**INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED'07**
28 - 31 AUGUST 2007, CITE DES SCIENCES ET DE L'INDUSTRIE, PARIS, FRANCE

# THE IDEA CONCEPT DESIGN PROCESS

**Andrew Masur[1], Dr. Filippo A. Salustri[2]**
[1]Ryerson University
[2]Ryerson University

## ABSTRACT
Concept design is one of the most important and intricate parts of engineering design. The authors have created the IDEA process to help manage complexity and improve concept design practices in engineering industry. An analysis of existing methods was conducted in order to identify weaknesses and areas for integration. The IDEA process was created to integrate many of these techniques together in a series of systematic steps to reduce designer confusion. The authors have created a supporting software tool to help minimize many of the bookkeeping tasks associated with IDEA and reduce the time required for concept design. The tool is built on the open source program Compendium. Three multi-disciplinary case studies were conducted to validate the process. The use of IDEA was found to lead to an increase in the number and quality of concepts generated.

*Keywords: Concept Design, IDEA, Compendium, Engineering Design, Design Software*

## 1 INTRODUCTION

Today's technological society depends on engineering, which helps us communicate with people over vast distances, move people and resources to locations around the globe, and provides many of the products that we use in our everyday lives. With the prevalence of engineering in today's society it would seem logical that people would have a basic understanding of exactly what it is that engineers do, however that is not normally the case. Some people see engineers as individuals who take advantage of a moment of inspiration in order to create a new product or process, while others see them as nothing more than human calculators who plug numbers into arcane formulae to generate new products. While both of these are important aspects of engineering, the reality is that engineers are nothing more than problem solvers. They take technical knowledge and apply it to a perceived problem in society in order to solve it.

The engineering process typically involves a number of phases. Figure 1 shows one of many representations of an engineering process. In this case, the figure is from a well-known method in aerospace engineering (our work was originally driven by aerospace engineering needs), but it is typical of many other methods (e.g. [1]). As can be seen in Figure 1, the detailed steps in the engineering process can differ between institutions; however, the overriding phases remain quite similar. The first phase is the *problem exploration and concept design phase*, where engineers gain a better understanding of the design problem and then generate initial solutions. This tends to be a very enjoyable process for engineers since this is where they can be creative, however it can also be quite confusing since there is very little concrete information available, making it difficult to identify relevant paths of thinking. The second phase in the design process is the *detailed design phase*, where engineers now take the best of their initial solutions and expand them into a real working design. The third stage deals with the actual *manufacture of the design*. Unforeseen problems will arise often at this stage, necessitating design revisions. The last phase of the engineering process is concerned with the *actual operation and disposal* of the product in question, where customer support and alterations to the as-built product may require engineering intervention.

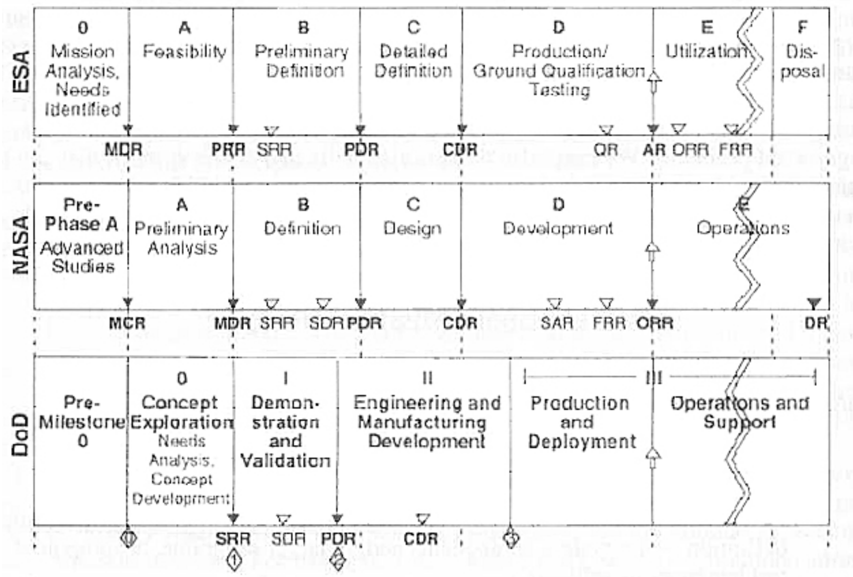


*Figure 1: Various Engineering Processes [2]*

Of all these phases, the concept design phase is the most critical. It is at this stage in the process that the *essence* of a product is determined. All subsequent phases assume that the design concept selected will remain essentially fixed throughout the rest of the development. Though one can optimize a design in downstream phases to make it even better, one cannot “manufacture out” the inherent flaws in a design. Thus, it is likely that a bad decision at the concept design phase will lead to a bad, or at least highly suboptimal, product. Delays in product delivery can accumulate quickly, but could have been avoided entirely by better concept design. On the other hand, the “early” phases of product development, including concept design, are usually far less cost-intensive than the resource-heavy phases downstream. Clearly, focussing on concept the design has the ability to benefit the overall product engineering process the most without constituting undue financial burdens on the enterprise.

## 2 CURRENT TECHNIQUES IN CONCEPT DESIGN

Many techniques have been developed to mitigate the lack of reliable, quantitative information so common in concept design. Here, the authors briefly introduce the techniques that we have incorporated into our new *Integrated Design Exploration and Analysis* (IDEA) method.

