionSets" table.

Pseudo Code:

```
For each Problem
{
    Output the Problem Name

    if (the Problem has Sub Problems)
    {
        Output link to Sub Problem Morph Chart
    }

    For each Solutions of this Problem
    {
        Output the details of Solution
    }

    For each Solution Sets from sub-problems
```

```
        {
                Output the details of manual solution sets
        }
}
```

6.3.6 Browsing Mechanisms/COTS

Mechanisms or COTS can be browsed at anytime by the user. The results are shown as per the input from the user. While browsing mechanisms/COTS user can select multiple values for multiple attributes. This use case explains browsing newly added mechanisms and COTS ideation methods.

Algorithm:

1. Save the multiple values for all the different attributes in multiple arrays (For example, from attribute list A user selects values X,Y,Z and from attribute list B user selects values L,M,N. Save X,Y,Z and L,M,N in two different arrays).

2. Go the mechanism/COTS database and fetch the data for selections X,Y,Z. Take union of that data set. Store it in a temporary dataset named XYZ.

3. Go the mechanism/COTS database and fetch the data for selections L,M,N. Take union of that data set. Store it in a temporary dataset named LMN.

4. Take the intersection of the dataset XYZ and LMN. Present the output to the user.

Pseudo Code:

```
function BrowseMechanisms/COTS( )
```

```
{
        Save the selected multiple values of the attributes as -
        For attribute A save X,Y,Z.
        For attribute B save L,M,N.

        Fetch the data from the tables "mechanims"/"cots" for six different values
        X,Y,Z,L,M, and N.

        XYZ = X U Y U Z (store the union of the fetched data)
        LMN = L U M U N (store the union of the fetched data)

        XYZLMN = XYZ ∩ LMN (store the intersection of two data sets)

        Output the entries from the XYZLMN dataset.

}
```

6.4 Query Calls in V.3

Every time user clicks on a button in the UI provided by View a query is generated by the Controller and Model makes that query call on the database to get the related data and show it to the user. In this section we will discuss different queries and query calls used in V.3. The query calls are written in italics.

6.4.1 Query Calls for Creating a CEMC

1. Problem decomposition (this step is done by designer in pre-analysis);

2. At the time of entering a problem into the CEMC check if the user defines a parent problem or not. In case of the user defining a parent problem, store it as a Sub-problem in the database;

3. Create a problem in the UI;

3.1. If parent problem is not selected create new problem:

INSERT INTO morph_chart_problems (session_id, name) VALUES (<session id>, “<problem name>”);

3.2. If parent problem is selected create new sub problem:

INSERT INTO morph_chart_problems (session_id, morph_chart_problem_id, name, root_id) VALUES (<session id>, <parent problem id>, “<problem name>”, <root problem id>);

4. Enter/edit the name of the problem - Enter/edit the name of the problem into the database;

5. Enter/edit the description of the problem - write/update the description of the problem into the database;

UPDATE morph_chart_problems SET morph_chart_problem_id = <parent problem id>, name = “<problem name>”, root_id = <root problem id>

6. Repeat step 2 to step 4 whenever a problem is added;

7. Add solution for selected problem — create space in database for this solution including text, images, and source of inspiration;

INSERT INTO morph_chart_solutions (session_id, morph_chart_problem_id, name, text_document, graphic_document, soi) VALUES (<session id>, <problem id>, “<solution name>”, “<description>”, “<sketch data>”, “<source of inspiration>”);

7.1. If an image is uploaded:

INSERT INTO morph_chart_images (morph_chart_solution_id, file_name) VALUES (<solution id>, “<image file name>”);

8. Repeat step 7 whenever a solution is added.

6.4.2 Query Calls for Automatic Population of Solutions

1. In the CEMC, choose from which database the solution needs to be added.
2. Select a solution from the database, enter a name for the solution and click submit.
3. Create a new entry in the database in the table "morph_chart_solutions" with source of inspiration as the database from which the solution was selected.

INSERT INTO morph_chart_solutions (session_id, morph_chart_problem_id, name, soi) VALUES (<session id>, <problem id>, “<solution name>”, “<source of inspiration>”);

6.4.3 Query Calls for Creating Solution Sets

1. Create a new "manual_solution_set" for the parent problem.

2. Save the "solution ids" of the selected combination of the solutions in the database table "morph_chart_manual_solutions".

INSERT INTO morph_chart_manual_solutions (session_id, morph_chart_problem_id, morph_chart_solution_id) VALUES (<session id>, <problem id>, <solution id>);

3. For each set a "*manual solution set id"* would be generated automatically and will be associated to the solution ids.

6.4.4 Query Calls for Creating Complete Solution

1. User first creates a complete solution set for the highest level morph chart by choosing a combination of its solutions.

2. Create a new entry in the database table "solution_sets".

3. Save the combination of "solution ids" of this solution set in the database table "morph_chart_solutions_solution_sets".

INSERT INTO solutions_sets (session_id, name) VALUES (<session id>, "<solution set name>");

INSERT INTO morph_chart_solutions_solution_sets (session_id, name) VALUES (<session id>, "<solution set name>");

6.4.5 Query Calls for Find Mechanisms Module

1. By Keyword - Take the input keyword from the user to search the database table "mechanisms" and fetch the results by matching the keyword with the "NAME" column in the table.

*SELECT * FROM mechanisms WHERE NAME LIKE "%<keyword>%";*

2. By Name, Group, Machine Components Involved, Functions - Choose one or more values from each category to search the database table "mechanisms" and fetch the results by matching the values with their corresponding columns in the table.

2.1 If no filter is selected:

*SELECT * FROM mechanisms;*

2.2 If one or more filers are selected, query the database by adding 'where conditions' in select query:

*SELECT * FROM mechanisms WHERE NAME = "<name>" and GROUP = "<group>";*

6.4.6 Query Calls for Select Mechanisms Module

1. By Input Type, Output Type, Input Speed, Output Speed, Input Velocity Direction, Output Velocity, Reversibility, Relation between Input and Output Line of Motion, Dimension, and Degree of Freedom - Choose one or more values from each category to search the database table "mechanisms" and fetch the results by matching the values with their corresponding columns in the table.

1.1 If no filter is selected:

*SELECT * FROM mechanisms;*

1.2 If one or more filers are selected, query the database by adding 'where conditions' in select query:

*SELECT * FROM mechanisms WHERE IPTYPE = "<input type>" and IPSPEED = "<input speed>";*

6.4.7 Query Calls for Find Machine Elements Module

1. By Keyword - Enter a keyword to search the database table "cots" and fetch the results by matching the keyword with the "NAME" column in the table.

*SELECT * FROM cots WHERE NAME LIKE "%<keyword>%";*

2. By Name, Category, Machine Element Category, Functions - Choose one or more values from each category to search the database table "cots" and fetch the results by matching the values with their corresponding columns in the table.

2.1. If no filter is selected:

*SELECT * FROM cots;*

2.2. If one or more filers are selected, query the database by adding 'where conditions' in select query:

*SELECT * FROM cots WHERE NAME = "<name>" and CATEGORY = "<category>";*

6.4.8 Query Calls for Select Machine Elements Module

1. By Input Type, Output Type, Input Speed, Output Speed, Input Velocity Direction, Output Velocity, Reversibility, Relation between Input and Output Line of Motion - Choose one or more values from each category to search the database table “cots” and fetch the results by matching the values with their corresponding columns in the table.

1.1. If no filter is selected:

*SELECT * FROM cots;*

1.2. If one or more filers are selected, query the database by adding ‘where conditions’ in select query:

*SELECT * FROM cots WHERE IPTYPE = “<input type>” and IPSPEED = “<input speed>”;*

CHAPTER 7

USER STUDIES

7.1 Objective Of The Study

In order to gauge the usability of the tool, evaluate newly added ideation methods and user tendencies, the user studies were conducted. From the user studies, some initial observations can be made on these aspects. In this chapter we will present the detail procedure of the user studies conducted.

7.2 Procedure

The user studies were conducted with graduate students from the Design Automation Lab who were also enrolled in Advanced Product design course (Shah J. , Advanced Product Design Methodology, 2011). The choice of the participants was made such that they have some preliminary knowledge about conceptual design, different ideation strategies and the holistic ideation tool. Three participants (Designer A, B, and C) were chosen to work on three problems (Problem 1, 2, and 3). To make them conversant with the tool, they were given a demo, a tutorial and some basic training. Participants A and B were given problem 1 and participant C was given problem 2 and 3. For each problem, a participant was allowed to work for 1 hour, after which, the designers were allowed to complete his/her sketches and label them properly for 10 minutes.

Problems used in the studies were as follows:

Problem 1 (Mechanical Harvesting for Orange trees)-

Design a mechanical device which can be used to collect oranges from the trees in a farm. Your device should collect and store oranges from the trees. Device should not damage the oranges or the tree in the process.

Problem 2 (8 Pattern Vehicle) -

Design an autonomous surveillance vehicle to automatically and periodically tour the perimeter of the two buildings as shown in Figure 95. The vehicle should start and stop at a point midway between the two buildings. The vehicle should travel in a figure eight pattern.

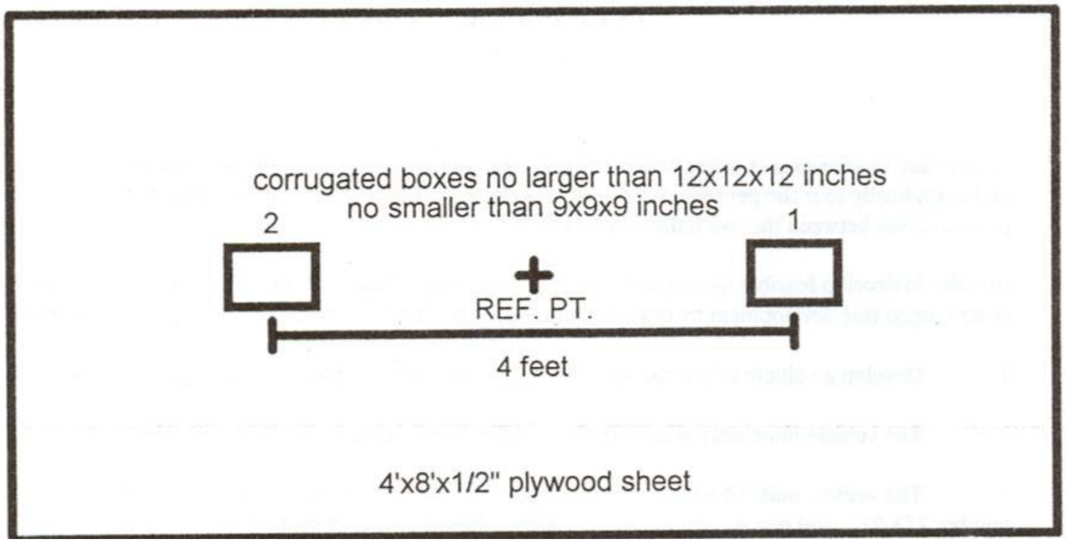

corrugated boxes no larger than 12x12x12 inches
no smaller than 9x9x9 inches
2
1
REF. PT.
4 feet
4'x8'x1/2" plywood sheet

Figure 95: Problem 2

Problem 3 (Golf collection game device) -

Design a device that can collect and store more golf balls than the opponent's device. The playing field is 5 ft x 12 ft, surrounded by a 1 ft fence, as shown in Figure 96. Collected balls need to be stored in the respective silos. There will be one red ball (50 points), 5 yellow balls (10 pts. ea.), and 15 white balls (1 pt. ea.). You are allowed to steal balls from your opponent and to interfere with the operation of their device, without destroying it. Your turn will last two minutes. To win, your device must get more points than the opponent's and follow all the rules listed below.

Rules:

1. The device must fit in a box 15 x l5 x l5 inches when in fully retracted position.

2. During a game, the only connection between the device and the operator will be via electric and/or pneumatic power cords.

3. No part of the power cord or device can cross the center line.

4. Any balls thrown out of the playing field will be awarded to the opponent.

5. You cannot cut your opponent's power cord.

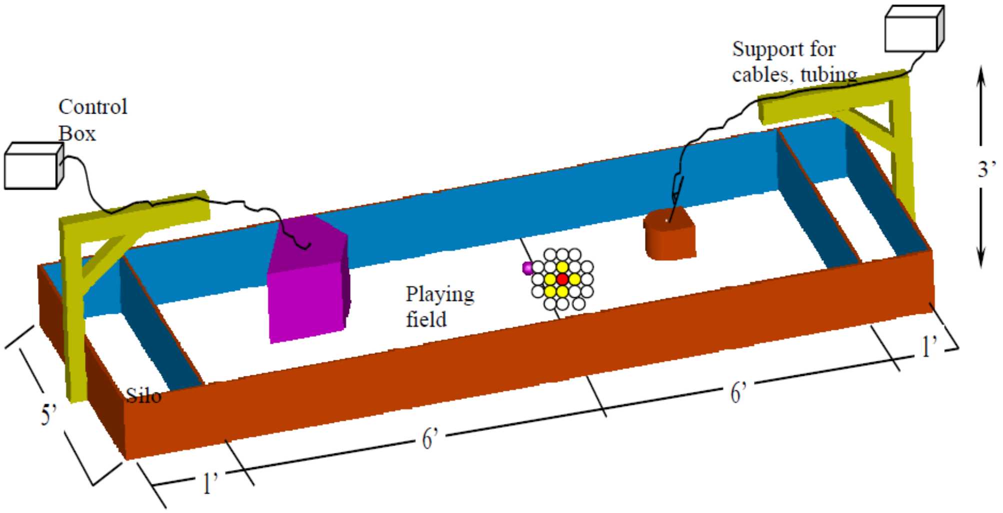


Figure 96: Problem 3

After this experiment, they were asked to fill following survey sheet to rate usability of each toolset provided in V.3. The designers were asked to rate usability in the scale of 1 - 10. They were also asked to provide the reason behind the rating. The blank survey sheet is shown in table 25.

Apart from the survey from the designers, we collected the data regarding the number of solutions that were inspired from different tool sets. This data was collected from V.3 using the "source of inspiration" field linked with each solution. This data was then categorized into different fields as shown in Table 26. Furthermore, while the designers were using the tool, we also captured the time when each of the tool set was used and solutions were generated by the designer.

Table 25: Survey Sheet for Experiments

Participant _ Survey Sheet: Holistic Ideation Tool V.3 (Use Scale of : 1 - 10)		
Please rate Usability/User-Friendliness for each toolset provided in the tool.		
Toolset	Rating	Why
1. Problem Formulation Methods		
2. Re-formulation		
3. Standard Solutions		
4. Generative methods		
5. External resources		
6. Solution Synthesis and		
7. process monitoring		
Please Provide any suggestions or feedback regarding the tool		

Table 26: Post Experiment Data Collection from the Tool

Case study No.	Total number of solutions generated	Number of solutions generated manually	Number of solutions generated using the toolset -		
			Standard Solution	Generative Methods	External Resources
1					
2					
3					
4					

From the above mentioned data collection, this study hopes to gain insight in the ideation process and to find areas of improvements needs to be done in the V.3. This study also tries to answer the question whether the new organization of the tool complement the ideation process or not. As the design problems are chosen in way that problem solving will involve mechanisms and COTS selection, we were also hoping to get feedback in the survey about the mechanism/COTS databases and the respective modules from the participants.

7.3 User Studies

7.3.1 User Study #1: Participant A - Problem 1

Participant A decided to start with function decomposition. However, designer A did not choose Function CAD module from the tool and resorted to hand-drawn function decomposition which is shown in Figure 97. Designer A included three level deep function decomposition including two disjunctions, which took him 11 minutes. Due to the time constraint, designer A decided to move on. He then added this function structure in the morph chart. Designer A, spent 7 minutes to add problems one by one into the CEMC. The final CEMC structure created by the designer A is shown in Figure 98.