The first task during concept design is to gain a better understanding of the problem since it is difficult to provide a reasonable answer when one does not know the question. There are a number of different techniques which can be used to do this. The first of these is customer interviews in order to prioritize design requirements. Advocated by Ralph L. Keeney [3], this technique involves the use of personalized interviews, group interviews, and mass distributed electronic interview forms in order to determine the needs of customers. By determining the needs of average customers, designers are better able to determine exactly what features and characteristics will maximize a product’s chance of success.

Another popular technique for gaining a better understanding of a design problem is *trade space analysis*. A trade space is nothing more than a theoretical construct which represents possible solutions to a problem. One possible representation of a trade space can be seen in Figure 2.

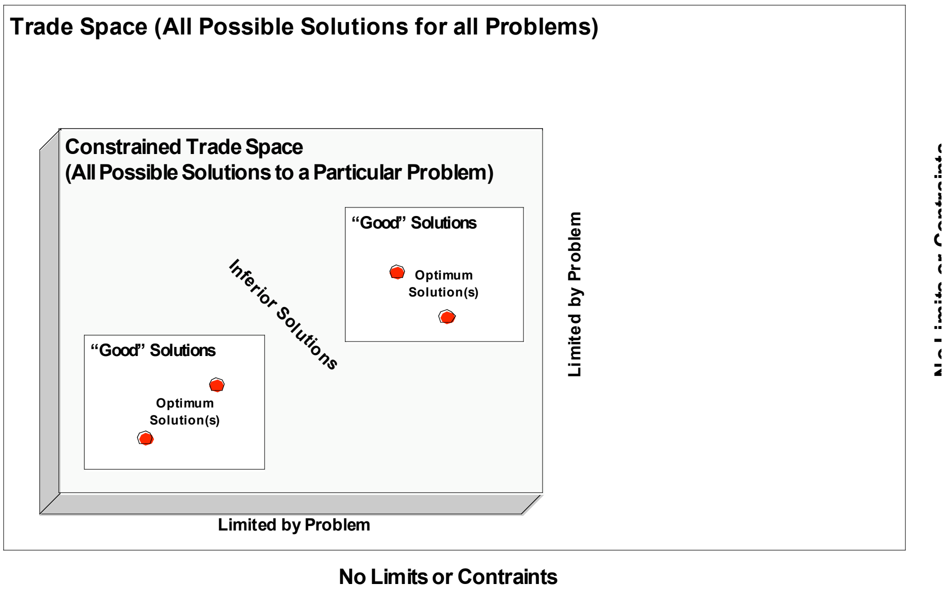


*Figure 2: Trade Space Representation*

The large square represents the total trade space, a theoretical construct which contains all possible solutions to all possible design problems. Theoretically, this space is infinite in size, which makes locating the best solution for a particular design problem intractable. Through trade space analysis designers explore the limitations on possible solutions (designs) for a particular problem. These limitations help to define a sub-space, represented in Figure 2 by the raised grey region. This sub-space contains the reasonable possible solutions to that particular problem. Using other techniques such as brainstorming sessions, interviews with customers and stakeholders, and analysis on the feasibility of certain options, engineers are further able to reduce the size of the sub-space and converge on the "best" solution with respect to the criteria they have set.

While an exploration of the trade space is a meaningful and beneficial exercise, eventually the design team will have to decide how to proceed with the design problem. Again, there are a number of tools available to designers to assist with this task. One of the simplest is *pairwise comparison*. Several variations of pairwise comparison exist, but some of these have problems because they violate Arrow's Theorem [4]. While a detailed discussion of the reasons for these violations is beyond the scope of this paper, the consequences can be quite severe. Chief among them is the possibility of *rank-reversal*, in which removal of one alternative substantively alters the relative ranking of the other alternatives. This is clearly unacceptable because it means that preference of alternatives depends on the mere *existence* of other alternatives. Details of this and other issues in the use of pairwise are discussed in detail in [5].

An alternative pairwise comparison method uses what is known as the pairwise comparison chart. A pairwise comparison chart is a graphical pairwise comparison technique that produces results identical to the well-known Borda count [4]. This is beneficial to designers since the Borda count is insensitive to violations of Arrow's Theorem, and rank-reversal in particular, making it a much more robust method for engineering decision making. This makes the pairwise comparison chart ideal for calculating the level of preference between simple items.

While pairwise comparison methods are quite useful for voting between simple alternatives with only a single criterion, multiple criteria decision making problems require a more robust method. The solution to this problem is a *decision matrix*. A decision matrix is a decision making tool based on a weighted rankings of alternatives against defined characteristics, and with respect to known reference solutions, and is presented as a table of values [6]. It is particularly useful when choosing amongst different concepts whose performance varies with respect to many characteristics. The decision matrix can be used with a standardized scale to rate how well each of the concepts achieves each of the different characteristics. Figure 3 shows an example of a completed decision matrix for three different concepts for a fictional garden tool. It is assumed that the first concept is the reference, or baseline, concept.