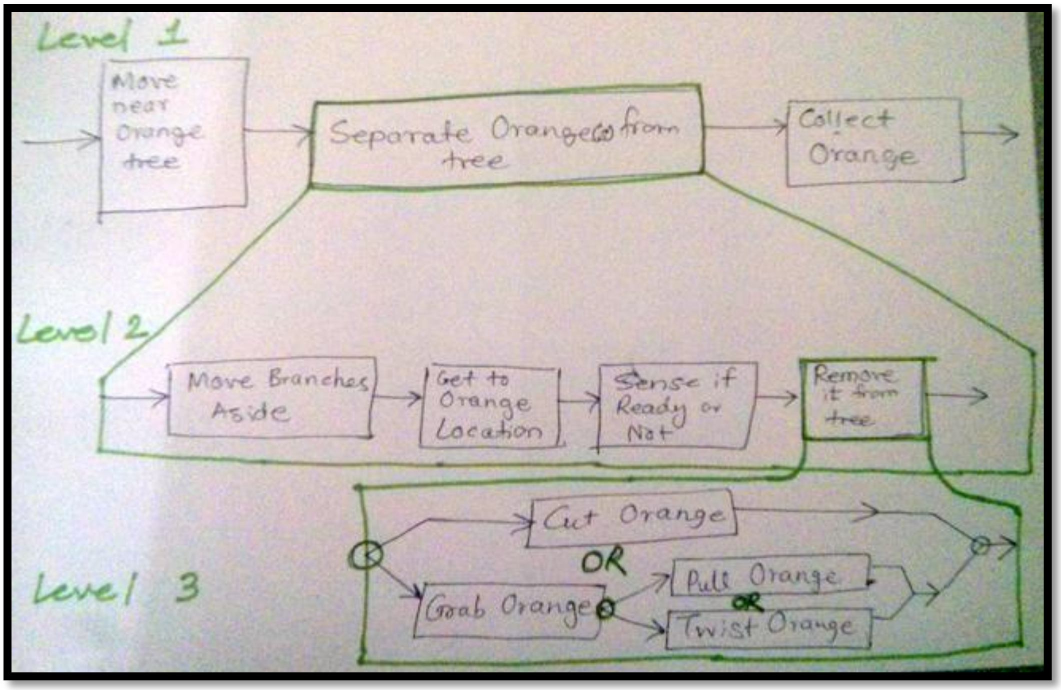


Figure 97: Function Decomposition in Case Study #1

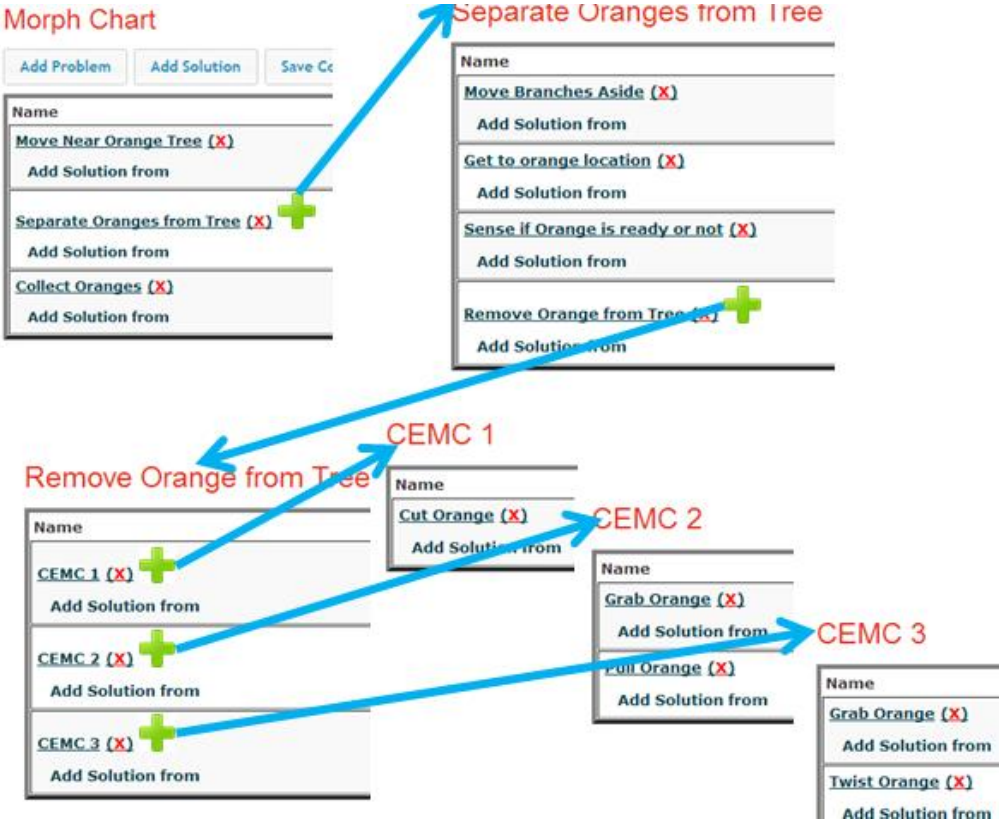


Figure 98: CEMC Created in Case Study #1

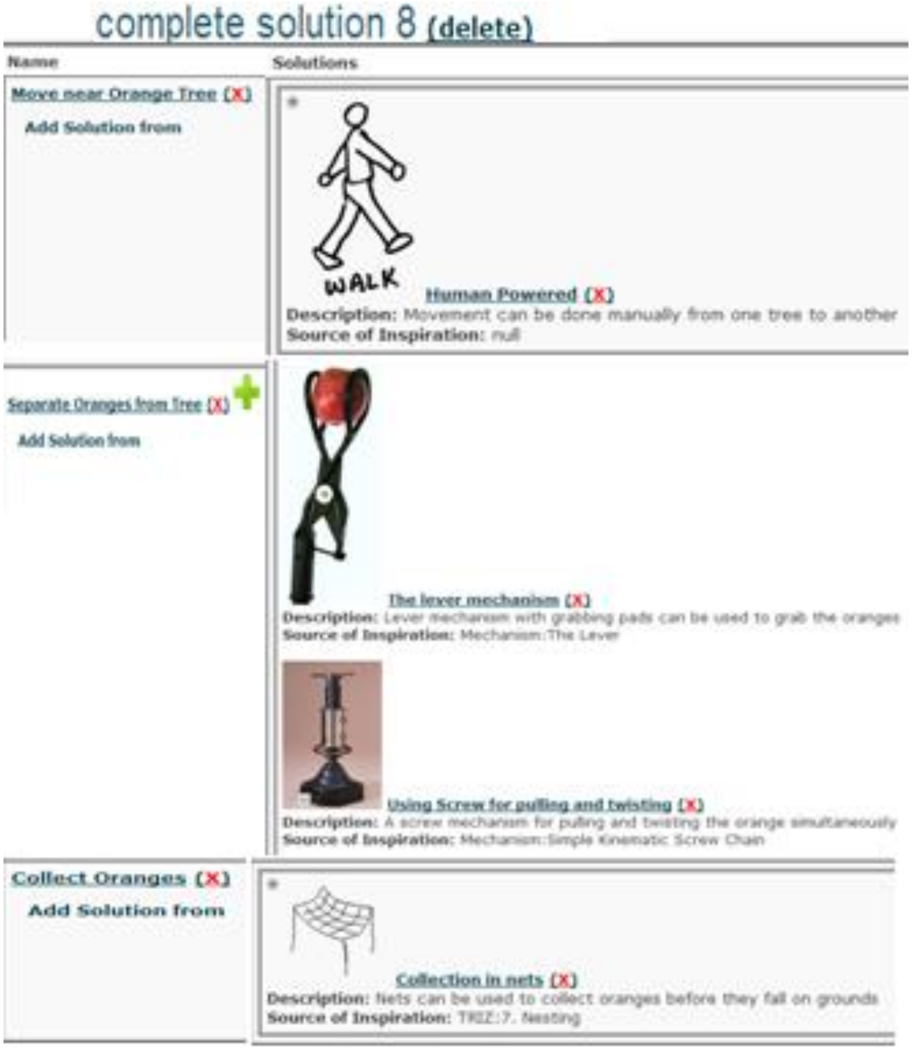
The CEMC structure created by the designer had 10 nodes (sub-problems). After adding all the problems and sub-problems to CEMC, the designer started to work on these leaf problems one by one. The designer first tackled all the leaf problems in the third level of CEMC hierarchy, and then went on to find solutions for the leaf problems in second level hierarchy and lastly, the leaf problems in the first level. While generating solutions, the designer started to work on problems without using the tool for 7-8 minutes.

After that the designer decided to use the tool sets for generating new ideas, which he did for 25 minutes. In this stage the designer mainly focused on

"Standard Solutions" toolset at first. Searching the COTS database through the related physical effect of "Bend" he found the machine element "Flexure Bearing". It inspired him for using a "Flexible Rod" as a solution for the problem "Move Branch Aside".

For the problem "Cut Orange", the designer was inspired to use "Knife" as solution from the "Artifact" database. It is noteworthy that while working on the CEMC-2 and CEMC-3 (Figure 98), which had the functions "Grab, Twist" and "Grab, Pull" respectively (Figure 98), the designer extensively searched the COTS and Mechanisms database to come up with a new mechanism which can perform these two functions at the same time. The designer worked about 10 minutes to couple these functions. After that the designer came up with a solution which can perform "Grab, Twist and Pull" at the same time using "Lever Mechanism" and "Screw Kinematic Pair". After searching through the standard solutions the designer moved on the find solution using "Generative Methods" toolset. Here "Gravitational Effect" PE inspired him for the problem "Collect Oranges" by using a cushion to collect oranges which fall from the tree without damaging them.

After this, the designer decided to synthesize complete solutions using CEMC. This activity took him 5-6 minutes. Lastly, the designer used the process monitoring tool for 3-4 minutes. The complete solutions generated by the designer are shown in Figure 99. Figure 100 shows the design sketches generated by the designer.



complete solution 8 (delete)

Name	Solutions
Move near Orange Tree (X) Add Solution from	Human Powered (X) **Description:** Movement can be done manually from one tree to another **Source of Inspiration:** null
Separate Oranges from Tree (X) Add Solution from	The lever mechanism (X) **Description:** Lever mechanism with grabbing pads can be used to grab the oranges **Source of Inspiration:** Mechanism:The Lever Using Screw for pulling and twisting (X) **Description:** A screw mechanism for pulling and twisting the orange simultaneously **Source of Inspiration:** Mechanism:Simple Kinematic Screw Chain
Collect Oranges (X) Add Solution from	Collection in nets (X) **Description:** Nets can be used to collect oranges before they fall on grounds **Source of Inspiration:** TRIZ:7. Nesting

Figure 99: Sample Complete Solution - Case Study #1

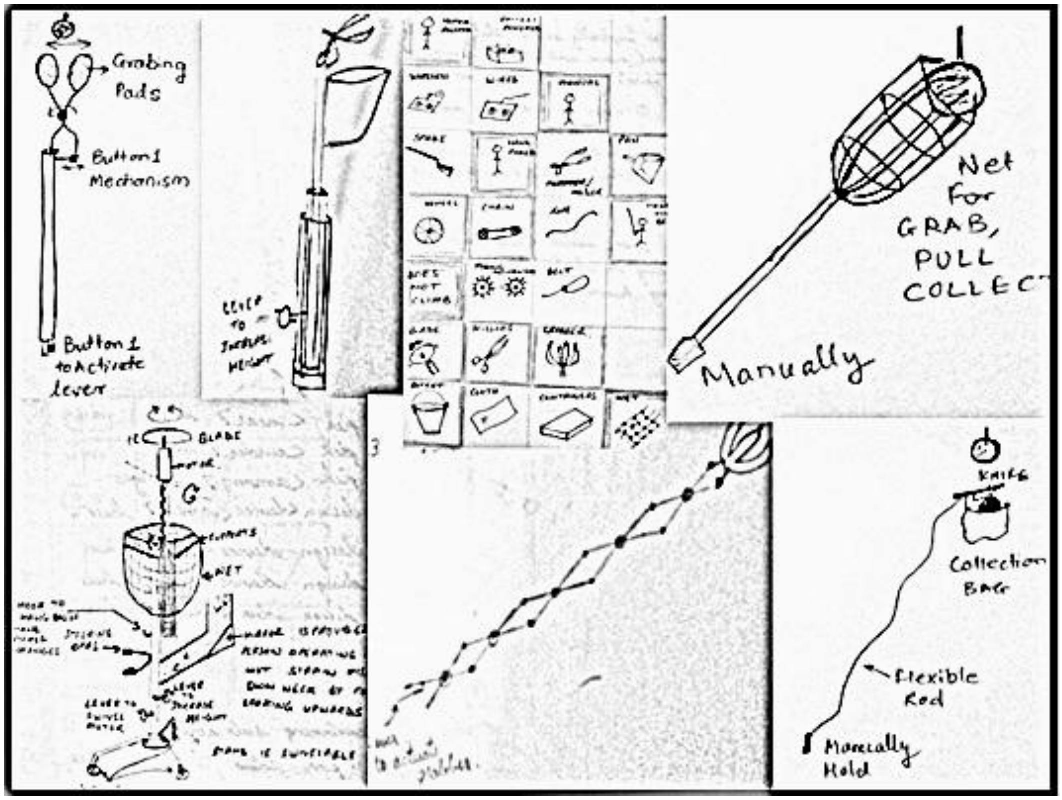


Figure 100: Case Study #1 - Sketches

7.3.2 User Study #2: Participant B - Problem 1

Participant B also decided to start with function decomposition for which he used the Function CAD module from the tool. The function decomposition is shown in Figure 101. Designer B took 9 minutes to include a two level deep function decomposition. He did not include any disjunctions. At this stage, the designer decided to move on to next step, where he added this function structure in the morph chart. The CEMC structure created by the designer B is shown in Figure 102 and this process took 5 minutes.

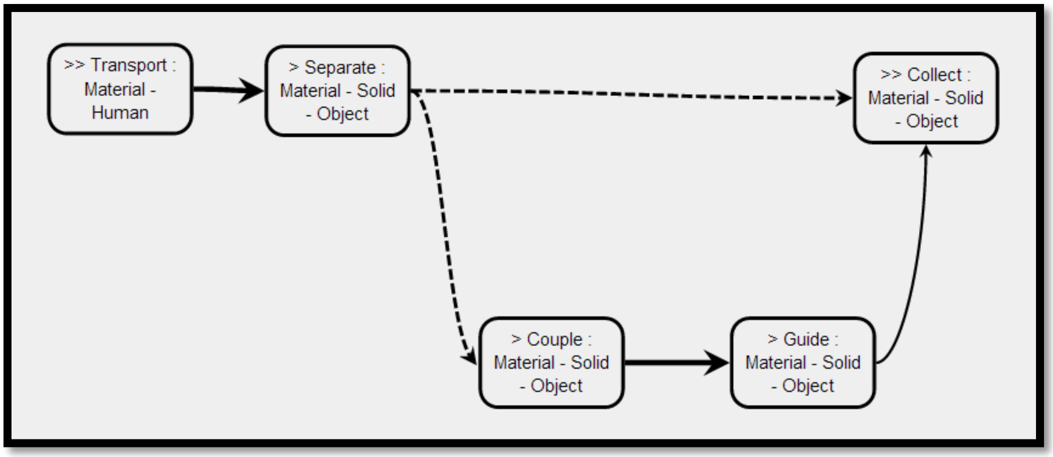
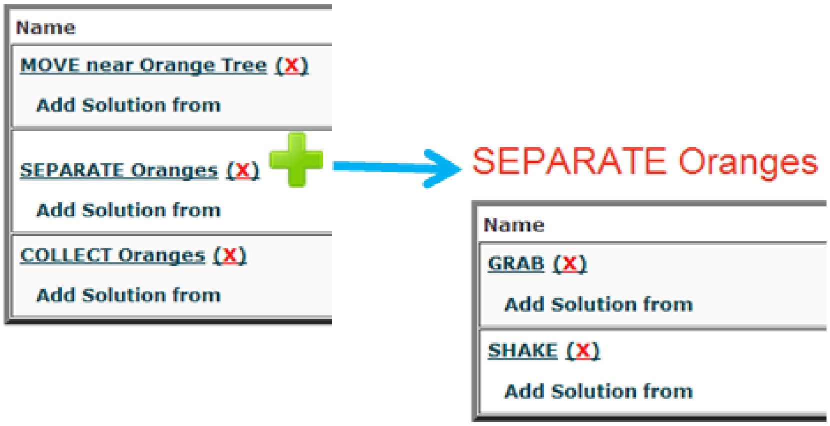

>> Transport : Material - Human
> Separate : Material - Solid - Object
>> Collect : Material - Solid - Object
> Couple : Material - Solid - Object
> Guide : Material - Solid - Object

Figure 101: Function Decomposition in Case Study #2



Name

MOVE near Orange Tree (X)

Add Solution from

SEPARATE Oranges (X)

Add Solution from

COLLECT Oranges (X)

Add Solution from

SEPARATE Oranges

Name

GRAB (X)

Add Solution from

SHAKE (X)

Add Solution from

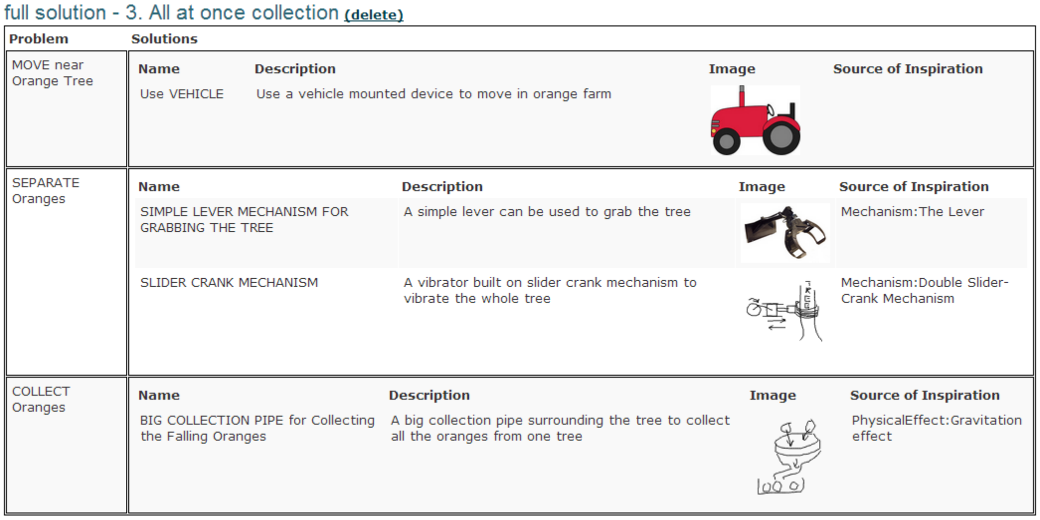
Figure 102: CEMC Created in Case Study #2

The CEMC structure created by the designer had only 4 leaf problems, which he tackled first. While generating solutions, the designer started to work on problems without using the tool for 10 minutes.