| | | Concept | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | A | | B | | C | | D | |
| | | Hoe (Baseline) | | Modified Hoe | | Rotating Mixer | | Electric Autotiller | |
| **Design Characteristic** | **Relative Weight** | **Rank** | **Score** | **Rank** | **Score** | **Rank** | **Score** | **Rank** | **Score** |
| Mass | 0.2 | 0 | 0 | 2 | 0.4 | 1 | 0.2 | 0 | 0 |
| Speed | 0.1 | 0 | 0 | -2 | -0.2 | 2 | 0.2 | 2 | 0.2 |
| Operating Comfort | 0.25 | 0 | 0 | 0 | 0 | 1 | 0.25 | 2 | 0.5 |
| Ease of Operation | 0.15 | 0 | 0 | 1 | 0.15 | 1 | 0.15 | 2 | 0.3 |
| Reliability | 0.15 | 0 | 0 | 2 | 0.3 | 1 | 0.15 | 0 | 0 |
| Cost | 0.15 | 0 | 0 | 2 | 0.3 | 1 | 0.15 | -1 | -0.15 |
| **Total Score** | | | 0 | | 0.95 | | 1.1 | | 0.85 |

*Figure 3: Completed Decision Matrix*

Each of the concepts is judged by the participants (a design team of some sort) against a baseline design concept, and with respect to an identified set of characteristics. The characteristics are listed on the left side of the chart. They can be anything that the team determines is important, but typically deal with things like mass, speed, and capacity. This allows for meaningful comparisons between the concepts since all are measured against a common baseline. The relative weights listed along the left of the matrix are essential because not all characteristics are equally important in a given problem. The values for these relative weights are most effective when they reflect the opinions of the entire design team. One of the simplest ways to accomplish this is though the use of the pairwise comparison chart previously described. The pairwise comparison chart fosters discussion amongst team members, enabling multiple viewpoints to be represented; however, it also facilitates the convergence of these multiple viewpoints into a single cohesive set.

The ranking scale captures uniformly how each concept meets the characteristics. Figure 3 uses a -2 to +2 scale, which is common in the automotive industry [6]. In reality, one can use almost any rating scale as long as it remains consistent across all concepts. The final score for each concept is calculated using a Simple Additive Weighting (SAW) aggregation function with the following form where $w_i$ represents the relative weight of a particular design characteristic and $R_i$ represents the rating for that characteristic for the current concept, as shown in Equation 1.

$$score = \sum_{i=1}^{n} w_i R_i \tag{1}$$

SAW generates a total score for each concept by summing the products of the relative weight for a design characteristic by the rating for that characteristic. The concept with the highest total score best meets the needs of the design project. This very simple additive function has been proven [7] to generate results that are comparable to more complex multiplicative functions.

## 3 THE IDEA PROCESS

Each of the above techniques, taken individually, has weaknesses. First, while each of these techniques is very good at a particular task, none of them presents a complete solution. For example, trade space analysis is very good at exploring the possible solutions to a problem, but it does not provide a way to help designers select one of those solutions. Conversely, while a decision matrix is good at selecting amongst a set of concepts, it does not really help with the initial generation of those concepts.

The second problem with these techniques is that they can actually increase confusion for designers during concept design. The techniques presented are only a small fraction of all the tools available to engineers. It is difficult to know which tool is best for a particular situation, and each tool has a number of variations.  This can lead to a kind of information overload for the engineers who must not only execute a design, but also select the tools they will use. This can have a negative impact on an engineer's creativity since they spend most of their time worrying about *how* to tackle the problem rather than actually *doing* it.

The Integrated Design Exploration and Analysis (IDEA) method attempts to address these problems. Rather creating a new technique for concept design, IDEA integrates one set of techniques into a concept design *process*. The overriding goal for IDEA is to reduce designer confusion and to help them generate "better" designs.  IDEA does this by first using concept design techniques which engineers are already familiar with, so they do not have to relearn something that already works. Secondly, IDEA follows a *standardized* procedure that helps to reduce confusion since designers know what technique they should be using and when. An overview of IDEA is given in Figure 4.

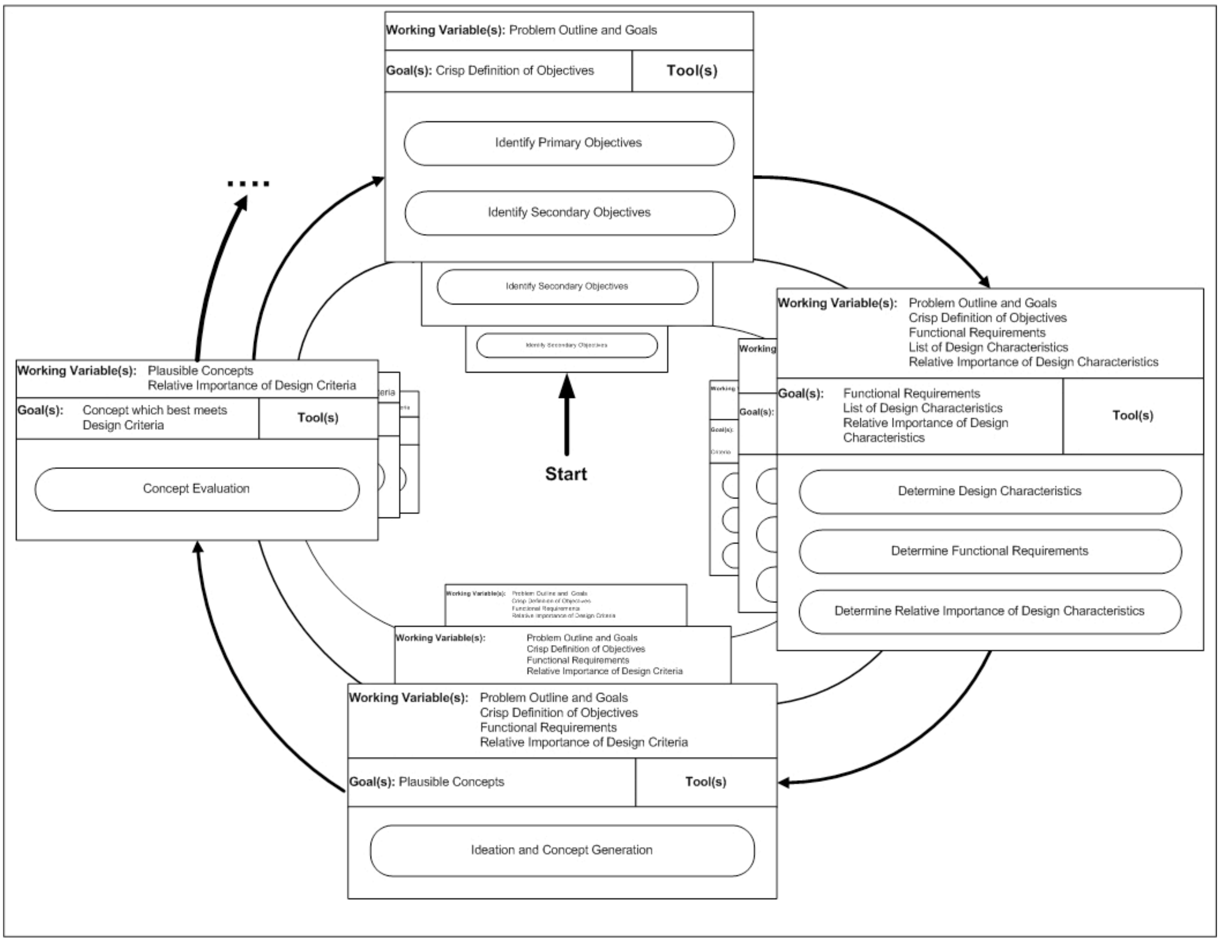


*Figure 4: IDEA Overall Outline*

As shown in Figure 4, IDEA is broken down into four main modules arranged in a spiral pattern starting from the bottom and continuing in a repeating loop. This structure illustrates the iterative nature of concept design. Looking at Figure 4 we can see that the text within the boxes is not clear. This was done intentionally to show that during the early stages of concept design designers often have very little hard information available, and only a vague understanding of the design problem. As designers iterate through the IDEA process their understanding of the problem increases and the information available become less vague. This is demonstrated in the diagram by the increasing clarity of the text within each of the modules. Each of these modules is responsible for a different aspect of the concept design process.

The first module, called *Identify*, uses brainstorming and meetings with the various project stakeholders to determine the primary and secondary objectives for the design problem. The primary objectives are things that the design must accomplish in order to be considered successful, while the secondary objectives are optional things which would be nice to have, but are not mission-critical.

The *Determine* module helps determine the design characteristics for the product, their relative importance, and the functional requirements for the design. The design characteristics define what the product must *be*. They are typically found through brainstorming sessions and meetings between design team members and stakeholders, and use a list of standardized design characteristics. The relative importance of the design characteristics is determined by using pairwise comparison charts. The use of the chart not only makes this process more systematic, but also helps to foster discussion amongst team members and stakeholders about the final rankings. The functional requirements define what the product must *do* and are derived from the primary and secondary objectives through brainstorming and meetings.

The purpose of the *Explore* module is to find solutions to the problem. After completion of the *Identify* and *Determine* modules, the design team has a better understanding of the boundaries of the problem trade space, and the *Explore* module helps them locate concepts within this space. This is accomplished with *trade space analysis* (sometimes called *ideation*). In trade space analysis, the design team generates different options for each of (a) the functional requirements or (b) subsystems (if known), or (c) the design characteristics, and plots them on a trade space diagram as shown in Figure 5. The design team then tries different combinations of options until they generate 5-10 plausible concepts that will undergo more detailed analysis.

**Trade Space Diagram**

| | | | | |
|---|---|---|---|---|
| **Bus Interface** | 601 comp. 64-bit | 601 comp. 32-bit | 601 comp. 64/32-bit | New Prot. 32-bit |
| **Branch Unit** | -issue early decode and fo | 2-Issue through main | 3-issue through main | |
| **Cache Organizati** | Split 4K 2-way | Split 4K 4-way | Split 8K 2-way | Split 8K 4-way |
| **Dispatch Units** | Dual Issue In-Order | Dual Issue Out-of-order | Single-Issue | |
| **No. of Completio** | 2 | 3 | 4 | 5 |
| **Instructions per** | 1 Instruction | 2 Instructions | 2 Instruction + 1 Retire | |
| **Execution Units** | 2 + 1 Reservation | 2 + 2 Reservation | 3 + 3 Reservation | 3 + 1 Reservation |

*Figure 5: Trade Space Diagram*

The authors note that much of the work in using a Trade Space Diagram involves discarding unacceptable concepts. Figure 5, for instance, implicitly represents 29,160 distinct concepts. By "casting such a wide net," this technique virtually guarantees the capture of the "best" concept. It is, however, surrounded by many unacceptable solutions. The team must quickly eliminate most of the concepts. There are various ways to do this. One obvious way is to search for idea pairs that are known to be unacceptable for some technical or other reason. For each pair thus discovered, every concept containing the pair can be eliminated. Partial computerization of this process can greatly expedite this elimination process. The end result is a small set of concepts, that the team is confident may be the "best" of all the concepts. A more detailed discussion goes beyond the scope of this paper, but some further details are available in [5].