After this, the designer B decided to use the tool sets for generating new ideas, which he did for 24 minutes. To do so, the designer mainly focused on the standard solution toolset. After looking at various mechanisms, he decided to use

"Lever mechanism" for gripping the tree. Using "Slider crank mechanism" a solution for vibrating the tree was generated. From generative methods toolset, the PE of "Gravitation effect" and TRIZ principle of "Nesting" inspired the designer to come up with the solutions "A large collection pipe" and "using nets" for the problem of collecting the oranges respectively.

After this, the designer decided to synthesize complete solutions using the CEMC, which took him 7 minutes. Lastly, the designer used the process monitoring tool for 3 minutes. Figure 103 and 104 shows the complete solutions and design sketches generated by the designer.



full solution - 3. All at once collection (delete)

Problem	Name	Description	Image	Source of Inspiration
MOVE near Orange Tree	Use VEHICLE	Use a vehicle mounted device to move in orange farm		
SEPARATE Oranges	SIMPLE LEVER MECHANISM FOR GRABBING THE TREE	A simple lever can be used to grab the tree		Mechanism:The Lever
	SLIDER CRANK MECHANISM	A vibrator built on slider crank mechanism to vibrate the whole tree		Mechanism:Double Slider-Crank Mechanism
COLLECT Oranges	BIG COLLECTION PIPE for Collecting the Falling Oranges	A big collection pipe surrounding the tree to collect all the oranges from one tree		PhysicalEffect:Gravitation effect

Figure 103: Sample Complete Solution - Case Study #2

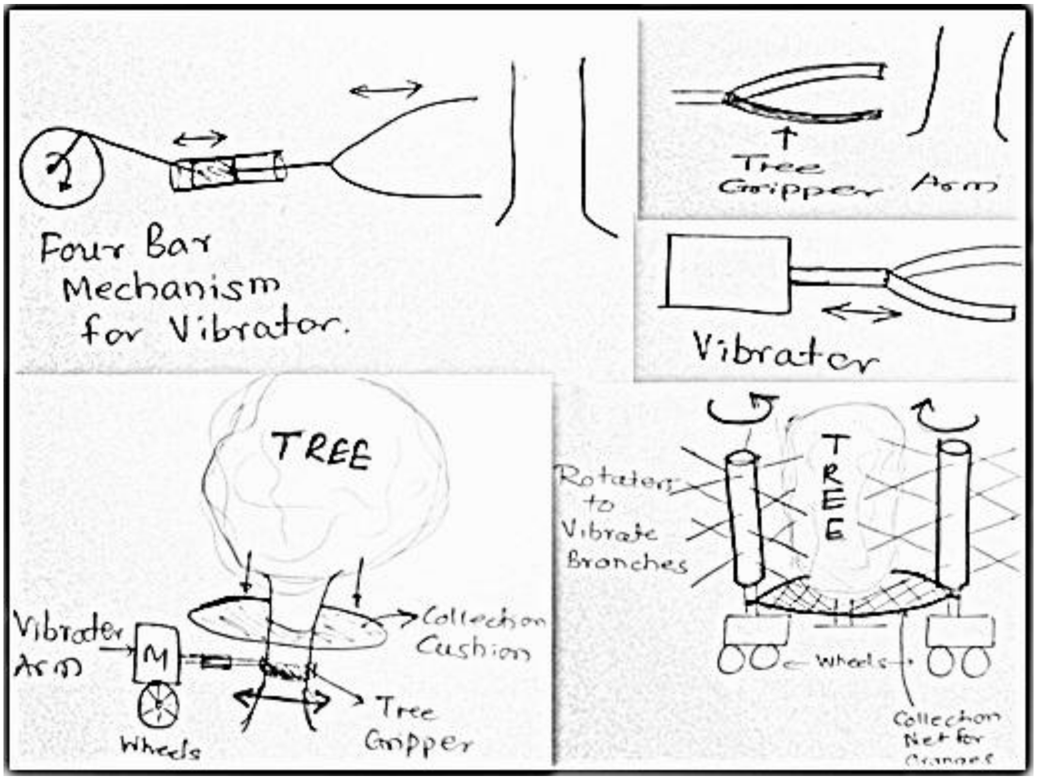


Figure 104: Case Study #2 - Sketches

7.3.3 User Study #3: Participant C - Problem 2

Participant C decided to choose the starting point as function decomposition for which he did not use the Function CAD module from the tool. The function decomposition is shown in Figure 105. Designer C included two level deep function decomposition and had no disjunctions. This activity took him 8 minutes. At this stage, the designer C decided to move on to next step, where he added this function structure in the morph chart. After adding all the functions to the CEMC, he also added requirement "Pattern 8" to the CEMC (Figure 106). This process took 10 minutes.

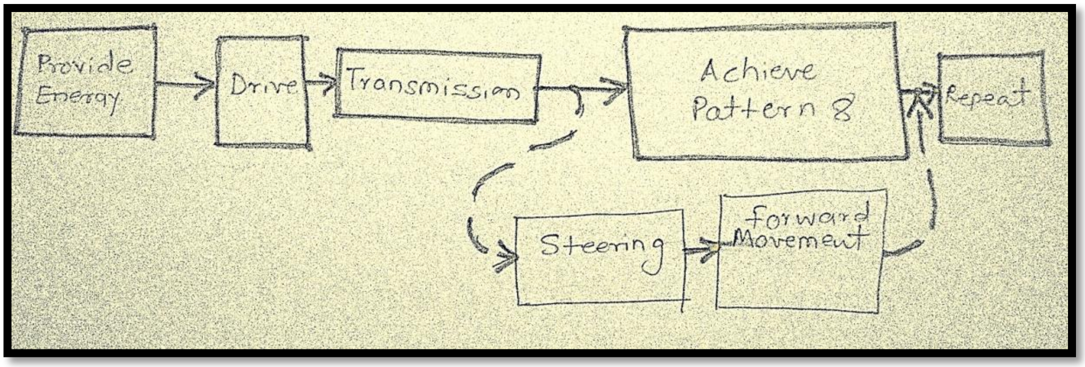


Figure 105: Function Decomposition in Case Study #3

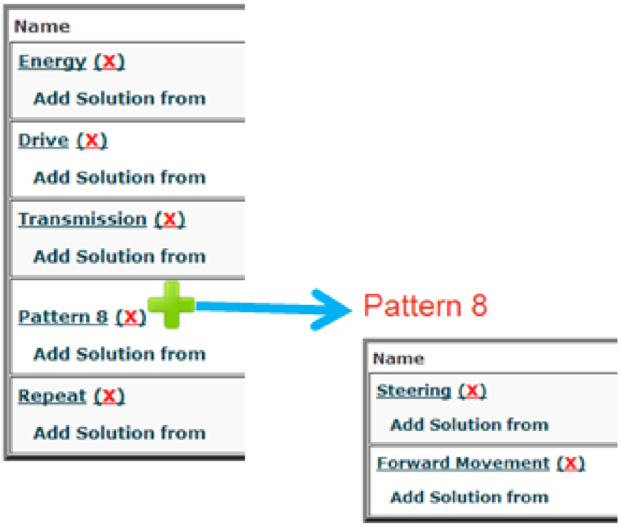


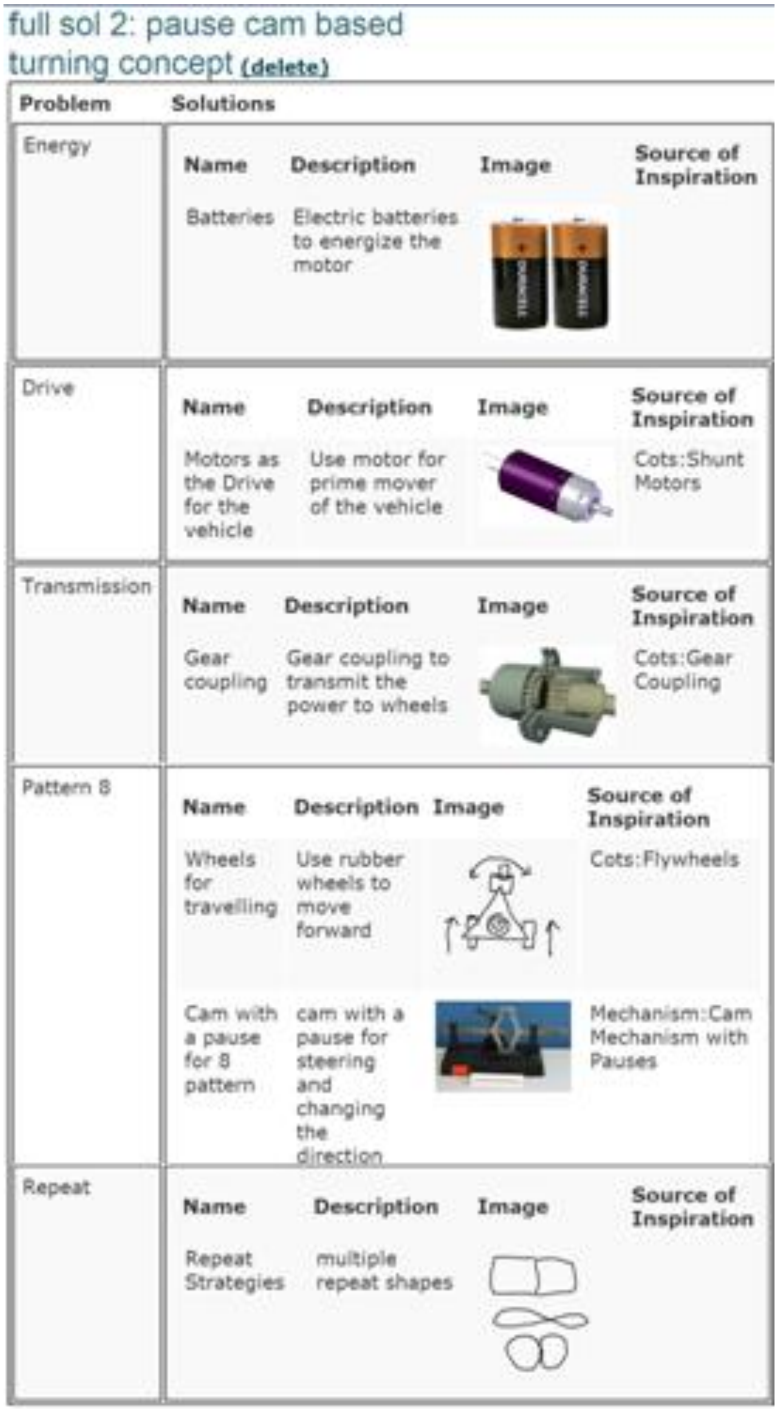
Figure 106: CEMC Created in Case Study #3

The CEMC structure created by designer C had 5 leaf problems. At start for generating solutions he started to work on the problems without using the tool for 10 minutes.

After that he decided to use the tool sets for generating new ideas. This activity took him 22 minutes.

In this process, the designer allotted most of his time to find a suitable standard solution for "steering" the mechanism. He decided to use the "Cam mechanism with pause" to steer the vehicle in patter 8. For all the problems he was able to find a solution using standard solutions toolset. The designer used most of his time looking through various mechanisms and COTS because he thought these methods provided easy way of searching the databases. It is noteworthy, that Generative Methods and External Resources were used for very less time compared to time invested in searching Standard Solution toolset.

Finally, the designer synthesized complete solutions using CEMC for 8 minutes and used the process monitoring tool for 2 minutes. Figure 107 and 108 shows complete solutions and design sketches generated by the designer.



full sol 2: pause cam based turning concept **(delete)**

Problem	Name	Description	Image	Source of Inspiration
Energy	Batteries	Electric batteries to energize the motor		
Drive	Motors as the Drive for the vehicle	Use motor for prime mover of the vehicle		Cots:Shunt Motors
Transmission	Gear coupling	Gear coupling to transmit the power to wheels		Cots:Gear Coupling
Pattern 8	Wheels for travelling	Use rubber wheels to move forward		Cots:Flywheels
	Cam with a pause for 8 pattern	cam with a pause for steering and changing the direction		Mechanism:Cam Mechanism with Pauses
Repeat	Repeat Strategies	multiple repeat shapes		

Figure 107: Sample Complete Solution - Case Study #3

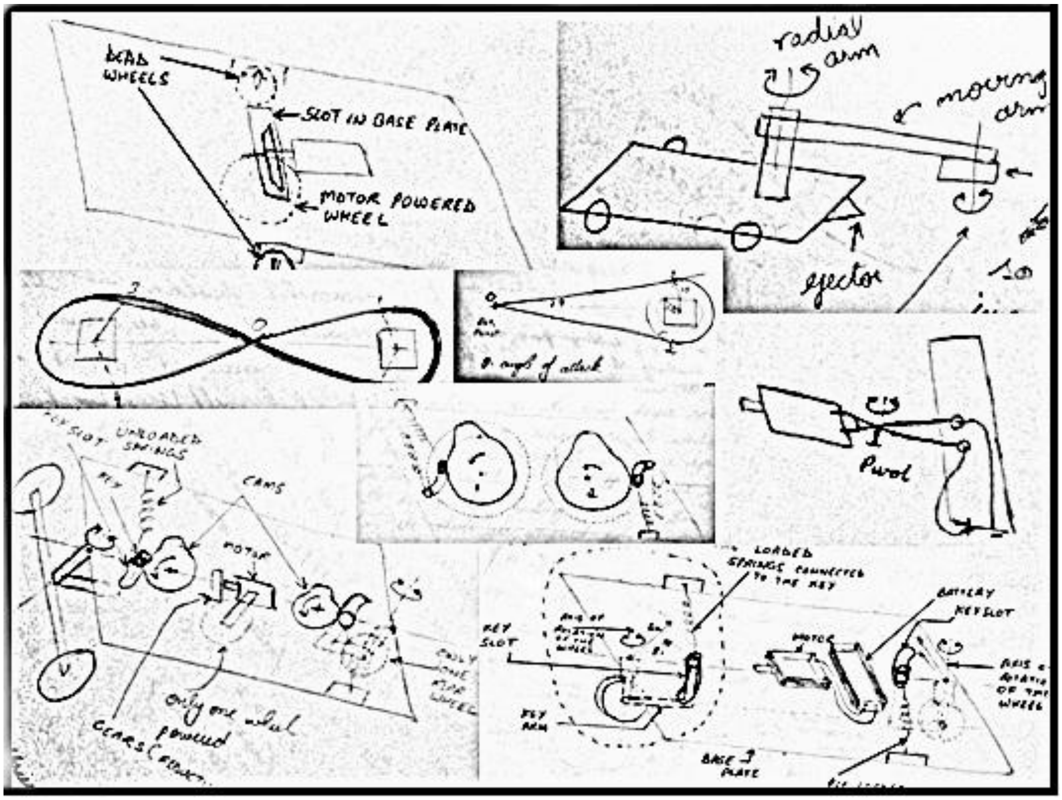
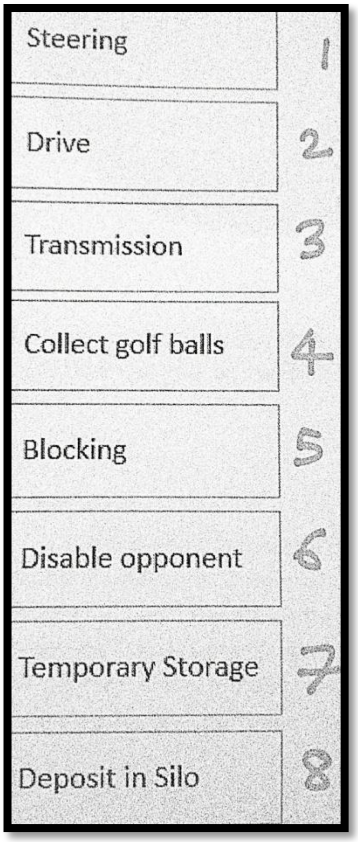


Figure 108: Case Study #3 - Sketches

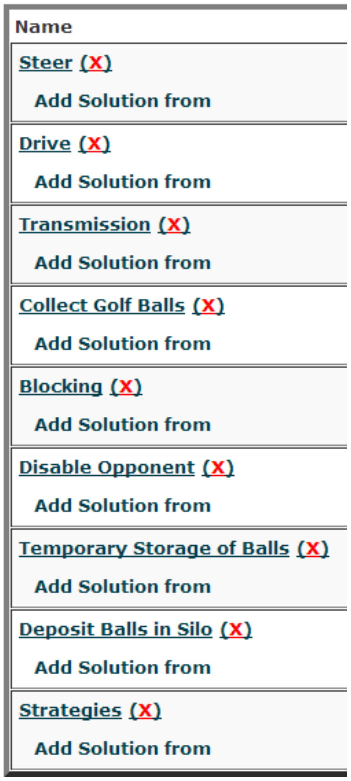
7.3.4 User Study #4: Participant C - Problem 3

Unlike all other problems, participant C for problem 3 did not choose the starting point as function decomposition. The reason he recorded was that the problem 3 involved strategy which required use of different functions at different times. Because of this the designer created a flat list of required functions as shown in Figure 109. This activity took him 11 minutes. At this stage, the designer C decided to move on to next step, where he added this list in the morph chart. The CEMC structure created by the designer C is shown in Figure 110. This process took 8 minutes.