The final module of the IDEA process is the *Analyze* module; the purpose of which is to analyze the remaining concepts and determine which should move on to detailed design. This is accomplished with a decision matrix as described in the previous section. A decision matrix is not only perfectly suited for this type of task, but its use allows the participation of many different individuals in the decision making process. This allows multiple points of view to be represented in the final selection and improves the probability that those selections are truly the "best" ones. In addition, because the decision matrix also utilizes the relative weights generated in the *Determine* module, the viewpoints of other stakeholders may also be represented.

## 4 THE IDEA SOFTWARE INTERFACE

While the IDEA process was designed to be a "lightweight" process, there is still considerable bookkeeping involved. It requires that engineers record a lot of information as they complete the various steps within the process, and while this is not difficult, it is time consuming. This is tedious – and therefore error-prone – work, and the time spent recording information is time *not* spent *designing.* Thus, the authors reasoned that IDEA users need a software tool to assist with managing all of the information generated during the process. Computerized recording of information used during IDEA also improves traceability of the process. If any questions arise about the choices made during concept design, the team only has to recall the appropriate information from the computer tool.

The IDEA software tool was built on top of an open source program called Compendium [8]. It became apparent that the tool would be graphical in nature. Creating a custom software engine was quite difficult and far beyond current interests of the authors. After extensive research, Compendium was selected for a number of reasons. First, Compendium has an extensible graphical interface that would eliminate the need to write code. Secondly, Compendium is an open source program, so there are no licensing costs; this is hoped to attract early adopters to the tool. Thirdly, Compendium is already being used in a wide variety of different engineering settings. One of the most interesting is at the Jet Propulsion Laboratory (JPL) in California where they use a custom interface to operate the Mars Exploration Rovers currently on the Martian surface [8]. The fact that Compendium was being used for such a mission-critical task spoke very highly of its versatility and robustness.

The final reason for the selection of Compendium was the fact that it was originally created as a knowledge mapping tool. The basic purpose behind Compendium both as a tool and a method is organizing seemingly disparate pieces of information into graphical maps that show the connections between these pieces of information. This is precisely the type of structure that is traditionally used in brainstorming and other design techniques such as the IBIS method. The tools for conducting brainstorming were already part of Compendium and refined through many years of research and development. By using Compendium we were able to leverage this refinement and incorporate it into the IDEA process, making it far more streamlined and powerful.

The current implementation of the IDEA software closely mimics the overall outline diagram presented in Figure 4. The IDEA Home Screen lays out each module similar to Figure 4, using light blue to distinguish them from other entities. Each of these areas contains icons that represent the tasks that the design team must accomplish in order to complete that module. The tasks are activated by simply clicking on the appropriate icon, which opens a separate window not only outlining what users must do to complete that task, but also providing tools to complete it. The Home Screen is presented in Figure 6, and an example of a task window is shown in Figure 7.

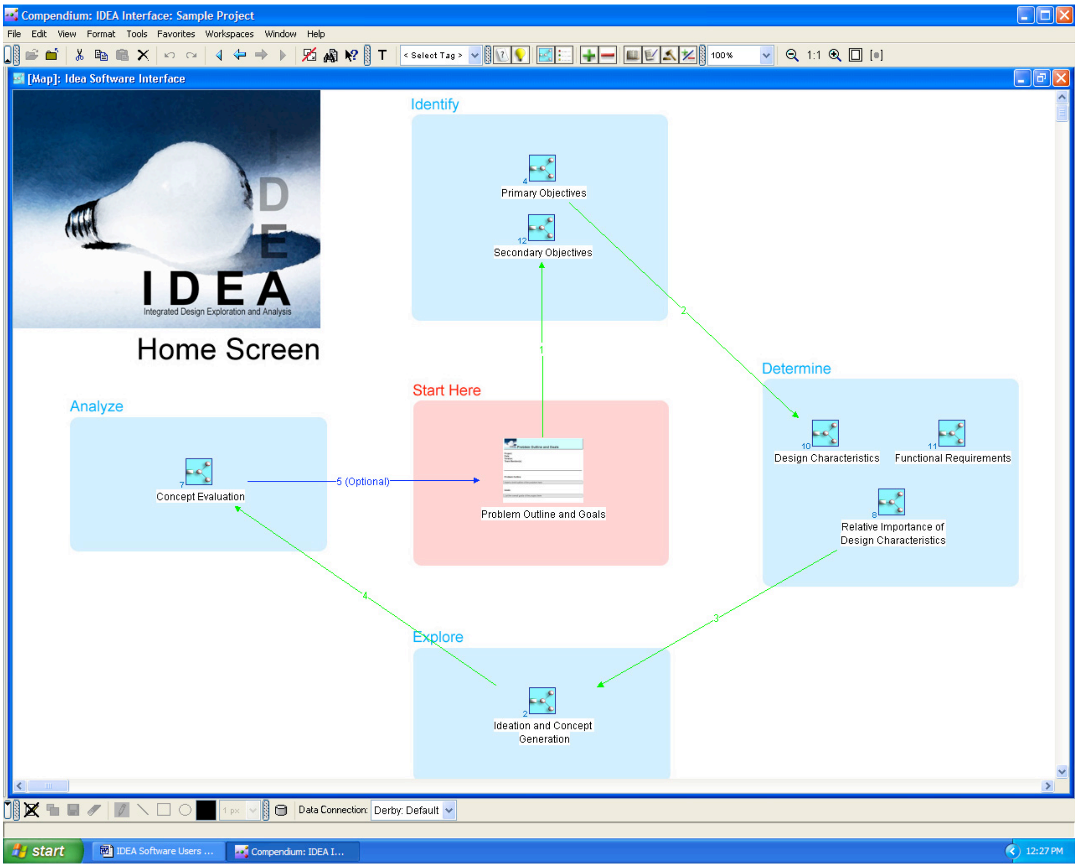


*Figure 6: IDEA Home Screen*

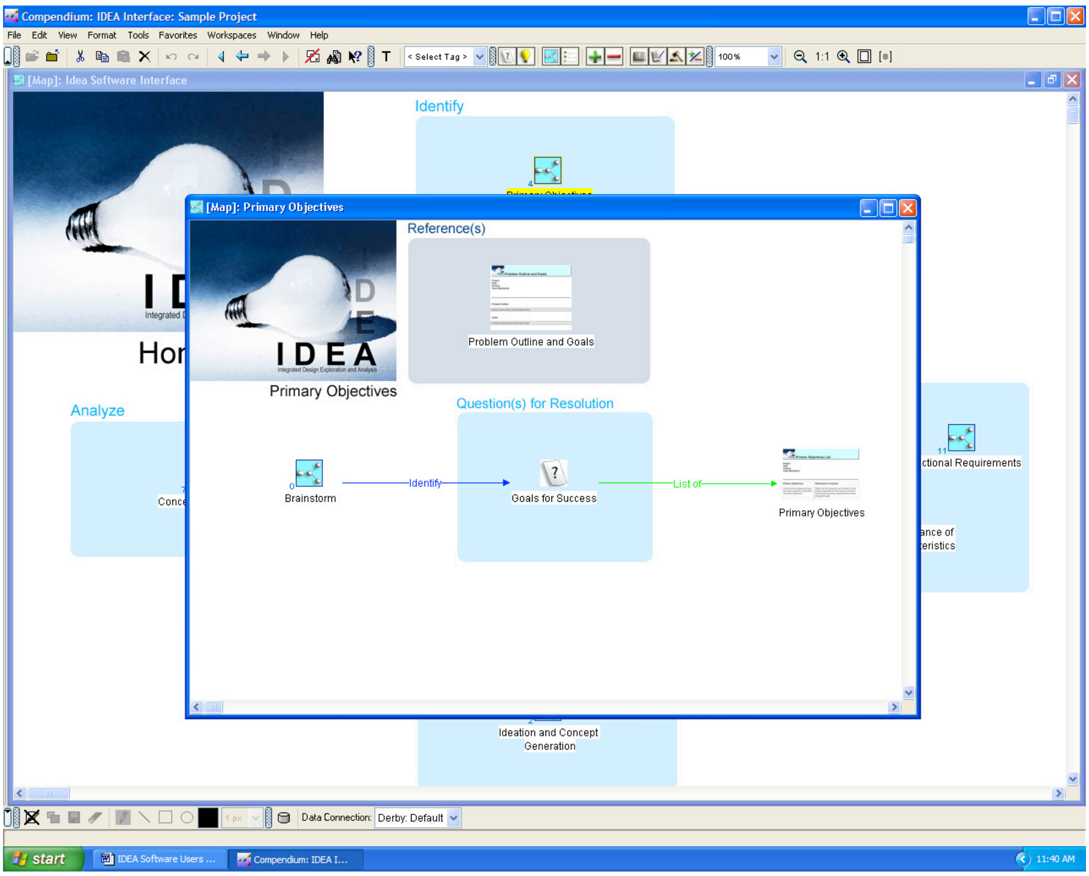


*Figure 7: IDEA Task Window*

In both figures above, we can see icons connected by arrows. In the Home Screen, the arrows guide the user from one module to another, whereas the arrows in the task window graphically describe activities that must be done to complete a task. Figure 7 has an example of this. There, it states that the design team must "**Brainstorm** to **identify goals for success** and add these to the list of **primary objectives**." This structure is the same in all of the task windows, informing users as to the next step. We can also see that distinct icons represent both the brainstorming and primary objectives lists. When activated, these icons open blank templates for the design team to use. While the team is still required to record manually the results of a design session, the use of standardized templates does save some time during the recording process.

While pairwise comparison and trade space analysis (previously presented) are very useful, they can also be time consuming to implement. In a complex design problem, the pairwise comparison chart can become especially troublesome. The chart itself can become very large which not only creates difficulties in creating it, but the large size can also be very time consuming to fill out. The large size can also create difficulties when it comes time to calculate the relative weights of the various design characteristics. It can become very easy to lose track of data, leading to errors in the calculations.

IDEA's *Concept Evaluation Workbook* lessens these problems by generating the empty pairwise comparison chart, simplifying data entry, and automatically calculating the relative importance of the design characteristics and displaying them in real time both numerically and graphically. The IDEA tool reuses this information in building the decision matrix, which is also largely automated. The design team simply fills in the appropriate ratings for each of the concepts in the decision matrix and the workbook automatically calculates the aggregate scores, while simultaneously generating a line graphs and pie charts of the strengths and weaknesses of each concept.

Furthermore, IDEA's *Trade Space Analysis* tool semi-automatically generates trade space diagrams. Designers simply enter the various options available for the functional requirements and the trade space tool automatically generates a trade space diagram, which can save a great deal of time in complex problems. The authors are still working on new modules to help eliminate "bad" combinations from the trade space diagram.