Steering	1
Drive	2
Transmission	3
Collect golf balls	4
Blocking	5
Disable opponent	6
Temporary Storage	7
Deposit in Silo	8

Figure 109: Flat List of Functional Requirements - Case Study #4



Name
Steer (X) Add Solution from
Drive (X) Add Solution from
Transmission (X) Add Solution from
Collect Golf Balls (X) Add Solution from
Blocking (X) Add Solution from
Disable Opponent (X) Add Solution from
Temporary Storage of Balls (X) Add Solution from
Deposit Balls in Silo (X) Add Solution from
Strategies (X) Add Solution from

Figure 110: CEMC Created in Case Study #4

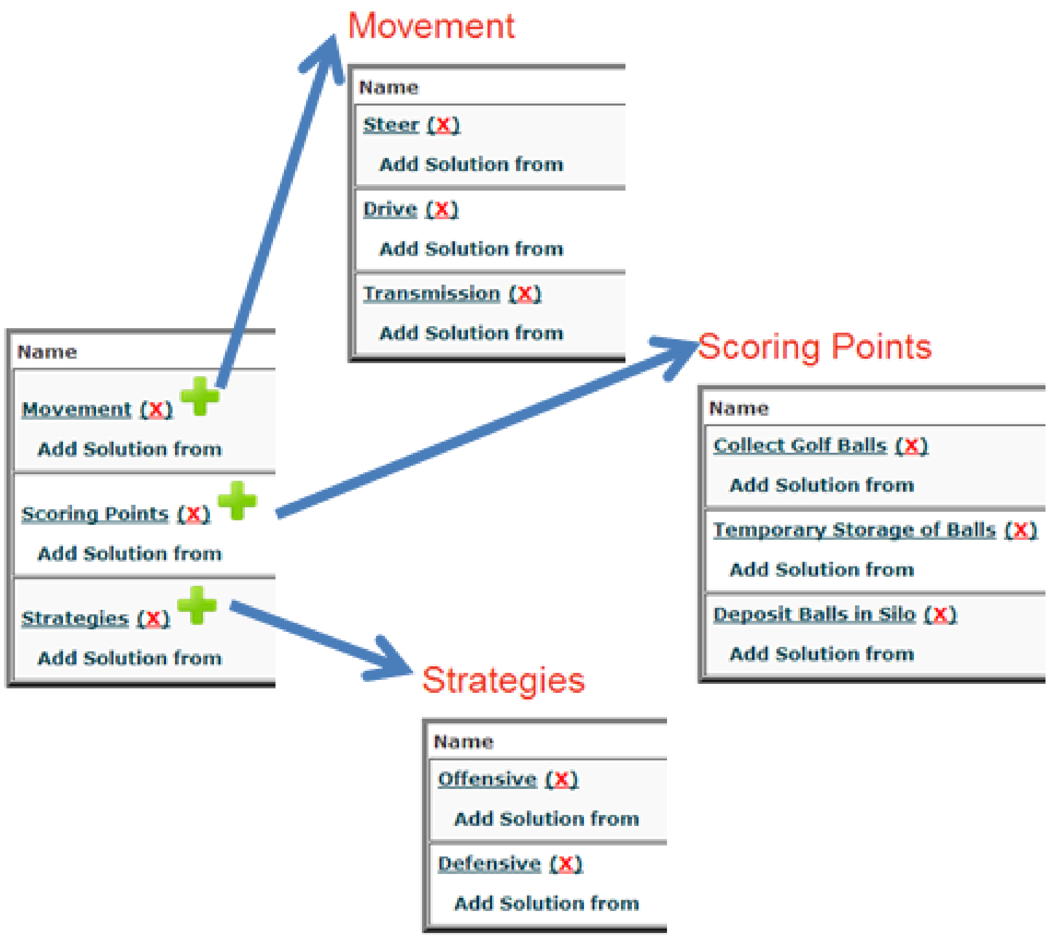
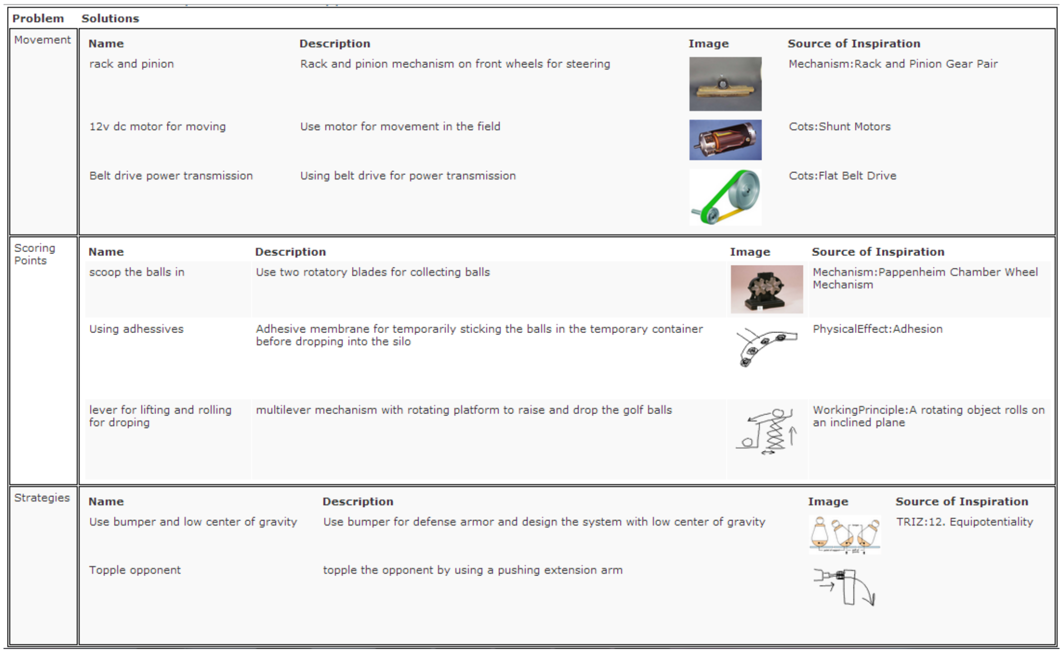


Figure 111: Reframed CEMC Structure - Case Study #4

At first, the CEMC structure created by the designer had only one level which included 8 leaf problems. However, after working on for some time, he was able to reframe the flat CEMC structure into a two level deep CEMC as shown in Figure 111. The designer recorded that easily restructuring problem hierarchy gave him a chance to play with the problem structure. It also provided him with intuitive understanding of the whole design problem. At the time of generating solutions for these problems the designer worked on the problems without using the tool for 8 minutes.

After that he decided to use the tool sets for generating new ideas, which took 23 minutes. In this step, the major time was given for the high level problem of "Strategies". The designer started by searching solutions for the sub problems - "offensive" and "defensive". To these problems TRIZ principle of "Equipotentiality" inspired him to come up with solutions - "Toppling the opponent" and "Low center of gravity vehicle" respectively. The designer generated solutions using many mechanisms such as "rack and pinion", "belt drive", "lever mechanism" for different problems.

After this, the designer synthesized complete solutions for 5 minutes. And lastly, the designer used the process monitoring tool for 5 minutes. Figure 112 and 113 shows complete solutions and design sketches generated by the designer.



Problem	Solutions			
Movement	Name	Description	Image	Source of Inspiration
	rack and pinion	Rack and pinion mechanism on front wheels for steering		Mechanism:Rack and Pinion Gear Pair
	12v dc motor for moving	Use motor for movement in the field		Cots:Shunt Motors
	Belt drive power transmission	Using belt drive for power transmission		Cots:Flat Belt Drive
Scoring Points	Name	Description	Image	Source of Inspiration
	scoop the balls in	Use two rotatory blades for collecting balls		Mechanism:Pappenheim Chamber Wheel Mechanism
	Using adhessives	Adhesive membrane for temporarily sticking the balls in the temporary container before dropping into the silo		PhysicalEffect:Adhesion
	lever for lifting and rolling for droping	multilever mechanism with rotating platform to raise and drop the golf balls		WorkingPrinciple:A rotating object rolls on an inclined plane
Strategies	Name	Description	Image	Source of Inspiration
	Use bumper and low center of gravity	Use bumper for defense armor and design the system with low center of gravity		TRIZ:12. Equipotentiality
	Topple opponent	topple the opponent by using a pushing extension arm		

Figure 112: Sample Complete Solution - Case Study #4

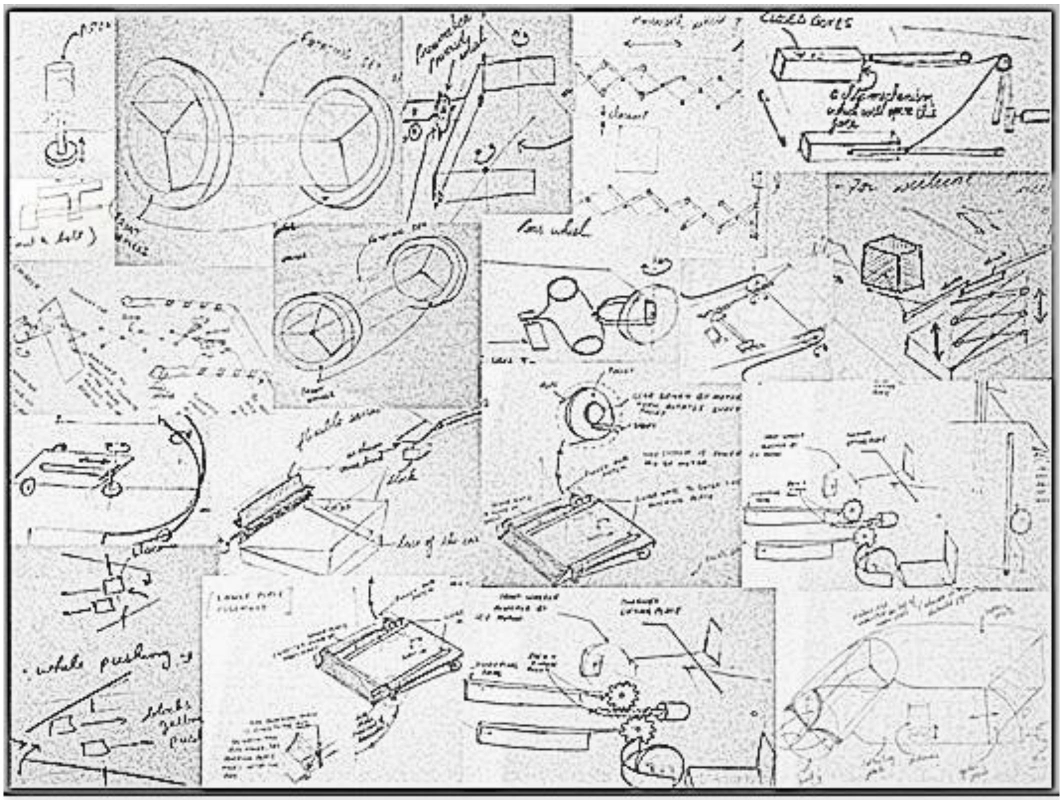


Figure 113: Case Study #4 - Sketches

7.4 Data Collection

After the case studies, participants A, B, and C were asked to fill in the survey sheets. Table 27, 28, and 29 are the result of the survey.

Table 27: Survey Sheet by Participant A

Participant A : Problem 1 Evaluation Work Sheet: Holistic Ideation Tool V.3 (Use Scale of : 1 - 10)		
Please rate Usability/User-Friendliness for each toolset provided in the tool.		
Toolset	Rating	Why
1. Problem Formulation Methods	7	Function CAD module is not very easy to use for doing complex function

		decomposition
2. Re-formulation	8	It would be good to see automatic restructuring of the CEMC by the tool
3. Standard Solutions	9	Standards solutions were useful, but more COTS needs to be added into the database
4. Generative methods	8	PE and WP description should have more images and video for quick understanding
5. External resources	5	External resources are not integrated so traversing from tool KB to external resource KB was not possible
6. Solution Synthesis and	9	Tool could not provide all the possible complete solutions automatically
7. process monitoring	3	These tools were not used much.
Please Provide any suggestions or feedback regarding the tool		
• Because of the difficulty in use I could not use Function CAD tool and chose to do function decomposition by hands. It needs improvements. • Looking at the solution space and problem space through CEMC was helpful towards understanding of the overall problem. I could see high level morph chart as well as low level problems simultaneously which was very helpful. • I could not couple multiple functions and store solutions for them in one place because each problem has its own solution list, but sometimes designer might want combine solutions by coupling different problems which don't have the same parent problem.		

Table 28: Survey Sheet by Participant B

Participant B : Problem 1 Evaluation Work Sheet: Holistic Ideation Tool V.3 (Use Scale of : 1 - 10)
Please rate Usability/User-Friendliness for each toolset provided in the tool.

Toolset	Rating	Why
1. Problem Formulation Methods	6	• Though I used Function CAD, it took much more time than it should require because of the difficulty in using UI. • Navigating through CEMC was very simple and intuitive, I have to refresh page each time I made a change into CEMC.
2. Re-formulation	8	Relational Algorithm some time provides pictures which I could not related with the word I selected.
3. Standard Solutions	8	Select Mechanism and Select COTS was easier to use, but I would like to see both the output lists of Mechanisms and COTS with same inputs.
4. Generative methods,	6	The list of related PE and WP was static and I had to manually go and find the respective related entity.
5. External resources,	8	External resources were useful but more external resources should be added. However, Google and Wikipedia were not useful to me for finding any solution.
6. Solution Synthesis and	10	NO FEEDBACK
7. process monitoring.	4	Find Ideation State results were not useful to me.
Please Provide any suggestions or feedback regarding the tool		
• In general CEMC was the tool I spent the most time working on. Whenever I have to add a problem I had to go to project homepage and select parent problem each time. It would be easy to work if those options are provided on every single CEMC so that user does not have to specify parent problem each time.		

• Also, to use any ideation method I had to go to the project homepage, it would save time for the designer if the ideation methods are provided with each of the low level CEMC.

Table 29: Survey Sheet by Participant C

Participant C : Problem 1 and 2 Evaluation Work Sheet: Holistic Ideation Tool V.3 (Use Scale of : 1 - 10)		
Please rate Usability/User-Friendliness for each toolset provided in the tool.		
Toolset	Rating	Why
1. Problem Formulation Methods	4	In this toolset I found CEMC very easy to use and at the same time Function CAD was very difficult to use.
2. Re-formulation	7	Similarly as the image net tool, it would be nice to see more number of images.
3. Standard Solutions	6	• I had to go to KMODDL website to look at the video simulations. It would be nice to have them embedded in the holistic Ideation tool itself which could save the time of the designer. • Artifacts module took too much time to load and hence could not be used much.
4. Generative methods	6	More PE and WP needs to be added to the database. Even though, I looked into all the entries but could not generate many solutions.
5. External resources	5	Could not use external resources much as these resources were not linked with functions I am using.
6. Solution Synthesis and	9	Random complete solutions generated by the tool does not let user specify which solutions are not compatible with each other and hence sometimes generate non useful combinations.

7. process monitoring	8	Some of the Ideation methods suggested by the tool are not implemented yet, while some are group based.
Please Provide any suggestions or feedback regarding the tool		
• Sometimes the tools gets very slow, especially while uploading the artifacts list. • I could not save a back up morph chart and went on modifying the CEMC in multiple different ways. • Sometimes I felt lost in the system and it would be easy to work with if a guiding tool can tell what to do next. • I could not attach a video simulation which I thought was related to the solution. • I would like to see the CEMC in the tree format.		

Apart from collecting the survey responses, the post experiment data collected from the tool also contains the following: 1. Total number of solutions generated, 2. Number of solutions generated manually and 3. Number of solutions generated using different toolsets (Table 30).

Table 30: Post Experimental Data Collection

Case study No.	Total number of solutions generated	Number of solutions generated manually	Number of solutions generated using the toolset -		
			Standard Solution	Generative Methods	External Resources
1	31	18	8	4	1
2	15	8	4	2	1
3	16	8	6	2	0
4	19	7	7	3	2
Total	81	41	25	11	4

Furthermore, while the designers were working on the tool, we also captured the time when different toolsets were used and when the solutions were generated by the designers. This data was collected without disturbing the participant's work flow. In the next section, we will represent analysis of the data collected during the experiments.

7.5 Analysis Of Data

We collected the following data in the experiments using V.3: 1. Which toolsets designers utilize during the ideation process, 2. which tool sets inspired designers to come up with solutions, and 3. At what time the solutions were generated. For better understanding of collected data, we plotted it for each case study in Figures 114, 115, 116 and 117. In these graphs, the X-axis corresponds to the time and Y-axis shows which toolset was being used at that time. Graph also shows which toolset was used to come up with a solution. Table 31 represents the values of the numbers on Y-axis and related toolset.

Table 31: Toolset Represented by Values

Value on Y-Axis	Represented Toolset
7	Process Monitoring
6	Solution Synthesis
5	External Resources
4	Generative Methods
3	Standard Solution
2	Reformulation
1	Problem Formulation
0	Without using the Tool

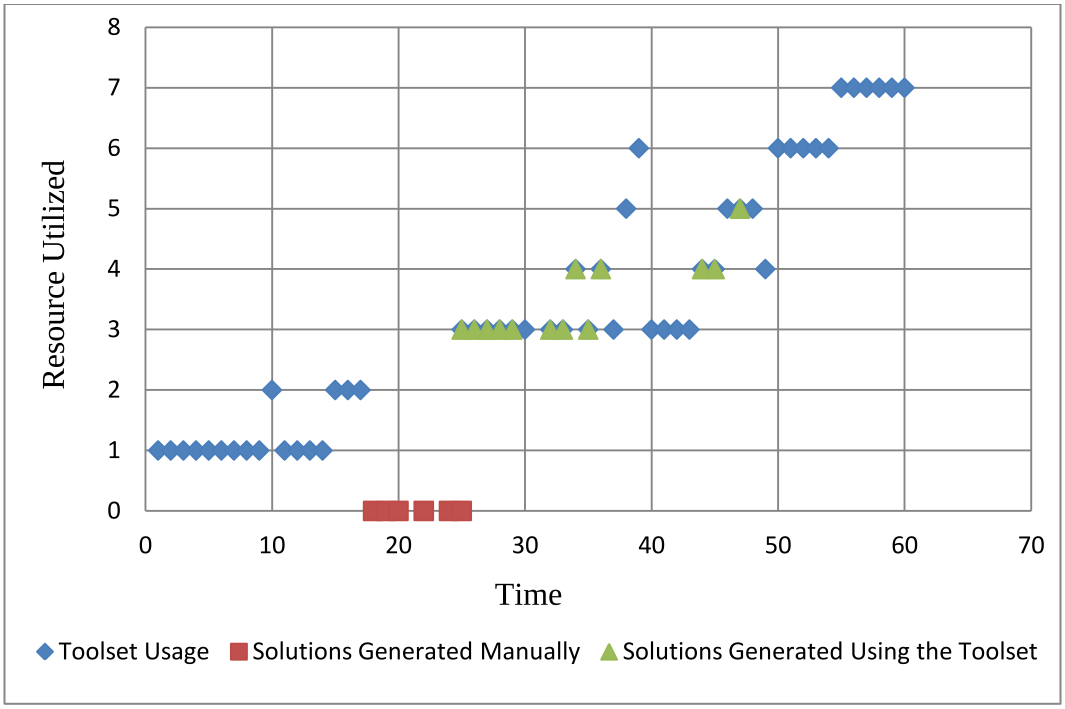


Figure 114: Case Study 1 : Toolset Usage Analysis

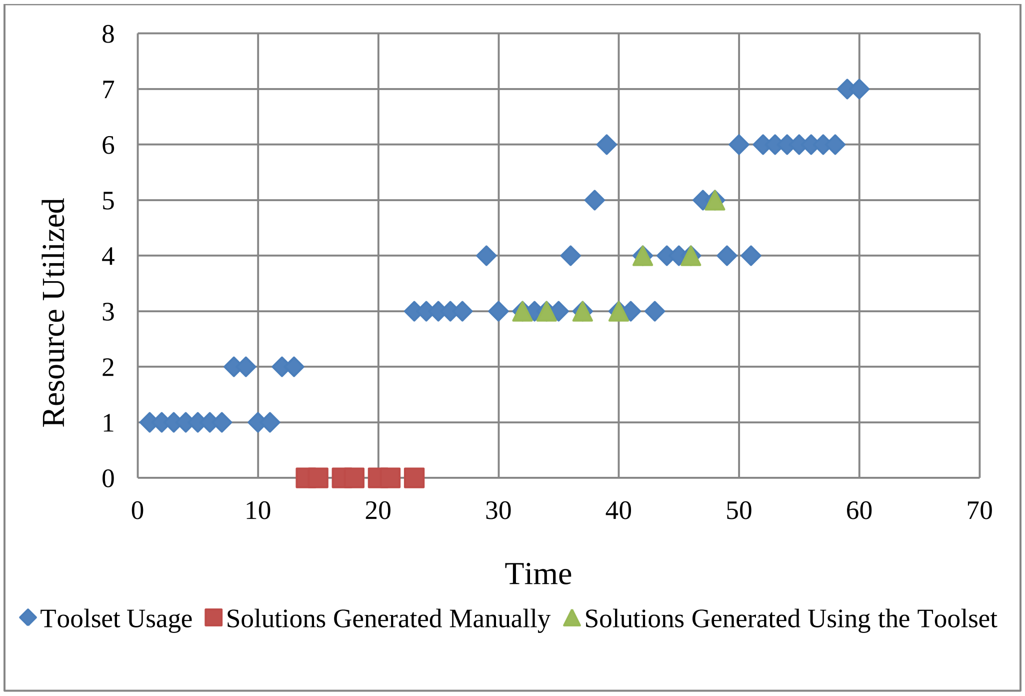


Figure 115: Case Study 2 : Toolset Usage Analysis

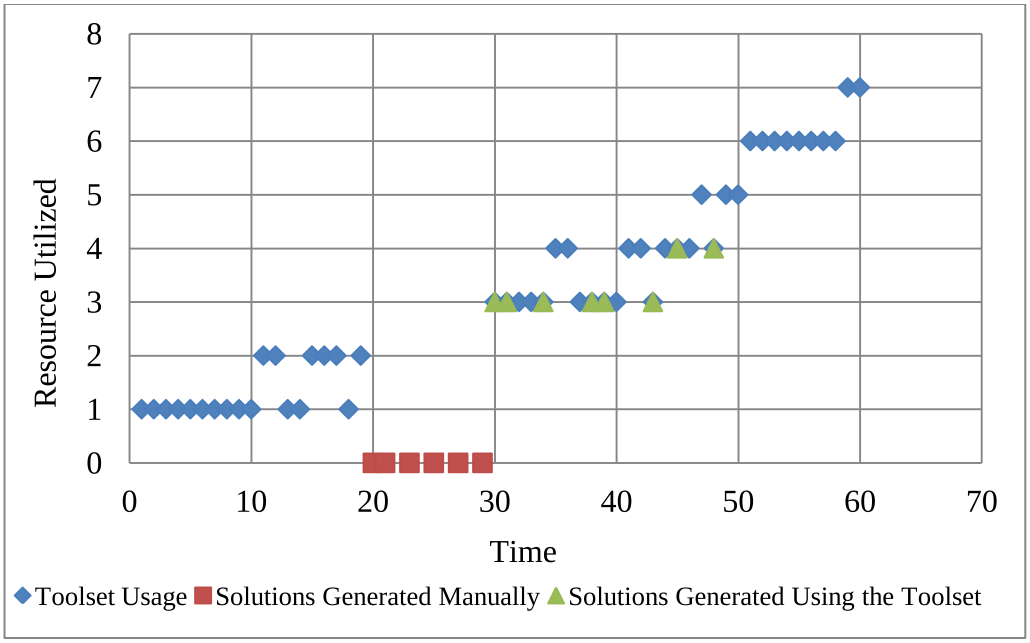


Figure 116: Case Study 3 : Toolset Usage Analysis

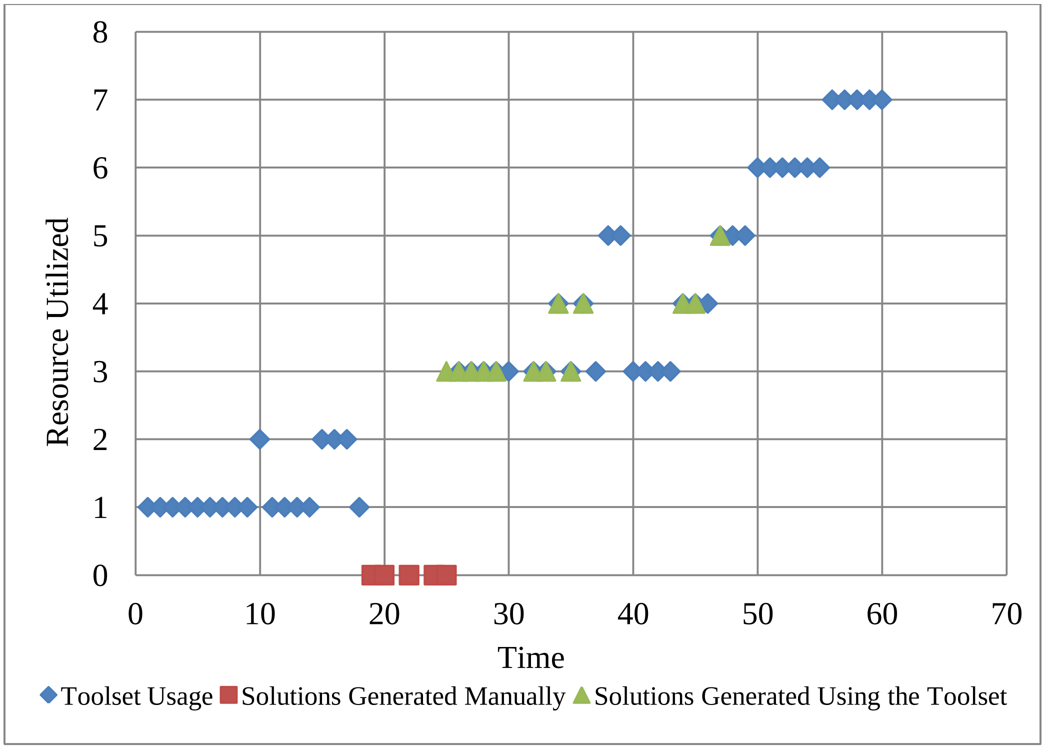


Figure 117: Case Study 4 : Toolset Usage Analysis

7.6 Conclusions From User Studies

After analyzing the data, a common structure in the usage of toolset was observed. Each designer followed common sequence of steps listed below:

Step 1: All the users started with using Problem formulation and reformulation tools.

Step 2: Then, all the users started working on their own to find solutions for the problems. Hence, all the designers did not use the tool in this step. The designers did this until they felt that they are exhausted and they have encountered an Ideation Block. During this time the designers recorded many solutions they could think of -off the top of their head- very quickly.

Step 3: After step 2, all the designers started to work with the toolset of Standard Solutions. Here they tried to find existing solutions that they could not think of in step 2. In this step they generated on an average half the solutions they generated in step 2. They spent more time on this step than the time spent on step 2. After searching the Standard Solutions toolset for all the functions, designers moved on to step 4.

Step 4: In this step every designer tried to generate solutions by using the PE, WP and TRIZ modules. In this stage they came up with very few but novel solutions, however this step took around same time as the step 3.

Step 5: Finally, when the designers felt exhausted, they retorted to the external solutions for very little time and managed to come up with couple of solutions.

Step 6: After feeling satisfied with the solutions generated for each of the problem, designers synthesized full solutions for the design problem.

Step 7: The designers had their preferences for the ideation methods. Because of that inclination, they gave very little time towards Finding Ideation State module. Also, because some designers felt the time pressure, they did not choose to use the last toolset.

The common structure and steps, which were derived from the case studies is represented graphically in Figure 118.

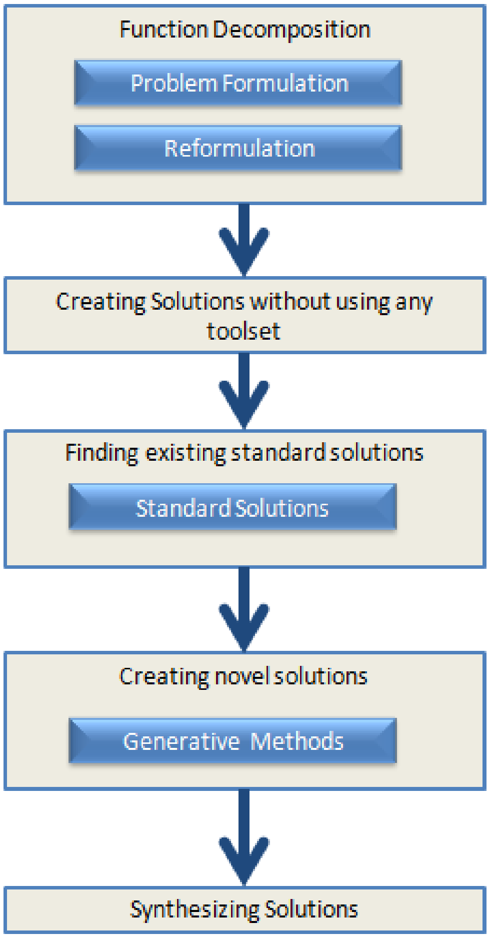


Figure 118: Common Structure Found from the Case Studies

After looking at the survey taken from the designers and the case studies results, we suggest following improvements in the tool:

1. The designers had encountered major issues in using the Function CAD tool. For better usability, improvements are needed in the UI implementation of the Function CAD module.

2. The designers suggested that they would like to look at greater number of Mechanisms and COTS. So, the Mechanisms and COTS databases need to be increased in volume.

3. One of the reason behind the designers finding the CEMC user friendly is because of the automatic solution population feature of the tool where user does not have to specify parent problem. On the same lines, designers suggested that every time they had to add a new sub-problem, they have to navigate to project homepage, click on "Add Problem" button and also select a Parent Problem from the list. Instead of this, each CEMC should contain its own "Add Problem" button so that a user can add a sub-problem to that CEMC without specifying the parent problem.

4. As the toolsets are organized in the vertical manner, the designers felt that they need to access all the ideation methods one by one. To alleviate this problem, we suggest the vertical arrangement of the toolsets should be changed.

5. The designers struggled to create disjunctions with the CEMC, which reduced the usability of the tool. For that reason a simple and intuitive way should be designed and implemented in the tool to create disjunctions easily.

6. Some of the designers suggested that they would also like to see the CEMC in tree structure. We suggest a tree format of the CEMC should be designed and implemented to increase the usability of the tool.

7. One designer felt that he lost the track of the process which played a significant factor in reducing the perceived usability of the tool. To negate this problem we suggest that designers should be guided while using the tool.

8. Because of missing features such as save, load, copy, cut, paste, undo, and redo, the designers felt constrained while using the CEMC. Addition of these features in the tool will significantly contribute towards the usability of the tool.

9. Some designers clearly tended not to use the "external resources" toolset, because these resources are not integrated with the KB of the tool. Therefore, the external resources should be coupled with the current KB of the design.

CHAPTER 8

CONCLUSION AND FUTURE WORK

8.1 Conclusion

The main objective of this thesis was to evaluate and improve the holistic approach for the conceptual design stage represented in the V.1 and V.2, and to implement a new version (V.3). After reviewing previous versions, additional ideation methods were investigated and added. These additional ideation methods included the mechanisms repository and the machine elements repository. The CEMC organizational framework was improved. To increase the usability of the tool a new UI was designed and implemented. After the user studies some valuable insights into the user tendencies about the new organizational framework and some usability issues were found. The strong point of the new tool is that, though it provides a user with a large amount of information, at the same time it also presents the user with an organizational framework to manage this information in an organized way. The available modules implemented in new holistic ideation V.3 are following:

1. Function CAD
2. CEMC
3. Word-Diamond
4. Word-Net
5. Relational Algorithm
6. Find Mechanisms
7. Select Mechanisms

8. Find Machine Elements

9. Select Machine Elements

10. Artifacts

11. Physical Effect

12. Working Principle

13. Bio-TRIZ

14. TRIZ

15. ImageNet

16. AskNature

17. Wikipedia

18. Google

19. Find Ideation State

20. Feedback Module

21. Random Connection

A framework was designed and implemented to provide the vast amount of the functionalities to the designer without reducing the usability of the tool. Another important contribution of this research is providing a structure for the conceptual design stage without imparting any constraints on the designer. At any point user could use any combination of ideation methods without losing his track. Moreover, the starting point is not limited to functions any more. Features, requirements, material and even components could be the starting point of CEMC. Also, with the use of this framework, there is a vast scope for the future work.