## 5 CASE STUDIES

One good way to validate the effectiveness of a new method is with case studies. Case studies allow one to go through a problem by actually using the process in question. This can uncover weaknesses in the process which may not have been previously evident. In attempting to validate the IDEA process, the authors conducted three case studies that covered a range of topics. Case study topics were selected to demonstrate the versatility of the IDEA method, and increase the probability of finding any potential problems with the process.

The first case study covers work previously done by Masur [9]. That work dealt with the design of a Sample Processing Unit (SPU) for use in the upcoming Mars Science Laboratory (MSL) mission. The main requirement of the SPU was to take solid or loose regolith samples from the Martian surface and reduce them down to a fine particulate suitable for use in NASA-supplied scientific instruments. This problem was selected for three reasons. Firstly, the authors are very familiar with the subject matter, and so could focus on applying IDEA carefully. Secondly, the authors were interested in seeing how IDEA might alter the solution that Masur had previously generated without IDEA. Thirdly, the original work received high marks for originality and creativity. If applying IDEA led to a better design, then one would have anecdotal evidence that IDEA promotes creativity.

The results of this case study are summarized as follows. In the original work, Masur generated only three concepts, and this was achieved with difficulty because no systematic method for exploring the design space was used. On the other hand, we generated over 756,000 concepts with the first three modules of IDEA, and included a much broader variety of concepts. This suggests it is more likely that the "best" concept was indeed contained in the rather large initial design space.

Furthermore, IDEA identified the same concept as the original work, as the best design concept. The authors could not predict this outcome during the execution of the IDEA *Analyze* module because the actual calculations are done "off-screen," and rendered only when the ranking task is complete. Indeed, it was our intention that IDEA act this way, to lessen the possibility of inadvertent bias by the designers on seeing partial results displayed (an analogy to the phenomenon of "strategic voting" in national elections in some countries). In addition, the assessors of the original design solution deemed it quite innovative. The authors suggest that this supports our proposition that a systematic method like IDEA can facilitate innovation.

Finally, we found the level of detail about the concept design process far greater in IDEA than even the design notebooks kept by Masur during the original work. The design history represented in the IDEA worksheets and other documents was much more concise and better organized than the original work. Also, the IDEA information stores can be searched easily and quickly; the original notes, written on paper, are nearly impossible to search quickly. In summary, while IDEA resulted in the same selection as the original work it is more important to note that its use lead to generation of many more concepts and consequently a much better exploration of the design space. This increases the probability that the optimum solution or solutions will be located.

The second case study considers a problem given to Harvard Law School students known as the Case of the Speluncean Explorers [10]. It is a fictional problem set in the state of Newgarth in the year 4300. In it, four male members of the Newgarth Speluncean Society become trapped while exploring a cave. Since the trip was originally not supposed to last longer than a day, they did not pack provisions for a long internment. Using a portable radio, they are able to contact the rescue team that the Society sent to retrieve them. Subsequent to these discussions, the team realizes that they will not survive until rescue unless they kill and eat one of their numbers. One of the men develops a lottery system using dice to decide who will be killed. However, this individual reneges on the agreement during the lottery. The others do not accept this and the roll happens to go against the man, who is cannibalized in due course. The rest of the team is subsequently rescued, but are charged with murder and eventually found guilty and sentenced to death. The verdict has been appealed and it is the job of the reader to act as the head of Newgarth's legal system and decide whether to uphold it.

The structure of the case precludes any truly satisfactory answer; it is instead an exercise in problem solving and ethics. The authors chose this problem to test IDEA because it would provide an interesting contrast to conventional, technologically oriented cases, as well as demonstrating the versatility of the method.

Once again, we used the IDEA tool to conduct the case study. It became apparent immediately that we could use only some elements of the IDEA tool. A problem such as this is more of a moral and ethical dilemma as opposed to an engineering problem that tends to be more exact. This limited many of the tools that we could use since many of them operate on concrete variables. Most of the work was done with the *Identify* and *Determine* modules, which we used to gain a better understanding of the problem and to more clearly outline all of the facts.

The use of IDEA led to the identification of one major factor that would have a significant impact on the outcome. While at first glance there may seem to be few stakeholders in the case, the IDEA analysis revealed that there are in fact many people with a stake in the case. Not only were there the interests of the families and friends of the men who would be put to death, but there were also the interests of the Speluncean Society, which bankrupted itself rescuing the men, and the families of the rescue workers, some of whom died during the rescue. Furthermore, there are also the needs of the State, which include upholding the integrity of the law so that it does not lose strength and validity in future cases. The needs of these stakeholders have a drastic effect on the final verdict; to ignore them would lead to highly questionable solutions. The fact the IDEA identifies these issues helps to ensure a more *just* solution.

Using IDEA, we found that the men should be found guilty, to uphold the laws of the State, but that their sentence should be commuted, to account for their lack of malicious intent. Once again, the actual result is of less importance than the *process* that was used to achieve it. The use of IDEA lead to a more systematic analysis of the problem and its variables, in this case the various stakeholders. Different stakeholders can have a dramatic effect on the final decision and identifying all of them is important in rendering a just decision. This demonstrates the versatility of the IDEA process since the logic can tackle a wide variety of problems outside of engineering.

The third case study was a problem based on the design of the PowerPC 603 processor. The main reason for choosing this case study was the vast amount of information available on the development of the PowerPC 603. Originally released in the early 1990's, the 603 was the follow-up to the PowerPC 601, which was the first microprocessor designed to process PowerPC code. The 601 was developed for Apple Computers, Inc. to compete against Intel's original Pentium chip from a performance standpoint, while using less power and generating less heat. While the 601 was indeed competitive on performance, its power consumption and heat generation were not much better than that of the Pentium. The PowerPC 603 was an attempt to create a processor that offered equivalent performance to the 601, ran cool enough for use in laptops, and used significantly less power. For these reasons, the 603 became known as a revolutionary chip.