8.2 Future Work

Though V.3 was improved in many aspects from previous versions it still needs some modifications. The very important modifications are the changes needed to implement in order to improve the usability of the tool. V.3 should be improved to let user load, save, copy and paste any morph chart easily. One more important improvement is in the direction of storing time related data during user sessions and using it for data mining later. The tool with little modifications can be improved to add team work capability. Also the tool output can be improved from a pdf file to full report of the user activities. The tool can also easily incorporate a guidance module to guide the designers during the ideation. It will always be good to keep adding entities to the existing databases making them richer.

8.2.1 Data Collection Using Web-Based Holistic Ideation

A radical paradigm shift is needed to collect massive amounts of data from diverse set of users over an extended period of time to truly get into deeper issues and solution strategies employed by the real designers instead of the timed experiments on exercises given to the undergraduate students which were used in this thesis (Mohan M. , 2011). Using this web enabled Holistic Ideation Testbed we can automatically collect vast amount of user data in a structured way which is suitable for data mining later.

8.2.2 Experiments for Ideation Paths

Our main motif for this free web service is to capture ideation mechanisms and paths followed by different users. The ideation paths could be understood as the order and combination of ideation methods or ideation strategies (Mohan M. , 2011). Studying ideation paths in conceptual design has not been previously explored and researched in a formal manner. As the web-based Holistic Ideation framework is implemented, it will become possible to get a large number of users to test it. Several experiments could then be designed to find and summarize the ideation paths.

BIBLIOGRAPHY

(n.d.). Retrieved 2013-14, from A Cornell University Web site: http://kmoddl.library.cornell.edu/

(n.d.). Retrieved 2014, from WordNet, Princeton University: http://wordnet.princeton.edu/

Akin, O. (1994). Creativity in design. *Performance Improvement Quarterly , 7* (3), 9-21.

Altshuller, G. (2001). 40 principles: TRIZ keys to technical innovation. *Tech Innov Center .*

Artobolevskiĭ, I. I. (1979-82). *Mechanisms in Modern Engineering Design Vol1-4.* Mir Publishers.

AskNature. (n.d.). Retrieved from http://www.asknature.org/

Beitz, W., & Pahl, G. (1996). *Engineering Design-A Systematic Approach.* London: Speinger.

Boden, M. A. (1996). *Dimensions of Creativity.* A Bradford Book.

Bogatyreva, O. &. (March,2009). TRIZ evolution trends in biological and technological design strategies. . *19th CIRP design conference* , 293.

Bogatyreva, O. S. (2004). Patterns in TRIZ Contradiction Matrix: integrated and distributed systems. *4th ETRIA Conference* , 305-313.

Bogatyreva, O., & Bogatyreva, N. (2009, March). TRIZ evolution trends in biological and technological design strategies. *19th CIRP design conference* , 293.

Bogatyreva, O., Shillerov, A., & Bagatyreva, N. (2004). Patterns in TRIZ Contradiction Matrix: integrated and distributed systems. *4th ETRIA Conference* , 305-313.

Bohm, M., Stone, R., Simpson, T., & Steva, E. (July 2008). Introduction of a data schema to support design repository. *Computer Aided Design, Vol 40, Issue 7* .

CakePhp. (n.d.). Retrieved 2014, from CakePHP Cookbook 2.x - http://book.cakephp.org/2.0/en/contents.html

Chen, Y. (2012). Cascading Evolutionary Morphological Charts for Holistic Ideation Framework.

Chikofsky, E. J., & James, H. (1990). Reverse Engineering and Design Recovery: A Taxonomy. *IEEE Software , 7* (1), 13-17.

DeBono, E. (1984). Tactics: The Art and Science of Success.

Dorst, K., & Cross, N. (2001). Creativity in the design process: co-evolution of problem solution. *Design Studies* , Vol.22 No.5, pages 425-437.

Duncker, K. (1945). On Problem-Solving. *Psychological Monographs , 58* (5).

Faltings, B. (1992). A symbolic approaach to qualitative kinamatics. *Artificial Intellegence No. 56* , 139-170.

Faltings, B. (1990). Qualitative kinamatics in mechanisms. *Artificial Intellegence No. 44* , 89-119.

Forbus, K., Nielsen, P., & Faltings, B. (1991). Qualitative spatial reasoning: the CLOCK project. *Artificial Intellegence No.51* , 417-471.

Gantner, D. (2001). The Analogical Mind: Perspectives from Cognitive Science.

Gick, M., & Holyoak, K. J. (1983). Schema induction and analogical transfer. *Cognitive Psychology , 15* (1), 1-38.

Google. (n.d.). Retrieved from https://www.google.com/

Gordon, W. (1961). Synectics. *Harper and Row* .

Grossman, S., & Wiseman, E. (1993). Seven Operating Principles for Enhanced Creative Problem Solving Training. *Journal of Creative Behavior , 27* (1), 1-17.

Ham, C. W. (1958). *Mechanics of Machinery.* McGraw-Hill.

Hey, J., Linsey, J., Agogino, A., & Wood, K. (2012). Analogies and Metaphors in Creative Design. *Department of Mechanical engineering , Berkeley, CA* .

Hirtz, J., Stone, R., McAdams, D., Szykman, S., & Wood, K. (2002). A functional Basis for Engineering Design: Reconciling and Evolving Previous Efforts. *Research in Engineering Design 13* , 65-82.

Horton, H., & Newell, J. (2004). *Ingenious Mechanisms for Designers and Inventors.* Industrial Press.

Hunt, K. H. (1978). *Kinematic Geometry of Mechanisms.* New York: Oxford University Press.

ImageNet. (n.d.). Retrieved from http://www.image-net.org/

Joskowicz, L. (1989). Reasoning about the kinematics of mechanical devices. *Arteficial Intellegence No.4* , 22-31.

Joskowicz, L., & Sacks, E. (1991). Computatinal Kinematics. *Artificial Intellegence No. 51* , 381-416.

Keveney, M. (n.d.). Retrieved 2013-14, from 507movements web site: http://507movements.com/

Kota, S., & Chiou, S. (1992). Conceptual design of mechanisms based on computational synthesis and simulation of kinematic building blocks. *Research in Engineering Design No. 4* , 75-87.

Li, C., Tan, S., & Chan, K. (1996). A qualitative and heuristic approach to the conceptual design of mechanisms. *Engineering Applications Artificial Intellegence Vol. 9 No. 1* , 17-31.

Liu, B., & Hsu, W. (2000). Conceptual Design: Issues and Challenges. *School of Computing, National University of Singapore, S16 Computer Aided Design 32* , 849-850.

Mohan, M. (2011). A Framework for Holistic Ideation in Conceptual Design.

Mohan, M., Chen, Y., & Shah, J. (2011). Towards a framework for holistic ideation in conceptual design. *IDETC/CIE.* Washington D.C.

Nagel, R., Perry, & Stone, R. (2009). FunctionCAD: A functional modeling application based on function design framework. *IDETC/CIE* .

Neimark, E. D., & Kagan, J. (1987). *Adventures in thinking.* San Diego: Harcourt Brace Jovanovich.

Nix, A., Sherrett, B., & Stone, R. (2011). A function based approach to TRIZ. *IDETC/CIE* .

Osborn, A. (1979). Applied Imagination, Scribners, New York, NY. *Scribners* .

Parnes, S. (1987). Visioneering – State of the Art. *Journal of Creative Behavior , 21* (4), 283-99.

Russell, J., & Cohn, R. (2012). *Slicing (interface design).* Bookvika Publishing.

Schon, D. A. (1984). Problems, Frames and Perspectives on Designing. *Design Studies , 5* (3), 132-136.

Shah J., S. (1998). Experimental Investigation of Collaborative Techniques for Progressive Idea Generation.

Shah, J. (2011). *Advanced Product Design Methodology*.

Shah, J. (1998). Experimental Investigation of Collaborative Techniques for Progressive Idea Generation. *Proceedings of ASME Design Theory and Methodology Conference, DETC98/DTM-5676, .*

Shah, J. J. (1998). Experimental investigation of progressive idea generation techniques in engineering design. *ASME DETC Design theory and Methodology* .

Shah, J. J., Smith, S. M., & Vargas-Hernandez, N. (2003). Metrics for Measuring Ideation Effectiveness. *Design Studies , vol. 24, no. 2*, pp. 111-134.

Shah, J. J.-H. (2003). Metrics for Measuring Ideation Effectiveness. *Design Studies , 24* (2), 111-134.

Shah, J., Vargas, N., Summers, J., & Kulkarni, S. (2001). Collaborative Sketching (C-Sketch) – An Idea Generation Technique for Engineering Design. *Journal of Creative Behavior , 35* (3), 168-198.

Shigley, J. E. (2003). *Mechanical Engineering Design.* New Zealand: McGraw-Hill.

Szykman, S., Racz, J., & Sriram, R. (1999). The Representation of Function in Computer-based Design. *ASME Design Engineering Conferences(11th International Conference on Design Theory and Methodolofy), Paper No. DETC99/DTM-8742* .

THOMASNET. (n.d.). Retrieved 2014, from http://www.thomasnet.com/

Tractinsky, N., Katz, A. S., & Ikar, D. (2000, March 2). What is Beautiful is Usable. *Interacting with Computers* , 127-145.

TRIZ. (2006). Retrieved 2014, from IdeationTRIZ: http://www.ideationtriz.com/triz.asp

Vargas, H., Shah, J., & Smith, S. (Las Vegas, September 2007). Cognitive models of design ideation. *ASME DTM conference* .

Vattam, S. S., Helms, M. E., & Goel, A. K. (2010). A content account of crative analogies in biologically inspired design. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* , 467-481.

VDI. (1997). Konstruktionsmethodik/1 VDI-Handbuch Konstruktion. 2222,1,Methodisches Entwickeln von Losungsprinzipien.

Weakland, J., Fisch, R., & Watzlawick, P. (n.d.). Change: Principles of Problem Formulation and Problem Resolution. *New York: WW Norton Co.*

Wikipedia. (n.d.). Retrieved from https://www.wikipedia.org/

Zwicky, F., & Wilson, A. (1967). New Methods of Thought and Procedure: Contributions to the Symposium on Methodologies. *Springer: Berlin* .

APPENDIX A

LIST OF MECHANISMS

ID	NAME
1	Screw Kinematic Pair
2	Revolute or Turning Kinematic Pair
3	Prismatic or Sliding Kinematic Pair
4	Duangle in an Equilateral Triangle
5	Curved Triangular Rotor in a Square Chamber
6	Curved Triangle in a Rhombus
7	Curved Triangle Variation in a Square
8	Four-bar Linkage
9	Slider Crank Mechanism
10	Spherical Four-bar Linkage
11	Slider Crank Mechanism with Higher Order Pair
12	Inversion of Slider-Crank Mechanism with Higher Pair Joints
13	Slider Crank Mechanism
14	Simple Spur Gear Mechanism
15	Simple Planet and Ring Gear Wheel Train
16	Endless Screw or Worm Drive Mechanism
17	Slider-Crank Linkage With Variable Angle Slider
18	Swinging Block Slider Crank Mechanism
19	Turning Block Slider-Crank -Inversion
20	Slider-Crank Mechanism-Inversion
21	Four-Bar Linkage with Reduced Centrodes
22	Slider-Crank Linkage with Reduced Centrodes
23	Double Slider-Crank Mechanism with Circular Guideway
24	Double Slider-Crank Mechanism
25	Double Slider-Crank Mechanism-Inversion
26	Double Slider Ellipsograph of Leonardo da Vinci
27	Double Slider-Crank Mechanism-Inversion
28	Skewed Double Slider-Crank Mechanism

29	Annular Slider-Crank Mechanism
30	Eccentric slider crank Mechanism
31	Eccentric Slider Crank Mechanism
32	Eccentric-Slider Mechanism
33	Eccentric-Slider Mechanism
34	Eccentric Slider Crank Mechanism
35	Eccentric-Slider Mechanism
36	Simpson and Shipton Chamber Crank Mechanism
37	Cochrane Chamber Crank Mechanism
38	Beale Chamber Crank Mechanism
39	Ramelli Chamber Crank Mechanism
40	Wedding Chamber Crank Mechanism
41	Davies Spherical Engine
42	Compound Gear Train
43	Planetary Gear Wheel Train
44	Planetary Gear Train
45	Double Planet and Ring Gear Train
46	Double Planet and Gear Trains on Parallel Axes
47	Compound Gear Train with Reversing Pinion Gear
48	Two Coupled Spur Gear Sets for Reverse Motion
49	Model Support Pedestal
50	Pappenheim Chamber Wheel Mechanism
51	Root's Blower Chamber Wheel Mechanism
52	Co-rotating Spiral Pump
53	Evrard Ventilator
54	Repsold's Pump
55	Dart's Chamber Wheel Mechanism
56	Revillion Screw Chamber Wheel Mechanism
57	Galloway Chamber Wheel Mechanism

58	Simple, Crossed, Slider Kinematic Chain
59	Double Crossed Slider Kinematic Chains
60	Positive Return Mechanism with Curved Triangle
61	Positive Return Cam
62	Positive Return Cam Mechanism with Two Twin Curved Square Cams
63	Curved Pentagon Positive Return Cam
64	Positive Return Cam
65	Positive Return Cam
66	Simple Kinematic Screw Chain
67	Simple Kinematic Screw Chain
68	Reversing Screw Mangle
69	Reversing Screw of Whitworth
70	Double Screw of Napier
71	Differential Screw Measurement Mechanism
72	Differential Screw Mechanisms with Two Spur-Gear Pairs
73	Cylindrical Drilling Machine Mechanism with Reversing Gear Wheels
74	Ratchet and Pawl Mechanism
75	Centrifugal Unlocking Ratchet Coupling
76	Coupling With Ratchet and Pawl
77	Langen Overrunning Clutch
78	Internal Ratchet Coupling
79	Geneva Wheel Intermittent Mechanism
80	Rotating Digital or Intermittent Wheel Mechanism
81	Intermittent Mechanism with Cylinder Ratchet
82	Intermittent or Digital Clock Hand Mechanism
83	Intermittent Gear-Lever Mechanism
84	Locking Ratchet After Chubb
85	Intermittent Mechanism
86	Simple Ratchet Mechanism with Four-Bar Linkage

87	Continuous Running Ratchet of Lagarousse
88	Crown Wheel Ratchet Lifting Mechanism
89	Double Acting, Friction Wheel Levered Ratchet
90	Mudge's Escapement
91	Alternating-clamp, Friction Clutch Ratchet
92	Power Escapement Ratchet With Detachable Regulator
93	Controller Escapement after Farey
94	Power Escapement
95	Double Acting Power Escapement with Watt-type Automatic Regulator
96	Flywheel Machine Element
97	Planetary Gear with Four-bar Linkage
98	Planetary Gear with Slider Crank
99	Planetary Gear with Slider Crank
100	Planetary Gear with Slider Crank
101	Planetary Gear, Slider-Crank Kinematic Chain
102	Universal Joint
103	Double Universal Joint
104	Clemen’s Jointed Coupling
105	Basic Model of Clemen’s Coupling I
106	Basic Model of Clemen’s Coupling II
107	Spur Gears Mixed-teeth Profiles
108	Gear Teeth Profiles: Four-Tooth Spur Gear With Circular Rack
109	Spur Gears, Involute Teeth Profiles
110	Involute Gear Tooth Profile
111	Thumb Shaped Gear Teeth Profiles
112	Thumb-shaped Teeth, Rack and Pinion
113	Shield Gearing, (Disc Wheels)
114	Disc Wheels with Pin Teeth
115	Spherical Cycloid for a Plate Rolling on a Cone

116	Rolling Cones
117	Spherical Cycloidal Rolling
118	Rolling Cones with Point Paths for 1:3 Diameter Ratio
119	Circular Plate Rolling on a Cone
120	Point Path for a Rolling Cone on a Plane Surface
121	Exact Slider Crank Ellipse and Straight-line Linkage
122	Double Slider Trammel Ellipse Tracing Linkage
123	Oblique Slider Crank Straight-line Linkage
124	Oblique Double Slider Ellipse and Straight-line Mechanism
125	Four-bar Linkage Straight-line Mechanism after Evans
126	Approximate Four-bar Straight-Line Linkage of the Second Kind After Evans
127	Approximate Four-Bar, Ellipse Linkage of the Third Kind
128	Reuleaux Straight-line Mechanism
129	Double Slider Straight-line Mechanism of Reuleaux
130	Inversion of an Ellipse Linkage of the Second Kind by Nehrlich
131	Inversion on Ellipse Linkage of the Third Kind of Nehrlich
132	Inversion of an Ellipse Linkage of the Third Kind by Nehrlich
133	Inversion of an Exact Ellipse Linkage with Large Motion
134	Approximate Four-bar Straight-line Linkage of Roberts
135	Inversion of a Roberts' Straight-line Linkage
136	Hypocycloid Straight-line Mechanism
137	Inversion of Hypocycloid Gear Train Ellipse and Straight-line Mechanism
138	Epicycloid Straight-Line Linkage
139	Cross Link Straight-line Mechanism
140	Conchoidal Straight-line Linkage of the First Kind
141	Conchoidal Straight-Line Linkage of the Third Kind after Reichenbach
142	Conchoidal Straight-Line Linkage of the Third Kind after Reichenbach

143	Inversion of the Conchoidal Straight-Line Linkage of the Third Kind
144	Lemniscoidal Linkage of the First Kind by Watt
145	Lemniscoidal Linkage of Watt (II. and III. Kind)
146	Inversion of the Lemniscate Straight-Line Linkage
147	Sector Straight-Line Cycloid Linkage of Reuleaux
148	Sector Straight-Line Cycloid Linkage of Reuleaux
149	Sector Straight-Line Involute Linkage of Reuleaux
150	Six Link, Straight-Line Mechanism of Chebyshev
151	Six Link, Straight-line Mechanism of Harwey
152	Six Link, Straight-Line Mechanism of Reuleaux
153	Straight-line linkage of Maudslay
154	Cartwright Straight-line Mechanism
155	Peaucellier Straight-line Mechanism
156	Parallelogram Straight-line 'Stork Bill' Mechanism
157	Four-bar Linkage with Parallelogram Straight-line Mechanism
158	Half “Stork’s Bill” Straight-Line Mechanism
159	Rhombus Straight-Line Linkage
160	Parallel Mechanisms: Single and Compound Chains
161	Parallel Link Mechanism
162	Parallel Guide of Schwilgue
163	Parallel Stage of von Schönemann
164	Parallel Guide of Milward
165	Approximate Lever Parallel Scale Mechanism of Pfitzer
166	Parallel Guide of Georges
167	Parallel Guide of Roberwal
168	Parallel Guide of Quintenz
169	Parallel Guide of Beranger
170	Compound Screw Parallel Guide of Cadiat
171	Compound Crank Parallel Guide of Redtenbacher

172	Compound Rack and Pinion Parallel Guide of Redtenbacher
173	Parallel Guide for Paddle Wheel Mechanism
174	Four-arm Paddle Wheel Orienting Mechanism
175	Drag-link Orientation Mechanism
176	Belt Drive Mechanism
177	Belt Drive Between Circular and Conical Cylinders
178	Belt Drive with Variable Axle Direction Pulleys
179	Belt Guide for Parallel Axis Pulleys
180	Adjustable Belt Drive for Parallel Axis Pulleys
181	Belt Drive with Non-Parallel Axes
182	Belt and Pulley Drive Between Two Non-parallel Axes
183	Horrizontal to Vertical Axes Belt-Pulley Transmission
184	Belt Drive to Demonstrate Slipping in a Pair of Pulleys and a Rubber Belt
185	Belt Drive to Demonstrate Motion Transmision and Slipping Using Five Pairs of Pulley Wheels
186	Friction Wheels with Parallel Axes
187	Friction Wheels with Parallel Axes
188	Friction Wheels with Parallel Axes and Grooved Wheel
189	Grooved Friction Wheels with Parallel Axes
190	Friction Wheels with Transverse Axes and Grooved Wheel
191	Grooved Friction Wheels with Parallel Axes
192	Verge and Foliot Escapement
193	Cylinder Clock Escapement
194	Three-tooth Clock Escapement
195	Pin Escapement
196	Doublewheel Gravity Escapement
197	Anchor Deadbeat Escapement
198	Inverted Graham Anchor Clock Escapement Adapted by Reuleaux

199	Anchor Escapement with Lever
200	Chronometer Escapement
201	German Clock Striking Train
202	English Clock Striking Train
203	Reversing Belt Transmission Mechanism of Sellers
204	Three-Pulley Reversing Belt Transmission with Bevel Gears
205	Three-Pulley Reversing Belt Transmission with an Intermediate Gear Train after Schwarzkopf
206	Three-Pulley Reversing Belt Transmission with Planetary Gear Train
207	Reversing Belt Transmission with Friction Wheels
208	Variable Speed Friction Wheel Transmission of Sellers
209	Differential Gear Train Reversing Mechanism
210	Cylinder Coupling of Fossey
211	Cylinder Coupling of Köchlin
212	Power Machine Coupling of Uhlhorn
213	Power Machine Coupling of Pouyer
214	Oldham’s Coupling
215	Grooved Disc Coupling of Reuleaux
216	Cone Coupling
217	The Inclined Plane
218	The Lever
219	The Wedge
220	The Screw
221	Belt Drives
222	Chain Drives
223	Slotted Connecting Rod & Treadle Drive
224	Slotted Bell Crank Drive
225	Slotted Yoke Drive
226	Universal Joints

227	Out Of Line Drive
228	Out Of Line Drive
229	Scotch Yoke
230	Eccentric Drive
231	Eccentric Drive
232	Eccentric Drive
233	Pulley Lifts
234	Cone Pulleys
235	Straight Line Drive
236	Multiple Straight Line Drive
237	Straight Line Motions
238	Rotary Into Rectilinear Motion
239	Variable Speed and Reverse Drive
240	Reciprocating Rectilinear Motion
241	Ratchet Wheels (Boston) or Ratchet Pawls & Stops (Newark)
242	Ratchet Wheels and Drivers (Boston) or Pawl Drives & Stops (Newark)1
243	Geneva Movement
244	Contionous Rotary Into Intermittent Motion
245	Wave Wheel For Oscillating Motion
246	Oscillating Into Intermittent Circular Motion
247	Square Gears
248	Elliptical Gears
249	Reverse From Rotary Motion
250	Crown Wheel and Pinion
251	Worm and Gear
252	Miter and Bevel Gears
253	Worms and Gears
254	Swash Plate Gears
255	Variable Reciprocating Movement

256	Reverse Motion
257	Sewing Machine
258	Automobile Engine Starter
259	Universal Joint
260	Auto Timer and Distributor
261	Multiple Disk Clutch
262	Multiple Gear Drive
263	Lazy tongs motion
264	Old Oaken Bucket
265	Bailing or Lifting Scoop
266	Balance Pump
267	Force Pump (Boston) or Lift and Force Pump (Newark)
268	Overshot Water Wheel
269	Vertical Paddle Propeller Wheel
270	Baling Press
271	Gravity Trip Hammer
272	Gravity Drop Ore Stamp
273	Gravity Drop Hammer
274	Rotary Conveyor
275	Trunk Type Engine
276	Oscillating Cylinder Engine
277	Oscillating Piston Engine
278	Double Quadrant Steam Engine
279	Elliptical Gear Engine or Pump
280	Horizontal Slide Valve Engine
281	Spur Gear Unit with a 4-teeth Pinion
282	6-part Geneva Wheel Intermittend Mechanism
283	Heart-shaped gear wheel pair (Example 1)
284	Spur Gear Unit with a Quadrangular Wheel Pair

285	Heart-shaped Gear Wheel Pair (Example 2)
286	Non-uniform Periodic Motion Transmission by Non-circular Gears
287	Axial Displacement of a Gear by Trapezoidal Thread
288	Gear Drive with Spiral Characteristic
289	Worm and Two-wheel Drive Chain
290	Extreme Gear Reduction by Two Series-connected Worm Gear Drives
291	Change 3-speed Gearbox with Rotation Angle Indication via Screw Spindles
292	Gear Device with Racks for Superposition of Two Translational Motions
293	Gear Transmission
294	Gear Device with Flat Belt Drive
295	Gear Transmission for Superposition of Two Rotational Motions
296	Differential Bevel Gear with Fixed Carrying Wheel
297	Gear Train with Fixed and Swaying Axes
298	Gear Train of 7 Wheels Producing a rotational Motion with Stop
299	Gear Wheel Train Driving a Pinion Supported by Wheel Engagement only
300	Gear Wheel Train for Superposition of Two Rotational Motions of a Sphere
301	Double Planetary Gear with Revolvable Sun Wheel
302	Fou-bar Linkage (left)
303	Spatial Four-bar Linkage (right)
304	Mechanism illustrating Theorem of Roberts
305	Dual Parallel Four-bar Mechanism
306	Four-bar Linkage with Fly Wheel
307	Four-bar Linkage with Two Fly Wheels
308	Slider Crank Mechanism Containing an Antiparallel Crank Device with Dead Center Aid
309	Sliding Crank/Rocker Mechanism
310	Double Sliding Crank Mechanism with Adjustable Crank Length

311	Slider Crank Mechanism
312	Slider Crank Mechanism with Automatic Stroke Adjustment during each Revolution
313	Cross-guided Slider Crank Mechanism for Speed Transmission 1:2 without Gears
314	Parallel Mechanism with Pole Illustration
315	Spare Gear for Slider Crank Mechanism
316	Slider Crank Mechanism Driven by Excentric/Oval Wheel Combination
317	Slider Crank Mechanism Driven by an Elliptic Wheel Pair
318	Double Slider Crank Mechanism (right)
319	Double Slider-crank Mechanism
320	Double Slider-crank Mechanism with Pin Expansion
321	Excentric Slider Mechanism (left)
322	Eccentric Slider-crank Mechanism (right)
323	Heart-profiled Cam Mechanism
324	Cam Mechanism with Heart-shaped Profile
325	Watt´s Planetary Gear Slider-crank Mechanism
326	Peaucellier Straight-line Mechanism
327	Hypocycloid Two-gear Straight-line Mechanism
328	Reciprocating Gear/Slider Mechanism (Mangle Drive)
329	Reciprocating Gear-wheel/Rack Mechanism
330	Reciprocating 3-teeth Spider/Slider Mechanism (Mangle Drive)
331	Slider Crank Mechanism with Stroke Doubling by Wheel/Rack Combination
332	Straight-line Mechanism with a Twin Gear Driving Two Slider Cranks
333	Variable Stroke Motion by Two Cams Adjustable against Eachother
334	Cam Mechanism Adjustable by Spatial Profile
335	Cam Mechanism with Pauses
336	Cam Straight-line Mechanism with Uniform Bar Motion

337	Cam Straight-line Mechanism with Spiral Cam Profile
338	Balance-beam Straight-line Mechanism
339	Morgan´s Paddle Wheel Drive
340	Reciprocating Gear/Rocker Mechanism
341	Reciprocating 2-gear Mechanism
342	Reciprocating Gear Mechanism
343	Switching Mechanism with Gear Rack
344	Inverse Rotational Motion by Three Bevel Gear Segments
345	Geneva Wheel Intermittend Mechanism with a 2-step Switching
346	Differential Gear with Overrunning Clutch
347	Differential Bevel Gear with Strap Brake
348	Three-wheel Bevel Gear Drive for Belt Drive Reversal
349	Four-wheel Bevel Gear Drive for Belt Alternation
350	Gear for Belt Alternation with Planetary Gear Reduction
351	Jaw and Cone Clutch
352	Belt Drive with Return Stop
353	Bevel Gear Disengaging Device with Two Claw Couplings
354	Spur Gear Wheel Disengaging Device
355	Planetary Gear Train with Strap Brake
356	Stephenson Valve Gear Mechanism (Example 1)
357	Stephenson Valve Gear Mechanism (Example 2)
358	Planar Display Model of Steam Engine with Valve Control
359	Valve Control for Steam Engine
360	Speed Governor with Bevel Gear Drive
361	Spatial Bar-linkage Joint
362	Ellipsograph
363	Screw Spindle Drive by Differential Thread
364	Mechanism of Excentrically Supported Gears and a Double Crank for Drawing an Indicator Diagram (Example 1)

365	Mechanism of Excentrically Supported Wheels and a Double Crank for Drawing an Indicator Diagram (Example 2)
366	Trajectory Plotter for Piston Stroke versus Rotation Angle
367	Trajectory Plotter with Two Slider Crank Mechanisms
368	Four-bar Mechanism with Centrode
369	Multi-bar Linkage
370	Dynamometer with Bevel Gears
371	Drive with 3-fold Universal Joint
372	Curved Triangle in a Square Chamber
373	Double -helical Gear Pair
374	Eccentric Spur Gear Pair
375	Helical Rack and Pinion Gear Pair
376	Helical Worm Gear Pair [Endless Screw]
377	Herringbone Gear Pair
378	Intermittent Spur Gear Pair
379	Rack and Pinion Gear Pair
380	Right Angle Straight Bevel Gear Pair
381	Right Angle Worm Gear Pair
382	Worm Driven Rack and Pinion
383	Double Slider-crank Mechanism
384	Whitworth Slider-Crank, Quick-Return Mechanism
385	Maltese Cross Intermittent Mechanism
386	Elliptical Gear Pair

APPENDIX B

LIST OF COTS

ID	NAME
1	Shaft
2	Plain Coupling
3	Toothed Coupling
4	Flange Coupling
5	Bostflex Coupling
6	Jaw Coupling
7	Universal Coupling
8	Oldham Coupling
9	Fluid Coupling
10	Magnetic Coupling
11	Gear Coupling
12	Chain Coupling
13	Steel Grid Coupling
14	Beam Coupling
15	Square Key
16	Round Key
17	Gib Head Key
18	Woodruff Key
19	Round Pin
20	Tapered Pin
21	Split Pin
22	Hollow Pin
23	Spline Connection
24	Flat Belt Drive
25	Round Belt Drive
26	V Belt Drive
27	Timing Belt Drive
28	Sprocket Chain Drives

29	Silent Chain Drives
30	Spur Gear Drive
31	Helical Gear Drive
32	Herringbone Gear
33	Bevel Gear Drive
34	Worm Gear Drive
35	Face Gear Drive
36	Double Helical Gear Drive
37	Spiral Bevel Gear Drive
38	Hypoid Gear Drive
39	Crown Gear Drive
40	Non Circular Gear Drive
41	Rack and Pinion Gear Drive
42	Cage Gear Drive
43	Magnetic Gear Drive
44	Planetary Gear Drive
45	Harmonic Gear Drive
46	Epicyclic Gear Drive
47	Single Reduction Worm
48	Double Reduction Worm
49	Triple Reduction Worm
50	Bevel Gearbox
51	Concentric Helical Reducer
52	3 Stage Helical Reducer
53	Worm and Helical Reducer
54	Disc Brakes
55	Band Brakes
56	Drum Brakes
57	Self Lubricating Bearings/Plain Bearings

58	Hydrodynamic Bearings
59	Single Row Deep Groove Ball Bearing
60	Single Row Cylindrical Roller Bearing
61	Single Row Needle Roller Bearing
62	Single Row Tapered Roller Bearing
63	Double Row Ball Bearing
64	Single Row Thrust Ball Bearing
65	Jewel Bearings
66	Fluid Bearings
67	Magnetic Bearings
68	Flexure Bearings
69	Disc Clutches
70	Cone Clutches
71	Electro-Magnetic Clutches
72	Springs
73	Flywheels
74	Shunt Motors
75	Permanent Magnet Motors
76	Series Motors
77	Compound Motors
78	Single Phase Induction Motors
79	Polyphase Induction Motors
80	Synchronous Motors
81	Stepper Motors
82	Linear Motors
83	Two Phase AC Servo Motors