A wealth of information is available about the development of the 603. Although many papers highlight different aspects of the chip, of greater interest are a number of papers [11,12] that describe the concept design process followed by its developers. This was a very compelling repository of information since it would allow direct comparison between the results and decisions of the original engineers, and the results and decision generated by the IDEA method. Furthermore, the papers also contained a great deal of theoretical performance data for different processor configurations. The original designers used a simulation package called BRAT [11] to simulate the performance of different theoretical configurations for the PowerPC 603. By using this information in the case study, the authors could generate rough estimates of the performance of many different processor configurations. We can then use this performance data in the IDEA modules to generate and select viable design options.

The IDEA method allowed the authors to generate eight different concepts for the final design of the PowerPC 603. In the end, we identified two concepts that struck superior balance between speed, power, and the other design characteristics, with respect to design goals derived from the existent documentation. One of the concepts was very similar to the actual 603 design. The other, however, was quite different; it emphasized lower power consumption and heat generation. Although the theoretical performance of the second concept was somewhat lower, the presumed errors between theoretical and actual performances were small enough to lead us to believe it may have been a very reasonable alternative.

The fact the IDEA was able to identify a second alternative design demonstrates that the process has lead to a thorough exploration of the design space. In engineering there is rarely a single "best" solution to a design problem. All solutions are compromises designed to suit a given set of criteria. Ultimately there can only be a single selection, however identifying a number of equally suitable alternatives can be very useful if the concerns driving the design problem change, or if a similar problem arises in the future. At the very least these alternative designs can be used to justify the decision on the ultimate design if questions arise.

Complete details of all the case studies are available in [5].

## 6 CONCLUSIONS

Concept design is critical to product development and engineering. It is, however, not especially well supported either computationally or methodologically. This paper has introduced the IDEA method of conducting concept design. IDEA integrates a set of well-known techniques, and provides a software tool to facilitate its use.

The purpose of IDEA is to provide a systematic method for concept design that helps (a) control methodological complexity, and (b) ensure a thorough exploration of the design space by facilitating designers to study as many aspects of a design problem as possible. The software tool enhances IDEA by making it easier to use and implement in practical settings.

In the future, we hope to incorporate features of other methods for concept design, such as TRIZ[X]. It is easy to see how this might work. First, one would have to identify "contradictions" (per TRIZ). Designers would use Compendium's graphical interface to develop a visual representation of the contradictory elements. The IDEA software could then render a visualization of the various TRIZ principles that address the identified contradictions. Of course, TRIZ is more complex than may be implied here, but the essential features should be relatively easy to embed into IDEA.

Although more evaluation of IDEA is needed, the results of three case studies tentatively demonstrate its usefulness. Not only has the use of IDEA lead to a greater exploration of the design space in all three of the case studies, but the use of the interface has also created permanent documentation which outlines not only what decisions were made, but why designers made them.

## REFERENCES


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

[2] Wertz, James R. and Larson, Wiley J., *Space Mission Analysis and Design 3rd Edition*, Microcosm, El Segundo, 2003, p. 7

[3] R.L. Keeney, "Stimulating Creative Design Alternatives Using Customer Values", *IEEE Transactions On Systems, Man, and Cybernetics – Part C Applications and Reviews*, Vol. 34, No. 4, November 2004, pp. 450-459

[4] M.J. Scott, and E.K. Antonsson, "Arrow's Theorem and Engineering Decision Making", *Research in Engineering Design*, 1999, pp. 218-228

[5] A. Masur, "The Integrated Design Exploration and Analysis (IDEA) Process", Master's Thesis, Ryerson University, Department of Mechanical and Industrial Engineering, 2006.

[6] D.O. Anderson, *Making Engineering Design Decisions*, Louisiana Tech University, 2000, pp. 1-9

[7] S.H. Zanakis, et. al., "Multi-attribute Decision Making: A Simulation Comparison of Select Methods", *European Journal of Operational Research*, Issue 107, 1998, pp. 507-529

[8] Compendium Institute, "Compendium Community Showcase", http://www.compendiumsinstitute.org/community/showcase.html, (April 21st, 2006)

[9] L.L. Fuller, "The Case of the Speluncean Explorers", *Harvard Law Review*, Vol. 62, 1949, pp. 616-645

[10] A. Masur, "Design of a Sample Processing Unit for the Mars Science Laboratory Mission", Bachelor of Engineering Thesis, Ryerson University, Department of Aerospace Engineering, 2004.

[11] B. Burgess, et. al., "The PowerPC 603 Microprocessor", *The Making of the Power PC*, Communications of the ACM, Vol. 37, No. 6, 1994, pp. 34-42

[12] B. Burgess, et. al., *The PowerPC 603 Microprocessor: Performance Analysis and Design Trade-Offs*, Motorola and IBM, 1994, pp. 316-323



Contact: Filippo A. Salustri, PhD, PEng
Ryerson University
Department of Mechanical and Industrial Engineering
350 Victoria Street
Toronto, Ontario
Canada
Phone: 001 416 979 5000 ext 7749
Fax: 001 416 979 5265
e-mail: salustri@ryerson.ca
URL: http://deseng.ryerson.ca/~fil