APPENDIX C

IMPLEMENTATION CODE

1. MECHANISMS CONTROLLER CODE

```
/* Filter Options */
var $filters = array(
  /* Find Mechanisms Filter options */
  'find' => array(
    'Mechanism' => array(
      'Mechanism.Name' => array(
        'condition' => 'like',
        'label' => 'Find Mechanisms by Keyword'
      ) ,
      'Mechanism.NAME' => array(
        'type' => 'select',
        'selectOptions' => array(
          'order' => 'Mechanism.NAME ASC'
        ) ,
        'required' => false,
        'multiple' => true,
        'label' => 'Find Mechanisms by Name',
        'filterField' => 'Mechanism.NAME'
      ) ,
      'Mechanism.GROUP' => array(
        'type' => 'select',
        'selectOptions' => array(
          'order' => 'Mechanism.GROUP ASC'
        ) ,
        'required' => false,
        'multiple' => true,
        'label' => 'Find Mechanisms by Group',
        'filterField' => 'Mechanism.GROUP'
      ) ,
      'Mechanism.MCI' => array(
        'type' => 'select',
        'selectOptions' => array(
          'order' => 'Mechanism.MCI ASC'
        ) ,
        'required' => false,
        'multiple' => true,
        'label' => 'Find Mechanisms by Machine Components 
Involved',
        'filterField' => 'Mechanism.MCI'
      ) ,
      'Mechanism.Function' => array(
        'type' => 'select',
        'selectOptions' => array(
        'order' => 'Mechanism.Function ASC'
      ) ,
      'required' => false,
      'multiple' => true,
      'label' => 'Find Mechanisms by Function',
      'filterField' => 'Mechanism.Function'
    )
  )
) ,
/* Select Mechanisms Filter options */
'select' => array(
  'Mechanism' => array(
    'Mechanism.IPTYPE' => array(
      'type' => 'select',
      'selectOptions' => array(
        'order' => 'Mechanism.IPTYPE ASC'
      ) ,
      'required' => false,
      'multiple' => true,
      'label' => 'Select Mechanisms by Input Type',
      'filterField' => 'Mechanism.IPTYPE',
    ) ,
    'Mechanism.OPTYPE' => array(
      'type' => 'select',
      'selectOptions' => array(
        'order' => 'Mechanism.OPTYPE ASC'
      ) ,
      'required' => false,
      'multiple' => true,
      'label' => 'Select Mechanisms by Output Type',
      'filterField' => 'Mechanism.OPTYPE'
    ) ,
    'Mechanism.IPSPEED' => array(
      'type' => 'select',
      'selectOptions' => array(
        'order' => 'Mechanism.IPSPEED ASC'
      ) ,
      'required' => false,
      'multiple' => true,
      'label' => 'Select Mechanisms by Input Speed',
      'filterField' => 'Mechanism.IPSPEED'
    ) ,
    'Mechanism.OPSPEED' => array(
      'type' => 'select',
      'selectOptions' => array(
          'order' => 'Mechanism.OPSPEED ASC'
        ) ,
        'required' => false,
        'multiple' => true,
        'label' => 'Select Mechanisms by Output Speed',
        'filterField' => 'Mechanism.OPSPEED'
      ) ,
      'Mechanism.IPVELOCITYDIRECTION' => array(
        'type' => 'select',
        'selectOptions' => array(
          'order' => 'Mechanism.IPVELOCITYDIRECTION ASC'
        ) ,
        'required' => false,
        'multiple' => true,
        'label' => 'Select Mechanisms by Input Velocity
Direction',
        'filterField' => 'Mechanism.IPVELOCITYDIRECTION'
      ) ,
      'Mechanism.OPVELOCITYDIRECTION' => array(
        'type' => 'select',
        'selectOptions' => array(
          'order' => 'Mechanism.OPVELOCITYDIRECTION ASC'
        ) ,
        'required' => false,
        'multiple' => true,
        'label' => 'Select Mechanisms by Output Velocity
Direction',
        'filterField' => 'Mechanism.OPVELOCITYDIRECTION'
      ) ,
      'Mechanism.REVERSIBILITY' => array(
        'type' => 'select',
        'selectOptions' => array(
          'order' => 'Mechanism.REVERSIBILITY ASC'
        ) ,
        'required' => false,
        'multiple' => true,
        'label' => 'Select Mechanisms by Reversibility',
        'filterField' => 'Mechanism.REVERSIBILITY'
      ) ,
      'Mechanism.RELBETWNIPAX' => array(
        'type' => 'select',
        'selectOptions' => array(
          'order' => 'Mechanism.RELBETWNIPAX ASC'
        ) ,
        'required' => false,
        'multiple' => true,
        'label' => 'Select Mechanisms by Relation Between 
Input and Output Line of Motion',
        'filterField' => 'Mechanism.RELBETWNIPAX'
      ) ,
      'Mechanism.Dimension' => array(
        'type' => 'select',
        'selectOptions' => array(
          'order' => 'Mechanism.Dimension ASC'
        ) ,
        'required' => false,
        'multiple' => true,
        'label' => 'Select Mechanisms by Dimension',
        'filterField' => 'Mechanism.Dimension'
      ) ,
      'Mechanism.DOF' => array(
        'type' => 'select',
        'selectOptions' => array(
          'order' => 'Mechanism.DOF ASC'
        ) ,
        'required' => false,
        'multiple' => true,
        'label' => 'Select Mechanisms by Degree of Freedom',
        'filterField' => 'Mechanism.DOF'
      ) ,
    )
  )
);

/* find mechanism method */
public function find($session, $pid = null)
{
    $this->Mechanism->recursive = 0;
    $this->set('mechanisms', $this->paginate());
    /* To display auto-populate solution form */
    if (!is_null($pid))
    {
        $this->set('pid', $pid);
        $this->set('session', $session);
    }
}

/* select mechanism method */
public function select($session, $pid = null)
{
    $this->Mechanism->recursive = 0;
    $this->set('mechanisms', $this->paginate());
    /* To display auto-populate solution form */
    if (!is_null($pid))
    {
        $this->set('pid', $pid);
        $this->set('session', $session);
    }
}
```

2. COTS CONTROLLER CODE

```
/* Filter Options */
var $filters = array(
  /* Find Machine Elements Filter options */
  'find' => array(
    'Cot' => array(
      'Cot.Name' => array(
        'condition' => 'like',
        'label' => 'Find Machine Elements by Keyword'
      ) ,
      'Cot.NAME' => array(
        'type' => 'select',
        'selectOptions' => array(
          'order' => 'Cot.NAME ASC'
        ) ,
        'required' => false,
        'multiple' => true,
        'label' => 'Find Machine Elements by Name',
        'filterField' => 'Cot.NAME'
      ) ,
      'Cot.CATEGORY' => array(
        'type' => 'select',
        'selectOptions' => array(
          'order' => 'Cot.CATEGORY ASC'
        ) ,
        'required' => false,
        'multiple' => true,
        'label' => 'Find Machine Elements by Category',
        'filterField' => 'Cot.CATEGORY'
      ) ,
      'Cot.MACHINEELEMENTCATEGORY' => array(
        'type' => 'select',
        'selectOptions' => array(
          'order' => 'Cot.MACHINEELEMENTCATEGORY ASC'
        ) ,
        'required' => false,
        'multiple' => true,
        'label' => 'Find Machine Elements by Machine Element
Category',
        'filterField' => 'Cot.MACHINEELEMENTCATEGORY'
      ) ,
      'Cot.FUNCTION' => array(
        'type' => 'select',
        'selectOptions' => array(
        'order' => 'Cot.FUNCTION ASC'
      ) ,
      'required' => false,
      'multiple' => true,
      'label' => 'Find Machine Elements by Function',
      'filterField' => 'Cot.FUNCTION'
    )
  )
) ,
/* Select Machine Elements Filter options */
'select' => array(
  'Cot' => array(
    'Cot.IPTYPE' => array(
      'type' => 'select',
      'selectOptions' => array(
        'order' => 'Cot.IPTYPE ASC'
      ) ,
      'required' => false,
      'multiple' => true,
      'label' => 'Select Machine Elements by Type',
      'filterField' => 'Cot.IPTYPE'
    ) ,
    'Cot.OPTYPE' => array(
      'type' => 'select',
      'selectOptions' => array(
        'order' => 'Cot.OPTYPE ASC'
      ) ,
      'required' => false,
      'multiple' => true,
      'label' => 'Select Machine Elements by Output Type',
      'filterField' => 'Cot.OPTYPE'
    ) ,
    'Cot.IPSPEED' => array(
      'type' => 'select',
      'selectOptions' => array(
        'order' => 'Cot.IPSPEED ASC'
      ) ,
      'required' => false,
      'multiple' => true,
      'label' => 'Select Machine Elements by Input Speed',
      'filterField' => 'Cot.IPSPEED'
    ) ,
    'Cot.OPSPEED' => array(
      'type' => 'select',
      'selectOptions' => array(
          'order' => 'Cot.OPSPEED ASC'
        ) ,
        'required' => false,
        'multiple' => true,
        'label' => 'Select Machine Elements by Output
Speed',
        'filterField' => 'Cot.OPSPEED'
      ) ,
      'Cot.IPVELOCITYDIRECTION' => array(
        'type' => 'select',
        'selectOptions' => array(
          'order' => 'Cot.IPVELOCITYDIRECTION ASC'
        ) ,
        'required' => false,
        'multiple' => true,
        'label' => 'Select Machine Elements by Input
Velocity Direction',
        'filterField' => 'Cot.IPVELOCITYDIRECTION'
      ) ,
      'Cot.OPVELOCITYDIRECTION' => array(
        'type' => 'select',
        'selectOptions' => array(
          'order' => 'Cot.OPVELOCITYDIRECTION ASC'
        ) ,
        'required' => false,
        'multiple' => true,
        'label' => 'Select Machine Elements by Output
Velocity Direction',
        'filterField' => 'Cot.OPVELOCITYDIRECTION'
      ) ,
      'Cot.RELATION' => array(
        'type' => 'select',
        'selectOptions' => array(
          'order' => 'Cot.RELATION ASC'
        ) ,
        'required' => false,
        'multiple' => true,
        'label' => 'Select Machine Elements by Relation
Between Input and Output Line of Motion',
        'filterField' => 'Cot.RELATION'
      )
    )
  )
);
```

```
/* find machine elements method */
public function find($session, $pid=null) {
    $this->Cot->recursive = 0;
    $this->set('cots', $this->paginate());
    /* To display auto-populate solution form */
    if(!is_null($pid)){
        $this->set('pid', $pid);
        $this->set('session', $session);
    }
}

/* select machine elements method */
public function select($session, $pid=null) {
    $this->Cot->recursive = 0;
    $this->set('cots', $this->paginate());
    /* To display auto-populate solution form */
    if(!is_null($pid)){
        $this->set('pid', $pid);
        $this->set('session', $session);
    }
}
```

3. USER SESSIONS CONTROLLER CODE

```
/* Show Morph Chart method */
public function morph_chart($id)
{
    $session = $this->Session->read();
    $this->MorphChartProblem-
>setEncryptionKey($session['cryptkey']);
    $this->MorphChartSolution-
>setEncryptionKey($session['cryptkey']);
    $this->SolutionSet-
>setEncryptionKey($session['cryptkey']);

    // retrieve the user session and set it to a variable
accessible in the view
    $UserSession = $this->UserSession->findById($id);
    parent::log_entry($UserSession['UserSession']['id'],
"Loaded morph chart page");
    $this->set(compact('UserSession'));
    $this->set('usersessions', $UserSession);

    // images
    $data = $this->MorphChartImage->find('all');
    $this->set('datas', $data);
    $rootProblems = $this->MorphChartProblem->find('all',
array(
      'conditions' => array(
        'MorphChartProblem.morph_chart_problem_id' => '',
        'MorphChartProblem.session_id' => $id
      )
    ));
    $this->set('rootproblems', $rootProblems);
    $this->set('uid', $id);
    $manualSolutions = $this->MorphChartManualSolution-
>find('all', array(
      'order' => array(
        'MorphChartManualSolution.manualSolutionSet ASC'
      )
    ));
    $setid = 0;
    $probid = 0;
    $manualSolutionSet = array();
    $sol_id = - 1;
    foreach($manualSolutions as $manualSolution) {
        $prid =
$manualSolution['MorphChartManualSolution']['morph_chart_pro
blem_id'];
        if ($setid != $prid)
        {
            $i = - 1;
            $setid = $prid;
            $manualSolutionSet[$setid] = array();
        }

        if ($sol_id !=
$manualSolution['MorphChartManualSolution']['manualSolutionS
et'])
        {
            $i++;
            $manualSolutionSet[$setid][$i] = array();
            $sol_id =
$manualSolution['MorphChartManualSolution']['manualSolutionS
et'];
        }

        array_push($manualSolutionSet[$setid][$i],
$manualSolution['MorphChartManualSolution']);
    }

    $this->set('manualsolutionset', $manualSolutionSet);
    $this->set('manualsolutions', $manualSolutions);

    // this is for JSON and XML requests.
    $this->set('_serialize', array(
      'UserSession',
      'rootproblems'
    ));

    // fetch solution sets
    $solnsets = $this->SolutionSet->find('all', array(
      'conditions' => array(
        'SolutionSet.session_id' => $id
      ) ,
      'order' => array(
        'SolutionSet.id DESC'
      )
    ));
    $this->set('solnsets', $solnsets);
}
```

```
/* Show Subtable method (Sub-Problems Morph Chart) */
public function show_subtable($id, $pid){
    $session = $this->Session->read();
    $this->MorphChartProblem-
>setEncryptionKey($session['cryptkey']);
    $this->MorphChartSolution-
>setEncryptionKey($session['cryptkey']);

    // retrieve the user session and set it to a variable
accessible in the view
    $UserSession = $this->UserSession->findById($id);
    parent::log_entry($UserSession['UserSession']['id'],
"Loaded morph chart page");
    $this->set(compact('UserSession'));

    // this is for JSON and XML requests.
    $this->set('_serialize', array('UserSession'));
    $this->set('usersessions',$UserSession);

    // images
    $data = $this->MorphChartImage->find('all');
    $this->set('datas',$data);
    $parentProblem = $this->MorphChartProblem-
>findById($pid);
    $childProblems=$this->MorphChartProblem-
>find('all',array(
        'conditions' =>
array('MorphChartProblem.morph_chart_problem_id' => $pid)
    ));

    $this->set('parentProblem',$parentProblem);
    $this->set('childproblems',$childProblems);

    $manualSolutions = $this->MorphChartManualSolution-
>find('all',array(
        'order' =>
array('MorphChartManualSolution.manualSolutionSet ASC')
    ));

    $setid = 0;
    $probid = 0;
    $manualSolutionSet = array();
    $sol_id = -1;
    foreach($manualSolutions as $manualSolution){
          $prid = 
$manualSolution['MorphChartManualSolution']['morph_chart_pro
blem_id'];
        if($setid != $prid){
            $i = -1;
            $setid = $prid;
            $manualSolutionSet[$setid] = array();
        }

        if($sol_id != 
$manualSolution['MorphChartManualSolution']['manualSolutionS
et']){
            $i++;
            $manualSolutionSet[$setid][$i] = array();
            $sol_id = 
$manualSolution['MorphChartManualSolution']['manualSolutionS
et'];
        }


array_push($manualSolutionSet[$setid][$i],$manualSolution['M
orphChartManualSolution']);
    }

    $this->set('manualsolutionset',$manualSolutionSet);
    $this->set('manualsolutions',$manualSolutions);

    $this->set('uid',$id);
    $this->set('pbid',$pid);
}

/* Auto-Populate Solution in Morph Chart from different 
internal databases */
public function auto_add_solution($session_id, $pid, $sname, 
$type, $soi){
    $session = $this->Session->read();
    $this->MorphChartProblem-
>setEncryptionKey($session['cryptkey']);
    $this->MorphChartSolution-
>setEncryptionKey($session['cryptkey']);
```

```
    $data = array('session_id'=> $session_id,
'morph_chart_problem_id' => $pid, 'name' => $sname, 'soi' =>
$type.':'.rawurldecode($soi));
    if ($this->MorphChartSolution->save($data)) {
        $this->Session->setFlash(__('Solution was saved
successfully'));
    } else {
        $this->Session->setFlash(__('Error: Solution was not
saved'));
    }
    $this->autoRender = false;
}
```

4. MORPH CHART MANUAL SOLUTIONS CONTROLLER CODE

```
/* Save Manual Solution Sets method */
public function save_solutions_manually($id, $pid, $sid,
$mid){
    $session = $this->Session->read();
    $data = array('session_id'=> $id,
'morph_chart_problem_id' => $pid, 'morph_chart_solution_id'
=> $sid, 'manualSolutionSet' => $mid);
    // Create new manual solution set
    $this->MorphChartManualSolution->save($data);
    $this->Session->setFlash(__('Solution set created.'));
    $this->autoRender = false;
}

/* Delete Manual Solution Set method */
public function delete_solution_set($set_id){
    // Delete all the data that has the set id
    if($this->MorphChartManualSolution-
>deleteAll(array('MorphChartManualSolution.manualSolutionSet
' => $set_id), false)){
        $this->Session->setFlash(__('Morph chart manual
solution set was deleted'));
        $this->autoRender = false;
    } else {
        $this->Session->setFlash(__('Morph chart solution
set was not deleted. Please try again.'));
    }
}
```

5. SOLUTION SETS CONTROLLER CODE

```
/* Save Complete Solution Set method */
public function add() {
    if($this->request->is('post')){
        $this->SolutionSet->create();

        if($this->SolutionSet->saveAll($this->request-
>data)){
            $this->Session->setFlash('New solution set has
been created.');
            $message = $this->SolutionSet->read();

        }
        else
            $message = "Error";

        $this->set(compact('message'));
        $this->set('_serialize', array('message'));
    }
    $this->autoRender = false;
}

/* Delete Complete Solution Set method */
public function delete($id) {
    $this->SolutionSet->id = $id;
    $solution = $this->SolutionSet->read();
    $session = $solution['SolutionSet']['session_id'];

    // delete the problem and redirect to user session.
    if ($this->SolutionSet->delete($id)) {
        $this->Session->setFlash('The solution set with id:
' . $id . ' has been deleted.');
        $this->redirect(array('controller' =>
'user_sessions', 'action' => 'morph_chart', $session));
        $message = 'Deleted';
    } else {
        $message = 'Error';
    }

    // this is for JSON and XML requests.
    $this->set(compact("message"));
    $this->set('_serialize', array('message'));
}
```